

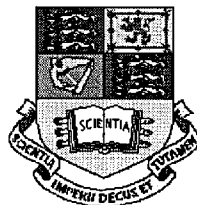
Fast and Automatic Techniques for 3D Visualisation of MRI Data

Norman Ronald Smith

A thesis submitted for the degree of Doctor of Philosophy

January 1998

Department of Electrical and Electronic Engineering



Imperial College of Science, Technology and Medicine
UNIVERSITY OF LONDON

Abstract

Since the advent of Magnetic Resonance Imaging (MRI), technological advances have made it possible to acquire high resolution, 3D anatomical, image data. As a result, processing techniques have emerged which enable the contents of such data to be visualised within a virtual three dimensional environment. However, existing approaches require user interaction and long processing times.

In this thesis, methods for automating and minimising the computation time required for the 3D visualisation process are developed. A preliminary study is presented to explore the capability of acquiring data, from both conventional MRI and interventional MR (IMR) scanners, for subsequent 3D visualisation. However, the main focus of this work is on the essential data classification and image segmentation stages of the visualisation process.

A new fuzzy clustering approach, the fast fuzzy *c*-means (FFCM) algorithm, is described for tissue classification. Using this method, accurate classifications of both MRI and IMR data have been obtained. Unlike existing techniques which require several hours to process 3D data sets, the FFCM algorithm takes less than one second.

In order to isolate anatomical structures from FFCM classified data, quickly and accurately, an adaptive refinement (AR) process has been developed. This was shown to extract 3D bone structures from MRI knee data in only a few seconds.

The performance of different rendering schemes, applied to FFCM classified, and AR segmented, data, is investigated. It was found that both surface and binary volume renditions of segmented data provide very clearly defined displays. Volume rendering FFCM classified data yielded poorly defined results. However, these were improved by applying appropriate matting operations.

Finally, the clinical applications of these methods, with particular attention to 3D image guided surgery, are examined and overall conclusions are made.

Preface

Statement of Originality

In this thesis, methods are investigated and developed for fast and automatic 3D visualisation of MRI data. To the best of my knowledge, the main contributions of this work are original. These are:

Chapter 5:

- The Fast Fuzzy c -Means (FFCM) classification algorithm.
- The colour blending scheme for displaying fuzzy classified data.

Chapter 6:

- The Adaptive Refinement (AR) process and its incorporated methods:
 - (i) Boundary tracking and labelling
 - (ii) The hierarchical approach to region merging:
 - Simple homogeneity testing
 - Markov Random Field (MRF) based homogeneity testing
 - (iii) 3D region extraction.

Notation

Throughout this thesis, standard mathematical notation is adopted. When citing a reference from the bibliography, the following convention is adopted: The first

letter of each author's surname is quoted followed by the year of publication. For example, the paper authored by Smith and Kitney in 1997 is cited as [SK97]. For papers written by only one person, the first three letters of the author's surname are quoted. For example, Besag's paper of 1974 is referred to as [Bes74]. Finally, where a paper has been co-authored by more than four people, the initials of the first three authors are followed by a + sign.

Acknowledgements

The research conducted for this thesis was supported by the Engineering and Physical Sciences Research Council (EPSRC) and Smith and Nephew Plc. I am indebted to Professor Richard Kitney for his supervision throughout the entire project. I am also grateful to Dr Peter Cashman for his general advice and useful tips, and to Dr Ioannis Matalas and Dr Joao Batista for their helpful discussions in the early stages of the project. In addition, thanks go to Dr Wady Gedroyc for his clinical appraisal of the results and to Jeremy Gerber and Nick Shaw for their help in acquiring the MRI data. Thanks are also due to Ravi Bickram, Dustin Green and Matthew Lip-trot for volunteering to be subjects for MR imaging studies. Finally, special thanks go to my wife Shoshana for her continual encouragement and support.

Norman Smith

January 1998

Contents

Preface	1
List of Figures	7
List of Tables	11
List of Abbreviations	12
1 Introduction	15
1.1 Magnetic Resonance Imaging	15
1.2 3D Visualisation of MRI Data	16
1.3 Direct 3D Visualisation: Clinical Applications	17
1.3.1 Diagnosis	17
1.3.2 Pre-surgical Planning and Rehearsal	17
1.3.3 Radiotherapy Planning	18
1.3.4 Surgical Guidance	18
1.4 3D Image Guided Surgery	19
1.4.1 Motivation	19
1.4.2 Implementation	20
1.5 Interventional Magnetic Resonance Imaging	22
1.6 3D Visualisation Methods: Speed Limitations	23
1.7 Thesis Objectives	24
1.8 Thesis Outline	24
2 Image Segmentation	26
2.1 Introduction	26
2.2 Pattern Recognition Techniques	28

2.2.1	PR Systems and Image Segmentation	29
2.2.2	Hard and Fuzzy Sets: A Framework for PR Systems	30
2.2.3	Supervised Methods	31
2.2.4	Unsupervised Methods	35
2.2.5	Feature Reduction	38
2.3	Spatial Methods	38
2.3.1	Boundary Identification	39
2.3.2	Morphological Techniques	42
2.3.3	Region Directed Methods	45
2.3.4	Fuzzy Region Extraction	47
2.4	Hybrid Approaches	48
2.4.1	Relaxation Labelling	49
2.4.2	Markov Random Field Modelling	49
2.5	Knowledge Based Schemes	55
2.5.1	Rule Based Systems	55
2.5.2	Knowledge Guided Methods	56
2.5.3	Deformable Atlas Matching	56
2.6	Methods for MR Image Segmentation	57
2.6.1	Conventional Techniques applied to MRI Data	57
2.6.2	Fast and Automatic Segmentation of MR Image Data	60
3	3D Rendering Techniques	61
3.1	Surface Rendering	61
3.1.1	Surface Extraction	62
3.1.2	Rendering	63
3.2	Volume Rendering	65
3.2.1	Preparatory Processing	66
3.2.2	Rendering	71
3.2.3	Matting	76
3.3	Surface versus Volume Rendering	76
3.3.1	Computational Efficiency	77
3.3.2	Pre-processing Requirements and Quality of Display	78
3.3.3	Interactive Manipulation	78
3.4	Final Remarks	78

4	MR Image Acquisition	79
4.1	Methods for Data Acquisition	79
4.1.1	Imaging Requirements	79
4.1.2	Imaging Protocols	80
4.2	Results and Discussion	82
4.2.1	Image Quality	82
4.2.2	Acquisition Times	84
4.3	Conclusions	85
5	Fast Fuzzy Classification of MR Image Data	86
5.1	Introduction	86
5.2	The New Classification Algorithm	87
5.2.1	Fast Fuzzy <i>c</i> -Means (FFCM) Clustering	87
5.2.2	Display of Fuzzy Segmented Data: Colour Blending	90
5.3	Alternative Classification Methods	93
5.4	Results and Discussion	94
5.4.1	The New Algorithm: Initial Conditions	94
5.4.2	Performance of the New Algorithm	95
5.4.3	Appraisal of Alternative Methods	102
5.5	Conclusions	107
6	Adaptive Refinement of Classified MRI Data	109
6.1	Introduction	109
6.1.1	Motivation	109
6.1.2	Overview of the Adaptive Refinement Process	110
6.2	Preliminary Processing Tasks	112
6.2.1	Data Defuzzification	112
6.2.2	Connected Component Labelling	113
6.3	Region Selection and Preparation	116
6.3.1	Hole Removal	116
6.3.2	Boundary Tracking and Labelling	117
6.4	Adaptive Region Refinement	121
6.4.1	Region Merging and Splitting	122
6.4.2	A Hierarchical Approach to Region Merging	123

6.4.3	Simple Homogeneity Testing	124
6.4.4	MRF Based Homogeneity Testing	126
6.5	3D Implementation of the AR process	130
6.5.1	The Problems of Direct 3D Data Processing	130
6.5.2	3D Region Extraction	132
6.5.3	Adaptive Refinement of Extracted 3D Region Data	135
6.6	Results and Discussion	135
6.6.1	Preliminary Processing Tasks	136
6.6.2	Region Selection and Preparation	137
6.6.3	Adaptive Region Refinement	139
6.6.4	3D Implementation of the AR Process	146
6.6.5	Computational Efficiency of the AR Process	148
6.7	Conclusions	149
7	Fast 3D Visualisation Techniques: Performance and Applications	151
7.1	Evaluation of 3D Rendering Methods	151
7.1.1	Methods	152
7.1.2	Results and Discussion	153
7.2	3D Image Guided Surgery	158
7.2.1	Experimental Methods	159
7.2.2	The Registration of World-Space with Image-Space	160
7.2.3	Results and Discussion	161
7.3	Other Applications	165
7.4	Conclusions	166
8	Final Discussion, Conclusions and Future Work	167
8.1	Summary of Main Results	167
8.2	Final Conclusions	168
8.3	Future Work	169
8.3.1	Image Acquisition	169
8.3.2	Image Segmentation	170
8.3.3	3D Rendering Methodologies	172
8.3.4	3D Image Guided Surgery	173

A	Fundamentals of MRI	174
A.1	Basic NMR Theory	174
A.1.1	Spins in a Magnetic Field	174
A.1.2	The Effect of Radiofrequency (RF) Pulses	175
A.1.3	Relaxation: The Return to Equilibrium	176
A.2	MR Image Formation	178
A.2.1	2D Slice Acquisition	178
A.2.2	3D Volume Acquisition	180
A.3	MR Image Properties	182
A.3.1	Image Contrast	182
A.3.2	Image Quality	183
B	MRF Based Classification of MRI Data: A Numerical Solution	184
B.1	Process Models	184
B.1.1	The Multilevel Logistic (MLL) Model	184
B.1.2	Univariate Gaussian Model	185
B.2	Numerical Solution of the MAP Criterion	186
B.2.1	Parameter Estimation	188
B.2.2	Iterated Conditional Modes	189
C	Seed Fill Algorithm	192
D	Least Squares Fitting of two 3D Point Sets	194
	Bibliography	196

List of Figures

1.1	The Process of 3D Visualisation of MRI Data.	16
2.1	Different approaches for segmenting image data into region data. . . .	27
2.2	Image segmentation using a pattern recognition scheme.	29
2.3	An example of the resulting segmentation after performing the first two stages of a split and merge algorithm	46
2.4	Neighbour sets of orders 1 to 5.	50
2.5	The second order neighbourhood set and its associated clique types .	51
3.1	The main stages involved in rendering a surface model.	63
3.2	The Phong reflection model.	64
3.3	Opacity assignment function weighted by surface strength.	68
3.4	Colour assignment within a voxel	70
3.5	Ray casting through coloured and opacity voxel array.	72
3.6	Projection of a voxel onto the viewing plane.	74
4.1	Image data acquired by conventional IGE 1.5 Tesla scanner.	82
4.2	Image data acquired by 0.5 Tesla IMR scanner.	83
5.1	Colour spaces for segmented images with (a) 4 clusters and (b) 6 clusters	91
5.2	The New Classification Algorithm	93
5.3	Plots of histograms from T1 knee data (as shown in figure 4.1(b)) from (a) original 12 bit image and (b) reduced resolution 8 bit image. .	95
5.4	Plots of Cluster Memberships after performing FFCM on T1 knee data (as shown in figure 4.1(b)) using grey scale histogram generated from original 12 bit image and from reduced resolution 8 bit image. .	96
5.5	Colour blended display of classified knee data (as shown in figure 4.1(b)) after 25 iterations, $c = 5$, $m = 2.0$	97

5.6	Colour blended display of classified head data after 25 iterations, $c = 4$, $m = 2.0$	98
5.7	Colour blended display of classified IMR head data, with FFCM applied to entire volume, after 100 iterations, $m = 2.0$	99
5.8	Plots of $J_m(W, \mathbf{v})$ throughout 25 iterations of the FFCM algorithm. .	101
5.9	Colour blended output of EM algorithm applied to conventional MRI knee data (slice15 - as shown in figure 4.1(b)), $c = 5$	103
5.10	Plots of estimated gaussian mixture distributions superimposed over normalised grey scale histograms from T1 knee data (as shown in figure 4.1(b)) after applying EM algorithm to data.	104
5.11	Colour blended output of EM algorithm applied to conventional MR head data (slice 62 - as shown in figure 4.1(a)), $c = 4$, number of iterations=200.	105
5.12	Slice 62 of conventional MR head data set after performing HCM and FFCM classification. $c = 4$, number of iterations=25.	106
6.1	A flow-chart summary of the Adaptive Refinement (AR) Process. . .	111
6.2	Different possible sets of neighbour elements in 2D and 3D image arrays.	113
6.3	An example of the connected component labelling algorithm: (a) Hard classified image data; (b) Labelled 4-connected region data. . .	115
6.4	An example of a region (a) before and (b) after hole removal.	116
6.5	(a) Image and boundary lattices and (b) site indexing.	117
6.6	(a) A region in an image. (b) A blown up section of the contour which defines the boundary of the region. This is made up of internal contour (IC) pixels, external contour (EC) pixels and boundary contour (BC) sites.	118
6.7	An example of tracking the BC sites which define a region boundary.	120
6.8	(a) Closed contour. Co-ordinates and associated labels are reshuffled so that the first contour co-ordinate is the first detected junction in the clockwise direction. (b) Open contour. Co-ordinates and associated labels are reshuffled so that they begin from one end and finish at the other in clockwise order.	121
6.9	Examples of region merging and splitting.	122
6.10	Example of a boundary contour section and its associated neighbourhood ϱ_s	125
6.11	Examples of a BC section and its associated neighbourhoods (a) ϱ_s ; and (b) ϱ_{MRF} - dominant radius=3. The neighbourhoods comprise pixels from the selected region as well as from its adjacent regions. . .	127

6.12	Growing the neighbourhood ϱ_{MRF} from region k , the non-dominant region.	128
6.13	An example of a BC section and its associated neighbourhood ϱ_{MRF} whose dominant radius=3. The pixels inside ϱ_{MRF} are highlighted and numbered according to the different regions in which they belong. . .	129
6.14	Examples of problematic cases when attempting to fill a 3D region: (a) Poor inter-slice resolution; (b) Misclassification error.	131
6.15	The 3D region extraction process. All the shaded regions have been classified according to the same tissue type. Regions R_1 and R_2 in slice $k - 1$ are both adjacent to region R_s in slice k	134
6.16	3D implementation of the AR process: A pictorial summary.	135
6.17	Preliminary processing tasks of the AR process applied to FFCM classified knee data (as shown in figure 4.1(b)).	136
6.18	Region selection: (a) Before hole removal; (b) After hole removal. . .	138
6.19	Magnified views of the selected region's: (a) Internal and external contours. The IC pixels are depicted by a darker grey scale intensity than their EC pixel counterparts; (b) Boundary contours.	139
6.20	Results from simple homogeneity testing: (a) Under merging; (b) Over merging.	140
6.21	Global MRF based classification of knee data (as shown in figure 4.1(b)).	141
6.22	(a) Selected region inside the tibial bone of the classified knee data shown in figure 6.17(a); (b) The ϱ_{MRF} neighbourhoods of the BC sections which make up the boundary contours of the selected region; (c) The relabelled pixels in the ϱ_{MRF} neighbourhoods.	143
6.23	Final result from the AR process after selecting the region illustrated in figure 6.18(a).	144
6.24	(a)-(c) Extracted 2D regions. (d)-(e) Adaptively refined 2D regions. .	147
6.25	Surface rendition of segmented femoral and tibial bone data.	148
7.1	Surface and volume renditions of segmented IMR brain data	154
7.2	Volume renditions of FFCM classified IMR head data.	156
7.3	(a) Surface rendition of knee phantom, localisation markers (yellow) and registered geometric probe model (blue). (b) Magnified surface rendered view of segmented knee bone data with simulated probe positioned inside the joint.	163

A.1	(a) Protons under the influence of an external field, $\mathbf{B}_0$. (b) An RF pulse is applied, which acts as a second field, $\mathbf{B}_1$, in the plane perpendicular to $\mathbf{B}_0$. This is portrayed within the <i>rotating frame</i> system of reference	175
A.2	The CPMG (Spin Echo) Sequence.	177
A.3	The external magnetic field with (a) no imposed gradient (b) a gradient introduced along a specific axis.	178
A.4	A summary of how gradient sequences are applied to generate (a) 2D slice images, and (b) 3D volumes.	180
A.5	The Spin Echo (SE) Pulse Sequence	181
C.1	An example of the seed fill algorithm as it progresses through a 4-connected region.	192

List of Tables

- 4.1 Acquisition times for conventional MR and IMR imaging sequences. . . 84
- 5.1 Performance times of FCM and FFCM algorithms 102
- 7.1 Tissue opacities and corresponding colour allocations and tissue types
resulting from FFCM classification. 155
- 7.2 Measured and registered co-ordinates of markers positioned around
the circumference of the “leg” of the knee model. 161
- A.1 Contrast Extremes for Spin Echo Imaging. 182

List of Abbreviations

AR	Adaptive Refinement
ASA	Adaptive Segmentation Algorithm
BC	Boundary Contour
CT	Computed Tomography
EC	External contour
EM	Expectation-Maximisation
FCM	Fuzzy <i>c</i> -Means
FFCM	Fast Fuzzy <i>c</i> -Means
FFT	Fast Fourier Transform
GD	Gibbs Distribution
GRASS	Gradient Recalled Acquisition at Steady State
HCM	Hard <i>c</i> -Means
IC	Internal contour
ICM	Iterated Conditional Modes
IMR	Interventional Magnetic Resonance
MAP	Maximum a posteriori
MIS	Minimally Invasive Surgery
ML	Maximum Likelihood
MPL	Maximum Pseudo-Likelihood
MR	Magnetic Resonance
MRF	Markov Random Field
MRI	Magnetic Resonance Imaging
NMR	Nuclear Magnetic Resonance
NEX	Number of Excitations
PR	Pattern Recognition

POS	Partial Optimal Solution
RF	Radio Frequency
RHS	Right hand side
SA	Simulated annealing
SNR	Signal to Noise Ratio
SPGR	Spoiled Gradient Recalled Acquisition at Steady State
TE	Echo time
TR	Repetition time (for MR imaging protocol)

Chapter 1

Introduction

1.1 Magnetic Resonance Imaging

The phenomenon of Nuclear Magnetic Resonance (NMR) in solids and liquids (see section A.1) was first observed in 1946 by Bloch and Purcell [BHP46, PTP46], for which they were both awarded Nobel prizes. The sphere of NMR applications was restricted to physics and chemistry problems until the early 1970s when the NMR signal was found to be sensitive to cancerous states in biological tissue [Dam71]. The development of methods for spatial localisation of the NMR signal followed shortly afterwards [Lau73], and in 1977 the first human MR images¹ were acquired [DGM77, MM77].

During the past twenty years, MRI technology has progressed enormously to become one of the most versatile and powerful diagnostic imaging techniques in medicine. There are several reasons for its huge popularity. First of all, unlike its main rival modality X-ray CT, MRI does not employ ionising radiation. Furthermore, MR signals are extremely sensitive to different biological tissues, thus allowing excellent tissue contrast to be achieved. Finally, high resolution images may be acquired by MRI systems, within any anatomical plane, in the form of ei-

¹The full abbreviation NMR has been more popularly shortened to MR when referring to its clinical application.

ther 2D tomographic, or 3D volume, data. For a more detailed discussion on the fundamentals of MRI, the reader is referred to appendix A.

Throughout this chapter and entire thesis, we will focus on the 3D imaging capability of MRI. In particular, we will be looking at techniques which best enable such data to be visualised.

1.2 3D Visualisation of MRI Data

The ability to acquire MR image data in three dimensions enables clinicians to view, in detail, entire anatomical regions. Such data is examined conventionally on a slice-by-slice basis. A sequence of consecutive slices is printed onto a single film and viewed by a light box. Although this may be very informative, its interpretation demands much skill. Furthermore, subtle variations between different image slices can be missed easily by even the most expert viewer.

Over the past two decades, image processing and computer graphics techniques have been developed which allow three dimensional MRI data to be visualised directly within a virtual 3D rendered environment. Figure 1.1 shows a flowchart

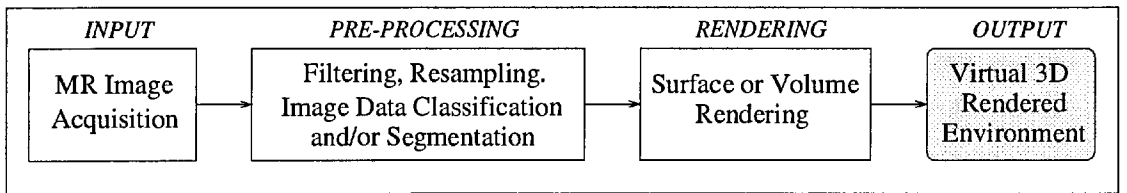


Figure 1.1: The Process of 3D Visualisation of MRI Data.

summary of the main stages of the three dimensional visualisation process. First and foremost, 3D image data of the anatomical region of interest must be acquired. This data is then input to the pre-processing stage of the visualisation process. Pre-processing operations include noise filtering, often used to smooth the data, and resampling which converts data to a higher or lower resolution. However, the most critical tasks to be performed during this stage are those of data classification and/or

image segmentation. The purpose of these essential steps is to identify the structures of interest so that they may be visualised subsequently. Finally, the pre-processed data is converted into a virtual 3D scene through the procedure of rendering.

1.3 Direct 3D Visualisation: Clinical Applications

The potential generation of a 3D rendered scene provides a wide range of applications within the clinical environment. Some of the most popular realms of application will now be discussed.

1.3.1 Diagnosis

High quality MR images enable radiologists to detect very subtle structural changes within anatomical regions. These allow the discovery and hence treatment of pathological abnormalities at the earliest possible stage. Hence, there is, nowadays, an increased emphasis on preventative, rather than curative, medicine.

As mentioned earlier, it can be extremely difficult to analyse 3D data through the examination of individual image slices alone. This is particularly true during the investigation of very slight structural abnormalities which might suggest the onset of disease. By employing 3D visualisation methods, the diagnostic process may be enhanced significantly. Small anatomical features, which could otherwise be missed, may be viewed clearly within a 3D rendered scene. Furthermore, the ability to manipulate three dimensional objects allows clinicians to uncover and inspect regions which are camouflaged by their surroundings.

1.3.2 Pre-surgical Planning and Rehearsal

Although there are many standard techniques which an experienced surgeon may perform almost blind-folded, situations often arise which are entirely new. Pre-

operative 3D visualisation of the structures on which the operation is to be performed provides additional, valuable information which may prevent such encounters from arising. Furthermore, through the employment of interactive manipulation and virtual cutting tools, it is even possible to rehearse surgery in advance. This eliminates, or minimises, the need to design complex plans in the operating theatre while a patient is under anaesthetic, possibly unnecessarily cut open. This, in turn, reduces the risk of complication and cuts the cost of surgery. In addition, 3D visualisation facilitates accurate measurement of structures of interest, hence enabling the precise design of surgical instruments and prosthetic implants.

A good review of different clinical examples in which pre-surgical planning and rehearsal have been applied may be found in [RHC96].

1.3.3 Radiotherapy Planning

The goal of radiotherapy is to eradicate tumours by means of either external radiation beams or internal radiation implants, while minimising the damage to normal tissue. The traditional approach to radiotherapy treatment planning relies on 2D image slices whose contents must be mentally reconstructed, into the actual 3D anatomy, by the clinician. Treatment planning systems have therefore been developed [MBB⁺88, RSF⁺89] which utilise 3D visualisation techniques. These simplify and enhance the planning procedure and consequently improve the performance of subsequent treatment.

1.3.4 Surgical Guidance

In addition to aiding the planning and rehearsal of interventional procedures, 3D visualisation methods may be used to guide actual surgery. This is an application of particular relevance to the work presented in this thesis. The following section therefore describes, in some detail, the motivation for, and implementation of, 3D

image guided procedures.

1.4 3D Image Guided Surgery

1.4.1 Motivation

The open and invasive nature of traditional surgery has been the cause of many consequent unexpected problems. Wound infections and long recovery times are but a few such complications. Hence, there has been a growing trend in recent years to develop and adopt *minimally invasive* procedures.

As its name suggests, the purpose of Minimally Invasive Surgery (MIS) is to allow the surgeon to operate on a patient without having to create major surgical incisions. This is accomplished by making minor incisions at specific positions on the patient's external anatomy. Through one incision, an endoscope is inserted, and through the other, a surgical instrument is passed. During a highly skilled process, known as *triangulation*, the instrument is positioned such that it may be viewed through the endoscope, thus allowing the operation to be performed.

Considerable expertise is required to conduct MIS routines. The surgeon must rely entirely on the *keyhole* view acquired from the end of an endoscope and combine this with an excellent knowledge of anatomy and much experience to navigate through the operation successfully. Furthermore, in some instances, such as in very confined regions, it is not possible to insert both an endoscope and a surgical tool. In such cases, the procedure must be either abandoned or performed blind.

The above described difficulties may be avoided through MR image guided surgery. This involves acquiring 3D MR image data of the global region in which the operation is being performed prior to surgery. During the operation, a continuously updated position of the surgical tool is superimposed on the image. Although such image data has until recently been viewed in 2D slice format, the advancement of

3D visualisation methods now enables the image and surgical position to be viewed within a 3D rendered environment. Thus, the combination of a high resolution microscopic view, acquired by the endoscope, together with a global 3D rendition of the overall anatomical scene which includes the positions of any tools, allows a surgeon to conduct even the most awkward and complex routines confidently and accurately.

1.4.2 Implementation

In order to perform image guided surgery, it is necessary to acquire MR image data, within the context of a fixed reference, prior to surgery. This may be achieved by either attaching MR sensitive markers to the skin of the patient or to an external structure, such as the stereotactic frame developed for neurosurgery. Alternatively, specific anatomical landmarks may be considered. During surgery, the co-ordinates of the tools are simultaneously tracked and registered with the image in order to provide the final display. The required processes of tracking and registration will now be discussed.

(a) Tracking

Tracking involves locating world-space co-ordinates, which define the position of the surgical tool, in real-time. A number of different approaches have been considered.

First attempts involved the use of a mechanical arm possessing angular sensors in six degrees of freedom [GELM93, GME92, HR94, LTG⁺92, ZJK94]. The most advanced system of this kind works to a resolution of about ± 2 mm [HR94]. Although this is a significant improvement over the early systems, which were accurate to only ± 5 mm, it requires a large working volume and weighs about 3.5 kg. During surgery, where the environment must be clean and sterile, such clumsy devices are awkward and difficult to use.

Other, less cumbersome and more accurate approaches have been adopted. The Polhemus Fast-track system is an electromagnetic system which was originally developed for flight simulation. It comprises a transmitter which produces an RF field and a small sensor whose location in space can be found automatically. However, this system suffers from inaccuracies due to field distortion caused by metal objects.

Horstmann and Reinhardt [HR94] have developed an ultrasonic based tracking system which they claim has a localisation accuracy of ± 0.4 mm. This requires an ultrasound receiver unit, mounted on a panel near the patient, and a transmitter unit, embedded in the handle of the pointing instrument.

The VISLAN project [Eva93, SHCG96] considers video technology for the purpose of localisation. This involves tracking a fixed pattern, attached to the surgical tool, by processing the images acquired by a pair of stereoscopic video cameras. The method is cheap and effective. However, the attachment of an appropriate pattern accessory can be a bit clumsy.

Optotrak and Pixsys have developed an LED based optical tracking system. It consists of a camera bar, placed above the operating area, which contains three sensors. These are able to track up to as many as 256 LED markers, which may be attached to hand-held devices, with an accuracy of ± 0.1 mm, at a rate of 10 positions per second. The disadvantage of this configuration is that if anything blocks one of the LED markers, its co-ordinates are not detected by the camera bar. Nevertheless, with the high degree of accuracy this system provides, it has proven to be one of the most popular methods.

(b) Registration

Registration involves computing suitable rotation and translation matrices such that any point existing in one co-ordinate system may be mapped to an equivalent point

in another. These transformations may be either linear or elastic. For most applications, including image guided surgery, a linear transformation may be assumed.

At the start of surgery, the co-ordinates of the fixed reference markers in world-space are registered with those in image-space. The resulting rotation and translation matrices are then used, during the actual procedure, to convert the tracked co-ordinates of surgical tool position in world-space to image-space.

1.5 Interventional Magnetic Resonance Imaging

During the past few years, General Electric has been running a trial and development programme on the prototype for a new type of MR imaging system, referred to as the Interventional Magnetic Resonance (IMR) scanner. The major novelty of this system is the architecture of its central magnet. Unlike the conventional closed-bore MRI magnet, the superconductive 0.5 Tesla magnet of the IMR scanner has an open configuration, built according to a “double doughnut” design. In addition to preventing patient claustrophobia and anxiety, this allows easy access to the patient during imaging.

The IMR system is designed with built in optical, LED based, tracking facilities, as described in section 1.4.2. A camera bar comprising infra-red detectors is built into the IMR unit and a hand-held device, to which LED markers are attached, is supplied. During imaging, the co-ordinates of the LED markers on the hand held device are tracked in real-time. These are registered with the acquired image without the need of any marker based fixation². Hence, the position of the tip of a probe or tool attached to the device is determined automatically. A different tracking system is also available which utilises direct MR signal based localisation [DSD93]. This works by incorporating one, or more, small RF coils into the tips of tools, which are then detected by the actual MR system in real-time. Although

²This is because the position of the field of view in which the image data is acquired is known in relation to the scanner unit and hence to the tracking tools.

this technique requires the tools to be especially modified, it is superior to the LED based approach if the tip of a curved or floppy tool is to be tracked.

IMR imaging facilitates all the clinical applications detailed in sections 1.3 and 1.4. In addition, due to the open nature of the magnet, it is possible to image a subject in non-supine positions. An even more significant breakthrough of the IMR system is its ability to simultaneously acquire image data and to allow clinicians access to the patient. This is particularly advantageous for image guided procedures since it enables accurate MR image guidance even through regions whose anatomy may change during the procedure.

1.6 3D Visualisation Methods: Speed Limitations

Conventional methods do not exist which enable the entire 3D visualisation process to be performed, from start to finish, in real-time. Although rendering techniques have been developed which are capable of extremely fast realisations, this is not the case for the earlier stages of the process. MR acquisition techniques are not yet sufficiently sophisticated to allow 3D volume data to be acquired in less than several minutes. The pre-processing stage is even more problematic. In particular, the classification and segmentation steps are extremely computationally involved. Furthermore, user interaction is invariably required.

The slow and user-dependent nature of current 3D visualisation techniques is a serious obstacle to their practical applicability. Although the benefits of creating a virtual 3D environment are substantial, as argued in section 1.3, clinicians do not have the time to spend long periods in front of a computer console preparing 3D data for subsequent rendition. In addition, slow methods preclude the possibility of simultaneous acquisition and 3D visualisation of IMR data during interventional procedures.

1.7 Thesis Objectives

The main objective of the research presented in this thesis is to develop fast and automatic techniques for 3D visualisation of MRI data in order to overcome the aforementioned limitations of existing approaches. The major focus of the study is on devising fast classification and segmentation methods which can be applied automatically to both conventional and interventional 3D MR image data.

A long term objective of the project is to enable MR image data to be visualised in three dimensions, from start to finish, in real-time.

1.8 Thesis Outline

The work presented in this thesis, excluding this introduction, is divided into the following chapters:

Chapter 2: Image Segmentation. An extensive literature review is presented on the different approaches to image segmentation. This is followed by a discussion on the pros and cons of applying these methods to MRI data. A brief introduction to the techniques described in chapters 5 and 6 is then detailed.

Chapter 3: 3D Rendering Techniques. Methods for surface and volume rendering are described. A comparison of the different techniques is then presented.

Chapter 4: MR Image Acquisition. A preliminary study is presented into acquiring 3D MR image data for the ultimate purpose of 3D visualisation, from both conventional MR and IMR scanners.

Chapter 5: Fast Fuzzy Classification of MR Image Data. A novel method for fast fuzzy classification of MRI data is described. In addition, a new “colour blending” approach for displaying the fuzzy classified data is detailed. Results are then compared with those obtained from alternative, conventional, techniques.

Chapter 6: Adaptive Refinement of Classified MRI Data. An original approach to the automatic refinement of fuzzy classified data (resulting from the method described in chapter 5) is developed. This enables accurately defined anatomical regions to be extracted automatically from both 2D and 3D data.

Chapter 7: Fast 3D Visualisation Techniques: Performance and Applications. The efficacy of applying 3D rendering techniques to data, which has been acquired and processed using the methods described in chapters 4-6, is investigated. Clinical applications are then explored.

Chapter 8: Final Discussion, Conclusions and Future Work. The main results from chapters 4-7 are reviewed and final conclusions are discussed. Suggestions for future research are then presented.

Appendices which include a discussion on MRI physics and some detailed mathematics may be found at the end of the thesis. These are followed by a comprehensive bibliography of the citations which are referred to throughout the thesis.

Chapter 2

Image Segmentation

2.1 Introduction

In this chapter, a comprehensive review of the different techniques for image segmentation is presented. The reader is also referred to [BHC93, CVC⁺95, HS85, PP93]. The chapter concludes with an appraisal of the various methods in relation to MRI data.

Before detailing any of these approaches, it is important to clarify the meaning of the term *segmentation* and to understand why it is such a useful and important image processing task.

A digital image is made up of a uniformly spaced grid of either 2D pixels or 3D voxels. Let us refer to each individual pixel or voxel as an image *element*, i_k , where k indexes the position of the element within the image. If there are n elements within an image, the entire image may be defined as the set I , where

$$I = \{i_1, i_2, \dots, i_n\} \quad (2.1)$$

When an image is segmented, in the conventional sense, the set I is partitioned into m *hard*¹ regions, such that

$$I = R_1 \cup R_2 \cup \dots \cup R_m \quad \text{and} \quad R_j \cap R_k = \emptyset \quad \forall j, k: 1 \leq j, k \leq m \quad (2.2)$$

¹See section 2.2.2 for a discussion on hard and fuzzy sets.

Each region R_j ($1 \leq j \leq m$) is a set of elements which are spatially connected to one another in the image grid, and $m < n$. Alternatively, the notion of *fuzzy* connectivity may be used to define *fuzzy* regions in an image [US96]. This will be discussed later in section 2.3.4.

The resulting segmented region data is of a higher level than the original image data. Every region in the segmented image represents a particular object within the original image. For example, in segmented medical image data, each region may correspond to a specific anatomical structure. It should be noted that an image may be segmented to different extents, depending on the application. Let us consider, as an example, a medical image of the head. We may want to segment the brain from its surrounding structures to produce an image containing two regions, i.e. brain and surrounding. Alternatively, we may wish to segment the actual brain into regions of grey matter and white matter, and so on.

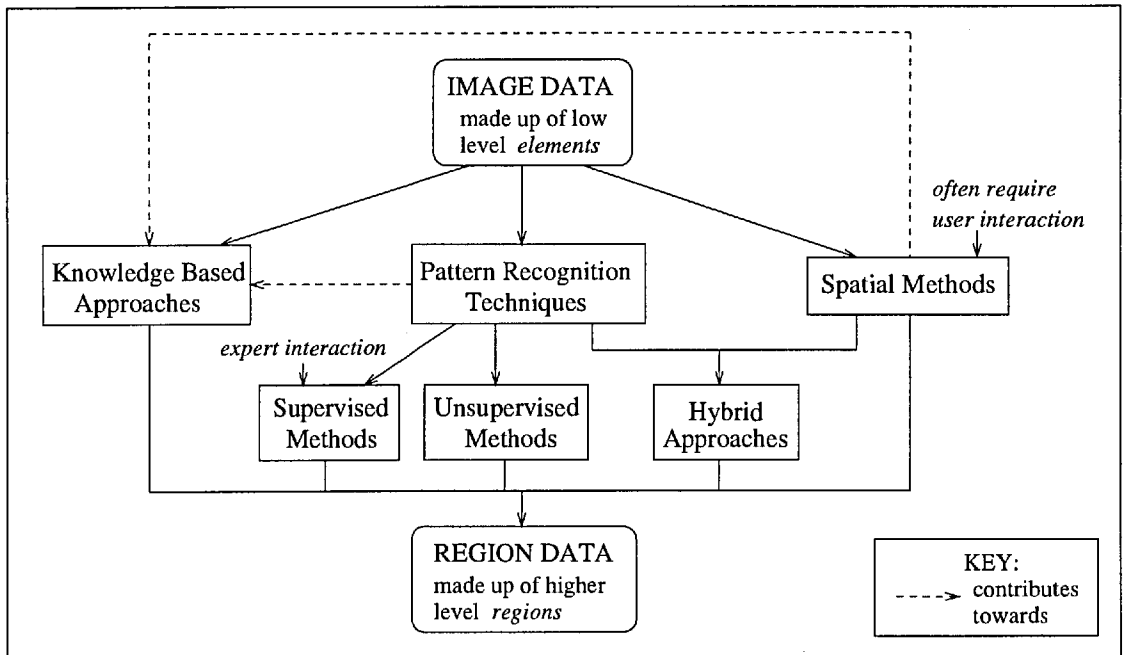


Figure 2.1: Different approaches for segmenting image data into region data.

There is a wide range of potential post-processing applications which may be performed on segmented image data. These may range from object recognition,

object analysis, e.g. computing the shape or size of an object, to 3D rendering. For any of these applications, the process of segmentation is a necessary first stage.

There is no absolute and universal approach to image segmentation. Instead, a specific segmentation technique is adopted according to the type of image and the application. Figure 2.1 shows a flow diagram highlighting the different possible strategies. These range from laboriously outlining region boundaries with a mouse to more sophisticated semi-automated and fully automated procedures. Often a combination of techniques yields the best results.

The following sections describe the theory and methodologies required for the different approaches outlined in figure 2.1.

2.2 Pattern Recognition Techniques

Pattern recognition (PR) systems, when applied to image data, perform a process called image element *classification*. This assumes that there are a certain number of region types or classes within the image, each class referred to by a specific *label*. For example, in a medical image, each class might correspond to a different tissue type. As a result of the classification process, each image element is either uniquely labelled, or is assigned *fuzzy* membership to more than one class (see section 2.2.2). The classified data may then be transformed into higher level regional data, thus producing a segmented image, by performing ^{connected-}component labelling², as described in section 6.2.2. However, for certain applications, e.g. volume rendering (see section 3.2.1), the classified data may be useful in its own right.

In this section, the fundamental theory required for PR systems, used to classify image data, will be described. This will then be followed by an extensive review of the various methods which may be adopted in a pattern recognition approach to

²If a fuzzy classification scheme is adopted, this step must be preceded by data *defuzzification*. This is described in section 6.2.1.

image segmentation.

2.2.1 PR Systems and Image Segmentation

An image is made up of a set of elements, as defined in equation 2.1. Pattern recognition systems receive *object* data as input, where each element or group of elements is referred to as an object. The object data may be defined by the $(p \times n)$ matrix $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, where n is the total number of objects within the image. The k 'th object, is represented by the feature vector $\mathbf{x}_k$, and x_{jk} is the j 'th characteristic or *feature* associated with object k , each object comprising p features. The features may be grey scale values of the original image elements, or may be different measures of image *texture* [DKS90, Gal75, HSD73, RD93, VDO85]. For MR images, they may also be the grey scale values associated with different MR parameters such as T1, T2 or proton density, etc..³ The choice of features strongly affects the outcome of the segmentation and in many cases it is necessary to first decorrelate and reduce the feature space. This will be discussed further in section 2.2.5.

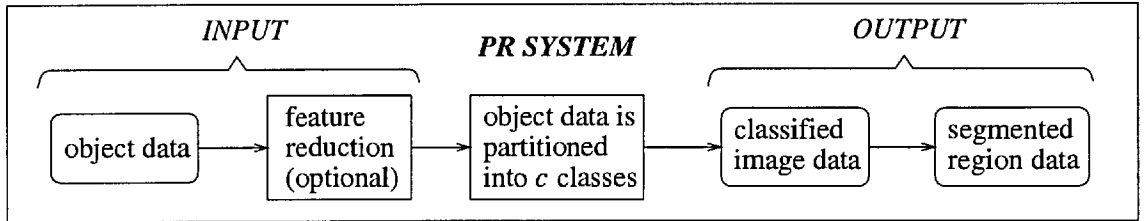


Figure 2.2: Image segmentation using a pattern recognition scheme.

The pattern recognition system partitions the object data into c clusters or classes. Using the partitioned data, each image element is *labelled*, i.e. assigned a label, in either a hard or fuzzy sense, according to the class in which its corresponding object belongs. The entire process is summarised in figure 2.2.

³For a discussion on MR contrast parameters, see appendix A, sections A.1.3 and A.3.1.

Before launching into a discussion on the different types of PR system, the mathematical framework required for these techniques will be described.

2.2.2 Hard and Fuzzy Sets: A Framework for PR Systems

Let $X = \{x_1, x_2, \dots, x_n\}$ be a set of finite objects. X may be partitioned into two different types of subset.

Conventional subsets, which are referred to as hard or crisp, will either contain or not contain objects of X . Let $P(X)$ be the power set of X , i.e. the set of all hard subsets of X , and let $A \in P(X)$ be any hard subset of X . The function $u_A : X \rightarrow \{0, 1\}$ defined by

$$u_A(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{otherwise} \end{cases} \quad (2.3)$$

is the characteristic function of hard subset $A \subset X$.

Fuzzy subsets may not necessarily uniquely contain objects of X . The theory for these was originally introduced by Zadeh [Zad65] as a way to represent vagueness in everyday life. Let $A \in P_f(X)$ be any fuzzy subset of X , where $P_f(X)$ is the set of all fuzzy subsets of X . The fuzzy subset A may be defined by its membership function

$$u_A : X \rightarrow [0, 1] \quad (2.4)$$

If $u_A(x) = 0$, the object x does not exist within subset A , whereas if $u_A(x) = 1$, it is fully contained within subset A . If $u_A(x)$ takes a value between 0 and 1, this signifies the degree of membership of x in subset A . For a more detailed discussion on fuzzy set theory, the reader is referred to [Bez81, BP92, Kan82].

We are now in a position to define a mathematical framework within which pattern recognition systems may be implemented. Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ be a finite set of object data; V_{cn} be the set of real $c \times n$ matrices; c be an integer

denoting the number of partitions in the data set, $2 \leq c < n$. X may be divided into either a hard or a fuzzy c -partition as follows:

Hard c -partition space for X is the set

$$M_c = \left\{ U \in V_{cn} \mid u_{ik} \in \{0, 1\} \forall i, k; \sum_{i=1}^c u_{ik} = 1; \quad 0 < \sum_{k=1}^n u_{ik} < n \forall i \right\} \quad (2.5)$$

where u_{ik} is the membership function of the k 'th member of the i 'th subset of the entire hard set. It should be noted that because u_{ik} may only take values of 0 or 1, as originally defined in equation 2.3, only a finite number of U matrices exist in hard partition space, M_c .

Fuzzy c -partition space for X is the set

$$M_{fc} = \left\{ U \in V_{cn} \mid u_{ik} \in [0, 1] \forall i, k; \sum_{i=1}^c u_{ik} = 1; \quad 0 < \sum_{k=1}^n u_{ik} < n \forall i \right\} \quad (2.6)$$

It should be noted that because u_{ik} may take any value between 0 and 1, as originally defined by equation 2.4, an infinite number of U matrices exist in fuzzy partition space, M_{fc} .

Each U matrix solution, satisfying equations 2.5 or 2.6, corresponds to a unique partition of the image data. The objective of any PR system is to obtain the best possible solution of U such that an optimal segmentation results.

The following sections describe the different approaches which may be adopted in order to partition image data optimally within either a hard or fuzzy⁴ framework.

2.2.3 Supervised Methods

Supervised methods are so named because they require the intervention of an expert operator or *supervisor*. The supervisor examines the image data visually and

⁴If probabilistic techniques are used, their outcomes may be represented within a fuzzy framework despite the conceptual and philosophical differences between membership values and probabilities. See [BHC93] for a more detailed discussion on this topic.

then labels specific image elements according to their corresponding classes. Thus a training data set is formed which is used to determine the characteristics of a classifier. This is then used to classify as yet unlabelled *test* data automatically.

Parametric Schemes

Parametric schemes seek to match the object data with a statistical distribution whose shape is governed by a parameter vector, θ . This distribution, $P(\mathbf{x} \mid \theta)$, is generally referred to as the mixture model and is defined as

$$P(\mathbf{x} \mid \theta) = \sum_{i=1}^c \alpha_i P(\mathbf{x} \mid \theta_i) \quad (2.7)$$

where:

α_i is the probability of any object belonging to class i ;

c is the number of different classes in the data;

$P(\mathbf{x} \mid \theta_i)$ is the probability distribution of class i ;

$\theta = (\theta_1, \dots, \theta_c)$ is the parameter vector which defines the shape of the c probability distributions.

The most popular choice of distribution for the mixture model is the Gaussian⁵. This is defined, for the i 'th class, as

$$P(\mathbf{x} \mid \theta_i) = (2\pi)^{-p/2} |\Psi_i|^{-1/2} \exp \left[-\frac{1}{2} (\mathbf{x} - \mu_i)^T \Psi_i^{-1} (\mathbf{x} - \mu_i) \right] \quad (2.8)$$

where:

$\theta_i = \{\mu_i, \Psi_i\}$ is the set of parameters which define the distribution of class i ;

μ_i is the mean of class i ;

Ψ_i is the co-variance matrix (see [Fuk90]) of class i ;

p is the number of features in the object data;

$|\cdot|$ denotes the determinant of a matrix.

⁵This is the most obvious choice, in particular when there is a large amount of data, due to the central limit theorem. This states that as the number of observations, for any type of data, tends to infinity, their distribution tends to be Gaussian.

The easiest way, in general, to compute the values of the parameter vector θ is by maximum likelihood (ML) estimation [Fuk90, RW84, The89]. This involves forming likelihood functions from the training data and maximising them. If a Gaussian mixture model is assumed, the resulting solutions (see [Rob95, Trå91] for proof) of the log-likelihood equations are:

$$\hat{\mu}_i = \frac{\sum_{k=1}^n p(i | \mathbf{x}_k) \mathbf{x}_k}{\sum_{k=1}^n p(i | \mathbf{x}_k)} \quad \hat{\Psi}_i = \frac{\sum_{k=1}^n p(i | \mathbf{x}_k) (\mathbf{x}_k - \hat{\mu}_i) (\mathbf{x}_k - \hat{\mu}_i)^T}{\sum_{k=1}^n p(i | \mathbf{x}_k)} \quad (2.9)$$

where:

$\hat{\mu}_i$ and $\hat{\Psi}_i$ are the ML estimates of the mean and co-variance matrix for class i respectively;

$p(i | \mathbf{x}_k)$ is the posterior probability of the k 'th object belonging to class i , given the values of its feature components;

n is the total number of objects in the data.

The estimated values for the parameter vector, θ , are computed directly from the *training* data alone, using equation 2.9. The required values for $p(i | \mathbf{x}_k)$ are deduced from the particular label of each *training* data object. The values of α_i are simply the proportions of training objects existing within each class. The *trained* classifier, i.e. the mixture model with computed parameter values, may then be used to classify the remaining *test* data. As a result of Bayes's theorem, the posterior probability, $p(i | \mathbf{x}_k)$, for each *test* data object is determined as follows:

$$p(i | \mathbf{x}_k) = \frac{\alpha_i p(\mathbf{x}_k | \theta_i)}{\sum_{j=1}^c \alpha_j p(\mathbf{x}_k | \theta_j)} \quad (2.10)$$

The computed posterior probability values of all the object data may be represented in the form of the $(c \times n)$ matrix $P = [p(i | \mathbf{x}_k)]$, where $1 \leq i \leq c$ and $1 \leq k \leq n$. The entries of this matrix may be considered as membership values with the overall matrix providing a fuzzy partition of the object data.

Non-parametric Schemes

Non-parametric methods do not require any knowledge or assumptions about the statistical properties of the data. One such approach is the adoption of the *k-nearest neighbour* (*k*-nn) rule [BHC93]. This method relies on a large number of labelled samples from each class in the training set. Each object in the test set is then visited and its *k* nearest labelled neighbours are checked. The *k* nearest neighbours are selected according to some distance measure, usually Euclidean, in *p*-dimensional feature space, $\mathbb{R}^p$. An appropriate voting scheme, often based on a straightforward majority, is then chosen to decide according to which class the test object of interest should be labelled. This can be performed within either a hard or fuzzy framework. Details of appropriate algorithms may be found in [BHC93, Fuk90].

Feed Forward Neural Network Methods

Artificial Neural Networks are composed of many non-linear computational elements operating in parallel and arranged in patterns attempting to simulate their biological equivalent. The most popular classifier nets used in image segmentation are multi-layer feed-forward neural networks [BHC93]. The networks comprise nodes or neurons which are interconnected. Associated with each interconnection, there is a value known as a *weight*. During the training phase of the network, labelled training data is fed into the network and the values of weights are updated iteratively until they converge sufficiently. The trained network is then used to classify the test data [Pao89]. Feed Forward Back Propagation (FF/BP) networks have been reported to give reasonable results with MR image data [BHC93]. However, they tend to be extremely sensitive to the test data. Feed Forward Cascade Correlation (FF/CC) networks self configure themselves. This avoids having to find the optimum configuration experimentally, which can often be time consuming. Hall *et al* [HBC⁺92] have found that FF/CC networks yield better results than FF/BP networks on MRI data.

2.2.4 Unsupervised Methods

Unsupervised techniques partition the data without the help of any pre-classified training data. Often referred to as clustering techniques, they search through the data and cluster it automatically into c classes. In general the number of classes, c , must be known in advance. However, some approaches have been developed which attempt to compute the optimum number of classes that exist in the data automatically. These will be discussed, in brief, at the end of this section.

Parametric Schemes

As for supervised parametric schemes, unsupervised approaches attempt to match the object data with a statistical distribution, as defined in equation 2.7. However, in the absence of training data, values for the ML parameter estimates cannot be computed directly. Instead, a numerical scheme must be adopted. One of the most popular approaches is known as the Expectation-Maximisation (EM) algorithm [AW80, DLR77]. Initial guesses are made for the posterior probabilities, $p(i | \mathbf{x}_k)$, where $1 \leq i \leq c$ and $1 \leq k \leq n$. First round estimates are then made for the mixture model parameters (as for M -step below). The algorithm then iteratively repeats the Expectation (E) and Maximisation (M) steps. The E and M steps are computed as follows:

E -step: The posterior probabilities, $p(i | \mathbf{x}_k)$, are computed with equation 2.10 using the parameter estimates from the previous M -step.

M -step: The values for α_i are

$$\alpha_i = \frac{1}{n} \sum_{k=1}^n p(i | \mathbf{x}_k) \quad (2.11)$$

The parameter values are computed using ML estimates. If a Gaussian mixture is assumed, the ML estimates for the mean and co-variance matrix may be found using equation 2.9.

The sequence of alternate E and M steps continues until a suitable stopping criterion is satisfied.

Non-parametric Schemes: Hard and Fuzzy c -Means Clustering

Non-parametric clustering methods do not attempt to match the data with any statistical distributions. Instead, objective functions are defined to guide the clustering process. An optimal partition results when the objective function has been minimised. The following techniques are the most commonly used in image segmentation:

1. *Hard c -Means (HCM)*: This algorithm minimises an objective function which measures the overall within-group sum of squared errors (WGSS) between the data and their cluster centroids. The objective function $J_W : M_c \times \mathbb{R}^{cp} \rightarrow \mathbb{R}^+$ is defined as

$$J_W(U, \mathbf{v}) = \sum_{k=1}^n \sum_{i=1}^c u_{ik} (d_{ik})^2 \quad (2.12)$$

where:

n is the number of objects;

$d_{ik} = \|\mathbf{x}_k - \mathbf{v}_i\|$ is the distance between data point $\mathbf{x}_k$ and cluster centroid $\mathbf{v}_i$,

$\|\cdot\|$ being an appropriate vector norm [GL89] on $\mathbb{R}^p$;

$U = [u_{ik}] \in M_c$ is a hard c -partition of X .

Details of an iterative algorithm which partitions X such that $J_W(U, \mathbf{v})$ is minimised may be found in [Bez81].

2. *Fuzzy c -Means (FCM)*: Bezdek [Bez81] defined the fuzzy objective function $J_m : M_{fc} \times \mathbb{R}^{cp} \rightarrow \mathbb{R}^+$ as

$$J_m(U, \mathbf{v}) = \sum_{k=1}^n \sum_{i=1}^c (u_{ik})^m (d_{ik})^2 \quad (2.13)$$

where:

n is the number of objects;

$d_{ik} = \|\mathbf{x}_k - \mathbf{v}_i\|$ is the distance between $\mathbf{x}_k$ and fuzzy cluster centroid $\mathbf{v}_i$ ($\mathbf{v}_i$ is defined in [Bez81]);

$U \in M_{fc}$ is a fuzzy c -partition of X ;

$m \in (1, \infty)$ is the fuzzy weighting exponent.

The solution which results from minimising the above defined objective function, $J_m(U, \mathbf{v})$, yields an optimal fuzzy segmentation. The FCM algorithm achieves this end and is described in [Bez81]. It should be noted that the value for m , the fuzzy weighting exponent, must be decided in advance. It has been suggested [PB95] that choosing m in the interval $[1.5, 2.5]$ provides optimal performance of the FCM.

One of the main limitations of the FCM algorithm is that it requires a huge amount of computation if there is a large amount of data to be processed. This is invariably the case for image segmentation. Cannon *et al* [CDB86] devised an approximated implementation (AFCM) in order to reduce the required processing time. This exploits a look up table approach for computing distances and exponentiation. The AFCM has been reported [CDB86] to perform up to six times faster than the FCM whilst producing similar, although not identical, results.

Bensaid *et al* [BBH⁺92, BHBC96] devised a partially supervised version of the FCM which they claim improves its performance. This utilises training data in guiding the clustering process.

Cluster Validity

One of the difficulties with unsupervised techniques is determining the number of clusters which exist within the data. There are various criteria which have been used which, given results from a range of partitions at different values of c , will determine which is the most valid solution. Amongst some of these measures are the partition coefficient, classification entropy, the proportion exponent, the separa-

tion coefficient [Bez81], the fuzzy hyper-volume measure [GG89] and data likelihood [Rob95]. However, there may be more than one value of c which give a good partition. Furthermore, the optimal number of clusters may depend on the application in question.

2.2.5 Feature Reduction

As mentioned in section 2.2.1, the choice of features strongly affects the outcome of any pattern recognition based image segmentation. In order to avoid including any features which may degrade the quality of the segmentation, it is important to select only those which contribute to a good partitioning of the object data. A related problem is that features may be highly correlated in the original space. Hence, before selecting the best features, it is important to first decorrelate the feature space.

Feature decorrelation involves performing a multi-dimensional rotation of the feature space. This may be achieved by the Karhunen-Loève expansion [Fuk90, The89].

Feature rejection may then be performed by measuring the distances between sets of features [CA79]. One of the most popular distance measures is the Bhattacharyya distance. A large distance implies a low classification error. Thus only those features with a large Bhattacharyya distance [CA79, Fuk90, The89] should be retained.

2.3 Spatial Methods

In the previous section, pattern recognition techniques, for image segmentation, were described. One of their main limitations is that they consider only the feature space of the object data regardless of their spatial location in the image. However, if there

are any spatial inhomogeneities between the object data in the image, errors will result. Methods which consider the spatial interaction between image elements may avoid this problem. Throughout the rest of this section, various different approaches to performing image segmentation using spatial information will be described. For the sake of simplicity, the following techniques will be described for 2D images only⁶.

2.3.1 Boundary Identification

Any region in an image can be clearly defined by its surrounding boundary. Thus, by identifying the regional boundaries which exist within an image, the image may be segmented. The simplest approach to boundary identification is for an operator to outline the desired boundaries manually with a mouse. However, this can be extremely time intensive and is not necessarily very accurate. Hence, any process which is able to improve on this, is to be desired. Different approaches which detect boundary contours will now be described.

Edge Detection and Linking

Edge detection techniques are able to detect the edges corresponding to regional boundaries automatically. The detected edges often need to be linked in order to extract complete boundary contours. Some of the most popular methods for edge detection and edge linking will now be reviewed.

1. *Edge Detection:* The most basic approach to edge detection is to apply an appropriate gradient operator. The earliest and simplest operator to be applied to images to extract edge information was designed by Roberts [Rob65]. This comprises two (2×2) masks which are convolved with the image to calculate the grey scale gradient across the edge in two diagonal directions. A more popular strategy is to compute edges in horizontal and vertical directions by using two (3×3) masks.

⁶In general, when it is possible to implement a technique directly in 3D, a straight forward extension of the 2D theory is possible.

Mask sets of this form, but each with different values, have been devised by Prewitt [Pre70], Sobel [Sob70] and Frei and Chen [FC77]. Kirsch [Kir71] and Robinson [RR77] have developed more complex gradient operators in the form of seven (3×3) compass masks each detecting an edge at a different angle and Nevatia and Babu [NB80] have formulated a set of six (5×5) compass masks.

The major limitation of gradient edge detectors is their general sensitivity to noise. In addition, they tend to produce thick edges, i.e. more than one pixel wide, at step transitions, thus reducing the accuracy of boundary definition. This may be avoided by utilising second derivative information. The second derivative is defined by the Laplacian function:

$$\nabla^2 I = \left(\frac{\partial^2}{\partial r^2} + \frac{\partial^2}{\partial c^2} \right) I = \frac{\partial^2 I}{\partial r^2} + \frac{\partial^2 I}{\partial c^2} \quad (2.14)$$

where r and c are the indices for the rows and columns respectively, of the image array, I . Operators which detect the zero-crossings of the Laplacian, which correspond to the maximal gradient values only, are able to find the actual edge locations. Details of appropriate masks to use for computing the Laplacian may be found in [GW92, HS92].

The main limitation of computing the Laplacian alone is that it is hypersensitive to noise. Marr and Hildreth [MH80] therefore suggest incorporating a Gaussian smoother. The resulting operator is called the Laplacian of Gaussian (LoG) zero-crossing edge detector. However, the LoG operator suffers from poor localisation, particularly for rounded edges with large curvature, and from false edge detection.

A different approach to edge detection involves fitting the underlying intensity function, i.e. grey scale values versus spatial co-ordinates, of the image, with a parametric surface. The edges of interest can then be extracted by measuring the gradient magnitude, orientation and second derivative zero-crossing location, of the fitted surface. Haralick [Har84] devised a cubic bivariate polynomial surface, the

cubic facet model, to approximate the sampled intensity values. Nalwa and Binford [NB86] fitted a 1D tanh function and a quadratic along the gradient direction to find the zero-crossing location. Huertas and Medioni [HM86] used a quadratic facet model to localise edge points with subpixel accuracy.

2. *Edge Linking*: Edge detection techniques alone are rarely able to identify complete regional boundaries. Invariably, sections of the boundary have poorly defined edges which are not picked up through edge detection. Thus, edge linking is usually required to complete the segmentation process.

One of the simplest approaches for linking edge points is to analyse the characteristics of pixels in a small neighbourhood around every pixel in the image which has undergone edge-detection. Assuming that a neighbouring pixel satisfies criteria based on features such as edge magnitude and direction, the two pixels are linked [GW92].

A more advanced approach is the matching of an active contour model to the edge data. Different approaches to achieve this will now be discussed.

Active Contour Models

Active contour models seek to fit a closed contour to the edges of a region boundary. One of the most popular models is known as a *snake*, originally devised by Kass *et al* [KWT87]. This is an energy minimising spline which locks onto nearby edges, localising them accurately. Cohen [Coh91] has developed the snake model to give more stable results. This treats the snake like a balloon which no longer needs to be placed near the solution in order to converge. More recently, Storvik [Sto94] has employed a Bayesian approach to obtain an accurate contour which is based on stochastic sampling. This allows much more general models than the original snake to be applied.

One of the limitations of the active snake, when used to localise objects in a

global fashion, is that the image may be very detailed and complicated. This may result in an energy function which has too many local minima, thus limiting the likelihood of obtaining an accurate global solution. Matalas [Mat96] has therefore proposed a more interactive method which first fits B-spline curves to all connected edges in the image. The user then selects pairs of curves which he wants to link. These are grown until they meet each other. The shape of each newly grown curve is then refined by minimising the energy of its associated B-snake. Unlike the edge linking approach mentioned above, this method is very robust and may connect edges accurately which are quite distant from each other. However, the main disadvantage of this technique is that it is extremely user intensive.

Falcão *et al* [FUSH96] have devised a completely different approach to the snake model. This requires any two points on the boundary to be selected by a user. The best boundary contour containing them is the minimum cost path, expressed as a sequence of oriented edge pixels, between them. This is computed via dynamic programming. Recently, they have extended this approach to extract the contours which define a 3D object inside a 3D data set[FU97].

2.3.2 Morphological Techniques

Mathematical morphology is a shape based approach which can be used to perform powerful tasks on digital images. One such application is image segmentation. Before reviewing morphological applications in image segmentation, we will first briefly describe some of the main operators used in mathematical morphology. Detailed texts on the subject may be found in [Ser82, HSZ87].

A. Basic Definitions

The most important morphological operations are *dilation* and *erosion*. These involve performing set operations on the image with what are known as *structuring*

elements. The operations may be defined for general N -space ($\mathbb{R}^N$) binary images as follows:

Definition 2.1 (Dilation) Let A and B be subsets of $\mathbb{R}^N$. The dilation of A by B is denoted by $A \oplus B$ and is defined by

$$A \oplus B = \{c \in \mathbb{R}^N \mid c = a + b \text{ for some } a \in A \text{ and } b \in B\} \quad (2.15)$$

Definition 2.2 (Erosion) The erosion of A and B is denoted by $A \ominus B$ and is defined by

$$A \ominus B = \{c \in \mathbb{R}^N \mid c + b \in A \text{ for every } b \in B\} \quad (2.16)$$

The set A corresponds to the image and B to the structuring element. Dilation performs a vector addition of the set elements and causes the binary regions in the image to grow or *dilate*. Erosion, its morphological dual, combines the two sets via vector subtraction and causes the binary regions to shrink or *erode*.

Combinations of dilation and erosion in a specific sequence affect the image in a unique way. The most popular combinations are known as *opening* and *closing*.

Definition 2.3 (Opening) The opening of image B by structuring element K is denoted by $B \circ K$ and is defined as $B \circ K = (B \ominus K) \oplus K$.

Definition 2.4 (Closing) The closing of image B by structuring element K is denoted by $B \bullet K$ and is defined as $B \bullet K = (B \oplus K) \ominus K$.

Opening causes any regions which are slightly connected to open and closing fills in any gaps or holes.

The above described operations are specific to binary images. With grey scale images, variants of these operations may be applied. Unlike their binary counterparts, these operate on the shape of the intensity surface defined by the values of the

image elements therein. As expected, these operators are referred to as grey scale dilation and erosion. Grey scale opening and closing operations are defined in the same way as their binary counterparts. For further details on grey scale operations, the reader is referred to [Ser82, HSZ87].

B. Applications in Image Segmentation

There is no formal approach to image segmentation when applying morphological operators. The correct combination of operations, as well as structuring elements used, is crucial to a successful outcome.

Matsopoulos and Marshall [MM94] describe a specific combination of both grey scale and binary morphological operations for the purpose of segmenting the head from fetal Ultrasound images. This involves first applying grey scale morphological operations in order to reduce the amount of noise in the image. The resulting image is then binary thresholded and a sequence of binary operations produces the segmented output. Höhne and Hanson [HH92] propose an interactive approach for 3D segmentation of CT and MRI volumes. This also involves first thresholding the data to produce a binary image. A sequence of binary operations is then performed on the entire 3D volume. However, after each stage, the resulting segmented region of interest is isolated from the remaining data and surface rendered (see section 3.1). Depending on the result, the operator chooses a further combination of operations until a satisfactory result is produced.

A major drawback of morphological techniques is the huge amount of computation required. Liang and Wong [LW93] have developed a hierarchical data structure to speed up the processing. They report a substantial reduction in computation when applying the structure and prove that their approach gives identical results to a conventional implementation.

2.3.3 Region Directed Methods

In this section, segmentation methods which delineate an image into region data directly are described.

Region Growing

Region growing, in its simplest sense, is the process of joining neighbouring image elements, or collections of elements, into larger regions. An entire image may be segmented by first dividing it into many small primitive regions. Adjacent regions are then iteratively merged if appropriate homogeneity criteria are satisfied, subject to predefined parameters or thresholds. Alternatively, a specific region may be grown by providing a user selected seed point from within the region of interest. A review of some of the basic approaches may be found in [Zuc76].

Chang and Li [CL94] propose an approach which does not require predefined global thresholds. Instead, a merge criterion is applied which guides the region growing process by assessing region homogeneity adaptively from region feature distributions. Adams and Bischof [AB94] have developed a method which is free of any tuning parameters. However, this requires the input of a number of seeds in order to control the formation of regions into which the image is segmented. Le Moigne and Tilton [LT95] have shown how by considering edge detected image information, the process of region merging may be refined.

Region Splitting

Region splitting techniques are the opposite of the above described region merging methods. They view the entire image as an initial segmentation and then successively split each current segment into quarters if the segment is not sufficiently homogeneous. In general this does not produce a useful segmented output. However, such methods may be used as a pre-processing stage to isolate the interesting areas

in an image from uninteresting background regions [Zuc76]. This enables further processing to proceed more easily.

Split and Merge Methodologies

Split and Merge techniques utilise both region growing and region splitting approaches in segmenting digital images. The overall approach was originated by Horowitz and Pavlidis [HP76] and works within a pyramidal data structure, or a *quad tree*. At the top level (root), there is only one region, R_0 , namely the entire image. At each level the regions are square or rectangular and are referred to as nodes. Progressing down the tree, each *parent* node of the higher level splits into four equal sized *children* nodes.

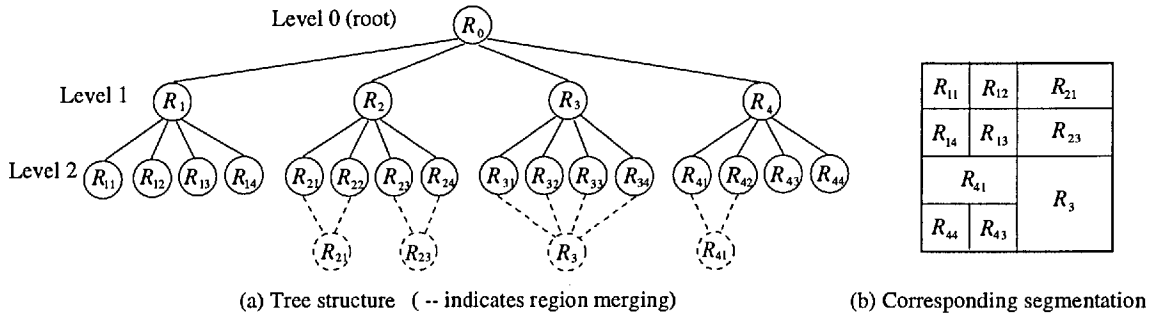


Figure 2.3: An example of the resulting segmentation after performing the first two stages of a split and merge algorithm

The process of segmentation is initiated by considering all the nodes of an appropriate tree level as the initial *cutset*. The following processing steps are then performed in sequence:

1. *Splitting Phase*: Within each node at the highest level of the initial cutset, an appropriate homogeneity test is performed. If the result is false, i.e. the region is not homogeneous, the region is split into four child nodes, referred to as leaves. Each new leaf node is in turn considered for further splitting and the process continues until no more false results occur.

2. *Merging Phase*: If on any branch, all the quad siblings are leaves, a uniformity test is performed to see whether they should be merged. The example shown in figure 2.3 demonstrates the effect of merging regions R_{41} with R_{42} (as well as R_{21} with R_{22} , etc.).
3. *Grouping adjacent nodes in different branches*: A homogeneity test is performed on any quad tree leaves which are spatially adjacent but reside on different branches. For example, regions R_3 and R_{41} in figure 2.3 will be tested for uniformity.
4. *Eliminating small regions*: In this final and optional stage, all regions smaller than a determined size are tested for homogeneity with all adjacent regions and merged with the one giving the highest homogeneity confidence.

Chen and Pavlidis [CP80] describe homogeneity tests which are appropriate for the first two stages of the algorithm. Chen and Lin [CL91] describe a method for computing all required parameters automatically.

One of the biggest limitations of the split and merge algorithm is the fact that the resultant regions are defined by very rough and often not very accurate boundaries. Pavlidis and Liow [PL90] have therefore developed a method which first over-segments the image using a conventional split and merge algorithm and then removes and modifies existing boundaries by considering edge information. Deklerck *et al* [DCB93] present a method which they call *edgmentation*. This deviates from the conventional split and merge technique by not splitting the image according to its quad tree but rather according to edge information.

2.3.4 Fuzzy Region Extraction

We have so far discussed only methods which delineate an image into hard regions. The concept of fuzzy sets was introduced in section 2.2.2 for the purpose of fuzzy *classification* of object data. However, the resulting data cannot be directly converted into fuzzy regional data in the spatial domain. Any methodology which is

able to represent poorly defined structures in an image as fuzzy regions, is likely to remove some of the inherent limitations of conventional hard region definition and produce a more accurate and realistic segmentation.

In order to develop appropriate methods for the extraction of fuzzy regions from digital images, a basic mathematical framework must first be defined. This should address issues such as: the definition of fuzzy objects; fuzzy connectivity; fuzzy boundary definition; efficient algorithms for fuzzy object extraction; etc.. There is only a very limited amount of published material which addresses any of the above issues [Blo93, Ros79, Ros91]. However, Udupa and Samarasekera [US96] have recently presented a theoretical framework which allows fuzzy object definition. In addition, they have described algorithms which are able to extract fuzzy objects from image data. They have successfully demonstrated the efficacy of their algorithms on a range of medical image data.

Although an approach for fuzzy object definition has been proposed for fuzzy region extraction [US96], the required theory is still in its infancy. The accuracy and realism of the extracted fuzzy regions remains to be assessed. Furthermore, the theory has not yet been developed to allow the definition of more than one fuzzy region in an image in the same way as for hard regions (see equation 2.2).

2.4 Hybrid Approaches

In sections 2.2 and 2.3, segmentation techniques have been presented which work in either feature space or the spatial domain. In this section, methods which classify the image data according to both their feature space characteristics and their spatial interactions, will be described.

2.4.1 Relaxation Labelling

Relaxation labelling works on the premise that the label or class of an object depends on a measure of compatibility of its potential labels with the labels of directly interacting objects. Even distant objects interact with each other as a result of constraint propagation. Normally, in order to simplify the required computation, only adjacent objects are considered to interact directly. Rosenfeld *et al* [RK82] describe how relaxation labelling may be used to segment image data. This involves first deriving estimates of the probabilities of elements belonging to each class by using an appropriate pattern recognition technique. Compatibility coefficients between adjacent image elements are derived and the relaxation process may then be applied. This is an iterative scheme which seeks to maximise the global “confidence” in the labelling.

The disadvantage of relaxation labelling is that it updates the labels using information from the current labelling only. The observed feature vectors are only used to initialise the relaxation network but are ignored during subsequent iterations. This can cause the system to converge to highly undesirable labellings.

2.4.2 Markov Random Field Modelling

Markov random fields (MRF’s) may be used to model spatial context and stochastic interaction among observable quantities. This has led to their application in image segmentation. In this section, the basic fundamentals of MRF theory, and how they may be used for image segmentation, will be presented. An excellent overview of MRF theory and applications in computer vision may be found in [Li95].

A. Basic Definitions

Let us consider a 2D random field which is defined over a finite $N_1 \times N_2$ uniform lattice, S , of image elements or *sites*, defined as

$$S = \{s = (i, j) : 1 \leq i \leq N_1, 1 \leq j \leq N_2\} \quad (2.17)$$

Definition 2.5 (Neighbourhood) A collection of subsets of S , described as,

$$\eta = \{\eta_s : s \in S, \eta_s \subseteq S\} \quad (2.18)$$

is a *neighbourhood system* of S if and only if:

- (i) $s \notin \eta_s$
- (ii) if $s \in \eta_r$ then $r \in \eta_s \forall s, r \in S$.

The neighbour set of s is defined as the set of nearby sites within a radius of R

$$\eta_s = \{s' \in S \mid \text{dist}(s, s')^2 \leq R, s' \neq s\} \quad (2.19)$$

where $\text{dist}(s, s')$ is the Euclidean distance between sites s and s' . We denote the neighbourhood system of order k , as the set $\{(s, \eta^k), s \in S\}$. Figure 2.4 shows neighbour sets of orders 1 to 5. Note that each neighbour set of order k also includes all the sites of lower orders.

5	4	3	4	5
4	2	1	2	4
3	1	(i,j)	1	3
4	2	1	2	4
5	4	3	4	5

Figure 2.4: Neighbour sets of orders 1 to 5.

Definition 2.6 (Clique) A *clique*, c , of the pair (S, η) is a subset of S such that

- (i) c consists of a single site, $c = \{s\}$, or
- (ii) c consists of a pair of neighbouring sites, $c = \{s, s'\}$; or a triple of neighbouring sites, $c = \{s, s', s''\}$, and so on.

Figure 2.5 shows the types of cliques associated with η^2 . The collections of single-

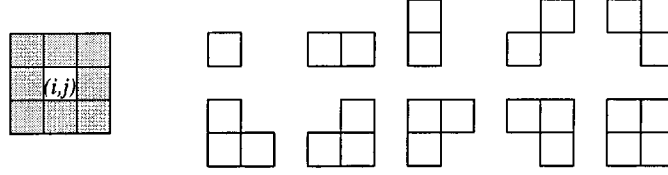


Figure 2.5: The second order neighbourhood set and its associated clique types

site and pair-site cliques may be defined as C_1 and C_2 , respectively, where $C_1 = \{s \mid s \in S\}$ and $C_2 = \{\{s, s'\} \mid s' \in \eta_s, s \in S\}$, and so on. The collection of all the cliques of (S, η) is denoted by $C = C_1 \cup C_2 \cup C_3 \cup \dots = C(S, \eta)$.

Definition 2.7 (Markov random field) Let $X = \{X_s : s \in S\}$ denote any family of random variables indexed by the lattice $S = \{s : 1 \leq s \leq n\}$, $n = N_1 \times N_2$. Let Ω be the set of all possible configurations of x , such that

$$\Omega = \{x = (x_1, \dots, x_n) : x_s \in \Lambda, 1 \leq s \leq n\} \quad (2.20)$$

where Λ is the set of possible labels or classes. For continuous fields, $\Lambda \subset \mathbb{R}^+$, whereas for discrete fields, Λ is the set of K possible labels, i.e. $\Lambda = \{1, 2, \dots, K\}$.

X is said to be a *Markov random field* (MRF), with respect to (S, η) , if

$$(i) \quad P(X = x) > 0 \quad \forall x \in \Omega \quad (\text{positivity}) \quad (2.21)$$

$$(ii) \quad P(X_s = x_s \mid X_{S-\{s\}} = x_{S-\{s\}}) = P(X_s = x_s \mid X_{\eta_s} = x_{\eta_s}) \\ \forall s \in S \text{ and } x \in \Omega \quad (\text{Markovianity}) \quad (2.22)$$

where $x_{S-\{s\}} = \{x_r : 1 \leq r \leq n, r \neq s\}$ and $x_{\eta_s} = \{x_r \mid r \in \eta_s\}$. In other words, the probability of a site having a particular label is dependent on those of its neighbouring sites only. It is always possible to select a sufficiently large η_s so that Markovianity holds. Note: The largest possible η_s is $S - \{s\}$.

Definition 2.8 (Gibbs distribution) A random field $X = \{X_s : s \in S\}$ has a *Gibbs distribution* (GD) or, equivalently, is a *Gibbs random field* (GRF) with respect to η , if and only if its joint probability distribution is of the form

$$P(X = x) = \frac{1}{Z} e^{-U(x)} \quad (2.23)$$

where $U(x)$ is the energy function, defined as

$$U(x) = \sum_{c \in C} V_c(x) \quad (2.24)$$

$V_c(x)$ is the potential associated with clique c and Z is the partition function, which is effectively a normalising constant, where

$$Z = \sum_{x \in \Omega} e^{-U(x)} \quad (2.25)$$

The origins of GD stem from statistical mechanics [DE87, GG84]. However, for computer vision applications, its main function results from the following equivalence theorem.

Theorem 2.1 (Markov-Gibbs equivalence) A random field X is a MRF on S , with respect to a neighbourhood system, η , if and only if X is a GD on S with respect to η . This theorem was originated by Hammersley and Clifford and proofs for it may be found in [Bes74, Li95].

B. The Hierarchical Image Model

The hierarchical image model was introduced by Derin *et al* [DC86, DE87] and has become widely used as a framework for MRF based image segmentation. The model considers the segmentation process as a restoration problem. This may be achieved by assuming the existence of two different images:

1. *Ideal Image*: A discretely labelled image which occupies a state space $\Lambda = \{1, 2, \dots, K\}$ where K corresponds to the number of region types or classes present

in the image⁷. This effectively corresponds to an ideally classified image.

2. *Actual image*: The image data to be segmented is considered to be a corrupted version of the original, where it has been degraded by either noise, texture, or both.

Thus, the segmentation procedure may take place by restoring the *actual* image data in order to retrieve the *ideal* labelled data. The hierarchical model is a framework within which this may be performed. It considers two separate models: a *high level* process which models the labelling of the ideal image and a *low level* process which models the noise or texture responsible for the image degradation. We shall refer to the high level process as X and the low level process as Y .

The most popular choice of model for the high level process is the *multilevel logistic* (MLL) model [DC86, DE87]. This assumes that the high level process $X = \{X_s\}$ is a second order discrete MRF. A detailed description of this model may be found in appendix B.

The low level process, $Y = \{Y_s\}$, adopts a pattern recognition framework by considering the image as multi-feature object data. Various different models may be chosen for Y_s , depending on the characteristics of the image data. The zero-mean Gaussian distribution is a common choice, in particular for noisy image data [DC86, DE87, WD92]. This takes the form

$$p(Y_s = \mathbf{y}_s \mid X_s = i) = (2\pi)^{-p/2} |\boldsymbol{\Psi}_i|^{-1/2} \exp \left[-\frac{1}{2} (\mathbf{y}_s - \boldsymbol{\mu}_i)^T \boldsymbol{\Psi}_i^{-1} (\mathbf{y}_s - \boldsymbol{\mu}_i) \right] \quad (2.26)$$

where p is the number of features in the object data and $|\cdot|$ denotes the determinant of a matrix. Each element of class i in the ideal image ($i \in \Lambda$, i.e. $1 \leq i \leq K$) is assumed to occupy the position $\boldsymbol{\mu}_i$ in feature space, where $\boldsymbol{\mu}_i$ is the mean of class i . However, in the actual image, it is superimposed by noise, characterised by covariance matrix $\boldsymbol{\Psi}_i$. A popular model for the low level process in textured images is known as the Gaussian Markov random field (GMRF) model whose details are described in [CC85, DC86, MSC90, WD92].

⁷We switch notation from c , as used in section 2.2, to K , in order to avoid confusion with the conventional clique notation.

C. Image Segmentation using Markov Random Fields

An image may be segmented by finding optimal label values, x^* , for the ideal image. The most popular approach is through maximising the posterior probability distribution, $P(X | Y)$. This method is known as *maximum a posteriori* (MAP) segmentation. Applying Bayes rule:

$$P(X | Y = y) = \frac{P(X, Y = y)}{P(Y = y)} = \frac{P(Y = y | X)P(X)}{P(Y = y)} \quad (2.27)$$

The MAP criterion may thus be formulated as

$$x^* = \arg \max_X \{P(Y = y | X)P(X)\} \quad (2.28)$$

since $P(Y = y)$ is constant and independent of the values assigned to X .

Another approach to computing x^* involves maximising the posterior marginal (MPM) distribution, $P(X_s | Y = y)$. Details of MPM segmentation may be found in [Bes86, MSC90].

In order to solve x^* by finding the global maximum of the RHS of equation 2.28, the computation required is enormous. This is because K^n possible solutions have to be considered, where K is the number of labels and n is the number of pixels in the image. Hence, various sub-optimal solutions have been proposed. Geman and Geman [GG84] applied a stochastic relaxation principle known as *simulated annealing* (SA). An alternative approach, known as the *iterated conditional modes* (ICM) algorithm, has been proposed by Besag [Bes86]. Unlike SA, this is a deterministic approach and does not guarantee convergence to a global minimum. However, it is much faster and has a clear stopping rule. Some comparative studies [DW87, KD89] have shown that except for low SNR images, the ICM method yields results as good as those produced by SA.

After computing the optimal values for x^* , the resulting classified image may be converted into region data by connected component labelling (see section 6.2.2).

2.5 Knowledge Based Schemes

The human process of object recognition invariably relies on specific knowledge in order to identify a particular component in a scene correctly. Thus, in order to solve an image segmentation problem, it is helpful and sometimes even necessary to incorporate knowledge into the process. To date, no precise techniques have been developed which are able to model the human cognitive process effectively. Instead, various heuristic formulations, which take advantage of the types of methods described in the previous sections of this chapter, have been formulated and tailored to specific problems. In this section, some of the most popular approaches will be reviewed.

2.5.1 Rule Based Systems

Rule based systems consist of three major components: a database or *short term memory* (STM) - this contains information obtained from the actual data; a knowledge base or *long term memory* (LTM) - this contains the knowledge based rules which guide the segmentation process; and a *rule interpreter* which controls the actual problem solving process.

Nazif and Levine [NL84, LN85] describe an image segmentation system in which knowledge is employed in the form of production rules regarding the low level features of regions, lines and focus of attention areas. The resulting actions of the rules are to merge and split regions; add, delete, join and connect lines; and create and update focus of attention areas. The system also stores a set of *meta rules* [Dav80] to control the segmentation process. Stansfield [Sta86] has originated a system called ANGY which is used to segment coronary vessels from digital subtracted angiograms. After performing spatial operations on the image data to produce a crude segmented output, a rule based expert system is employed. This involves two stages: A low level stage, used to refine the existing segmentation; followed by a high

level stage which utilises knowledge of cardiac anatomy and physiology to identify those objects which are vessels and eliminate all others. Raya [Ray90] has devised a method for segmenting MRI brain data. This utilises multi-spectral data and derives confidence functions for the population distribution of each multi-spectral feature. The rule based system consists of sequential stages in which partially segmented binary scenes of one stage guide the next.

2.5.2 Knowledge Guided Methods

It is often useful to incorporate specific knowledge into traditional segmentation procedures without the rigorous implementation of an expert rule based system.

Tenenbaum and Barrow [TB77] incorporate *a priori* knowledge to guide a region merging process for image segmentation. Li *et al* [LGH93] have developed an approach which is able to identify tumours present in MRI brain data by combining anatomical knowledge with results from fuzzy *c*-means (FCM) classification (see section 2.2.4) of the data. An adaptive scheme for segmenting MRI data has been devised by Wells *et al* [WGKJ95]. This incorporates knowledge of tissue intensity properties and intensity inhomogeneities with the EM algorithm (see section 2.2.4) to overcome spatial deviations in signal uniformity and produce an accurate segmentation. Fisher *et al* [FCT⁺97] use *a priori* knowledge of the intensity and 3D spatial relationships of structures in MRI head data to segment the brain from its surrounding structures. Hinshaw and Brinkley [HB97] describe a 3D shape model that uses prior knowledge of an object's structure to guide the search for its boundaries.

2.5.3 Deformable Atlas Matching

There is growing interest in using atlases to segment 3D image data of the brain into its different structures [BK89, GRB93, TT96]. In these techniques, a reference volumetric data set, or atlas, is developed which has all the prominent anatomical

information delineated into labelled regions. This is then deformed to match the individual characteristics of a patient's anatomy. Given an accurate correspondence between the atlas data and the patient data, the anatomy within is automatically delineated and identified.

2.6 Methods for MR Image Segmentation

In the previous sections of this chapter, a wide range of approaches to image segmentation has been presented. This review is by no means exhaustive. Nevertheless, the main conventional strategies have been highlighted. In this section, the performance of these methods, when applied to MRI data, will be discussed. This will then be followed by a brief outline of the new techniques which have been developed for our study.

2.6.1 Conventional Techniques applied to MRI Data

The performance of any segmentation technique is highly dependent on the properties of the image in question. As described in appendix A, the contrast of MR image data is controlled by three main parameters, namely: proton density; T_1 ; and T_2 . Furthermore, there are a huge variety of protocols which may be adopted to acquire the MRI data. All these factors affect the properties and quality of the acquired data. The amount of noise present in these images depends on several factors ranging from the type of acquisition, e.g. spin echo or gradient echo, to voxel size and strength of the main magnet in the scanner. We assume, for the purpose of the following discussion, that the grey scale distribution of the image data is fairly homogeneous throughout the image, the contrast between different tissues is good, and that minimal noise is present.

When adopting a PR approach to classifying the data, most researchers use a collection of multi-spectral data, e.g. T_1 , T_2 and proton density images (see ap-

pendix A), of the same anatomical slice or volume. Supervised methods have been reported to give reasonable results [BHC93], although they are very sensitive to the training data used. Unsupervised methods, in addition to not requiring an expert to generate training data, perform at least as well if not better than their supervised counterparts [BHC93, HBC⁺92]. However, all these methods suffer from speed deficiencies. Processing times of several hours are typical for large data sets. Furthermore, the acquisition of multi-spectral data is extremely time consuming in its own right. In addition, it is not always possible to obtain MR image data, of two different contrasts, in the same resolution. In such instances, resampling and registration methods must be applied subsequently so that all the multi-spectral data are aligned and of the same size. The following section will introduce the idea behind a new classification method, whose details are described in detail in chapter 5, which alleviates these problems and performs very quickly. There is one main limitation to adopting any PR approach when classifying MR image data. This is the global nature of its implementation. Although the image data may be of high contrast and good spatial homogeneity, it is not perfect. Thus, it is inevitable that there will be some image elements which are incorrectly classified.

The success of spatial techniques varies from one method to the next. Boundary identification methods tend to require user interaction if good results are to be obtained. It is rare for a boundary to be defined by its edges alone. Simple edge linking approaches are rarely sophisticated enough to be able to determine which edges to link. The application of active contour models is now an intensive area of research in image segmentation. As described in section 2.3.1, there are many different approaches which have been developed and many of these are capable of extracting fairly accurate boundary contours. However, as yet, none of these may be fully automated. A limitation of all boundary identification approaches is their ability to segment a 3D structure of complex shape on a slice by slice basis. A structure may be bounded by a single contour on one slice and by several on another. Careful expert interaction is required to identify the correct contours in such a case.

Morphological techniques are extremely powerful. However, these tend to be ad hoc and invariably require a large amount of user interaction, both in terms of choice of structuring element as well as which operations to choose and in what sequence they should be performed.

The performance of region directed approaches varies according to the techniques adopted. Region growing methods are limited by the size of regions into which the image is initially divided. If these regions are too large, very rough edged regions will result. Furthermore, regions which should otherwise be separated may end up combined. On the other hand, reducing the region size increases the required processing time. Split and merge techniques are able to overcome the under-segmentation problems and computational inefficiencies of region merging techniques. However, the resulting segmented region data tend to be very ragged. In addition, the final result is dependent on the order in which the image is traversed during processing. A general problem from which region directed approaches suffer is the appropriate determination of thresholds which decide whether to split or merge different regions. These usually need to be set interactively and different optimal values may be necessary for different parts of the image.

Fuzzy region extraction techniques, as already mentioned in section 2.3.4, are still in their infancy.

Hybrid approaches would appear to share the best features of both PR approaches and spatial techniques. However, this is not always the case. Relaxation labelling techniques are only appropriate for refining already segmented data. Markov Random Field labelling, in theory should produce excellent results. However, this assumes that a good low level process model may be found which closely resembles the imaging process. Furthermore, the MAP criterion of equation 2.28 must be correctly solved. As mentioned in section 2.4.2, an optimal solution is computationally prohibitive so a sub-optimal solution must be adopted. Such a solution will get trapped in incorrect local minima unless a good starting solution is chosen. Results

from MRF based segmentation will be presented and discussed in more detail in chapter 6.

Knowledge based techniques have been shown to perform well for specific applications and in these instances provide the benefit of complete automation. However, no generic knowledge based approach has yet been developed which is able to tackle any image segmentation problem.

2.6.2 Fast and Automatic Segmentation of MR Image Data

The important and vital role of classification and segmentation steps was discussed in chapter 1 within the context of 3D visualisation of volume data. In order for this process to be realisable in real time, it is necessary to develop both automatic and fast methods for this purpose.

None of the conventional methods described in this chapter are capable of satisfying the aforementioned criteria. As discussed in section 2.6.1, each technique demands either an enormous amount of computation or requires user interaction in order to achieve satisfactory results when applied to MR image data. We therefore propose original methods in later chapters of this thesis which are able to classify and segment MRI data quickly, automatically and accurately. In chapter 5 a novel method is proposed for fast fuzzy classification of MRI data. This considers only the grey scale distribution of the MR image data. In order to refine the resulting classification, an adaptive approach, which considers spatial and hybrid methodologies, is presented in chapter 6. This allows specific anatomical structures to be removed from the data and thereby facilitates their accurate definition within a 3D rendered environment.

Chapter 3

3D Rendering Techniques

Tomographic imaging systems, such as X-ray CT and MRI, are able to acquire volume data in the form of a 3D array of voxels. As discussed in chapter 1, such data is conventionally viewed on a slice-by-slice basis. By employing a process called *rendering*, it is possible to visualise this data directly in three dimensions. This involves transforming the contents of the 3D data, through a sequence of projection and shading operations, into a two dimensional image which provides an illusory 3D effect.

Two main approaches to rendering three dimensional data have been developed. These are known as *surface* and *volume* rendering. This chapter describes the various techniques proposed for each of these schemes.

3.1 Surface Rendering

Surface rendering methods consider only the surfaces of the structures which are to be viewed. The first step in this process is to extract any such surface. This is followed by the actual rendering stage.

3.1.1 Surface Extraction

When describing how to extract the surface from a specific structure, we assume that the structure has been segmented accurately from its surrounding 3D image data using an appropriate method (see chapter 2). The approach adopted for surface extraction depends on the surface's ultimate representation, for which there are two possibilities: *geometric modelling* and *direct voxel representation*.

Geometric Modelling

The most popular form of geometric model, with which surfaces are defined, is composed of interconnected triangular tiles. This is the standard model used by computer graphics hardware and software systems.

Different approaches for extracting such a model from medical data have been proposed. An early approach [FKU77, Kep75] starts with the contours which define the surface boundary of the object to be rendered. The contours of consecutive slices are then connected by triangles. However, this approach fails if more than one surface contour exists on a single slice. This is because of the confusion which arises when attempting to decide on which contours to connect. Lorensen and Cline [LC87] have therefore proposed a more robust technique for surface extraction which they refer to as *marching cubes*. This scans the entire segmented volume and at each potential surface voxel, a database of vertices, which defines all the possible combinations of surface tile and voxel intersection, is accessed. The matching vertices are stored and a triangular tiled surface model of the entire object results.

Direct Voxel Representation

Herman and Liu [HL79] introduced the *cuberille* approach for the display of surfaces. Each voxel is considered as a cube whose faces are made of square tiles. In order to retrieve smooth surface normal information, Chen *et al* [CHRU85] consider all

four faces of each voxel's adjoining neighbours. Subsequent rendering processing times may be reduced by merging adjacent cubes to form an oct-tree [Mea82]. More recently, Cline *et al* [CLL⁺88] devised the *dividing cubes* approach. This utilises an appropriate interpolation technique to subdivide the surface voxels until they correspond to the final rendered image resolution. At each subdivided voxel, gradient vectors are interpolated and the surface normal is computed. If high resolution data are used, this approach is more efficient than the marching cubes algorithm [LC87] which produces polygons whose sizes approach those of voxels.

3.1.2 Rendering

The process of rendering converts the extracted surface into a 3D rendered two dimensional image. The main steps involved, for both types of surface representation, are depicted in figure 3.1 and explained in more detail below.

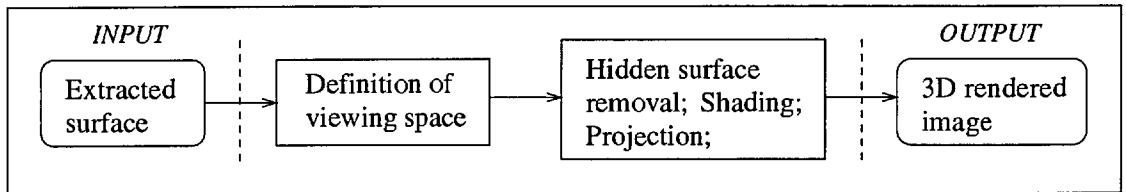


Figure 3.1: The main stages involved in rendering a surface model.

1. *Definition of viewing space*: This stage involves deciding on the position of the viewing plane in relation to the surface. The surface is then scaled, translated and rotated so that the perspective of interest faces the viewing surface directly.
2. *Hidden surface removal*: When viewing a surface, some parts of the surface are often obscured by others. The Z-buffer algorithm [Cat80] is the most popular approach for suppressing any hidden surfaces during rendering. The Z-buffer stores the smallest z value, i.e. distance from surface point to viewing plane, so far encountered for each projected point on the viewing plane. Thus, if a surface point is projected onto the viewing plane with a smaller associated z value than the current entry of

the Z-buffer, its value substitutes the existing entry and the Z-buffer is updated.

3. *Shading*: In order to view objects in the real world, they must be properly illuminated and reflect light to the eye of the observer. Thus, for a rendered image to produce a realistic effect, this process must be modelled. The most commonly

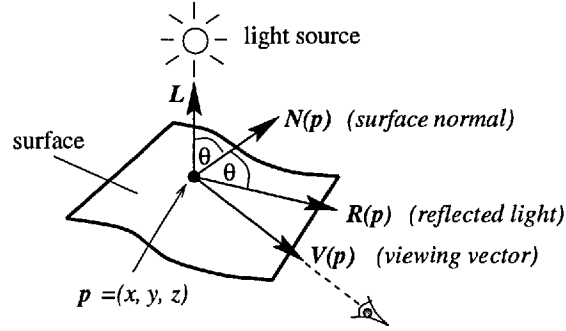


Figure 3.2: The Phong reflection model.

adopted model to be used in computer graphics was devised by Phong [Pho75]. This considers the illumination of a surface, as illustrated in figure 3.2. The light, reflected from a specific point, p , on the surface, as it reaches the viewing plane, which is perpendicular to $V(p)$, is computed as

$$I(p) = I_a k_a + \frac{I_i}{k_1 + k_2 d(p)} (k_d (L \cdot N(p)) + k_s (R(p) \cdot V(p))^n) \quad (3.1)$$

where:

I_a is the ambient lighting component (this approximates indirect global illumination and is usually assumed to be constant);

k_a is the ambient lighting coefficient;

I_i is the intensity of the point light source (this is assumed to behave as a spot light positioned at infinity - hence L is constant);

k_d is the diffuse reflection coefficient;

k_s is the specular reflection coefficient;

n is an index that simulates surface roughness;

$d(p)$ is the projected distance of surface point, p , to the viewing plane;

k_1 and k_2 are constants used for linear approximation of depth-cueing.

All the directional vectors, L , $N(\mathbf{p})$, $R(\mathbf{p})$ and $V(\mathbf{p})$, as shown in figure 3.2, are normalised.

It should be noted that the Phong model, as represented by equation 3.1, may incorporate a coloured light source as well as a coloured surface. The diffuse reflection component, k_d , is then a function of the surface colour and the specular component, k_s , is a function of the colour of the light. For details on more complex reflection models, the reader is referred to [Bli82, CT82].

4. *Projection*: This stage translates the 3D surface coordinates to the 2D viewing surface. This is achieved by projecting points from the surface onto a viewing surface. Either parallel or perspective projection is possible. However, due to its relative simplicity, parallel projection is invariably adopted.

It should be noted that steps 2-4 are not necessarily performed in the order presented and may vary according to the chosen rendering strategy. For a more detailed discussion on all the above steps, the reader is referred to [BG89, FV82, Udu91, Wat93].

3.2 Volume Rendering

A different approach to rendering 3D image data, is to consider the entire volume as a whole, rather than the surfaces of specific structures within. This allows the depiction of semi-transparent structures and poorly defined fuzzy surfaces. Before rendering can be performed, it is necessary to suitably pre-process the data. In this section, the different methods for preparatory processing of the volume will be detailed. This is followed by a description of the actual rendering stage. Finally, the need for performing matting operations, in order to isolate specific structures within the volume rendered environment, is discussed.

3.2.1 Preparatory Processing

All schemes for volume rendering require the preparation of two different volume arrays, one containing opacity values, and the other voxel colours. These two arrays are then input to the rendering stage. In order to obtain these arrays, it is first necessary to classify the data and to extract surface information from within. We therefore introduce this section with a review of voxel classification and surface extraction techniques.

(i) *Voxel Classification*: This involves deciding how many tissue types are present in the data, and to which tissue type each voxel belongs. It has been claimed [DCH88, Lev88] that the ideal strategy for a volume rendering scheme is to adopt a fuzzy classification approach. This provides more realism to the rendered output. A pattern recognition technique, such as described in section 2.2, may be adopted for this purpose. The output membership values of the classification process for a voxel, indexed by k in the 3D array, are $\{u_{ik}: 1 \leq i \leq c\}$, where c is the number of tissue types in the data (see section 2.2.2).

(ii) *Surface Extraction*: In order to extract surface information from the data volume, Levoy [Lev88] considered the central differences between the 6-neighbour voxels (see section 6.2.2) of each position in the volume array to obtain

$$\nabla g_{x,y,z} = \left[\frac{1}{2}(g_{x+1,y,z} - g_{x-1,y,z}), \frac{1}{2}(g_{x,y+1,z} - g_{x,y-1,z}), \frac{1}{2}(g_{x,y,z+1} - g_{x,y,z-1}) \right] \quad (3.2)$$

where $\nabla g_{x,y,z}$ is the gradient vector at position $\mathbf{p} = (x, y, z)$ in the voxel array and $g_{x,y,z}$ is the grey scale value at that point. The resulting surface strength, $S(\mathbf{p})$, and normal, $N(\mathbf{p})$, at position $\mathbf{p}$, are

$$S(\mathbf{p}) = |\nabla g_{x,y,z}| \quad \text{and} \quad N(\mathbf{p}) = \frac{\nabla g_{x,y,z}}{|\nabla g_{x,y,z}|} \quad (3.3)$$

Although more sophisticated methods for computing the surface gradient may be adopted (see section 2.3.1), these are more computationally involved. Furthermore, the visual difference between using such methods, as opposed to implementing equation 3.2, is virtually indistinguishable.

Drebin *et al* [DCH88] considered the material properties of the structures in the data whilst extracting the surface. Instead of using the grey scale values in equation 3.3, they used the values of a density volume, $D = \{d_k\}$. The density volume was computed by assigning to each tissue type, i , the density characteristic, ρ_i , and computing for each voxel

$$d_k = \sum_{i=1}^c u_{ik} \rho_i \quad (3.4)$$

Thus, by modifying equation 3.2, through substituting the values of $g_{x,y,z}$ with those of $d_{x,y,z}$ (k indexes the (x, y, z) position in the 3D grid), a surface, based on the density of the structures within, may be extracted.

The advantage of the latter scheme is that by considering the material properties of the data, the extracted surface portrays a more realistic representation of the actual anatomy. On the other hand, any imperfections in the classification process will generate errors during surface extraction.

We are now in a position to consider the different schemes which have been proposed for preparing the opacity and voxel colour arrays. Two main approaches have been developed, each making its own assumptions about the composition of the data. These are:

1. *Fuzzy Surface Model*: The volume is composed of light *reflecting*, fuzzy surfaces, whose opacities depend on the material properties of their associated tissue structures [Lev88].
2. *Translucent Coloured Voxel Model*: The volume is a block of translucent, coloured, light *emitting* structures which are bounded by light *reflecting*, fuzzy surfaces. Again, all these properties are determined by the characteristics of their corresponding tissue types [DCH88].

Throughout the rest of this section, the processing stages required for each of the above described volume models, will be described.

Fuzzy Surface Model

The two voxel arrays, one comprising opacity values, and the other shaded voxel colours, are generated as follows:

(a) *Opacity Assignment*: Each voxel in the volume is assigned an opacity value, α ($0 \leq \alpha \leq 1$), depending on its corresponding tissue type and surface strength. If $\alpha = 1$, this implies that the voxel is completely opaque. Conversely, if $\alpha = 0$, it is completely transparent. Figure 3.3 shows a two dimensional opacity function,

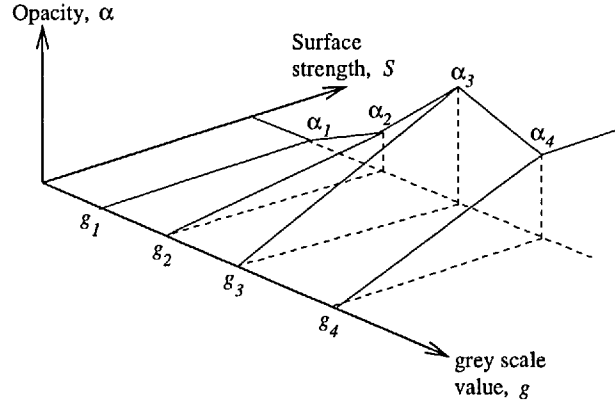


Figure 3.3: Opacity assignment function weighted by surface strength.

proposed by Levoy [Lev88], which assumes that each voxel, within a structure of tissue type i , has a specific opacity value, α_i , and an associated grey scale value, g_i . As one structure merges into another, the opacity values are assumed to change linearly, according to the changing grey scale values. In order to emphasise only the surfaces during rendering, the opacity values are scaled according to the surface strength, S .

The limitation of using such an opacity function is that it assumes that voxels can only contain mixtures of two tissue types. This is not always true. Furthermore, the values g_i ($1 \leq i \leq c$) must be known in advance. With X-ray CT data, it is possible to choose the values of g_i according to the Hounsfield scale. However, this is not the case for MRI data where no such scale exists due to the diversity of scanning protocols. We therefore propose, that by adopting an appropriate fuzzy classification

technique, the resulting membership values, u_{ik} , may be used to compute the opacity value, α , of a voxel at position $\mathbf{p}$, as

$$\alpha(\mathbf{p}) = S(\mathbf{p}) \sum_{i=1}^c u_{ik} \alpha_i \quad (3.5)$$

where:

$S(\mathbf{p})$ is the surface strength;

c is the number of tissue types in the data;

u_{ik} is the membership of the k 'th object (which corresponds to the voxel at position $\mathbf{p}$) in class i ;

α_i is the opacity value of tissue type i (this is set by the user in advance).

(b) *Voxel Colouring*: This process involves computing the lighting equations for each voxel and hence determining the amount of light reflected at that particular point in the 3D array¹. The Phong model [Pho75], as defined in equation 3.1, is typically adopted. At each point, the value for the surface normal, $N(\mathbf{p})$, as computed during the surface extraction stage is used. The value of $\mathbf{L}$ is set according to the position of the light source and the values for $R(\mathbf{p})$ and $V(\mathbf{p})$ may be found easily using vector algebra. An entire voxel array of shaded values results. If a coloured light source is considered, the shaded values are coloured and are referred to as $C(\mathbf{p})$.

Translucent Coloured Voxel Model

(a) *Opacity Assignment*: Opacities are assigned to each voxel in almost the same way as for the fuzzy surface model. However, for this model, these values are not scaled according to the surface strength. Thus the opacity, α , at any location $\mathbf{p}$ (indexed by k) may be computed as

$$\alpha(\mathbf{p}) = \sum_{i=1}^c u_{ik} \alpha_i \quad (3.6)$$

¹Only surface reflection of the illuminating light is considered. Any emission of light from the voxel data is neglected.

(b) *Voxel Colouring*: A colour, C_i , is assigned to each tissue type, i , within the volume data. Each voxel is then assigned a blend of colours, C , such that

$$C(\mathbf{p}) = \sum_{i=1}^c u_{ik} C_i \quad (3.7)$$

The colour, C_i , of each tissue type is, in general, represented by an (r, g, b) vector. A method for choosing values for C_i which optimises the contrast in the image data is described in [SK97] and in section 5.2.2.

If a surface passes through a voxel, as illustrated in figure 3.4, it will reflect light. We refer to the colours of the tissue structures in front of and behind the intersecting surface as C_F and C_B respectively. During the subsequent rendering process, the coloured light emitted by these structures must be considered, together with C_S , the colour of the light reflected from the surface. The colour, C_S , reflected

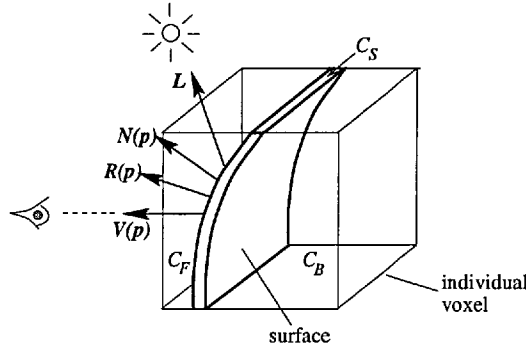


Figure 3.4: Colour assignment within a voxel

from a surface which intersects a voxel at position $\mathbf{p}$, may be computed as

$$C_S = [f_d(\mathbf{L}, \mathbf{N})C_D + f_s(\mathbf{R}, \mathbf{V})C_L] S(\mathbf{p}) + C_L k_a \quad (3.8)$$

where:

- $\mathbf{L}, \mathbf{N}, \mathbf{R}$ and $\mathbf{V}$ are as defined earlier in equation 3.1 (note that $(\mathbf{p})$ has been dropped from the end of the latter three terms for the sake of brevity);
- f_d and f_s are appropriate functions describing diffuse and specular reflection (such as those defined by the Phong model of equation 3.1);

C_D is the diffuse colour of the surface;

C_L is the colour of the light source;

k_a is the ambient lighting coefficient;

$S(\mathbf{p})$ is the surface strength at location $\mathbf{p}$ in the voxel array.

The simplest approach to computing the diffuse surface colour, C_D , is to set $C_D = C_F + C_B$. However, this results in the “bleeding” of colours at tissue junctions². Thus, it is preferable to set the value of C_D equal to C_B . This may be determined by checking the direction of $\mathbf{N}(\mathbf{p})$ [DCH88].

3.2.2 Rendering

This stage involves the projection and compositing of the colour and opacity data onto a viewing plane. This may be achieved by either *ray casting* or *voxel projection*.

Ray Casting

The process of ray casting is illustrated in figure 3.5. From each pixel on the viewing plane, a ray is cast into the voxel array and is sampled at N equidistant positions as it passes through. A back drop is positioned behind the voxel array to provide a background to the rendered image. At each sample point, the colour and opacity values, associated with the voxel through which the ray intersects, are considered. For more realistic, but more computationally expensive, renditions, the values taken at the sample points are calculated by interpolation of the values associated with the surrounding adjacent voxels. The colour of the ray, as it traverses the data, may be computed from the transparency equation [PD84]. This may be performed, in either back to front [Lev88] or front to back [Lev90b, UK88] order, as follows:

²If more than two tissue types exist within a single voxel, only the two tissues of highest membership, in the voxel in question, are considered.

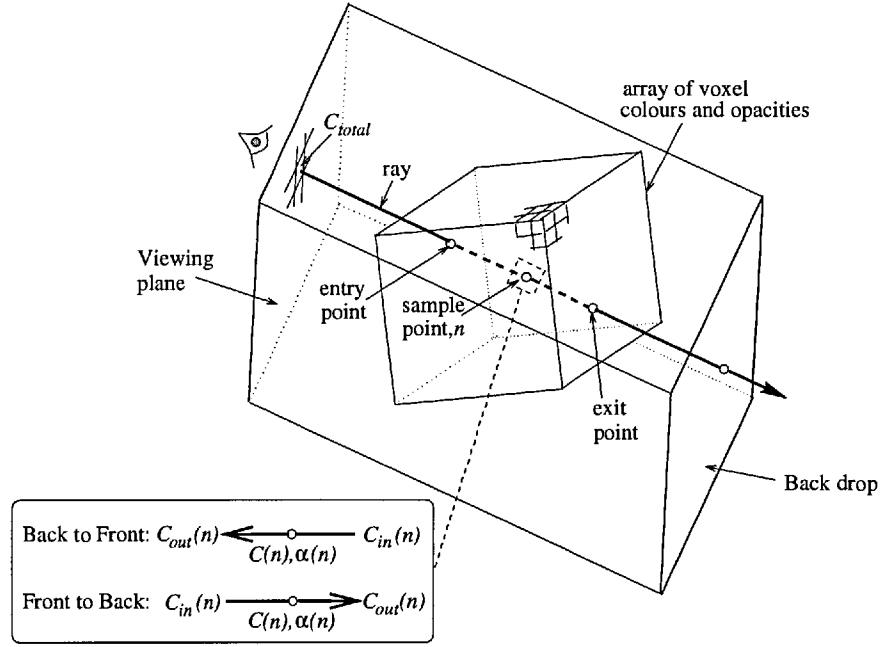


Figure 3.5: Ray casting through coloured and opacity voxel array.

1. *Back to Front Compositing:* The sample points are indexed as $n = 1$ at the exit point and $n = N$ at the entry point of the voxel array. Progressing along the ray in backwards order, the colour of the ray leaving each point, $C_{out}(n)$, is

$$C_{out}(n) = C_{in}(n)(1 - \alpha(n)) + C(n)\alpha(n) \quad (3.9)$$

where $C_{in}(n)$ is the colour of the ray before entering the sample point, and $C(n)$ and $\alpha(n)$ are the ray's colour and opacity values respectively. The final colour of the ray, seen from behind the viewing plane, is equal to

$$C_{total} = \sum_{n=0}^N \left[C(n)\alpha(n) \prod_{m=n+1}^N (1 - \alpha(m)) \right] \quad (3.10)$$

where $C(0)$ and $\alpha(0)$ are the colour and opacity of the back drop, respectively. A fully opaque background is usually considered, i.e. $\alpha(0) = 1$

If the coloured translucent voxel model is adopted, it is necessary to incorporate the surface colours within the compositing process. If the opacity, colour, and surface values for the sample points along the ray are interpolated from their adjacent neighbours, each interpolated sample point is considered as the centre of a new voxel.

It is therefore preferable, in this instance, to compute the shading calculations for any surfaces, as described at the end of section 3.2.1, after interpolation. If there is no surface present in a resampled voxel, i.e. $S(n)$ (as defined by equation 3.3) is equal to zero, the colour of the ray leaving the voxel is computed from equation 3.9. However, if a surface is present in the voxel, a three stage equation is required to compute $C_{out}(n)$, as follows:

$$C_{out}(n) = C_F(n) \text{ over } (C_S(n) \text{ over } (C_B(n) \text{ over } C_{in}(n))) \quad (3.11)$$

where $C_F(n)$, $C_S(n)$ and $C_B(n)$ are as defined in figure 3.4, and

$$C_a \text{ over } C_b = C_a(1 - \alpha_b) + C_b\alpha_b \quad (3.12)$$

as defined by Porter and Duff [PD84].

2. *Front to Back Compositing.* The colour and opacity values of the ray are composited as the ray progresses through the data set. Thus, the samples along the ray are indexed as $n = 1$ at the entry point and $n = N$ at the exit point. The colour and opacity of the ray leaving each sample point is computed using a reverse transparency formula to that of equation 3.9, such that

$$\begin{aligned} \hat{C}_{out}(n) &= \hat{C}_{in}(n) + \hat{C}(n)(1 - \alpha_{in}(n)) \\ \text{and } \alpha_{out}(n) &= \alpha_{in}(n) + \alpha(n)(1 - \alpha_{in}(n)) \end{aligned} \quad (3.13)$$

where $\hat{C}_{in}(n) = C_{in}(n)\alpha_{in}(n)$, $\hat{C}_{out}(n) = C_{out}(n)\alpha_{out}(n)$, and $\hat{C}(n) = C(n)\alpha(n)$.

After all the samples of a ray have been processed, the colour, C_{total} , of the ray is computed as

$$C_{total} = \frac{\hat{C}_{out}(N + 1)}{\alpha_{out}(N + 1)} \quad (3.14)$$

where $C(N + 1)$ is the colour of the back drop and $\alpha(N + 1)$ is its opacity. Clearly, if a fully opaque background is draped behind the data set, the normalisation step in equation 3.14 may be omitted.

If the coloured translucent voxel model is adopted, it is again necessary to perform compositing in three stages wherever a surface exists in the resampled voxel.

Back to front compositing has the advantage that it does not require an opacity buffer to store accumulated opacity values. However, in some situations, front to back compositing is superior. As soon as a ray reaches a sample point, n , where the accumulated value of $\alpha_{out}(n)$, has reached one, the ray need not be further projected. This may save a significant amount of processing for large volume sets [Lev90b].

Voxel Projection

Ray casting is a *backward* mapping algorithm which maps the viewing plane onto the voxel data. Conversely, voxel projection is a *forward* mapping algorithm which maps the voxel data onto the viewing plane [DCH88, UK88, Wes90, WV91]. As shown in figure 3.6, each voxel may be projected onto more than one pixel of the viewing

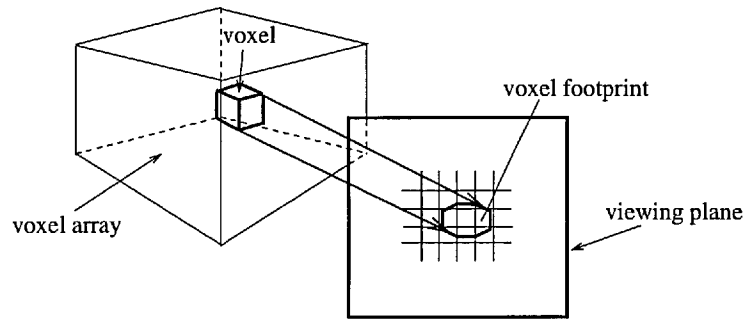


Figure 3.6: Projection of a voxel onto the viewing plane.

plane. The resulting projection is referred to as a *footprint*. Westover [Wes90] describes a method which calculates a viewing plane footprint for each individual voxel and uses this to spread the voxel’s “energy” onto the viewing plane. This process is also referred to as *splatting*.

Voxel projection proceeds in a slice by slice order. An efficient approach is to resample the voxel array and obtain a stack of image slices which are parallel to

the viewing plane [DCH88, Wes90]. The resampled voxels from each plane are then splatted onto a “sheet” which is subsequently composited onto the viewing plane. This procedure is often referred to as plane compositing. The order of processing may take place in either a back to front, or front to back, order, and the respective compositing operations are as described in equations 3.9 and 3.13.

Evaluation of Ray Casting and Voxel Projection Methods

The main advantage of ray casting over voxel projection is the ease with which it may be implemented. No footprint function needs to be considered. Furthermore, as mentioned earlier, if the ray data is composited from front to back, each ray may be terminated as soon as its opacity value reaches one. However, since ray casting is essentially a sampling technique, it can suffer from strong aliasing, particularly if only one ray per pixel is cast from the viewing plane. Levoy [Lev88] claims that by adopting a fuzzy classification scheme, the amount of aliasing is substantially reduced. This may be further decreased by blurring, or supersampling the voxel data, prior to ray casting. However, this does add to the overall computational burden. Alternatively, more than one ray can be cast per pixel in the viewing plane, whose values may be subsequently interpolated. The efficiency of this approach may be improved by varying the density of rays according to image complexity [Lev90c].

Voxel projection does not suffer from aliasing in the same way as ray casting. A further advantage of this approach is its amenability to parallel implementation. This is possible since, as each voxel is processed, only knowledge of its immediate neighbours is required. In addition, little memory is required, as its total overhead never exceeds the size of each plane to be composited. This compares favourably to ray casting where each ray requires random access to the entire voxel array.

3.2.3 Matting

One of the benefits of volume rendering is that it allows all the data to be viewed at once. However, the rendered image may, as a result, suffer from an overload of information. It may be confusing to identify the contents of the data, due to the complexity of the structures contained within. Furthermore, the view of certain objects within the data may be obstructed by neighbouring structures. Thus, the regions of primary interest may be completely inaccessible to the renderer.

This problem may be somewhat overcome by adjusting the opacity values, corresponding to particular tissue types, throughout the entire data. However, since more than one structure may be of the same tissue type, this does not allow the removal or isolation of an individual object in the rendered scene. This limitation will be demonstrated in section 7.1.

Matting is a method which is able to solve the problem of information overload. Specific sections of the data may be either entirely removed, or made less visible. Matte volumes are created for these operations. These may be simple shapes, such as wedges, or may correspond to segmented regions within the data. Matting operations involve either multiplying or complementing the matte volume with the volume opacity data prior to rendering. As a result, the visibility of either the matte shape, or its surrounding volume data, is reduced.

3.3 Surface versus Volume Rendering

The relative merits of surface and volume rendering have, in the past, raised some controversy [UH89]. Research has been performed [RNM⁺91, UHC91] in order to quantify and qualify the strengths and weaknesses of the two approaches. In section 7.1, we present a preliminary study to investigate their relative performances when applied to MRI data. In this section, some of the fundamental differences between

the two methodologies are discussed.

3.3.1 Computational Efficiency

Surface rendering strategies have the distinct advantage that only the surface of interest needs to be rendered. Thus, they perform significantly faster than basic volume rendering algorithms which process every voxel in the entire volume. Furthermore, when rendering geometric surfaces, conventional computer graphics software and hardware may be utilised.

During the past few years, much work has been done to improve the efficiency of volume rendering techniques. As mentioned earlier in section 3.2.2, voxel projection approaches may be adopted which can be implemented in parallel. Furthermore, methods have been proposed [Lev90b] to improve the efficiency of ray casting. These either employ a pyramid of binary volumes to encode spatial coherence in the data, or use an opacity threshold to terminate the process adaptively. Reductions in computation time of an order of magnitude have been achieved. A further development, in improving the efficiency of volume rendering, involves using a shear-warp factorisation of the viewing transformation [LL94]. This is claimed to perform at least five times faster than conventional approaches. A different approach has been pursued by Udupa and Odhner [UO93] who define a new data structure, which they call a *shell*, to represent fuzzy surfaces within a volume. This considers only those voxels which lie within the vicinity of the boundary of the structure to be rendered, and discards all other voxels in the volume. Udupa and Odhner [UO93] report an almost thousand fold increase in speed of performance as a result of applying the shell structure to represent a non-complex 3D object.

3.3.2 Pre-processing Requirements and Quality of Display

The amount of processing required for different rendering schemes and the resulting quality of the renditions depends on many different factors, including the type of image data to be rendered. For example, it is very unclear from the literature, whether data classification techniques alone are sufficient, or whether rigorous segmentation techniques are required, in order to produce a high quality volume rendered display from 3D MRI data. A study to evaluate a range of 3D rendering methods and to determine their corresponding processing requirements, when applied to MRI data, is presented in chapter 7.

3.3.3 Interactive Manipulation

It is often desirable to move structures, or even perform dissections, within the 3D rendered environment. This is a straightforward task within a surface rendered scene [BTU84, HTU88, Tri86] since each object is defined by a surface model. On the other hand, within a volume rendered scene, objects are often not absolutely defined and it is impossible to manipulate them using conventional methods. However, techniques have recently been developed [OU95] which allow manipulation to be performed on fuzzy structures which are visualised by adopting the shell rendering scheme [UO93].

3.4 Final Remarks

In this chapter, the different approaches to rendering 3D data have been presented and discussed. The following three chapters describe methods for MRI data acquisition and for fast and automatic pre-processing operations which facilitate the ultimate rendition of the data. This is followed by an evaluation study, in chapter 7, of the performance of the main rendering schemes detailed in this chapter, when applied to MRI data which has been acquired and processed using these techniques.

Chapter 4

MR Image Acquisition

The first step in the process of 3D visualisation of MRI data is to acquire the actual data. In this chapter, details of methods adopted for obtaining image data, from both conventional and interventional MR scanners, will be described. For a discussion on the basic fundamentals of MR imaging, the reader is referred to appendix A.

4.1 Methods for Data Acquisition

4.1.1 Imaging Requirements

When acquiring 3D image data with the eventual goal of visualising the data in three dimensions, it is necessary to consider whether subsequent segmentation and rendering stages may be implemented effectively. It is therefore important to ensure that the following criteria are satisfied during the acquisition stage:

- (a) *Image Resolution*: The resolution of the acquired data must be sufficiently large to include enough information to enable the anatomical regions of interest to be clearly delineated.
- (b) *Noise Suppression*: In order to classify and segment the data effectively, it is

important to minimise the amount of noise inherent in the acquisition process.

(c) *Tissue Contrast*: In our study we are primarily interested in visualising general anatomical structures, as opposed to specific pathologies or abnormalities. It is therefore essential that the image data acquired possesses good overall tissue contrast. This is a necessary criterion if good results are to be obtained from the classification and segmentation techniques described in the next two chapters.

A long-term objective of this project is to be able to visualise data in 3D, from start to finish, in real time. Thus, the most efficient protocol, which satisfies the above conditions, should be adopted.

4.1.2 Imaging Protocols

For the purpose of our study, we consider the acquisition of head and knee images only. The conventional scanner used is the IGE Signa 1.5 Tesla Advantage scanner. The interventional machine is the new GE 0.5 Tesla IMR scanner (see section 1.5).

In order to acquire images with good overall anatomical contrast, T_1 weighted sequences are used. Both spin-echo (MEMP) and Spoiled GRASS (SPGR) sequences have been adopted for this purpose. These sequences produce images with little noise. Furthermore, T_1 weighted spin-echo data is acquired faster than its T_2 counterpart due to the much shorter echo time (TE) used. The disadvantage of acquiring spin-echo data, for the purpose of subsequent 3D processing, is that the volume is obtained by multi-slice acquisition. As mentioned in appendix A, section A.2.1, after initiating a field gradient, G_{slice} , each slice is selected by applying an RF pulse of an appropriate range of frequencies. In order to excite only those protons which occupy the slice of interest, a sinc shaped signal must be applied. Due to the infinite extent of such a pulse, a truncated form is adopted. However, as a result, protons from outside the selected slice also contribute to the image. Also, slice thickness is necessarily several times the pixels size. SPGR volume acquisitions do not suf-

4.1. METHODS FOR DATA ACQUISITION

fer from this problem since the volume is acquired as a whole without requiring RF slice excitation. However, these images are slightly noisier than their spin echo counterparts.

The following protocols were used to acquire data from the conventional IGE 1.5 Tesla scanner:

Head data: Coronal SPGR volume (124 slices). Slice thickness = 1.5 mm (no gap), pixel size = $(0.937 \text{ mm})^2$. Flip angle = 60° , NEX=1.

Knee data: Sagittal, T_1 weighted, MEMP. Slice thickness = 3 mm (slice gap = 0.5 mm), pixel size = $(0.703 \text{ mm})^2$. NEX=2. This is a standard sequence used routinely for clinical applications.

The strength of the magnet of the new IMR scanner is only 0.5 Tesla, as opposed to the 1.5 Tesla magnet of the conventional scanner. Thus, the images acquired by the IMR scanner using the above sequences are more noisy. We have therefore experimented with the voxel size and NEX to obtain the following sequences:

Head data: The same protocol as used for the conventional MR scanner was adopted. However, the NEX is increased to 1.5.

Knee data: We implemented the following three sequences:

Sequence 1: Sagittal, T_1 weighted, MEMP. Slice thickness = 4 mm (slice gap = 0.5 mm), pixel size = $(0.703 \text{ mm})^2$. NEX=1.

Sequence 2: Sagittal SPGR volume. Slice thickness = 3 mm (no gap), pixel size = $(0.703 \text{ mm})^2$. Flip angle = 60° , NEX=1.5.

Sequence 3: Sagittal SPGR volume. Slice thickness = 2 mm (no gap), pixel size = $(0.703 \text{ mm})^2$. Flip angle = 60° , NEX=2.

All the conventional MRI data and the IMR head data were acquired using receive and transmit birdcage quadrature coils. The IMR knee data was obtained using wrap around receive coils.

4.2 Results and Discussion

4.2.1 Image Quality

The middle slices of the data sets acquired using the protocols described in section 4.1.2 are shown in figures 4.1 and 4.2, respectively.

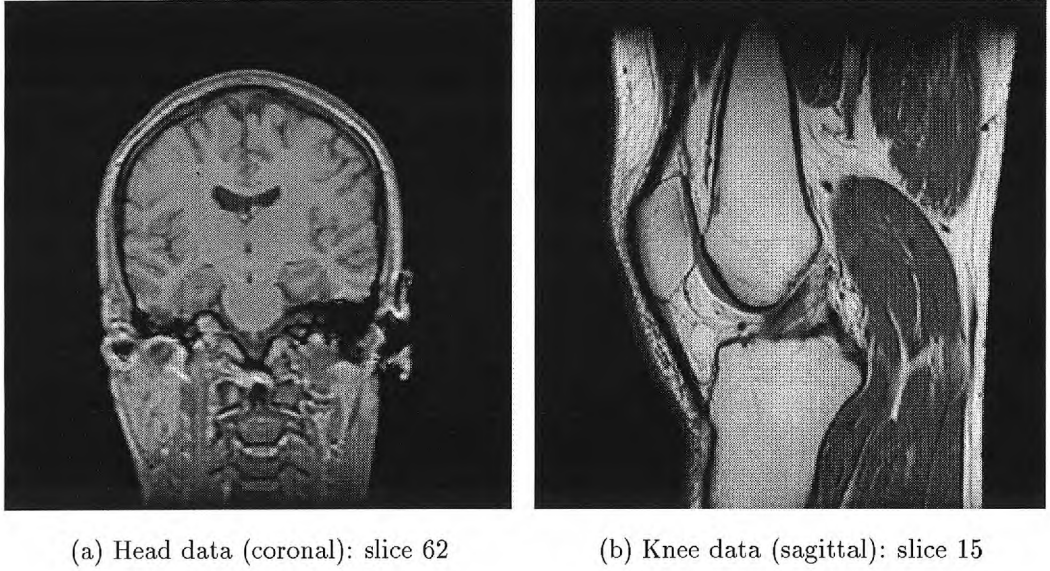
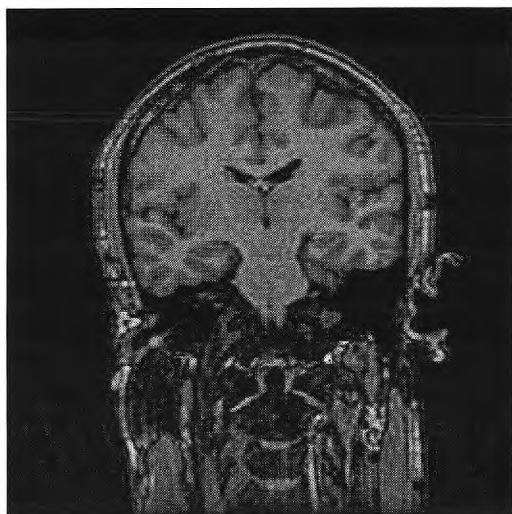


Figure 4.1: Image data acquired by conventional IGE 1.5 Tesla scanner.

All the images acquired by the conventional MRI scanner are of extremely high quality. It may be seen from figures 4.1(a) and (b) that the tissue contrast is very good and little noise appears to be present.

The head data acquired by the IMR scanner, shown in figure 4.2(a), is of surprisingly high quality considering the lower strength of the IMR magnet. Although its conventional MRI counterpart, shown in figure 4.1(a), is slightly less noisy and provides better tissue contrast, the difference in quality is only marginal. This is despite the fact that the only adjustment to the IMR imaging protocol is an increase in NEX from 1 to 1.5. The following chapters will compare how effectively subsequent processing tasks perform on these two data sets.



(a) Head data (coronal): slice 62



(b) Knee data (sequence 1): slice 11



(c) Knee data (sequence 2): slice 14



(d) Knee data (sequence 3): slice 30

Figure 4.2: Image data acquired by 0.5 Tesla IMR scanner.

The IMR knee data acquisition was not as successful as for the head data. Figures 4.2(b)-(d) demonstrate how the overall tissue contrast and grey scale spatial homogeneity of the images, from all three sequences, is of markedly lower quality than that of conventional MRI knee data (see figure 4.1(b)). The most likely cause for this is the localiser coils used during the imaging procedure. The current IMR system only provides wrap around receive coils for body scanning. Unlike the more rigid receive-transmit coils which are circular and therefore receive and transmit uniformly throughout the enclosed region of interest, the receive-only coils are simply positioned on top of the anatomy of interest. Thus, the spatial homogeneity of the resultant images is substantially reduced. The future development of improved localiser coils is likely to solve this problem.

4.2.2 Acquisition Times

The acquisition times of the different MR and IMR imaging sequences are displayed in table 4.1.

	Acquisition time (minutes and seconds)
Head data	9:35
Knee data	6:32

(a) Conventional MRI acquisition times

		Acquisition time (minutes and seconds)
Head data		20:03
Knee data	Sequence 1	5:21
	Sequence 2	5:11
	Sequence 3	30:06

(b) IMR acquisition times

Table 4.1: Acquisition times for conventional MR and IMR imaging sequences.

It is evident from these results that both the MRI and IMR imaging sequences used in this study are some way off from performing in real time. We have been rather conservative in this study and have attempted to acquire IMR data of the highest possible quality. It should be mentioned that we have not considered using the specially designed fast IMR sequences due to the noisy nature of the resulting

images. Future investigation needs to be performed into more efficient imaging sequences and the extent to which the quality of the data can be compromised in terms of corresponding reductions in acquisition times (see chapter 8).

It is important to remember that the IMR system used in this study is a first generation prototype. A number of different aspects of hardware, such as the speed and quality of gradient switching, as well as software, have been overlooked. There is no question that future generation systems will be superior. In particular, much work is being done to improve the quality of gradient imaging sequences. Furthermore, new sequences such as Single Shot Fast Spin Echo (SSFSE), Echo Planar (EPI) and Fast Spoiled GRASS (FSPGR) are being developed which reduce acquisition times considerably.

4.3 Conclusions

Imaging protocols have been described for the acquisition of head and knee data from conventional MRI and IMR scanners. Extremely good results were obtained from the conventional MRI system and good quality head image data was obtained from the IMR machine. Although the quality of the IMR knee data was somewhat lower, it has been argued that with the availability of new localiser coils, this can be significantly improved.

The acquisition times of the sequences used in this study, for both scanners, were of the order of several minutes. A combination of future improvements in scanner technology and refined sequences are likely to reduce the required imaging times substantially.

Chapter 5

Fast Fuzzy Classification of MR Image Data

5.1 Introduction

In order to visualise specific structures within MRI volume data, it is essential to delineate between the different objects within. In chapter 2, a comprehensive review of segmentation techniques was presented. Of the approaches described, unsupervised pattern recognition schemes demand the least user intervention. However, existing methods require several hours of processing time in order to classify 3D image data sets.

In this chapter, a new, fuzzy, unsupervised, clustering approach is described which is able to overcome the inefficiencies of conventional techniques and classify MR images very quickly and accurately. Furthermore, a novel colour blending scheme has been developed, which is able to display the fuzzy segmented data with no loss of information resulting from the classification. The following sections first describe the new algorithm. Next, methods for implementing traditional alternative algorithms are briefly described. Results from classifying both conventional MRI as well as IMR head and knee data are then presented and discussed. Finally, these are compared with results from the other techniques.

5.2 The New Classification Algorithm

Classical pattern recognition schemes consider the grouping of clusters within a multi-dimensional feature space. As mentioned in section 2.2.1, when considering a pattern recognition approach to segment an image, the different features which may be considered may include grey scale, texture, and for MRI data, grey scale values from a range of multi-spectral data. Batista [Bat96] analysed the performance of different features for classifying MR image data. He found that although considering texture features for natural texture generated images improved the resulting classifications, this did not occur with MR images. In fact, choosing grey scale as the only feature consistently yielded better classification results than when using any combination of different textures. When using multi-spectral features, e.g. T1, T2, proton density, etc., the resulting classifications were sometimes marginally improved. However, feature reduction was often necessary. Furthermore, both acquisition and processing times were significantly increased (see section 2.6.1).

It would therefore appear that in order to optimise the performance of clustering schemes for the purpose of MRI data classification, it is preferable to consider grey scale as the only feature. A new algorithm will now be described which, by using only grey scale data, is able to perform fuzzy classification of MRI data very rapidly.

5.2.1 Fast Fuzzy c -Means (FFCM) Clustering

The FFCM algorithm involves two stages. First of all, a reduced resolution grey scale histogram is produced from the original image data. Next, the values of the histogram are clustered using a simplified form of the traditional FCM algorithm. These stages will now be described in greater detail.

Generation of Reduced Resolution Grey Scale Histogram (STAGE 1)

Typically, MR images are acquired in a 12 bit format. In order to reduce the required computation during the clustering process, the grey scale value of each pixel is reduced to fit within an 8 bit grey scale resolution. A histogram, f , is then generated from the new pixel values. The value of each entry, $f(g)$, is equal to the frequency with which pixels, of grey scale value g , occur within the 8 bit image. This step reduces the size of the object data by a factor of 2^4 . This means that during the next stage of the algorithm, the amount of data to be clustered, and hence the required computation time, will be 16 times smaller.

Clustering the Histogram Data (STAGE 2)

In section 2.2.4, the fuzzy objective function, as defined by Bezdek [Bez81], was described in brief. By using the newly generated values, $f(g)$, of the reduced resolution histogram, an approximated¹ form of equation 2.13 may be written as

$$J_m(W, \mathbf{v}) = \sum_{g=G_{min}}^{G_{max}} \sum_{i=1}^c f(g)(w_{ig})^m (d_{ig})^2 \quad (5.1)$$

where:

$d_{ig} = |g - v_i|$ is the Euclidean distance between the grey scale value g and the fuzzy cluster centroid (v_i is defined in equation 5.4);

W is a fuzzy partition of the histogram data whose entries w_{ig} correspond to the membership values of each grey scale value g in cluster i of the image;

c is the number of clusters, or tissue types, in the image;

G_{min} and G_{max} are the minimum and maximum grey scale values in the 8 bit image respectively;

$f(g)$ is the frequency of grey scale value g in the 8 bit image;

$m \in (1, \infty)$ is the fuzzy weighting exponent.

¹If the values from a histogram generated from the original 12 bit image are used, the RHS of equations 2.13 and 5.1 are equivalent.

The resulting W partition² after minimising the function $J_m(W, \mathbf{v})$ gives an optimal fuzzy classification solution of the image data. This is obtained by using the following theorem:

Theorem 5.1 Fix $m \in (1, \infty)$ and let $f(g)$ have at least $c < (G_{max} - G_{min} + 1)$ distinct samples. Define $\forall g$ the sets

$$\begin{aligned} I_g &= \{i | 1 \leq i \leq c; d_{ig} = |g - v_i| = 0\} \\ \tilde{I}_g &= \{1, 2, \dots, c\} - I_g \end{aligned}$$

then $(W, \mathbf{v}) \in M_{fc} \times \mathbb{R}^c$ may be globally minimum for J_m only if either

$$I_g = \emptyset \Rightarrow w_{ig} = 1 \Bigg/ \left[\sum_{j=1}^c \frac{d_{ig}^{2/(m-1)}}{d_{jg}} \right] \quad (5.2)$$

or

$$I_g \neq \emptyset \Rightarrow w_{ig} = 0 \ \forall i \in \tilde{I}_g \quad \text{and} \quad \sum_{i \in I_g} w_{ig} = 1 \quad (5.3)$$

$$v_i = \sum_{g=G_{min}}^{G_{max}} (w_{ig})^m f(g) g \Bigg/ \sum_{g=G_{min}}^{G_{max}} (w_{ig})^m f(g) \quad \forall i \quad (5.4)$$

This theorem is derived by fixing $\mathbf{v} \in \mathbb{R}^c$ and applying Lagrange multipliers to the variables $\{w_{ig}\}$. The proof for Bezdek's theorem [Bez81] for the FCM algorithm, may be easily extended to theorem 5.1.

$J_m(W, \mathbf{v})$ may be minimised by implementing the following algorithm which satisfies theorem 5.1 at each iteration:

²We define W exactly the same way as U (see equation 2.6). However, since the partition does not directly correspond to the object data but rather to the histogram data, we change the notation from U to W in order to avoid confusion.

Algorithm 5.1 The following procedure is implemented:

1. (a) Fix c within the integer range $2 \leq c < (G_{max} - G_{min} + 1)$
(b) Fix $m \in (1, \infty)$
(c) Initialise $W^{(0)} \in M_{fc}$
(d) Set $l = 0$.
2. Calculate the c fuzzy cluster centres $\{v_i^{(l)}\}$ using equation 5.4 and the values of $W^{(l)}$.
3. Update $W^{(l)}$ using equations 5.2 or 5.3 and the values of $\{v_i^{(l)}\}$.
4. Compute $\epsilon = \|W^{(l+1)} - W^{(l)}\|$ where $\|\cdot\|$ is an appropriate matrix norm [GL89].
If $\epsilon \leq \epsilon_{min}$ stop; otherwise, set $l = l + 1$ and return to step 2.

The next stage of the new algorithm is to translate the values of the fuzzy histogram partition, W , into a meaningful visual output. A novel colour blending approach, which does this very task, is described in the following section.

5.2.2 Display of Fuzzy Segmented Data: Colour Blending

Colour coding has been the popular approach for the display of fuzzy classified MR image data. Bezdek *et al* [BHC93] describe an elementary method which first “defuzzifies” the fuzzy partition. This is done by assigning to each image element (i.e. pixel or voxel, as defined in section 2.1) unique membership of the cluster in which it has the highest fuzzy membership (see section 6.2.1). The defuzzified image is then displayed by assigning to each cluster a particular colour. This approach ignores any benefit gained by adopting a fuzzy approach to classification and has therefore been superseded by a method described by Hall *et al* [HBC⁺92]. They also defuzzified the classified image data and assigned a particular colour to each cluster. However, the brightness of the colour which they assigned to each image element was set proportional to its highest membership value.

The above described approaches are limited by the degree to which they are able to convey the full implications of the fuzzy classification in the final visual display. This is because a fuzzy classified image will have elements which may belong to more than one cluster. The approach developed by Hall *et al* only displays the degree to which any pixel belongs to the cluster of its highest membership and hides any information regarding the degrees of memberships in other clusters. It is for this reason that we propose the following colour blending scheme.

Our technique involves working with the RGB colour model and assigning to each cluster a particular colour. Figure 5.1(a) shows the colour space defined by the

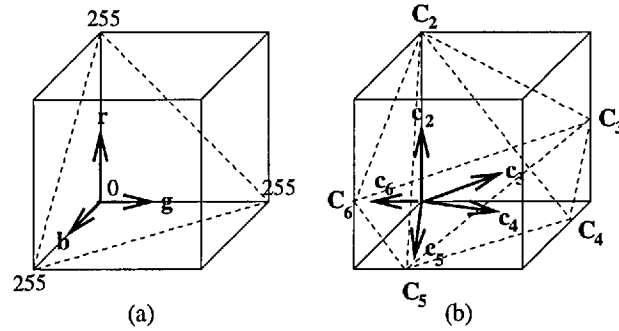


Figure 5.1: Colour spaces for segmented images with (a) 4 clusters and (b) 6 clusters

RGB model. The particular model used is made up of a discrete 8 bit space. It has three principal axes, defined by the unit vectors $\mathbf{r}$, $\mathbf{g}$ and $\mathbf{b}$, which correspond to the colours red, green and blue respectively. The origin of the space corresponds to the colour with no RGB components, namely black. Each principal axis has values ranging from 0 to 255. As the values increase along, for example, the $\mathbf{r}$ axis, the brightness of the colour red increases until it reaches its maximum when $r = 255$ and similarly for the other axes. The extremes of each axis may be defined by the vectors $\mathbf{R}$, $\mathbf{G}$ and $\mathbf{B}$. Where $r=g=b=255$, i.e. $\mathbf{R}+\mathbf{G}+\mathbf{B}$ - the corner farthest from the origin, the RGB space has reached full saturation and therefore corresponds to the colour white.

Consider an MR image after its data has been classified into four fuzzy clusters. One of these four clusters will correspond to the region of the image with a very low

MR signal³. There is no need to assign this region any colour at all. The other three clusters may then be arbitrarily assigned the colours red, green and blue. Thus, after performing the FFCM algorithm, any image element of grey scale value g will be assigned the colour defined by the vector $\mathbf{Y}_g$, where

$$\mathbf{Y}_g = w_{2g}\mathbf{R} + w_{3g}\mathbf{G} + w_{4g}\mathbf{B}, \quad 0 \leq \sum_{i=2}^4 w_{ig} \leq 1 \quad (5.5)$$

This allows elements which have memberships in more than one cluster to be viewed as such by the blending of the “basic” colours assigned to the individual clusters. Due to the constraints of a fuzzy partition, as defined by equation 2.6, not all the RGB space may be utilised for colour blending. The space used is the volume which contains all the possible vectors $\mathbf{Y}_g$. This is the volume confined by the tetrahedron joining the origin and the vertices of the three principal axes as shown in figure 5.1(a).

If MR image data is classified into more than four clusters, ideally a colour space which is of more than three dimensions should be used if it is to possess more than three linearly independent principal axes. However since the RGB model is only three dimensional, a pseudo multi-dimensional space must be defined within the existing 3D RGB space. Figure 5.1(b) shows an example of such a space which has been created for image data which has been classified into six clusters. As with the data classified into four clusters, the cluster corresponding to no MR signal is not assigned any colour. The other five clusters are assigned “basic” colours according to the axes $\mathbf{c}_2$ to $\mathbf{c}_6$, which are spaced equally apart along the faces of the r - g , g - b and b - r planes. A pixel with unique membership to cluster i , is assigned the colour defined by the vector $\mathbf{C}_i$. The entire space available for colour blending is the sum of all the tetrahedral volumes as depicted in figure 5.1(b). For the general case, where the image data is classified into c clusters, the colour assigned to any pixel of grey

³This is the region of the image with pixels of zero or very low grey scale intensity.

scale value g , $\mathbf{Y}_g$, is defined by the equation

$$\mathbf{Y}_g = \sum_{i=2}^c w_{ig} \mathbf{C}_i, \quad 0 \leq \sum_{i=2}^c w_{ig} \leq 1 \quad (5.6)$$

The overall process of the new algorithm is summarised below in figure 5.2.

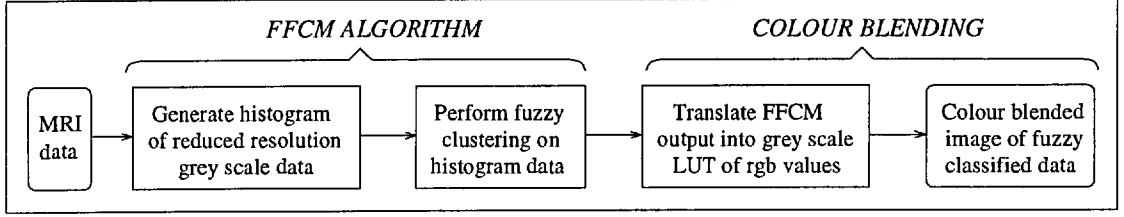


Figure 5.2: The New Classification Algorithm

5.3 Alternative Classification Methods

In order to assess the performance of our proposed classification algorithm, it is important to compare it with existing proven techniques. For this purpose, we have chosen two of the most popular unsupervised approaches. These are the EM algorithm, used to match the data to a Gaussian mixture model, and the HCM (often referred to as the K-means) clustering algorithm. In order for the results to be comparable to those from the FFCM algorithm, only grey scale data has been considered.

The EM algorithm may be used to solve the ML equations for the parameters defining a Gaussian mixture model which best fits the image data. This is described in detail, in section 2.2.4. The output values of this algorithm may be represented as a $(c \times n)$ matrix, $P[p(i | x_k)]$, where $1 \leq i \leq c$ and $1 \leq k \leq n$, c is the number of classes and n the number of image elements. As discussed earlier in section 2.2.3, these probability values may be considered as membership values. Thus, the output of this algorithm may be visualised using the colour blending scheme described in section 5.2.2.

The HCM algorithm partitions the data so that the objective function of equation 2.12 (see chapter 2, section 2.2.4) is minimised. The algorithm recomputes cluster centroids iteratively so that their distance from the cluster data is minimised. Details of the full algorithm may be found in [Bez81, DH73]. It should be noted that unlike the FFCM, the HCM only performs a hard classification. Thus, it is unnecessary to use colour blending to display the output. Instead, a specific colour or grey scale may be assigned to each element according to its corresponding tissue type.

5.4 Results and Discussion

In this section, results from performing the new algorithm, on head and knee data sets acquired from both conventional and interventional MR scanners, as described in chapter 4, are presented and discussed. These are then compared with results from alternative classification methods.

5.4.1 The New Algorithm: Initial Conditions

Before being able to implement the FFCM algorithm, it is necessary to choose a value for c , the number of fuzzy clusters, to result from the classification. In addition to setting a value for c , first guess values must be chosen for the fuzzy partition, $W^{(0)}$ (see step 1 of algorithm 5.1). It is possible simply to choose random values, which satisfy equation 2.6. However, we have found that by dividing the grey scale histogram into c equally spaced regions and assigning values to $W^{(0)}$, the algorithm converges smoothly towards a global minimum. Finally, it is necessary to set the value of the fuzzy index, m , and to decide on a value for the stopping criterion, ϵ_{min} , in stage 4 of algorithm 5.1.

5.4.2 Performance of the New Algorithm

Accuracy of Classification

In order to investigate the accuracy of the new algorithm, we will first consider the conventional MRI knee data shown in chapter 4, figure 4.1(b). Figure 5.3(a) shows the histogram, f , of the original 12 bit grey scale data from the knee image (as shown in figure 4.1(b)), and figure 5.3(b) shows the histogram, f , of the reduced resolution 8 bit data of the same image. It should be noted that the reduction in resolution does not have any significant effect on the shape of the histogram data.

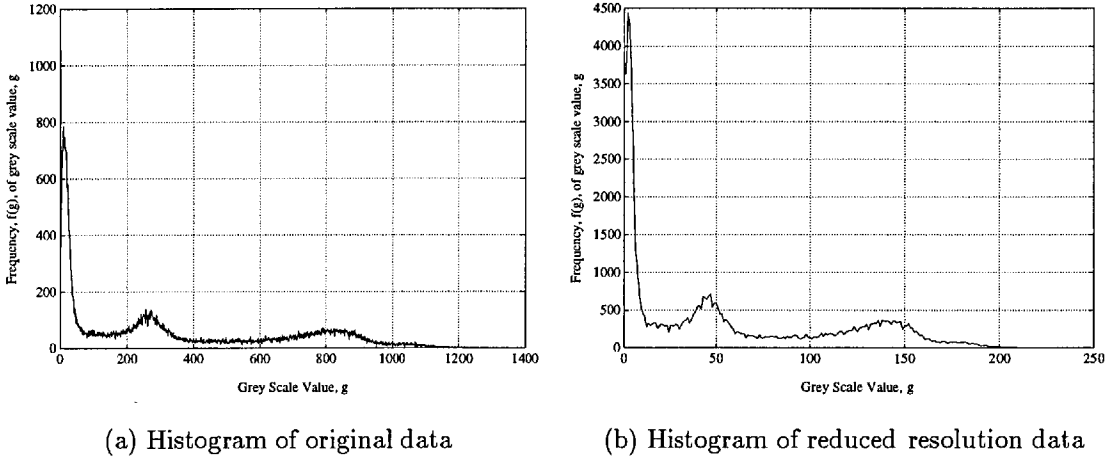


Figure 5.3: Plots of histograms from T1 knee data (as shown in figure 4.1(b)) from (a) original 12 bit image and (b) reduced resolution 8 bit image.

Figures 5.4(a) and (b) show plots of the final W partitions after the clustering stage of the FFCM has converged, after inputting 12 bit and 8 bit histogram data, respectively. The shapes of the two plots are very similar despite the lower resolution of the histogram of figure 5.3(b). This is because the essential shape of the histogram was preserved even after reducing the grey scale resolution. Finally, figures 5.5(a) and (b) show the colour blended results after using both the original 12 bit data and the reduced resolution 8 bit data. Visually, these images appear to be identical. The actual difference between the image values was computed and the insignificant error

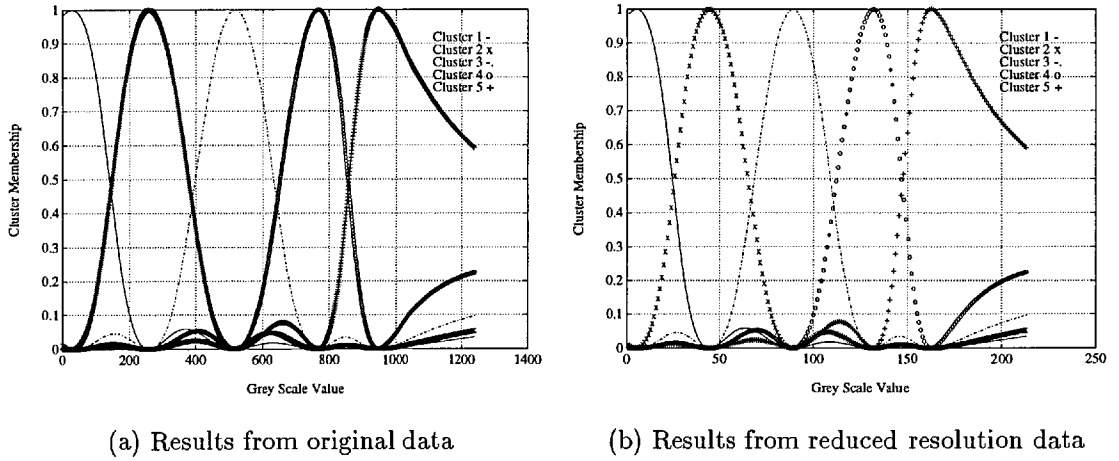
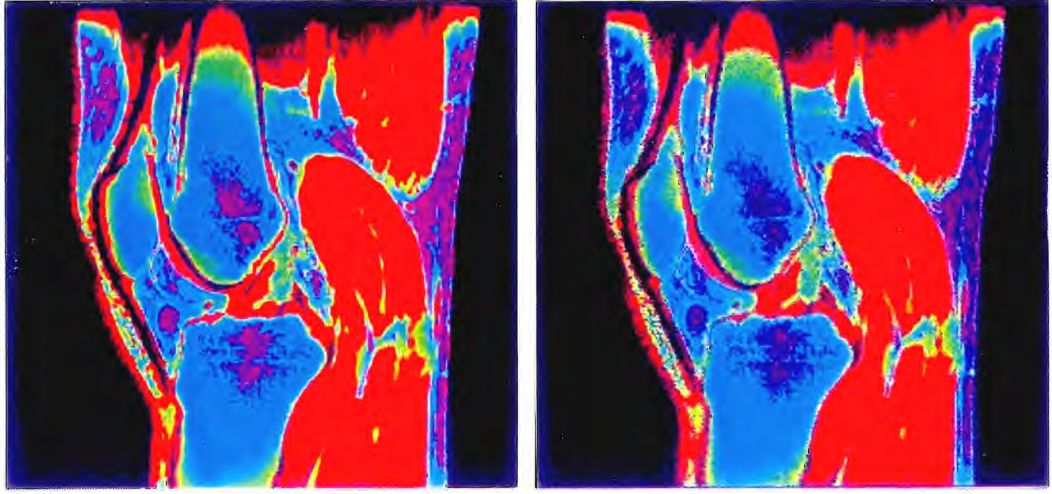


Figure 5.4: Plots of Cluster Memberships after performing FFCM on T1 knee data (as shown in figure 4.1(b)) using grey scale histogram generated from original 12 bit image and from reduced resolution 8 bit image.

of 0.77% was measured. It should be noted that the only error introduced to the FFCM algorithm is that resulting from reducing the grey scale resolution. Otherwise, it produces an identical result to its original FCM counterpart. The accuracy of the FFCM algorithm has been assessed on a wide range of both conventional and interventional MRI data. In all cases the FFCM produced visually identical results compared with those from its FCM counterpart. The actual quality of these results will be discussed shortly.

We have chosen to use the Euclidean distance measure for d_{ig} , the distance between grey scale value g and the centroid of the i 'th cluster, v_i . The Euclidean measure for distance is the simplest and least computationally involved measure. In our case where only single feature data is being clustered, the calculation of the Euclidean distance requires only a single arithmetic operation per iteration. Furthermore, the resulting classification is extremely good. When choosing the Mahalanobis distance [Fuk90] measure, as well as requiring more computation, we found that the iterative minimisation procedure became unstable and poor results were obtained.



(a) Slice 15 of knee data after FCM

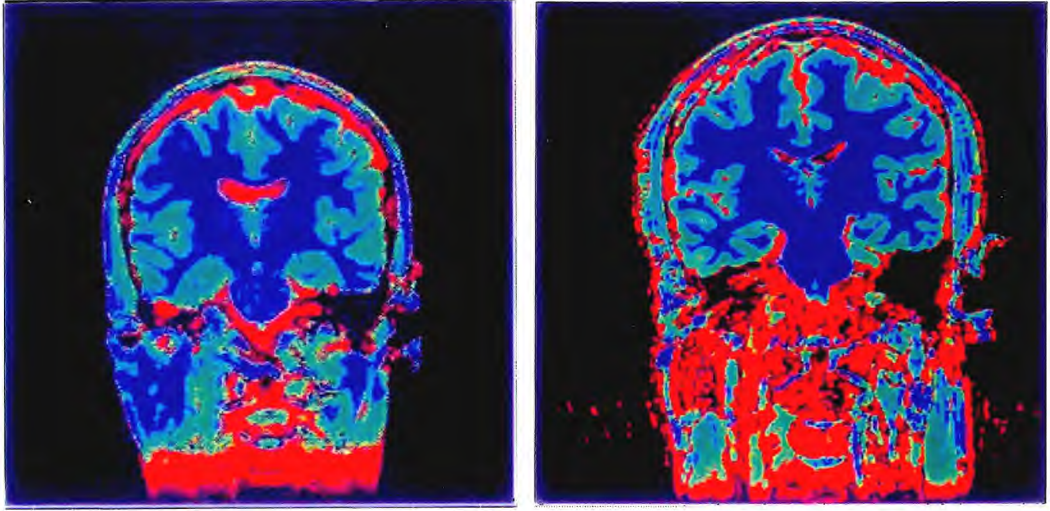
(b) Slice 15 of knee data after FFCM

Figure 5.5: Colour blended display of classified knee data (as shown in figure 4.1(b)) after 25 iterations, $c = 5$, $m = 2.0$.

Quality of Classification

In order to determine optimal values for the number of clusters, c , an expert radiologist examined segmented images, which were classified according to different values of c . It was found that for head data, depending on the image in question, setting the value of c to either 4 or 5 provides an output which differentiates between the main structures most clearly. However, for knee data, setting $c = 5$ consistently yields the best results.

Figures 5.6(a)-(b) show the colour blended outputs of classified head data acquired by both MRI and IMR scanners. Pixels corresponding to fluid tissue are assigned the colour red, those corresponding to grey matter, green and those to white matter, blue. The cluster corresponding to very low MR signal was not assigned a colour. It may be observed from figure 5.6(b) that despite the reduced SNR of the original IMR data, only slight misclassifications may be observed in the background. The results shown in figures 5.6(a)-(b) are from performing the FFCM on 2D image data only. When applying the FFCM to the entire 3D volume at once,



(a) Slice 62 of MR data after FFCM

(b) Slice 62 of IMR data after FFCM

Figure 5.6: Colour blended display of classified head data after 25 iterations, $c = 4$, $m = 2.0$.

it was found that almost identical results were obtained with the first MR head set. However, this was not the case with the data acquired by the IMR scanner. Figure 5.7(a) shows the colour blended output, of the same image slice as shown in figure 5.6(b), after applying the FFCM to the entire volume, setting $c = 4$. The resulting output is incorrectly classified, particularly in the brain area where the cerebrospinal fluid, grey matter and white matter are not correctly delineated. Figure 5.7(b) shows the equivalent output of 5.7(a), but with $c = 5$ instead of 4. In this case a very good classification results. Since $c = 5$, the resulting colours of the different tissues differ slightly. The pixels corresponding to fluid tissue are again assigned the colour red. However, those corresponding to grey matter are assigned the colour green/yellow, those corresponding to white matter, cyan, and a few isolated pixels outside the brain, purple. It may be seen that less background noise is prevalent in the resulting classification when performing the FFCM on the entire volume than when implementing it on a single image slice. The cause of the discrepancies between 2D and 3D classification, in this case, is the presence of an additional tissue type in the other slices of the volume. This is due to the particular

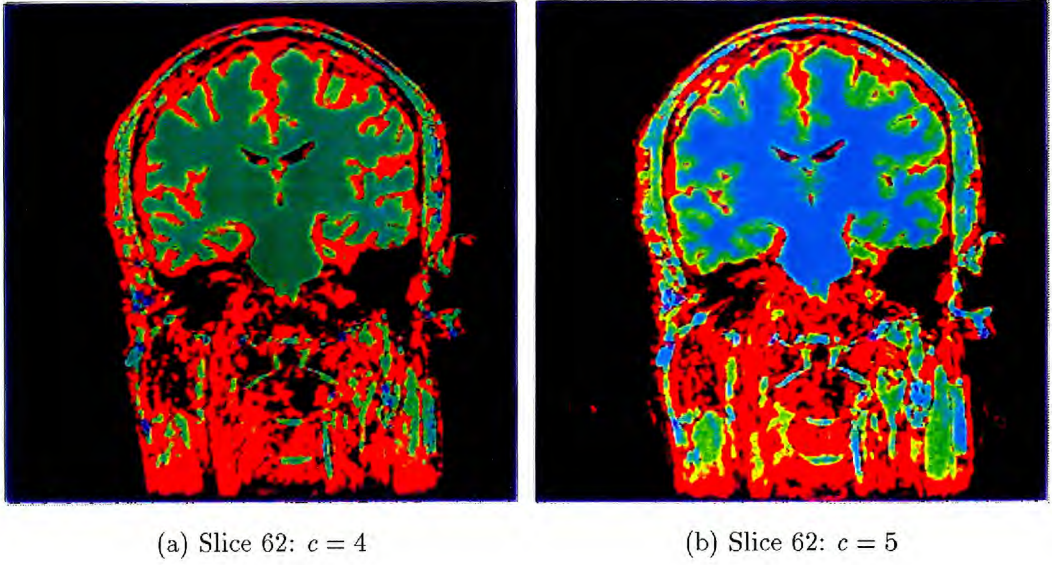


Figure 5.7: Colour blended display of classified IMR head data, with FFCM applied to entire volume, after 100 iterations, $m = 2.0$.

anatomy of the subject in question and is regardless of whether the data is acquired by either a conventional or interventional scanner.

The resulting allocation of the “basic” colours in the classified conventional MRI knee data, as shown in figures 5.5(a) and (b), is as follows: Muscle type tissue pixels are red, the pixels corresponding to high density marrow or fat tissue are purple, those corresponding to medium density marrow or fat tissue are cyan and those corresponding to low density marrow or fat tissue are coloured green/yellow. Finally, the pixels which correspond to tendon, cortex or air have not been coloured. When applying the FFCM to the IMR knee data shown in figures 4.2(b)-(d), the resulting tissue classification was not as accurate. This is due to the poor contrast and grey scale inhomogeneity of the acquired image data. As discussed in chapter 4, the quality of the IMR knee data is limited by the wrap around localiser coils used during imaging. It is almost certain that with the availability of superior coils, it will be possible to acquire higher quality knee data. Nevertheless, it has been demonstrated above, with the IMR head data, that IMR data, when acquired

properly, may be classified accurately by the FFCM algorithm.

It should be noted from figures 5.5, 5.6 and 5.7 how the colour blended display allows the various colours to blend into each other as structures transcend from one to another. It should be pointed out, that although the colour allocation used in our work was calculated from equation 5.6, the actual colour assignment may be chosen manually instead. Thus, clinicians can label each tissue according to the colour of their choice. So long as the contrast between the chosen colours within RGB colour space is reasonable, the fuzzy blending between the different structures will be maintained.

The value chosen for the fuzzy weighting index, m , alters the degree of “fuzziness” of the classified data. It has been suggested [PB95] that choosing $m \in [1.5, 2.5]$ produces an optimal classification. We found that setting $m = 2.0$ gave the best result. Smaller values of m led to a more “hard” classification, whereas higher values introduced noise into the segmented data. This may be easily understood in the light of equation 5.4. As $m \rightarrow \infty$, $w_{ig} \rightarrow (1/c) \forall i, g$, i.e. the image reverts to noise.

Finally, the results of the new algorithm, when applied to several conventional MRI and IMR knee and head image sets, were studied by a radiologist. His conclusions were, that with the exception of the IMR knee data which was poorly acquired, the structures delineated by the classification procedure corresponded extremely closely with the actual anatomy.

Convergence Properties

The way in which the objective function, $J_m(W, \mathbf{v})$, converges towards a global minimum is demonstrated in figure 5.8. It was found that a value for the stopping point, ϵ_{min} , which yields a good classification, varies from image to image. However, we found that by choosing a sufficiently large number of iterations for the cluster-

ing stage of the algorithm, regardless of the value of ϵ_{min} , $J_m(W, \mathbf{v})$ consistently converged with a wide range of different data.

It may be seen, from figure 5.8(a), that with the head data, after about 20 iterations, the objective function, $J_m(W, \mathbf{v})$, approaches very close towards its global minimum. Although it does progressively decay after further iterations of the clustering algorithm, the change is very minor and there is no visual difference in the resulting segmentation. Figure 5.8(b) shows how, with the knee data, $J_m(W, \mathbf{v})$ almost reaches its global minimum after only 10 iterations. The reason for this is that with the knee data, the first guess values for $W^{(0)}$ are closer to the final optimal values of W than with the head data.

It is impossible to predict exactly how many iterations are likely to be required for the algorithm to converge. However, because of the speed of our method, we can afford to be somewhat conservative and always allow for 30 iterations. This should guarantee convergence with all types of image data.

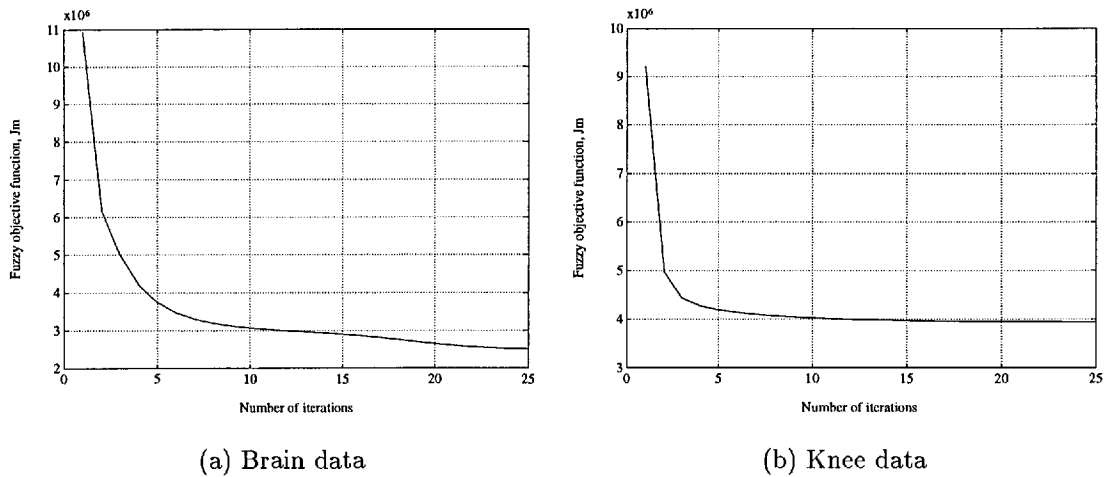


Figure 5.8: Plots of $J_m(W, \mathbf{v})$ throughout 25 iterations of the FFCM algorithm.

Computational Efficiency

The actual computation times of the new algorithm, as well as those of both the traditional FCM, and the FFCM without reducing the 12 bit resolution of the data, were measured when implementing them on a DEC Alpha AXP/200 workstation. Table 5.1 shows the computation times after 30 iterations with c set to 5. Since our algorithm uses the histogram data as input, as opposed to the FCM which uses actual image data, the computation time was independent of the amount of voxels in the 3D image data. Thus, the FFCM was able to segment large 3D data sets in under one second! Although, using 12 bit histogram data as input to the FFCM took only 2.5 seconds, the 16 times improvement in efficiency which results from using 8 bit data is significant, providing an almost instantaneous result. It should be noted that due to the spatial homogeneity of the magnetic field in an MR scanner, we are able consider the histogram of the entire data set all at once, rather than that of each slice individually. Although there may be some inhomogeneities towards the periphery of the image field, these are insignificant and hardly affect the overall grey scale distribution.

	Original FCM algorithm	FFCM (using 12 bit histogram data)	FFCM (using 8 bit histogram data)
2D Image (256x256)	90 seconds	2.5 seconds	< 1 second
3D Image (124x256x256)	> 3 hours	2.5 seconds	< 1 second

Table 5.1: Performance times of FCM and FFCM algorithms

5.4.3 Appraisal of Alternative Methods

A. EM Algorithm: Solving ML Equations for Gaussian Mixture Model

As for the FFCM, when initiating the EM algorithm, it is necessary to fix the value of c in advance. In addition, first guess values for $P[p(x_k | i)]$ must be chosen. These

may be set by either dividing the grey scale histogram equally into c clusters and setting $P[p(x_k | i)]$ to equal 0 or 1, depending on whether x_k lies in cluster i or not. Alternatively, values for $P[p(x_k | i)]$ may be set using the defuzzified values of the matrix W which is output from the FFCM algorithm. In either case, the values of α_i are determined according to the number of pixels existing in each cluster. The first round values for $P[p(i | x_k)]$ are then computed using equation 2.10. We have set the initial conditions according to both strategies.

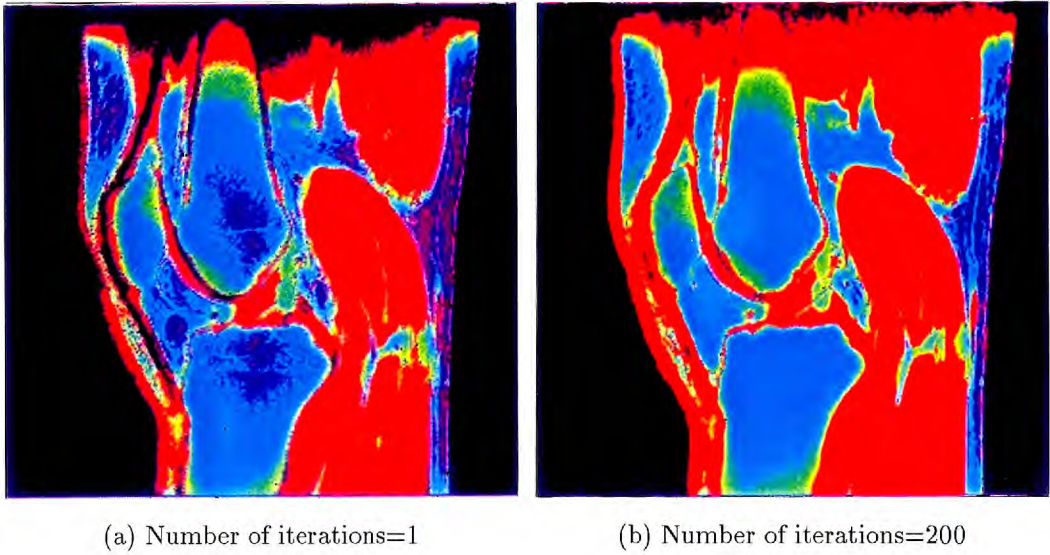


Figure 5.9: Colour blended output of EM algorithm applied to conventional MRI knee data (slice15 - as shown in figure 4.1(b)), $c = 5$.

Figure 5.9 shows colour blended outputs of the EM algorithms when applied to conventional MRI knee data. The result in figure 5.9(a) is obtained after only one iteration of the EM algorithm when using the defuzzified values of the output matrix W , produced by the FFCM algorithm, as first guess values for $P[p(x_k | i)]$. As one would expect, this result is extremely similar to that produced by the FFCM as shown in figure 5.5. However, as the EM algorithm iterates and the solutions for the ML estimates for the Gaussian parameters converge to a maximum, the resulting classification becomes steadily worse. Figure 5.9(b) shows the output of

the EM algorithm after 200 iterations have been performed. By this time, the solutions have completely converged. It may be seen that the resulting classification is unsatisfactory. Background noise is emphasised and the delineation of certain structures, particularly the cruciate ligaments and the patellar tendon, is poor.

In order to assess the accuracy of the EM algorithm quantitatively, we computed the Gaussian mixture distribution, as defined by equation 2.8, using the resulting estimated parameter values. Figure 5.10 shows plots of the computed Gaussian

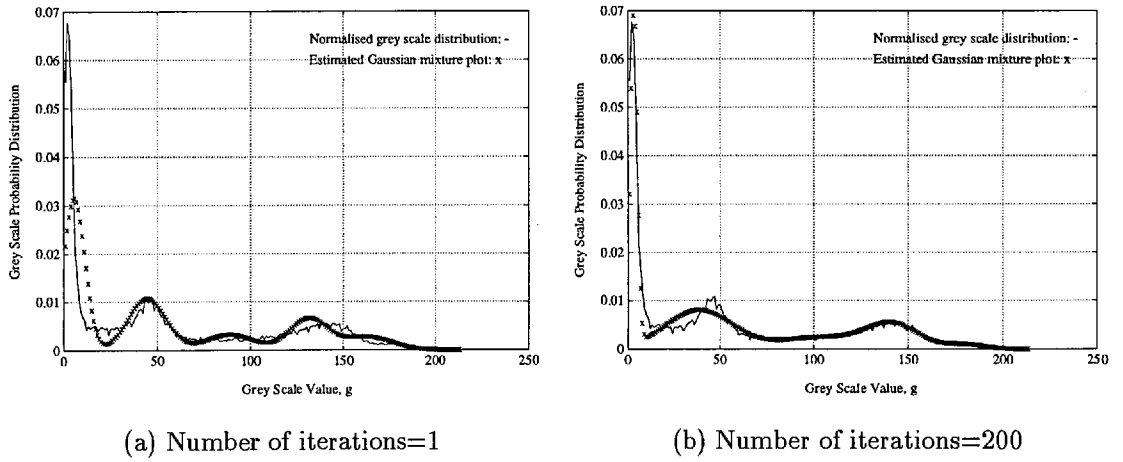


Figure 5.10: Plots of estimated gaussian mixture distributions superimposed over normalised grey scale histograms from T1 knee data (as shown in figure 4.1(b)) after applying EM algorithm to data.

distribution superimposed over plots of the normalised grey scale histogram. After only one iteration of the EM algorithm, the resulting Gaussian mixture plot does not fit the normalised histogram plot very well, as depicted by figure 5.10(a). In order to determine the discrepancy between the estimated Gaussian distribution and the normalised histogram, the percentage difference between the two plots was calculated. A discrepancy error of 43.37% was measured. As the EM algorithm iterates, this error reduces and the resulting Gaussian mixture fits the normalised histogram more closely. However, even when the algorithm has converged, a discrepancy error of 20.44%, between the two distributions, remains.

We have implemented the EM algorithm on a large range of both conventional and interventional MRI data. In all instances, the final result was visually extremely poor. In fact, the results displayed in figure 5.9 were better than those obtained from other image data. Figure 5.11 shows more typical results from the EM algorithm. In this case, the algorithm was applied to MRI head data, and was initiated using both schemes described above. In both cases, the resulting Gaussian mixture plots, after the algorithm had converged, were extremely similar. However, the estimated parameter values were quite different. This is evident by the clearly visible difference between the results shown in figures 5.11(a) and (b). This suggests that when applying the EM algorithm to MRI data, it is susceptible to getting trapped in local minima.

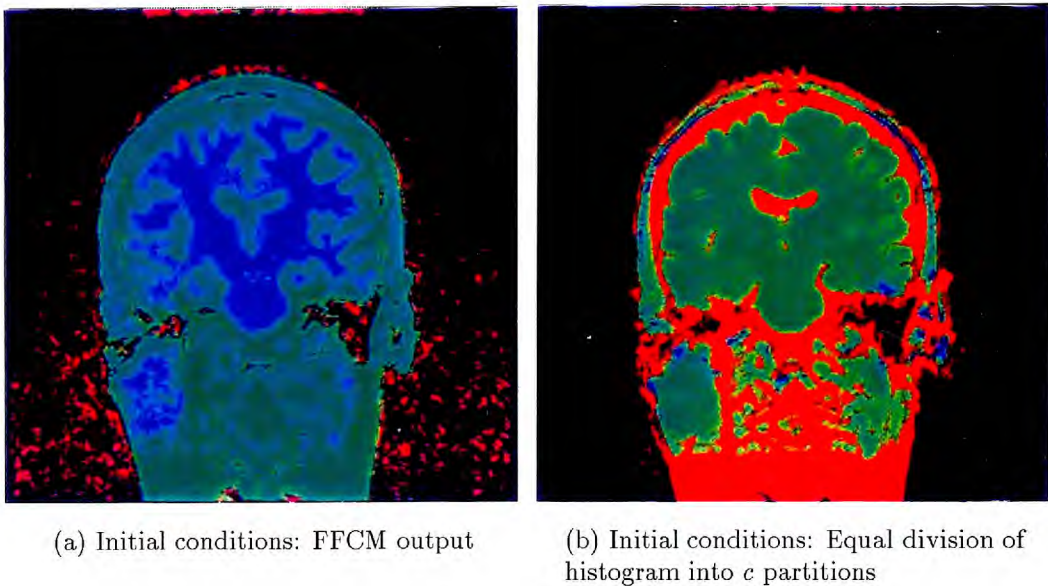


Figure 5.11: Colour blended output of EM algorithm applied to conventional MR head data (slice 62 - as shown in figure 4.1(a)), $c = 4$, number of iterations=200.

From the above results, we may deduce that the distribution of grey scale data in MR images is non Gaussian. The best classification results produce the poorest fit between their resultant Gaussian mixtures and the normalised grey scale histogram. Despite the widespread use of parametric techniques for image classification, our re-

sults suggest that non-parametric techniques are likely to produce the most accurate results for MRI data classification.

B. HCM Clustering

In order to assess the performance of the HCM algorithm, we compared its results with those obtained from FFCM classification after defuzzification (see section 6.2.1 for details). In many instances, the results were almost identical. However, in some cases the HCM gave a poor result. Figure 5.12(a) shows an example of where the HCM goes wrong when applied to the head data of figure 4.1(a). In this case the

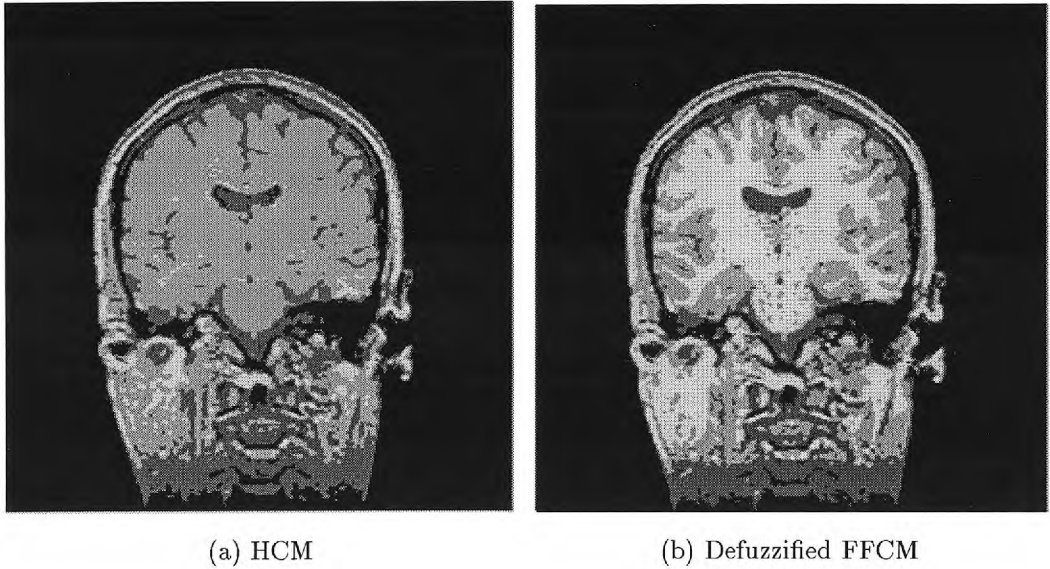


Figure 5.12: Slice 62 of conventional MR head data set after performing HCM and FFCM classification. $c = 4$, number of iterations=25.

structures of the brain have not been correctly delineated. Figure 5.12(b) shows the defuzzified result of the FFCM algorithm, applied to the same image data, which shows a clear distinction between the anatomy. The reason why the HCM lacks the consistency of the FFCM is that due to the finite nature of hard partition space, the objective function is liable to getting trapped in unwanted minima during the clustering process. The infinite nature of fuzzy partition space prevents such unde-

sirable solutions from occurring. Another basic limitation of the HCM algorithm, is that its hard output cannot capture the inherent fuzziness of the original image data.

5.5 Conclusions

The new FFCM algorithm has been shown to classify MRI data quickly and accurately. The best attempt by previous researchers [CDB86] to improve the speed of the FCM only did so by a factor of six. Our new method has reduced the computation of the FCM from several hours to less than one second, yielding indistinguishable results.

The choice of only the grey scale value as a single feature input to the clustering stage of the algorithm has proven to yield extremely good results. The algorithm has been tested on both conventional and interventional MR image data. In both cases, after assessment by an expert radiologist, the classified data was shown to correspond very closely to the anatomical tissue types within. Furthermore, the FFCM algorithm is able to capture the fuzzy nature of the data which may be best displayed by adopting the colour blending scheme. In addition, the fuzzy nature of the classification improves the quality of subsequent volume renditions.

We have compared the FFCM with the the EM and HCM algorithms. The FFCM algorithm has consistently been shown to produce superior results. Furthermore, the distribution of the grey scale data in MR images has been shown to be non Gaussian. This corroborates our results and explains why adopting a non-parametric approach, such as the FFCM, produces an optimal classification. It should be noted that we have only compared the FFCM with alternative unsupervised approaches. This is because we are only interested in automatic methods. Supervised approaches, by their very definition, require expert interaction. Thus, we have not explored the different available methods.

Although the new classification scheme has been shown to produce very accurate results, it is not always straightforward to extract anatomical structures from the classified data directly and accurately. This is partly due to misclassification errors, particularly in the periphery of the image, caused by the inhomogeneity of the field in these regions. However, a more significant factor is that a unique structure may comprise more than one tissue type. This is particularly true with knee data. It is therefore necessary to refine the results produced by the FFCM if an accurate segmentation is to be performed. The next chapter describes a novel method which, through a hierarchical process of refinement, is able to isolate anatomical structures from both 2D and 3D data sets, both accurately and quickly.

Chapter 6

Adaptive Refinement of Classified MRI Data

6.1 Introduction

This chapter describes a novel approach for fast and automatic extraction of anatomical structures from MR image data. This is achieved by the adaptive refinement of FFCM classified MRI data. The rest of this section will describe the motivating factors for adopting such an approach as well as an overview of the entire method.

6.1.1 Motivation

In order to visualise an anatomical structure clearly in three dimensions, it is first necessary to isolate it from its surrounding structures. This requirement is true, whichever 3D rendering scheme may be adopted subsequently, as explained in section 3.2.3 and demonstrated in the next chapter. In addition, the accurate definition of anatomical regions is necessary for several other applications such as quantitative image analysis and image registration.

In chapter 5, it was shown how the FFCM algorithm is able to classify MR image data very quickly and accurately. Nevertheless, as concluded in section 5.5,

it is not always possible to translate the resulting classified data into regional data, which accurately define the anatomical regions within the original MRI data, directly. This is because some anatomical regions comprise more than one tissue type. Furthermore, the FFCM method assumes that the distribution of grey scale values alone provides sufficient information for accurate tissue classification in MRI data. Although, this is a fair assumption, it is not always true. In some areas of an image, particularly in its periphery, the grey scale values may be spatially inconsistent due to the presence of noise or field inhomogeneities.

We therefore propose an original approach which refines the result of the FFCM to produce a more accurate segmentation. Since the regional data resulting from the FFCM is accurate in most parts of the image, it is only necessary to refine the incorrectly segmented regions. An adaptive technique is developed which, through a hierarchy of progressively more sophisticated image processing operations, is able to segment both 2D and 3D MRI data automatically, accurately and quickly.

6.1.2 Overview of the Adaptive Refinement Process

The entire adaptive refinement (AR) process for segmenting 2D image data is summarised by the flow diagram in figure 6.1. There are three main stages in the process. The first stage performs preliminary processing tasks on FFCM classified data to convert it into regional data. During the second stage, a region of interest is selected by a user and then prepared for the subsequent and final stage of adaptive region refinement. The last stage considers the boundaries of the selected region and determines whether or not to merge it with one or more of its neighbours or to split the region itself. The adaptive region refinement process is an iterative procedure which repeats itself until no more region merging or splitting needs to be performed. An accurately segmented region then results.

Sections 6.2 to 6.4 describe the above mentioned stages. The extension of the AR approach to 3D segmentation will be described in section 6.5.

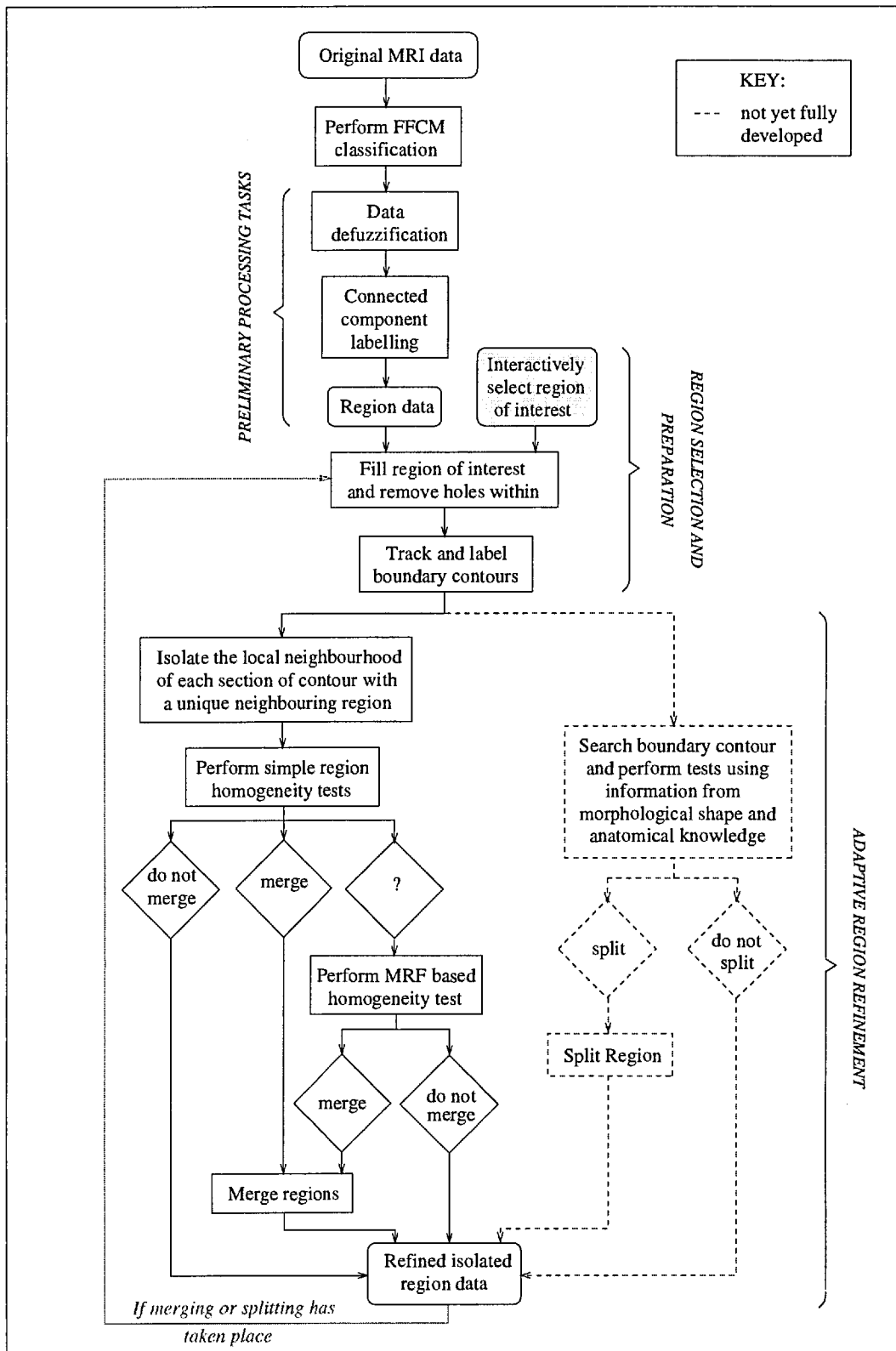


Figure 6.1: A flow-chart summary of the Adaptive Refinement (AR) Process.

6.2 Preliminary Processing Tasks

In order to implement the AR process, it is necessary to perform a number of processing tasks on the fuzzy classified MRI data in advance. In this section, the required steps of data defuzzification and connected component labelling will be described.

6.2.1 Data Defuzzification

Chapter 5 describes the FFCM algorithm and how it may be implemented on MR image data. One of the advantages of adopting a fuzzy approach, is that it enables the inherent fuzziness of the data to be captured by assigning to each image element memberships for more than one tissue type. However, although this is useful for subsequent volume rendering, this is not the case for extracting regional data in order to isolate anatomical structures uniquely. Udupa and Samarasekera [US96] have recently proposed a method which utilises fuzzy connectivity theory for the extraction of fuzzy regions (see chapter 2). However, it is not possible to convert fuzzy classified data to fuzzy region data directly. Furthermore, the theory for fuzzy region definition is still in its infancy and the extraction methods proposed are computationally intensive.

We therefore defuzzify the fuzzy classified data, so that it should be in the appropriate hard form for subsequent region data processing. This may be achieved as follows:

After applying the FFCM algorithm to the image data, a fuzzy grey scale partition results which is stored as a W matrix (see section 5.2.1). Each entry, w_{ig} , of the $(c \times n)$ W matrix is defuzzified by recomputing its value:

$$w_{ig} = \begin{cases} 1, & \text{if } w_{ig} > w_{lg}, \forall l \in \{1, \dots, c\} - \{i\} \\ 0, & \text{otherwise} \end{cases} \quad (6.1)$$

where:

c is the number of clusters in the image;

$n = (G_{max} - G_{min} + 1)$. G_{min} and G_{max} are the minimum and maximum grey scale values in the image respectively;

$i \in \{1, \dots, c\}$ indexes the cluster;

$g \in \{G_{min}, \dots, G_{max}\}$ is the image element grey scale value.

The defuzzified data is then obtained by labelling each image element according to the class in which its corresponding membership value $w_{ig} = 1$.

6.2.2 Connected Component Labelling

The output of the defuzzified FFCM is an image whose elements are hard labelled according to their corresponding tissue types. Connected component labelling is a process which converts this data into region data. This is achieved by assigning a unique label to each element corresponding to the region in which it belongs.

Image Element Relationships

Before describing the actual procedure of connected component labelling, it is necessary to define a few basic relationships between image elements. Figure 6.2 shows

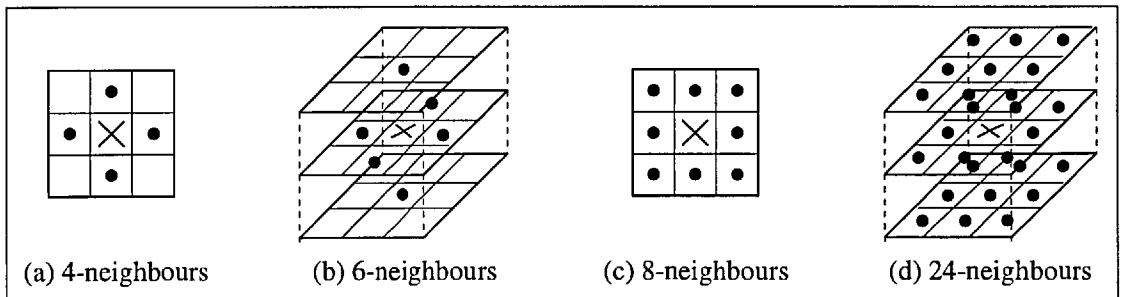


Figure 6.2: Different possible sets of neighbour elements in 2D and 3D image arrays.

different types of neighbour sets that are typically considered when defining regions

in 2D and 3D image arrays (Note: $\bullet$ denotes neighbour of element $\times$). If two image elements are neighbours and satisfy a criterion of similarity, e.g. they are labelled according to the same class, they are said to be *adjacent*. A *path* from element p with co-ordinates (x, y, z) to element q with co-ordinates (x', y', z') is a sequence of distinct elements with coordinates

$$(x_0, y_0, z_0), (x_1, y_1, z_1), \dots, (x_n, y_n, z_n) \quad (6.2)$$

where $(x_0, y_0, z_0) = (x, y, z)$ and $(x_n, y_n, z_n) = (x', y', z')$, (x_i, y_i, z_i) is adjacent to $(x_{i-1}, y_{i-1}, z_{i-1}) \forall i \in \{1, \dots, n\}$, and n is the length of the path. If p and q are elements of an image lattice S , then p is *connected* to q in S if there is a path from p to q . For any element p in S , the set of elements in S which are connected to p is called a *connected component* of S . Hence, any two elements of a connected component are connected to each other, and distinct connected components are disjoint. It is evident that a connected component corresponds to a *region* as defined by equation 2.2. Thus, a region is 4-connected if it is defined by 4-neighbour adjacency criteria. Similarly, a region may be 6-connected, 8-connected or 24-connected.

The Connected Component Labelling Algorithm

There are many different algorithms which have been described to perform connected component labelling on image data [HS92]. We propose a simple approach for 2D image data which utilises the seed fill algorithm described in appendix C.

Algorithm 6.1 The following steps are performed:

1. Set $l = 1$.
2. Perform a raster scan on the classified image data.
3. At each element which has not yet been visited, fill a uniquely class labelled 4-connected region. The element found by the raster scan is used as the seed.

6.2. PRELIMINARY PROCESSING TASKS

The corresponding elements of a label image, of the same size as the classified image, are filled simultaneously with the label l . As each region is being filled, its area and mean are computed as follows:

$$A_l = \sum_{s \in R_l} \quad \mu_l = \frac{1}{A_l} \sum_{s \in R_l} x_s \quad (6.3)$$

where A_l and μ_l are the area and mean of region R_l respectively, and x_s is the grey scale value of the element at site s in the image lattice. These values are required during a later stage of the AR process.

4. Continue the raster scan until a new site, not yet visited by the seed fill algorithm, is reached. Set $l = l + 1$ and go back to step 2 until the raster scan has completed.

This algorithm may be extended easily to detect 8-connected 2D regions, as well as 6-connected and 24-connected regions in 3D volumes. Figure 6.3 shows an example

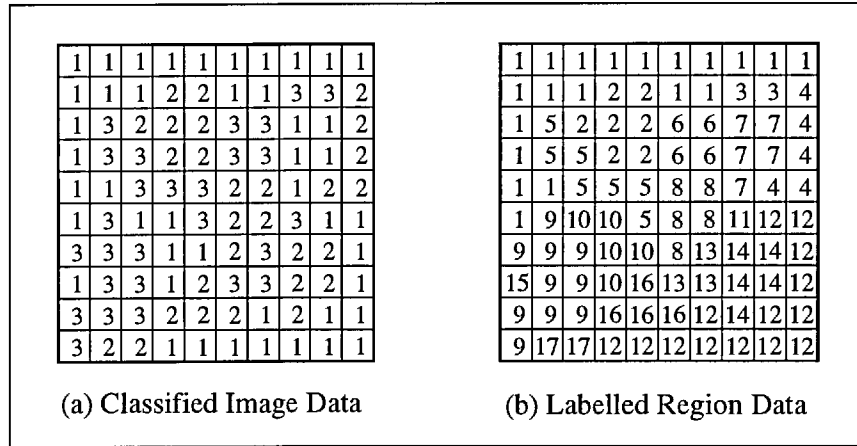


Figure 6.3: An example of the connected component labelling algorithm: (a) Hard classified image data; (b) Labelled 4-connected region data.

of applying algorithm 6.1 to hard classified image data. It should be noted that although only 3 class labels are present in the classified image of figure 6.3(a), 17 regions result as shown in figure 6.3(b).

6.3 Region Selection and Preparation

After the preliminary processing tasks have been performed, a specific region within the image is selected interactively by a user, e.g. with a mouse, so that it may then be refined. In this section, the various operations required to prepare the selected region, for the subsequent refinement stage, are described.

6.3.1 Hole Removal

The region data resulting from the processing stages described in the previous section is invariably rather grainy. Figure 6.4(a) shows an example of a region which suffers from this problem. This is a result of the global nature of FFCM classification

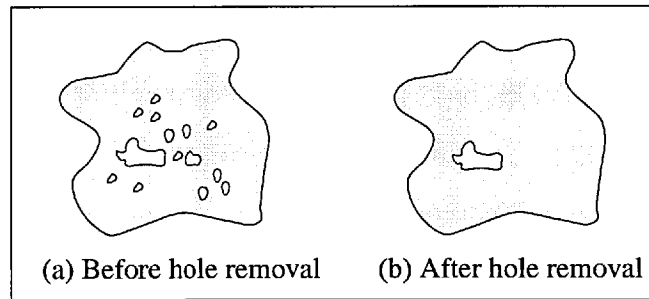


Figure 6.4: An example of a region (a) before and (b) after hole removal.

which does not account for spatial variations and noise in the original image data. In order to rectify this problem and to reduce subsequent processing, we perform the following simple hole removal algorithm.

Algorithm 6.2 (Hole removal) The following steps are performed:

1. Interactively select (e.g. with a mouse) a point within the region of interest.
2. Fill the selected region, R_k , using the algorithm described in appendix C. As the region is being filled, perform the following test on each newly filled region pixel:

6.3. REGION SELECTION AND PREPARATION

Check all the 4-neighbours of the new pixel (whose grey scale value is x_{new}):

If any of the neighbours belong to R_l , $l \neq k$:

If $A_l < H_{max}$ **and** $|x_{new} - \mu_l| < G_{max}$ (where H_{max} is the maximum size of hole to be removed and G_{max} is the maximum difference between x_{new} and μ_l for which hole removal is allowed)

Seed fill region R_l and relabel its pixels according to R_k .

Figure 6.4(b) illustrates the effects of algorithm 6.2 after being applied to the region shown in figure 6.4(a).

6.3.2 Boundary Tracking and Labelling

The AR process involves adaptively refining the boundaries of the regional data resulting from the processing steps described in sections 6.2 and 6.3.1. This is achieved by examining the outer contours of a user selected region and simultaneously performing different image processing tasks. These determine if and how to alter the shape and size of the region. Effective boundary tracking and labelling are an essential prerequisite to this process. This section describes an original method, comprising a host of novel algorithms, which tracks and simultaneously labels the sites defining the boundaries of the selected region.

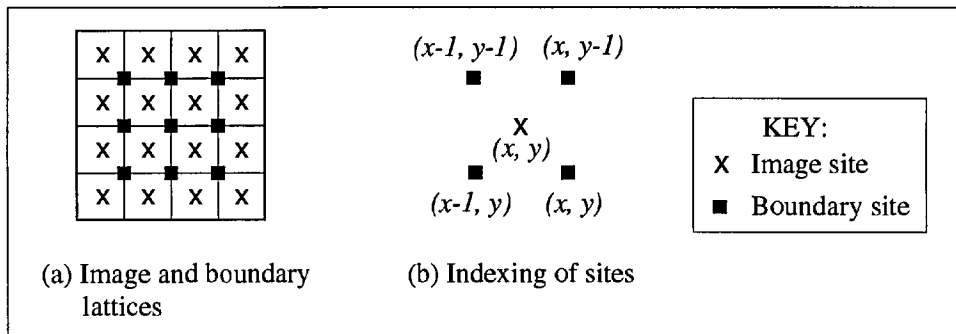


Figure 6.5: (a) Image and boundary lattices and (b) site indexing.

Before detailing any actual algorithms, it is first necessary to define what we will refer to as the *image lattice* and the *boundary lattice*. Figure 6.5 shows how

the boundary lattice is defined in relation to the image lattice. The image lattice represents the positions of the pixels within an image. Each pixel is considered as a small rectangle whose centre defines the location of its corresponding site on the original image lattice. The sites of the boundary lattice are positioned, as shown in figure 6.5(a), along the vertices of the pixel rectangles, each site defining an edge between its four neighbouring pixels. For an image lattice of dimensions $X \times Y$, there exists a corresponding boundary lattice of dimensions $(X - 1) \times (Y - 1)$. The sites of the boundary lattice are indexed in relation to the image lattice as shown in figure 6.5(b).

In order to track and label each boundary of a region, it is first necessary to identify the pixels which define its bounding contour(s). Figure 6.6 shows an example

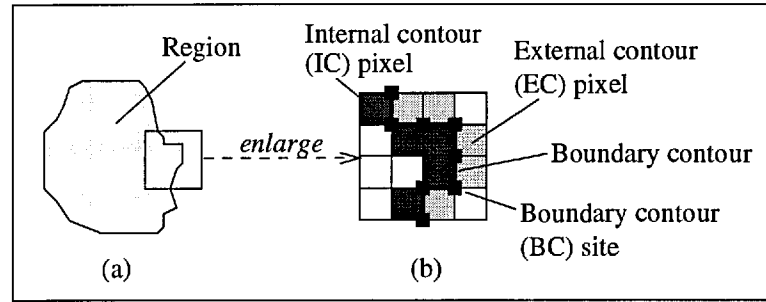


Figure 6.6: (a) A region in an image. (b) A blown up section of the contour which defines the boundary of the region. This is made up of internal contour (IC) pixels, external contour (EC) pixels and boundary contour (BC) sites.

of a region and a blown up view of a section of its boundary. Within the image lattice, it is not possible to define a contour which actually bounds the region. Image pixels may either define the outermost pixels of the region (internal contour (IC) pixels) or the neighbouring pixels belonging to adjacent regions (external contour (EC) pixels). It is therefore essential to define a contour on the boundary lattice, which is made up of boundary contour (BC) sites, in order to track these pixels reliably. With this in mind, the contour detection algorithm may now be described.

Algorithm 6.3 (Contour Detection) The following procedure is performed:

Fill the selected region (after hole removal) using the algorithm described in appendix C. As the region is being filled, perform the following test on each newly filled region pixel:

Check all the 4-neighbours of the new pixel:

If all the neighbours have the same region label as the new pixel

Continue filling the region.

Otherwise

- (a) Label the new pixel as an IC pixel.
- (b) Label each of the 4-neighbours belonging to an adjacent region as an EC pixel.
- (c) Label the boundary sites which intersect the detected IC pixel and its associated EC pixels as BC sites (e.g. see figure 6.6(b)).
- (d) Continue filling the region.

After the internal, external and boundary contours have been detected and marked, it is possible to track the sites of the boundary contour and label them.

Algorithm 6.4 (Boundary Tracking and Labelling) The following steps are performed:

1. Perform a raster scan along the boundary lattice until an unmarked BC site is detected.
2. **While** there exists an unmarked 4-neighbour BC pixel {

Check the 4-neighbour boundary sites, beginning with the eastern site, in clockwise order, as follows:

If a neighbour site is labelled as a BC site **and** it is unmarked **and** the

path of the contour reaching this site intersects an IC and EC pixel as demonstrated in figure 6.7

- (a) Store the co-ordinates of the neighbour BC site.
- (b) Mark the neighbour BC site.
- (c) Record a label for the newly stored BC site as follows:

All four adjacent pixels are checked:

If all the EC pixels belong to the same external region

Record the label of this region.

Otherwise

No label is recorded. (This denotes a junction between two neighbouring regions).

- (d) Stop the clockwise search.

}

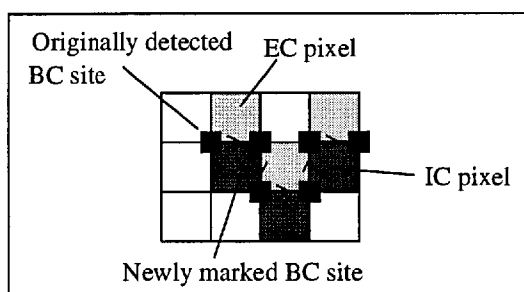


Figure 6.7: An example of tracking the BC sites which define a region boundary.

3. Check whether the tracked contour is closed or open:

If the originally detected BC site is identical to the most recently tracked site

The contour is *closed* as shown in figure 6.8(a).

Otherwise

The contour is *open*: Go back to the first detected BC site and repeat step 2 (see figure 6.8(b)).

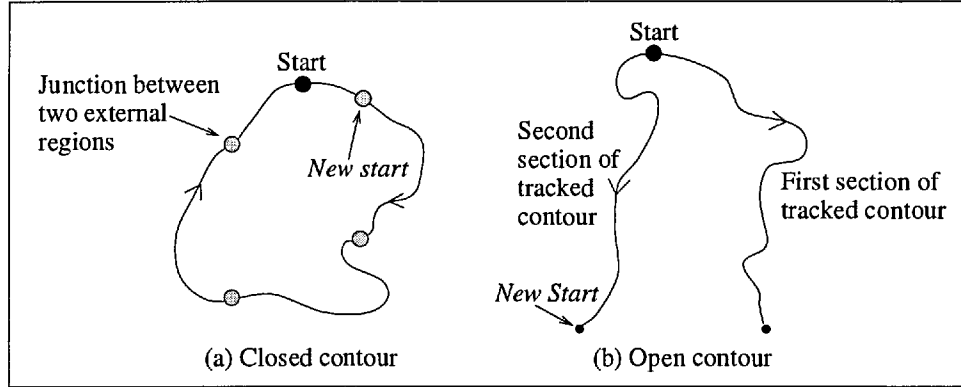


Figure 6.8: (a) Closed contour. Co-ordinates and associated labels are reshuffled so that the first contour co-ordinate is the first detected junction in the clockwise direction. (b) Open contour. Co-ordinates and associated labels are reshuffled so that they begin from one end and finish at the other in clockwise order.

4. Continue raster scan from first detected BC site. If a new unmarked BC site is detected, this indicates that a new contour has been detected. Return to step 2.
5. Go through each set of contour co-ordinates and associated labels and reshuffle them. Examples of closed and open contours which require reshuffling are illustrated in figures 6.8(a) and (b) respectively.

After the BC sites of the selected region have been tracked and labelled it is possible to proceed to the final stage of the AR process, namely adaptive region refinement.

6.4 Adaptive Region Refinement

Adaptive region refinement involves either merging a selected region with its neighbouring external regions or splitting the region itself. In this section the theory required for this process will be described.

6.4.1 Region Merging and Splitting

The purpose of region merging is to rectify over-segmentation inaccuracies which result from FFCM classification. Figure 6.9(a) shows an example of a selected

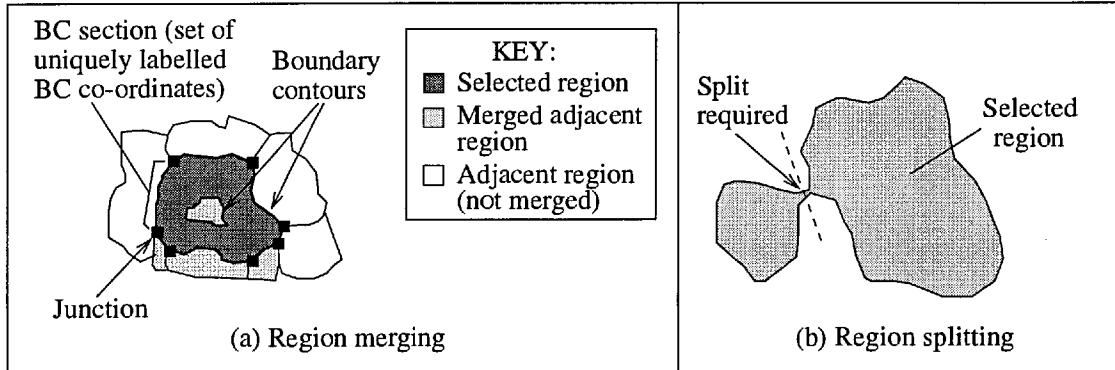


Figure 6.9: Examples of region merging and splitting.

region which is merged with some of its adjacent neighbours. The selected region is defined by two boundary contours. Each section of the outer contour which is adjacent to a specific neighbouring region is defined by a set of uniquely labelled BC co-ordinates which are bounded by a junction site at either end. The inner contour is adjacent to only one region and is therefore considered as a single section. In order to determine which regions should be merged, each contour section is studied and depending on the outcome of homogeneity tests performed in its vicinity, the selected region is either merged or not merged with the adjacent neighbour in question. A hierarchical approach to region merging will be formalised in the following section.

Figure 6.9(b) shows an example of a region which needs to be split. This is, in most instances, a result of two regions connected by a very small region, such as illustrated in this example. In order to determine whether, where and how to split a region, the contour needs to be examined in terms of its general morphological shape. Furthermore, the consideration of specific knowledge may be useful. We have not yet formulated a general method for region splitting. However, ideas for future development are described later in chapter 8.

6.4.2 A Hierarchical Approach to Region Merging

Region merging was discussed in general terms in section 6.4.1. In this section, a hierarchical approach to region merging will be described in detail.

Most region merging methods, as described in section 2.3.3, consider the global characteristics of each pair of regions which are under consideration for merging. We do not adopt this approach during adaptive region refinement. This is due to the nature of the FFCM based initial segmentation. Since FFCM classification is based on grey scale characteristics of the image, two neighbouring regions which have been classified differently will not obviously combine to form a single homogeneous region. Instead, we consider the region homogeneity within a local neighbourhood of each BC section and consequently decide whether or not to merge the two regions. This provides a further advantage of requiring less computation.

The region homogeneity within the vicinity of each contour section is measured by performing an appropriate homogeneity test. In order to optimise the speed and performance of homogeneity testing, we have developed a hierarchical approach. This involves first performing a simple region homogeneity test, $P_s(\varrho_s)$ which is carried out within ϱ_s , the immediate vicinity of the contour section in question. Depending on the outcome of $P_s(\varrho_s)$, the region is either merged, not merged, or if the outcome is undefined, is subjugated to a more sophisticated homogeneity test, $P_{\text{MRF}}(\varrho_{\text{MRF}})$. The test $P_{\text{MRF}}(\varrho_{\text{MRF}})$ performs MRF based labelling within ϱ_{MRF} , a specifically defined neighbouring vicinity of the contour segment, and decides whether or not to merge, according to the result. Full details of the homogeneity tests, $P_s(\varrho_s)$ and $P_{\text{MRF}}(\varrho_{\text{MRF}})$, and their associated neighbourhoods will be described in sections 6.4.3 and 6.4.4 respectively.

We are now in a position to formalise the hierarchical region merging approach as follows:

Algorithm 6.5 (Hierarchical region merging) The following steps are performed on each boundary contour:

1. Isolate each contour section (defined by a set of uniquely labelled boundary contour co-ordinates) as illustrated in the example of figure 6.9(a).
2. Within each contour section, perform the region homogeneity test, $P_s(\varrho_s)$:

If $P_s(\varrho_s) = \text{true}$

Merge regions as follows:

- (a) Remove holes in neighbouring region using algorithm 6.2
- (b) Fill the neighbouring region (see appendix C for algorithm)
and relabel its pixels according to the selected region.

If $P_s(\varrho_s) = \text{false}$

Do not merge regions.

If $P_s(\varrho_s) = \text{undefined}$

Perform MRF based homogeneity test $P_{\text{MRF}}(\varrho_{\text{MRF}})$:

If $P_{\text{MRF}}(\varrho_{\text{MRF}}) = \text{true}$

Merge regions (as for $P_s(\varrho_s) = \text{true}$).

If $P_{\text{MRF}}(\varrho_{\text{MRF}}) = \text{false}$

Do not merge regions.

6.4.3 Simple Homogeneity Testing

In order to define the homogeneity test $P_s(\varrho_s)$, it is first necessary to define the BC section neighbourhood ϱ_s and to compute statistical values within its confines.

The neighbourhood ϱ_s is extracted from the image by simply isolating all the pixels which are immediately adjacent to each BC site of the contour section. Figure 6.10 shows an example of a boundary contour segment and its associated neighbourhood ϱ_s .

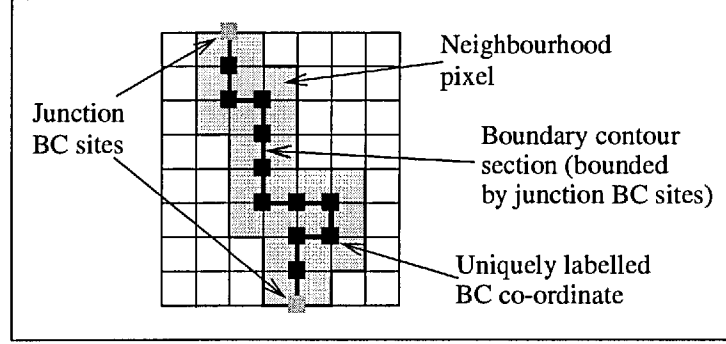


Figure 6.10: Example of a boundary contour section and its associated neighbourhood ϱ_s .

The following statistical values are computed within the neighbourhood ϱ_s :

$$\sigma_{\varrho_s}^2 = \frac{1}{N_{\varrho_s}} \sum_{x \in \varrho_s} (x - \mu_{\varrho_s})^2 \quad (6.4)$$

$$\mu_{diff} = \left| \mu_{R_{int}}(\varrho_s) - \mu_{R_{ext}}(\varrho_s) \right| \quad (6.5)$$

where:

$\sigma_{\varrho_s}^2$ is the grey scale variance of the pixels in the neighbourhood ϱ_s ;

N_{ϱ_s} is the number of pixels in ϱ_s ;

μ_{ϱ_s} is the grey scale mean of the pixels in the neighbourhood ϱ_s ;

μ_{diff} is the difference between the grey scale mean values of the internal (i.e. selected) region pixels, $\mu_{R_{int}}(\varrho_s)$, and the external region pixels, $\mu_{R_{ext}}(\varrho_s)$, which occupy the neighbourhood ϱ_s .

In addition to evaluating equations 6.4 and 6.5, minimum and maximum threshold values for $\sigma_{\varrho_s}^2$ and μ_{diff} are selected by the user.

The homogeneity test $P_s(\varrho_s)$ may now be defined as follows:

Definition 6.1 (Simple Homogeneity Test) The homogeneity test $P_s(\varrho_s)$ has the following three possible outcomes:

1. $P_s(\varrho_s) = \text{true} \Leftrightarrow \sigma_{\varrho_s}^2 \leq \sigma_{\min}^2 \text{ and } \mu_{diff} \leq \mu_{\min}$

2. $P_s(\varrho_s) = \text{false} \Leftrightarrow \sigma_{\varrho_s}^2 \geq \sigma_{\max}^2 \text{ or } \mu_{diff} \geq \mu_{\max}$
3. $P_s(\varrho_s) = \text{undefined} \Leftrightarrow \sigma_{\min}^2 < \sigma_{\varrho_s}^2 < \sigma_{\max}^2 \text{ and } \mu_{\min} < \mu_{diff} < \mu_{\max}$

If the maximum and minimum threshold values are set to be equal, i.e. $\sigma_{\min}^2 = \sigma_{\max}^2$ and $\mu_{\min} = \mu_{\max}$, the possibility of obtaining an **undefined** outcome from $P_s(\varrho_s)$ may be avoided. However, this requires the user to choose very finely tuned values for the threshold parameters. Furthermore, the optimal settings for threshold values may vary from one area of the image to another. We therefore choose very safe threshold values which give high confidence results for $P_s(\varrho_s) = \text{true}$ or **false**. In the event where $P_s(\varrho_s) = \text{undefined}$, we perform the more complex MRF based homogeneity test $P_{\text{MRF}}(\varrho_{\text{MRF}})$.

6.4.4 MRF Based Homogeneity Testing

The basic fundamentals of Markov random field theory are described in section 2.4.2. The overall image segmentation problem, in terms of the MRF model, is most popularly expressed by the MAP criterion

$$x^* = \arg \max_X \{P(Y = y | X)P(X)\} \quad (6.6)$$

where X is the high level process and Y is the low level process (as defined by the hierarchical image model - see section 2.4.2) and x^* is the optimal labelling for the segmented image. As mentioned in section 2.4.2, to find the optimal solution for the MAP criterion is computationally prohibitive. Various sub-optimal solutions have therefore been proposed [GG84, Bes86]. In appendix B we describe a numerical solution to this problem.

For the purpose of testing the homogeneity of the region in the vicinity of a contour section we propose an original method which considers the solution of equation 6.6 for the pixels in the neighbourhood ϱ_{MRF} . Depending on the final pixel labelling, $P_{\text{MRF}}(\varrho_{\text{MRF}})$ is set to either **true** or **false**. If all the IC and EC pixels in ϱ_{MRF} , or the majority of these pixels, have been assigned the same label, this

implies that the boundary is homogeneous, i.e. $P_{\text{MRF}}(\varrho_{\text{MRF}}) = \text{true}$. Otherwise $P_{\text{MRF}}(\varrho_{\text{MRF}}) = \text{false}$.

Before detailing how to relabel the pixels in the vicinity of a contour section using MRF based methods, it is necessary to define the neighbourhood ϱ_{MRF} . As mentioned above, we are only able to solve equation 6.6 by sub-optimal methods. It is therefore important to choose a first guess labelling which is fairly close to the final solution. Figure 6.11(a) shows an example of a BC section and its associated neighbourhood ϱ_s . Each pixel in ϱ_s is assigned a specific label according to the

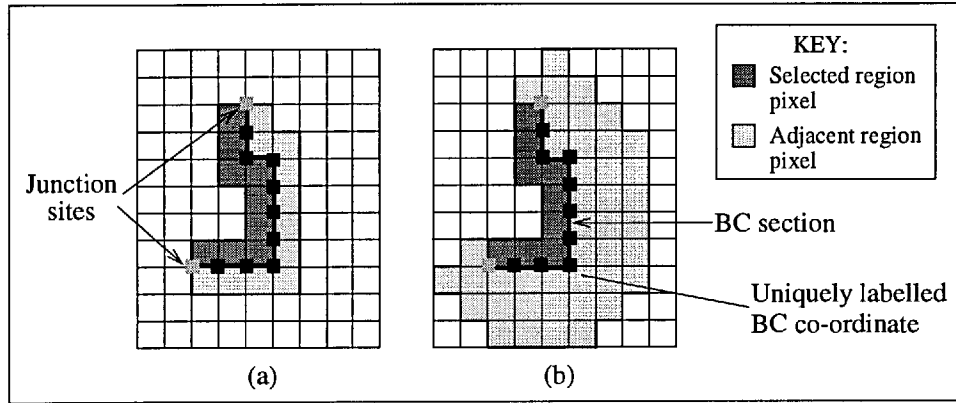


Figure 6.11: Examples of a BC section and its associated neighbourhoods (a) ϱ_s ; and (b) ϱ_{MRF} - dominant radius=3. The neighbourhoods comprise pixels from the selected region as well as from its adjacent regions.

4-connected region in which it belongs. It may happen that the optimal solution of the MAP criterion for the pixels in ϱ_s results in their being allocated the same label, i.e. ϱ_s is homogeneous and $P_{\text{MRF}}(\varrho_s) = \text{true}$. However, in such a case, the first guess MAP solution has approximately the same number of pixels labelled according to both regions. This is so far off from the optimal solution of equation 6.6 that a numerical sub-optimal solution will fail to approach a maximum. Instead it will get trapped in the initial guess labelling. This would imply the incorrect result $P_{\text{MRF}}(\varrho_s) = \text{false}$. It is therefore essential to choose a more appropriate BC section neighbourhood.

Figure 6.11(b) shows the neighbourhood ϱ_{MRF} for the same contour section

as shown in figure 6.11(a). As for ϱ_s , the four adjacent pixels of each uniquely labelled BC co-ordinate are isolated. Next, the *dominant* region, namely the larger of the two regions adjacent to the BC section, is determined (see figure 6.12 for an example in which the selected region is dominant). Additional pixels are then isolated from the side of the dominant region. The extent to which ϱ_{MRF} is grown from the dominant region is determined by a user defined *dominant radius*. This is achieved by an iterative filling process. Starting with the isolated neighbourhood ϱ_s , each not yet isolated 4-neighbour of every dominant pixel is filled. This process is repeated according to the size of the dominant radius less one. Thus, in the example of figure 6.11(b) where the dominant radius is set to 3, the iterative growing process is performed twice and the resulting neighbourhood ϱ_{MRF} is three pixels wide on the side of the dominant region.

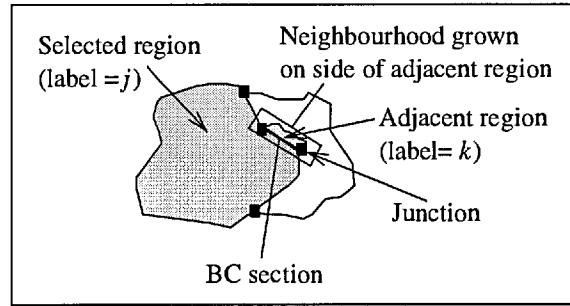


Figure 6.12: Growing the neighbourhood ϱ_{MRF} from region k , the non-dominant region.

Figure 6.12 illustrates why it is important to grow ϱ_{MRF} specifically from the side of the dominant region. In this example the region adjacent to the selected region is very thin. Thus, if ϱ_{MRF} were grown from the side of the adjacent region, the initial labelling of the pixels in ϱ_{MRF} would never favour an ultimate relabelling of all the pixels to a single region. However, by growing ϱ_{MRF} from the side of the dominant region, this problem is avoided.

Once the neighbourhood ϱ_{MRF} has been isolated, it is possible to initiate the conditions which enable MRF based labelling to proceed within its confines. Figure 6.13 shows an example of the neighbourhood ϱ_{MRF} of a BC section whose dominant

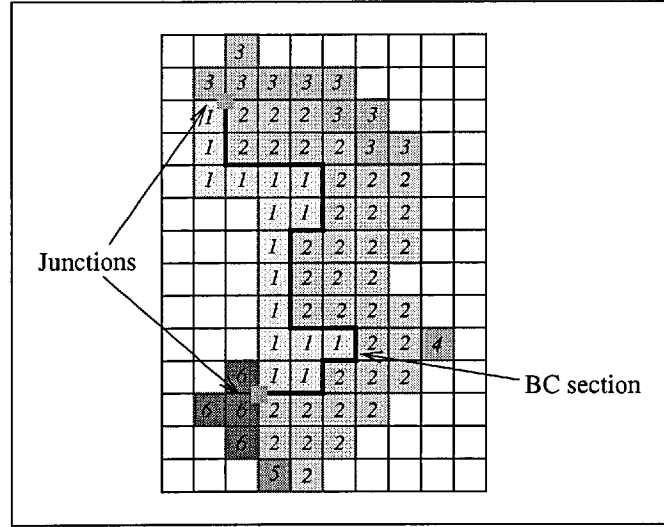


Figure 6.13: An example of a BC section and its associated neighbourhood ϱ_{MRF} whose dominant radius=3. The pixels inside ϱ_{MRF} are highlighted and numbered according to the different regions in which they belong.

radius is set to 3. Each pixel inside ϱ_{MRF} is labelled according to the 4-connected region in which it belongs rather than according to the tissue class to which it is allocated as a result of the defuzzified FFCM classification. In this example six different regions occupy ϱ_{MRF} . We consider the neighbourhood ϱ_{MRF} as an entity of its own in isolation from the rest of the image. Thus, we ignore all the regional statistics of the region image as a whole and consider the characteristics of regions within ϱ_{MRF} alone. The random fields, X and Y , are hence defined over the lattice $S = \varrho_{\text{MRF}}$. MRF labelling may take place within S by solving the MAP criterion of equation 6.6 using the numerical solution described in appendix B.

After the pixels in ϱ_{MRF} have been relabelled, the labels of the IC and EC pixels which occupy ϱ_{MRF} are examined as follows: Let N_c be the total number of IC and EC pixels present in ϱ_{MRF} . Let n_{max} be the largest number of uniquely labelled IC and EC pixels present in ϱ_{MRF} . The region homogeneity ratio, χ , is then computed:

$$\chi = \frac{n_{\text{max}}}{N_c} \quad (6.7)$$

Finally, a user defined threshold for χ_{min} is set.

The MRF based homogeneity test may now be instigated as follows:

Definition 6.2 (MRF based homogeneity test) The test $P_{\text{MRF}}(\varrho_{\text{MRF}})$ has the following two possible outcomes:

1. $P_{\text{MRF}}(\varrho_{\text{MRF}}) = \text{true} \Leftrightarrow \chi \geq \chi_{\min}$
2. $P_{\text{MRF}}(\varrho_{\text{MRF}}) = \text{false} \Leftrightarrow \chi < \chi_{\min}$

Thus the MRF based homogeneity test is able to definitely resolve the **undefined** situations which result from simple homogeneity testing.

6.5 3D Implementation of the AR process

In the previous sections, the AR process was detailed for the extraction of anatomical structures from 2D images. In this section, we describe how to extend this approach so that three dimensional objects may be isolated automatically from 3D image data.

6.5.1 The Problems of Direct 3D Data Processing

The most obvious way to extend the AR process to work for 3D data is to perform all the processing tasks, previously described in two dimensions, directly in 3D. This requires algorithms 6.1 to 6.5 to be adapted so that they consider 6-connected 3D regions instead of 4-connected 2D regions. However, in the majority of cases, this is not feasible due to either poor resolution of the image data in the slice plane and/or the propagation of misclassification errors resulting from the FFCM algorithm. Figure 6.14(a) illustrates how attempting to fill a specific 6-connected region within classified 3D image data, of low inter-slice resolution, does not always produce the desired result. In the example shown, all the pixels inside the highlighted regions have been classified according to the same tissue type. However, due to the

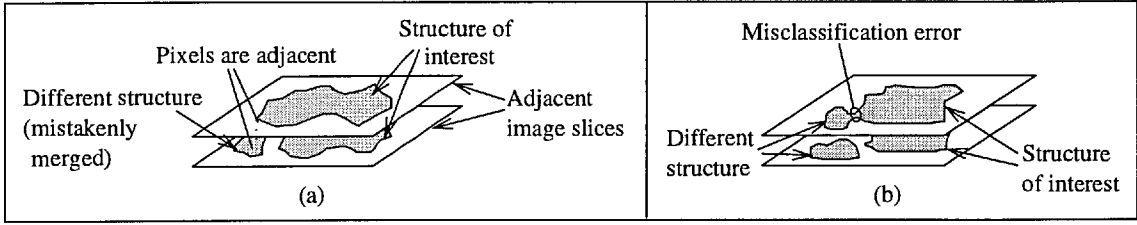


Figure 6.14: Examples of problematic cases when attempting to fill a 3D region: (a) Poor inter-slice resolution; (b) Misclassification error.

wide gap between the two image slices, there are pixels from the region in the top slice which are adjacent to those from a region belonging to a different structure in the lower slice. Hence, the 3D filling process will over-spill into several structures. Figure 6.14(b) highlights a different scenario in which 6-connected region filling goes wrong. As for the previous example, all the pixels within the highlighted regions have been classified according to the same tissue type. However, due to misclassification errors resulting from the FFCM algorithm, two regions belonging to different structures have become connected to form a single region in the top slice. As a result, over-filling occurs. Furthermore, each misclassification error propagates and spreads throughout the entire 3D volume.

The problems which result from poor inter-slice resolution may be circumvented by reconstructing the data in the inter-slice plane, using appropriate interpolation techniques, so that the inter-slice slice resolution approaches the intra-slice resolution. However, there are two limiting conditions on the mode of acquisition of the original data if subsequent reconstruction is to succeed:

1. The data must be *band-limited* in the inter-slice plane.
2. The *Nyquist criterion* must be satisfied. This requires the inter-slice sampling frequency, i.e. the Nyquist rate, to be greater than twice the the maximum frequency present in the inter-slice plane.

If these two conditions cannot be met, the reconstructed data will suffer from aliasing

artefacts and will not be correctly restored. A good theoretical discussion on this phenomenon may be found in [Wol94].

Appendix A describes how MR image data may be acquired as either a stack of individual 2D image slices or directly as a 3D volume. With multi-slice acquisition, the resulting 3D data is not band-limited in the inter-slice plane thus precluding the possibility of accurate data reconstruction. Although directly acquired 3D data is band-limited in the inter-slice plane, the inter-slice resolution must be fairly good if the Nyquist criterion is to be entirely satisfied. Thus, data reconstruction is rarely a viable option. Furthermore, the amount of data that would be generated if reconstruction could take place would add significantly to the computational burden of subsequent processing. In addition, the problems resulting from misclassification errors still remain even if the data has been perfectly reconstructed.

We therefore propose a 2D slice based approach for the implementation of the AR process in three dimensions. The following sections will describe how this achieved.

6.5.2 3D Region Extraction

In order to implement the AR process in three dimensions, it is first necessary to isolate a 3D region of interest from defuzzified FFCM classified 3D data. This is achieved by considering the 3D region of interest as a stack of 2D regions occupying a 3D data volume which is made up of N two dimensional image slices. Each 2D region is isolated individually to eventually form the entire 3D region. We propose the following algorithm for the automatic extraction of such a region:

Algorithm 6.6 (3D Region Extraction) The following steps are performed:

1. Fix the minimum adjacency threshold, α_{min} , such that $\alpha_{min} \in [0, 1]$. This is used later, in steps 7 and 8, to determine whether or not to extract a region.

2. Interactively select a 3D region of interest by choosing a single 3D point, $x(k)$, within the region, e.g. by using a mouse (see figure 6.15). Note: k indexes the inter-slice plane axis, or slice number, where $1 \leq k \leq N$.
3. Set $m = n = 1$.
4. Fill the 2D region, R_s , which contains the selected point $x(k)$ (see figure 6.15), using the algorithm described in appendix C, and compute its size A_s simultaneously. For each newly filled pixel, $x(k)$, in R_s :

If $k > 1$, check the pixel $x(k - 1)$ which is immediately above $x(k)$ {

If $x(k - 1)$ has been classified according to the same class as $x(k)$ and it has not yet been visited:

(a) Fill the region in which $x(k - 1)$ belongs and label it as R_m .

Whilst filling R_m , compute δ_{sm}^- , i.e. the number of pixels in R_m which are adjacent (i.e. 6-connected) to R_s .

(b) Set $m = m + 1$.

}

If $k < N$, check the pixel $x(k + 1)$ which is immediately below $x(k)$ {

If $x(k + 1)$ has been classified according to the same class as $x(k)$ and it has not yet been visited:

(a) Fill the region in which $x(k + 1)$ belongs and label it as R_n .

Whilst filling R_n , compute δ_{sn}^+ , i.e. the number of pixels in R_n which are adjacent to R_s .

(b) Set $n = n + 1$.

}

5. Compute adjacency ratios for slices $k - 1$ and $k + 1$ as follows:

$$\alpha_m^- = A_s / \delta_{sm}^-; \quad \alpha_n^+ = A_s / \delta_{sn}^+ \quad (6.8)$$

where:

α_m^- and α_n^+ are the adjacency ratios for regions R_m and $R_n \forall n, m$ in slices $k-1$ and $k+1$ respectively;

A_s is the area of region R_s ;

δ_{sm}^- and δ_{sn}^+ are as defined in step 4.

6. Determine the indices i and j of the regions with the maximum adjacency ratio in slices $k-1$ and $k+1$ respectively:

$$i = \arg \max_m (\alpha_m^-); \quad j = \arg \max_n (\alpha_n^+) \quad (6.9)$$

If $\alpha_i^- > 1$ then set $\alpha_i^- = 1/\alpha_i^-$.

If $\alpha_j^+ > 1$ then set $\alpha_j^+ = 1/\alpha_j^+$.

7. **If $\alpha_i^- > \alpha_{min}$**

Extract region R_i (in slice $k-1$) to form part of selected 3D region.

Otherwise

Do not extract any regions from slice $k-1$.

8. **If $\alpha_j^- > \alpha_{min}$**

Extract region R_j (in slice $k+1$) to form part of selected 3D region.

Otherwise

Do not extract any regions from slice $k+1$.

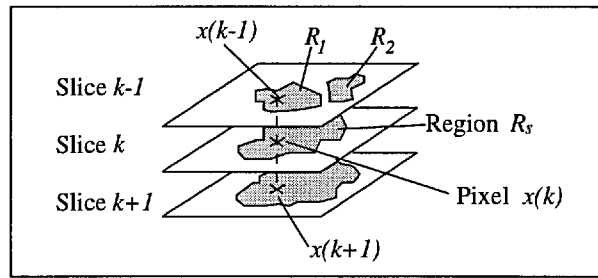


Figure 6.15: The 3D region extraction process. All the shaded regions have been classified according to the same tissue type. Regions R_1 and R_2 in slice $k-1$ are both adjacent to region R_s in slice k .

Figure 6.15 depicts an example where there are two adjacent regions in slice $k-1$. In this particular case, $\alpha_1^- > \alpha_2^-$ and $\alpha_1^- > \alpha_{min}$. Hence region R_1 is extracted as part of the entire 3D region.

9. If a region has been extracted from either slice $k - 1$ or slice $k + 1$, we consider this region as R_s and repeat the above described process, starting from step 3. However, we consider only the slice either above or below the region in question depending on which has not yet been examined. This process is repeated iteratively until the entire 3D region has been extracted.

6.5.3 Adaptive Refinement of Extracted 3D Region Data

After the 3D region of interest has been extracted, it is necessary to refine its boundaries so that a region results which corresponds accurately to the desired anatomical structure. This is achieved by simply processing each image slice of the 3D data array individually. Each slice, which contains a 2D region that has been extracted as part of the 3D region of interest, is considered on its own and the AR process is applied to it exactly as described in sections 6.2, 6.3 and 6.4. The entire procedure for the 3D implementation of the AR process is summarised in figure 6.16.

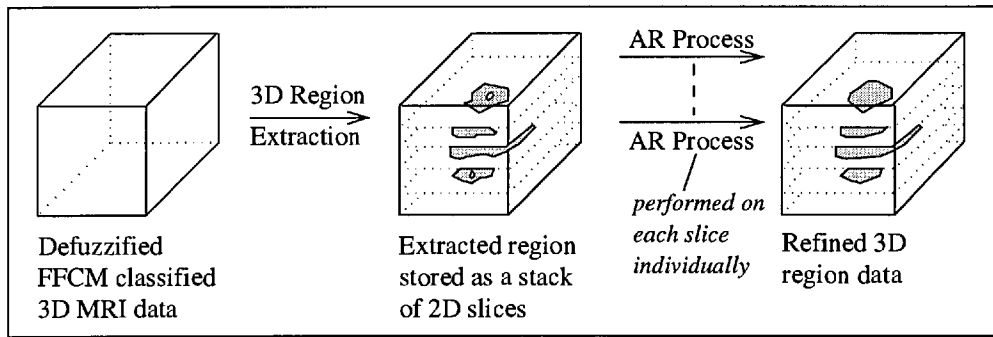


Figure 6.16: 3D implementation of the AR process: A pictorial summary.

6.6 Results and Discussion

In this section, results from applying the AR process to MRI data are presented and discussed. In sections 6.6.1, 6.6.2 and 6.6.3 the efficacy of the AR process, when applied to 2D image data, is demonstrated by examining its performance on the

knee data shown in figure 4.1(b). The behaviour of all the other data presented in chapter 4 is also discussed. Following this analysis on 2D image data, results from applying the AR process to 3D data volumes are presented in section 6.6.4. Finally, in section 6.6.5, the computational efficiency of the AR process is examined for both 2D and 3D image data.

6.6.1 Preliminary Processing Tasks

The results from FFCM classification of MRI data have been discussed in detail in chapter 5. Figure 6.17(a) shows the defuzzified output of the FFCM classified knee data of figure 5.5. It may be observed that although the inherent fuzziness

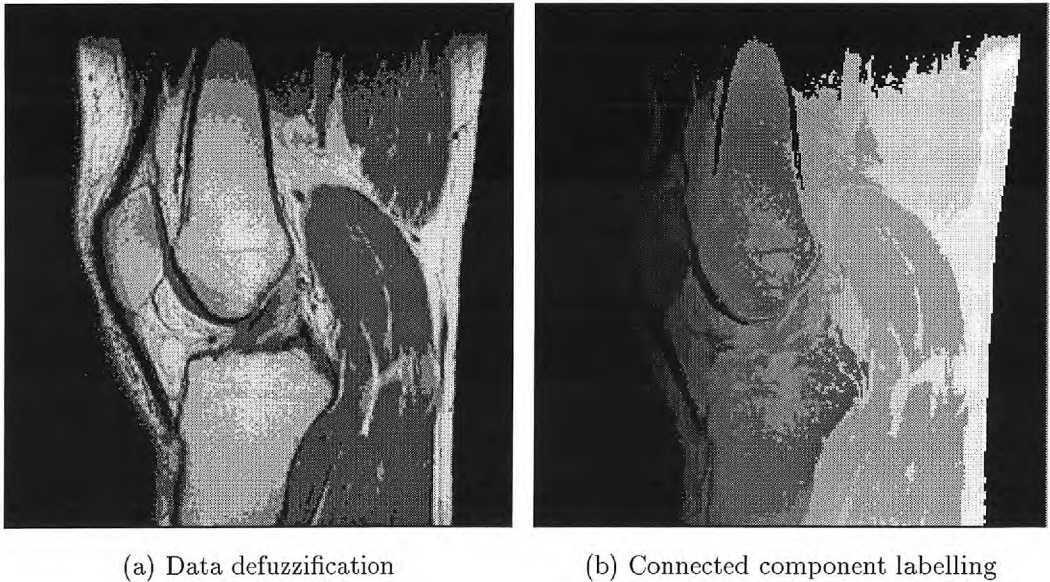


Figure 6.17: Preliminary processing tasks of the AR process applied to FFCM classified knee data (as shown in figure 4.1(b)).

present in the original data is discarded during the defuzzification process, the overall classification still corresponds closely to the actual anatomy. Note how the two main bony structures, i.e the femur (top) and the tibia (bottom), are composed of more than one tissue type. Furthermore, the classified data is rather “grainy”, i.e. there

are many holes present.

Figure 6.17(b) shows the connected component labelled output of the defuzzified data of figure 6.17(a). Each uniquely labelled region is represented by a different grey scale. Although only five tissue types exist in the original data, almost two thousand 4-connected regions result from the connected component labelling. We specifically label 4-connected regions to avoid unwanted connections between different regions which might occur when considering 8-connected regions.

6.6.2 Region Selection and Preparation

From this stage onwards in the AR procedure, the processing operations are confined to the vicinity of an interactively selected region in the image.

Hole Removal

Figure 6.18(a) shows the result of seed filling a selected region, highlighted in red, inside the tibial bone. There are several small holes present in the selected region which evidently do not represent structures in their own right. Figure 6.18(b) shows the selected region after it has been “cleaned” up by applying the hole removal algorithm (see algorithm 6.2). We have set the maximum hole size and maximum grey scale difference as $H_{max} = 4$ and $G_{max} = 20$ respectively. If H_{max} is set too high, holes, which may link the selected region to inappropriate neighbouring structures, are joined to the selected region. G_{max} must also not be given too high a value. This avoids holes which belong to other structures from being merged with the region of interest.

Boundary Tracking and Labelling

After the region has been selected and the undesirable holes within have been removed, the boundaries which define the region are tracked and labelled. A prerequisite

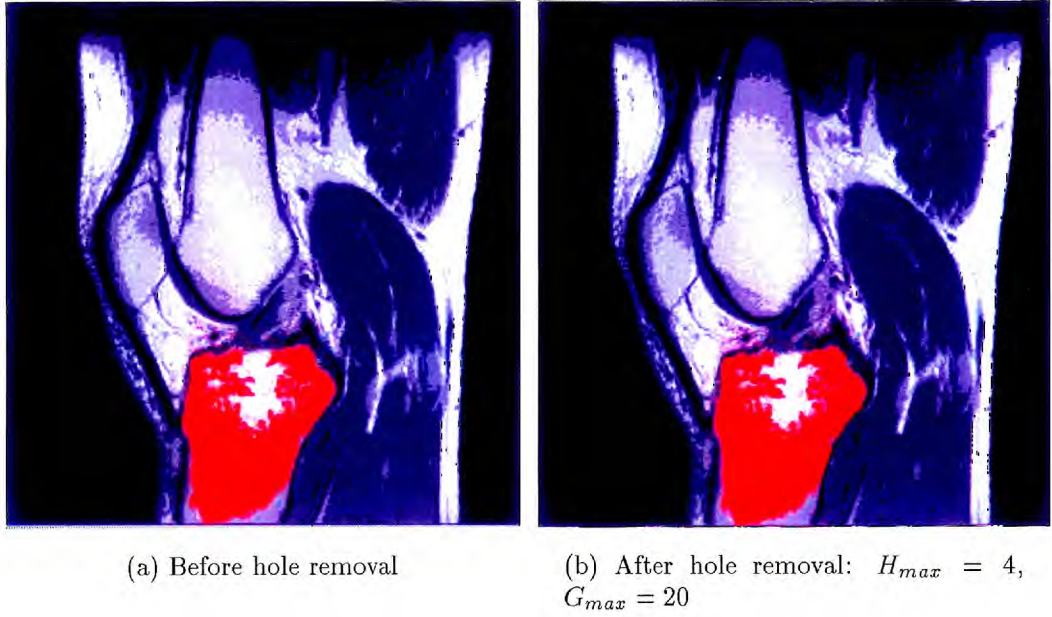


Figure 6.18: Region selection: (a) Before hole removal; (b) After hole removal.

to this process is to detect the regional boundaries. Figures 6.19(a) and (b) show magnified views of the internal and external contours as well as of the boundary contours of the selected region. These are defined on the image lattice and boundary lattices respectively as explained in section 6.3.2.

For the selected region shown in figure 6.18(b), 10 contours define the region's boundaries. These are adjacent to a total of 46 different regions. This contrasts significantly with the originally selected region before hole removal, as illustrated in figure 6.18(a). The latter is defined by 81 boundary contours with 145 adjacent regions! We may therefore observe that by performing the simple hole removal algorithm at an early stage of the AR process, the total amount of subsequent processing required is substantially reduced.

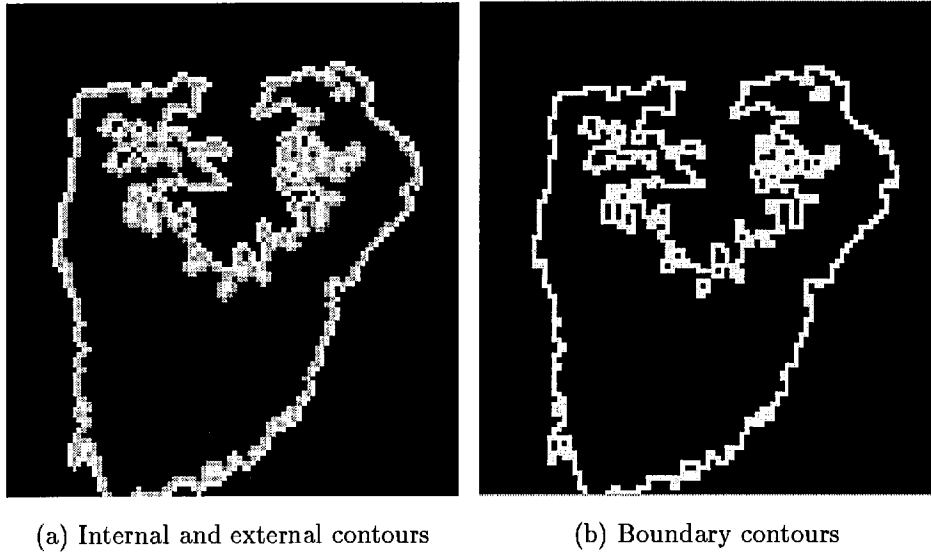


Figure 6.19: Magnified views of the selected region's: (a) Internal and external contours. The IC pixels are depicted by a darker grey scale intensity than their EC pixel counterparts; (b) Boundary contours.

6.6.3 Adaptive Region Refinement

In this section, results from adaptively refining the selected region of the knee data, which was analysed in section 6.6.2, are presented. As discussed in section 6.4.1, it is sometimes necessary to perform splitting operations on the selected region. However, since we have not yet developed a formal method for this, we present only results from algorithm 6.5 for hierarchical region merging. Ideas for future work on region splitting are described in chapter 8.

Simple Homogeneity Testing

In order to assess the performance of the simple homogeneity test $P_s(\varrho_s)$, as formalised by definition 6.1, we have set the values $\sigma_{min}^2 = \sigma_{max}^2$ and $\mu_{min} = \mu_{max}$ so that a definitive result of either **true** or **false** is obtained. We have ignored the effects of $\mu_{min} = \mu_{max}$ by assigning them a very high value. This is because we have found that computing the grey scale mean difference between two regions in the

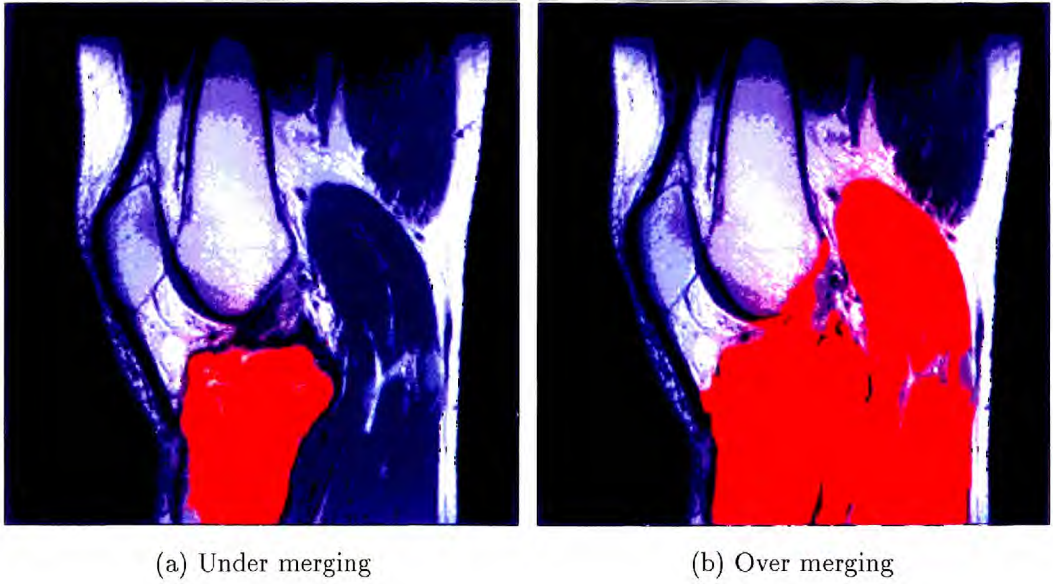


Figure 6.20: Results from simple homogeneity testing: (a) Under merging; (b) Over merging.

neighbourhood ϱ_s is a less effective measure of homogeneity than the overall grey scale variance of the pixels in the neighbourhood. This is a result of performing the simple homogeneity test on a wide range of different images.

Figure 6.20(a) shows the segmentation which results from setting $\sigma_{min}^2 = \sigma_{max}^2 = 5.0$. It may be seen from this result that insufficient merging has taken place. Figure 6.20(b) shows how setting $\sigma_{min}^2 = \sigma_{max}^2 = 25.0$ causes the opposite problem of over merging. In fact, definition 6.1 only yields accurate results if the values of $\sigma_{min}^2 = \sigma_{max}^2$ are fixed within the interval $[12, 18]$. Furthermore, this is only the case for the specific region of interest shown in figure 6.18(b). A different region within the same image may require completely different parameter settings to give a good result. For example, when extracting the region corresponding to the same tibial structure on slice 8 of the same 3D data set, good results are only obtained when setting $\sigma_{min}^2 = \sigma_{max}^2$ within $[18, 24]$.

It is therefore clear that although it is possible to obtain good results from $P_s(\varrho_s)$, user intervention is required to select optimal parameter values in order to

assure consistently accurate results.

MRF Based Homogeneity Testing

Before discussing the performance of the MRF based homogeneity test, $P_{\text{MRF}}(\varrho_{\text{MRF}})$, as formalised by definition 6.2, it is important to examine the behaviour of MRF based classification when applied globally to MRI data.

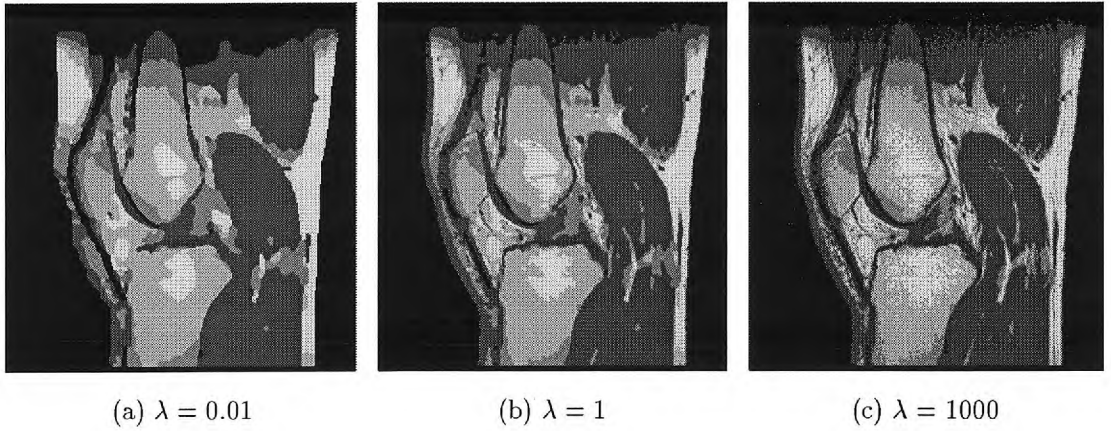


Figure 6.21: Global MRF based classification of knee data (as shown in figure 4.1(b)).

Figure 6.21 shows results from MRF based classification of the knee data of figure 4.1(b). The MAP criterion defined by equation 2.28 has been evaluated using the numerical solution described in appendix B. When obtaining these results, the parameter values for β_k , which define the clique potential in equation B.3, are confined¹ to the real interval $[-5, 5]$. The value for γ_{max} (see algorithm B.2, step 3) is set to 0.05. The ICM and the Metropolis algorithms (algorithms B.2 and B.3) both perform ten iterations during each round of the ASA, and the ASA (algorithm B.1) is stopped after three iterations. Finally, the initial labelling $x^{(0)}$ chosen in

¹These values are restricted in order to avoid a degenerative solution. If β_k were allowed to assume very large positive or negative values, the corresponding image classification would be forced to an extreme realisation, e.g. a single region or a striped or chess board realisation.

step 1 of the ASA is obtained by FFCM classification whilst setting the number of classes $c = 5$.

As described in section B.2.2, the parameter λ is introduced into equation B.16 so that the user may control the performance of the ICM algorithm by adjusting the influence of spatial interactions versus that of the global properties of the image data on the final classification. Figures 6.21(a)-(c) show the resulting classifications when λ is assigned a range of values from very low to very high.

When $\lambda = 1$, the influence of the spatial interactions within the image is equal to that of its global grey scale distribution. We may observe from figure 6.21(b), that in this instance, the resulting labelling is less grainy than the first guess solution, $x^{(0)}$, i.e. the original defuzzied FFCM classification. This is due to the influence of the MRF model used for the high level process X . However, in some ways, the quality of the segmentation is degraded due to the introduction of misclassification errors. This is particularly noticable in the region corresponding to the patellar tendon. The cause of these mistakes may be understood in light of the result shown in figure 6.21(c). When $\lambda = 1000$, the classification is influenced almost entirely by the low level field Y . This assumes that the global distribution of the MR image data is Gaussian. Thus, the resulting classification is equivalent to the result which would be obtained from solving the ML estimates for the Gaussian distribution. Therefore, as expected, figure 6.21(c) closely resembles the colour blended output of the EM algorithm shown in figure 5.9(b). This is less accurate than the output of the FFCM because the distribution of MRI grey scale data is not Gaussian, as proven in chapter 5. We may therefore understand why the influence of low level process Y may actually degrade the quality of the segmented output.

If no a priori knowledge of the data distribution is assumed, and only the high level process is considered, the final result is even worse than when choosing the imperfect Gaussian model to fit the low level process. This is illustrated by the result shown in figure 6.21(a) where $\lambda = 0.01$, i.e. the high level process, X ,

dominates. It is evident that far too much merging takes place and that a significant reduction in quality of segmentation results. If it were possible to find a generic model which closely fits the grey scale distribution of MRI data, then by choosing it for the low level process Y and setting $\lambda = 1$, an optimal MRF based labelling would be obtained.

With the results from the global implementation of MRF based labelling on MRI data in mind, it is possible to analyse the results from MRF based homogeneity testing intelligently. Figure 6.22(a) shows a magnified view of a selected region inside

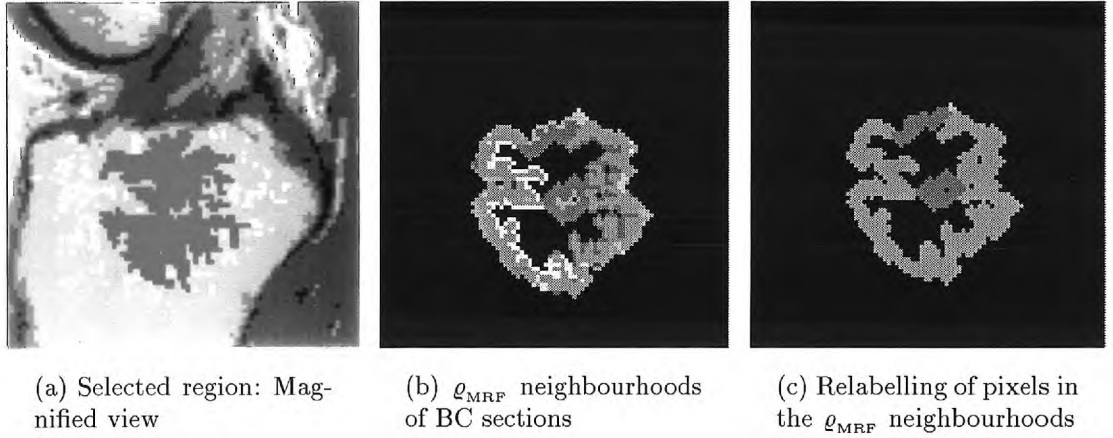


Figure 6.22: (a) Selected region inside the tibial bone of the classified knee data shown in figure 6.17(a); (b) The ϱ_{MRF} neighbourhoods of the BC sections which make up the boundary contours of the selected region; (c) The relabelled pixels in the ϱ_{MRF} neighbourhoods.

the tibial bone of the classified knee data shown in figure 6.17(a). Figure 6.22(b) shows the ϱ_{MRF} neighbourhoods of the BC sections which make up the region's two boundary contours. The dominant radii of the ϱ_{MRF} neighbourhoods have been set to 4. Within each ϱ_{MRF} , every pixel is assigned a specific label, which is depicted by a unique grey scale, according to the particular region in which it belongs. Note how the dominant region is sometimes the selected region and sometimes its adjacent neighbour, depending on their relative sizes. Figure 6.22(c) shows the final labelling of the neighbourhood pixels after MRF based labelling has been performed. In this

example, for each neighbourhood, all the IC and EC pixels have been relabelled according to a single region only. This indicates that the region within the immediate vicinity of each BC section is homogeneous. Thus, the homogeneity ratio, χ , as defined by equation 6.7, is equal to 1.0 for all four neighbourhoods. All four corresponding adjacent regions are therefore merged with the selected region.

It should be noted that although the Gaussian model chosen for Y does not match the MRI grey scale data distribution closely during global MRF based labelling, it does not introduce misclassification errors when relabelling the pixels within a BC section neighbourhood. This is because within the small confines of a ϱ_{MRF} neighbourhood, the global properties of the data become less prominent. In fact, the pixel data within ϱ_{MRF} is distributed in a fairly random nature. Thus, a Gaussian model is more likely to be appropriate for the low level process Y in a ϱ_{MRF} neighbourhood than when used to model the entire image. We have found that by slightly reducing the influence of Y by setting $\lambda = 0.5$, the consistency of accurate MRF based labelling within the ϱ_{MRF} neighbourhood is improved.

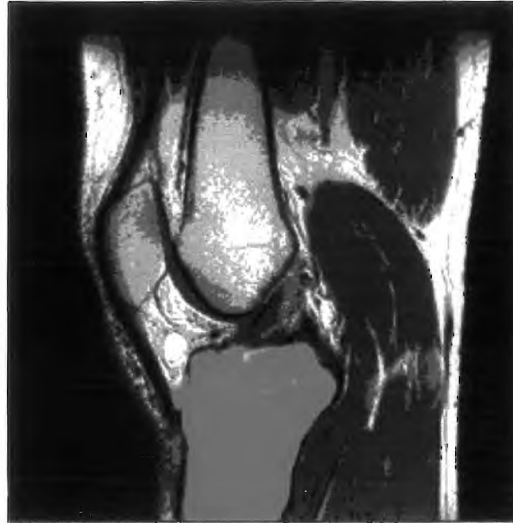


Figure 6.23: Final result from the AR process after selecting the region illustrated in figure 6.18(a).

Figure 6.23 shows the adaptively refined region which results from initially selecting the region depicted in figure 6.18(a). The simple homogeneity test parameter

values have been set as follows: $\sigma_{min}^2 = 5.0$ and $\sigma_{max}^2 = 25.0$. Thus, the regions adjacent to any BC sections, whose neighbourhood ρ_s grey scale variances are less than 5.0, are merged with the selected region. Those with grey scale variances of greater than 25.0 are definitely not merged. Any BC sections whose ρ_s grey scale variance values lie within the interval $[5, 25]$ are examined by the MRF based homogeneity test. We have set the dominant radius to 3 and have performed five iterations of the ICM and MA algorithms during each round of the ASA and 6 iterations of the ASA. The value of γ_{max} has been set to 0.1 and $\chi_{min} = 0.9$. We have found that this combination of parameter values yields accurate results in a wide range of situations. Suggestions for further improvements to the hierarchical merging process are presented in chapter 8.

The resulting region highlighted in figure 6.23 corresponds very closely to the tibial bone in the original image. It may be observed that a small region belonging to the fat pads (these rest on the top left corner of the tibia) is mistakenly connected to the segmented region. This error is not caused by any misclassifications resulting from the MRF based homogeneity test. Rather, it is due to an error resulting from the original FFCM output. This mistake can be rectified by splitting the region which was mistakenly connected using methods such as those suggested in chapter 8. If adaptive region refinement were to be performed iteratively, as portrayed by figure 6.1, such errors could spread and generate gross inaccuracies in the final segmented region. We have therefore selected the largest region within the structure of interest and performed adaptive refinement only once with no subsequent iterations. This avoids any propagation of errors and, since the selected region is adjacent to almost all the other regions of the structure of interest, a fairly accurate segmentation results. Clearly, the AR process will produce optimal results if accurate region splitting can be employed and adaptive region refinement is performed iteratively.

The AR process has been tested on all the images displayed in figures 4.1 and 4.2. The same parameter values used to obtain the result shown in figure 6.23

were adopted throughout. It was found that accurate segmentations were obtained consistently. Under merging almost never occurred, and in nearly all the cases in which over merging took place, FFCM misclassifications were the cause. These could have been avoided if region splitting was employed. It should be noted, that the original FFCM classification of the IMR knee data is not particularly good as discussed in chapter 5. Nevertheless, the AR process performed well wherever the initial FFCM result was oversegmented.

6.6.4 3D Implementation of the AR Process

The first and most crucial stage of applying the AR process to 3D data is that of 3D region extraction. Figures 6.24(a)-(c) show selected slices from the extracted 3D region which results by interactively choosing the region of figure 6.18(b) and applying algorithm 6.6. The adjacency threshold α_{min} (defined in algorithm 6.6, step 1) was set to 0.3. In the example shown, all the correct 2D regions were isolated to form the complete 3D region. This has been shown to work consistently with most structures. However, the value of α_{min} must sometimes be adjusted to avoid either extracting too few or too many 2D regions at either extreme of the 3D region. The main limitation of algorithm 6.6 becomes apparent when attempting to extract the 2D regions from a complex 3D structure which branches so that it occupies more than one region in a single slice. Ideas for extending the algorithm so that it can cope with such situations are described in chapter 8. Nevertheless, most structures, particularly within knee data volumes, do not branch. Hence, algorithm 6.6 performs well in the majority of situations.

Figures 6.24(d)-(e) show the adaptively refined outputs of the regions displayed in figures 6.24(a)-(c). These have been obtained by using the same parameter values as those used for producing the result displayed in figure 6.23 which are listed in the previous section. The importance of MRF based homogeneity testing becomes most apparent when implementing the AR process in 3D. If only simple homogeneity

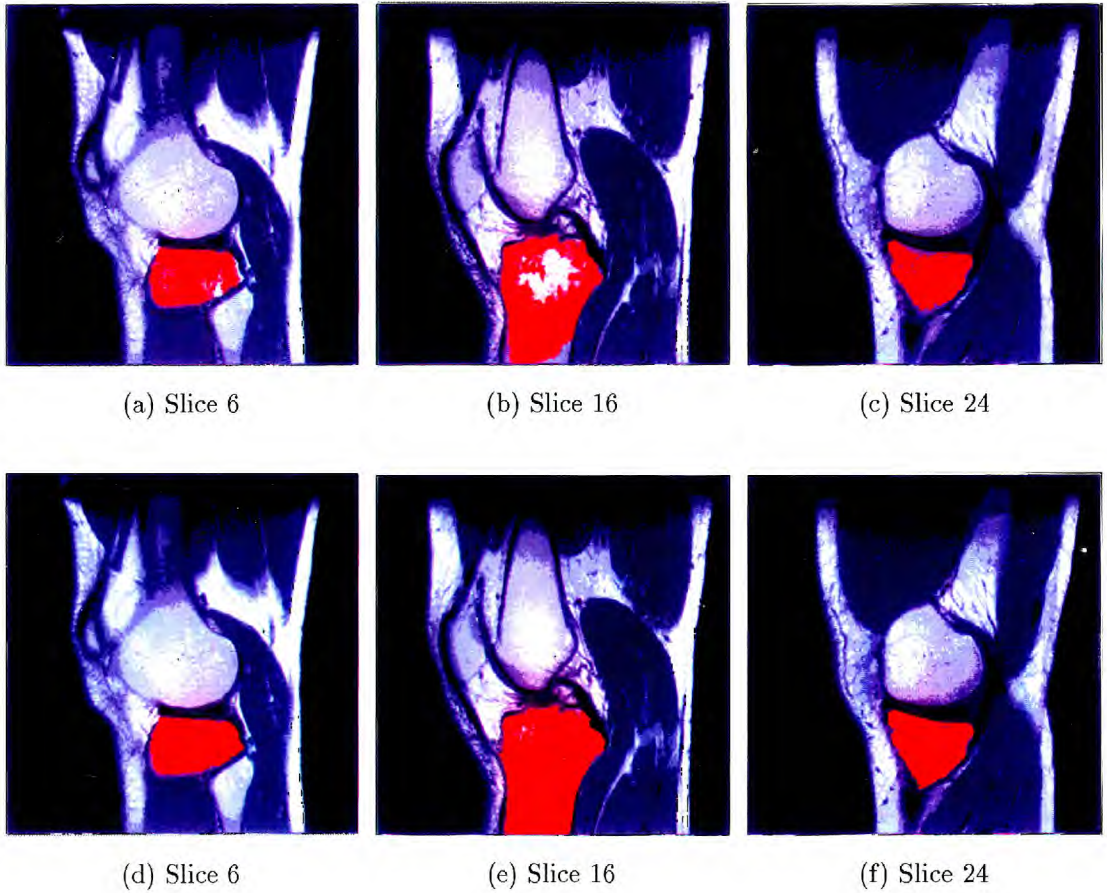


Figure 6.24: (a)-(c) Extracted 2D regions. (d)-(e) Adaptively refined 2D regions.

testing is used, it is never possible to refine a 3D region accurately with a single set of parameter values. This is because the optimal threshold values vary from slice to slice. By adopting the hierarchical framework defined by algorithm 6.5, an entire 3D region may be extracted from the image accurately whilst using a single set of parameter values. This enables the process of 3D segmentation to be fully automated.

The entire AR segmented 3D region of the tibial bone, a selection of whose slices were shown in figures 6.24(d)-(e), may be visualised most easily by generating a 3D surface or volume rendition. Figure 6.25 depicts the surface rendered output of this structure together with that of the femur, also extracted by the AR process.

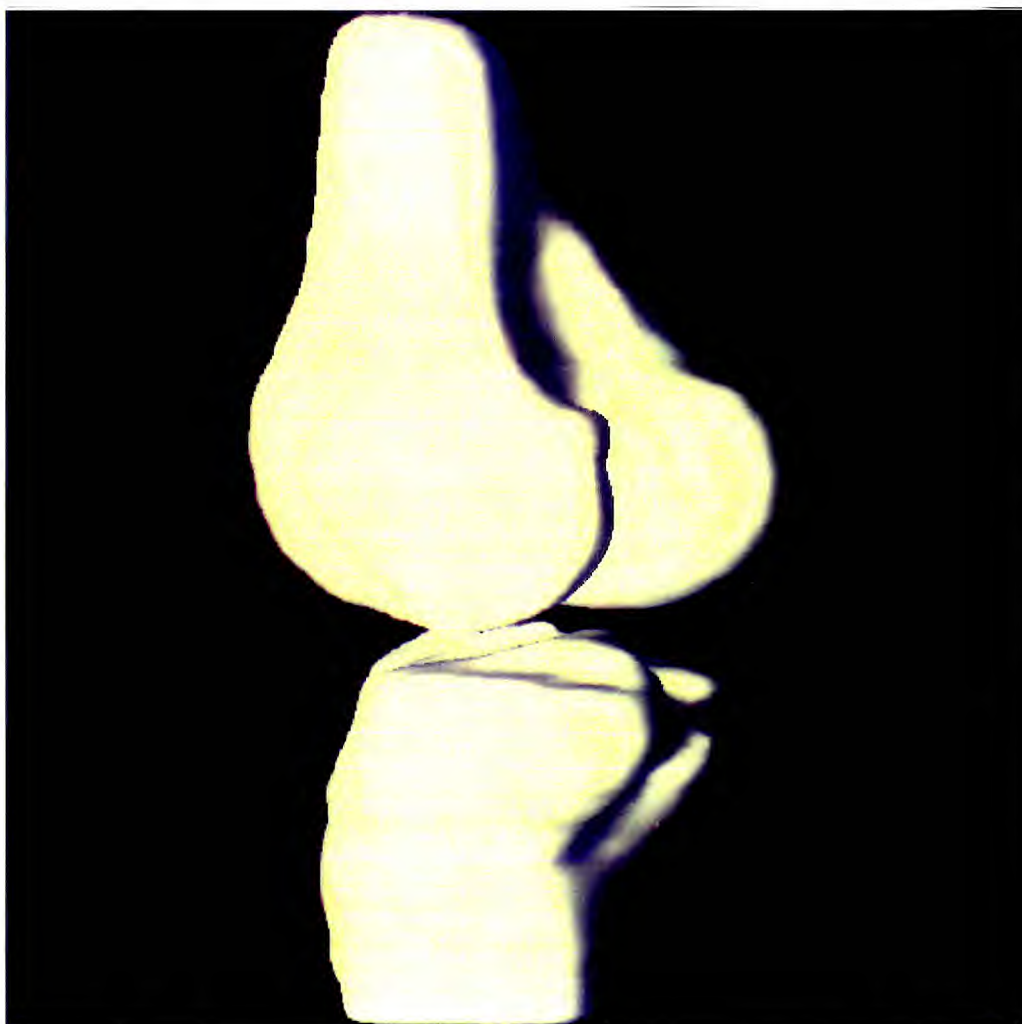


Figure 6.25: Surface rendition of segmented femoral and tibial bone data.

This has been achieved by first smoothing the ^{segmented} data, by convolving it with a $(4 \times 4 \times 4)$ mean averaging filter, and surface rendering the resultant data using the method described in section 7.1. In the next chapter, the performance of different rendering methods applied to segmented MRI data will be examined and discussed.

6.6.5 Computational Efficiency of the AR Process

One of the main strengths of the AR process is its computational efficiency. Due to its adaptive nature, its speed of performance varies from one structure to another. In order to give an indication of the general performance of the AR process, we will

first examine the processing times recorded for the 2D example of figure 6.23 on a DEC Alpha AXP/200 workstation.

When only simple homogeneity tests were considered, the final refined region was generated almost instantaneously. The implementation of the full hierarchical approach, including MRF based homogeneity testing, took just under a second. When MRF labelling was applied globally to an entire 2D image, the final classification was obtained after two minutes. Thus, by adopting an adaptive hierarchical framework for processing, the conventional global processing times have been reduced by a factor of more than one hundred. Furthermore, as demonstrated in section 6.6.3, the quality of MRF based segmentation is significantly improved when implemented within an adaptive framework.

When implementing the AR process on the entire 3D tibial bone structure, the region extraction process is almost instantaneous. The subsequent adaptive refinement stage, which processes each 2D region sequentially, takes approximately 20 seconds. This would be reduced if the AR process were implemented in parallel. Alternatively, a faster processor could speed up its performance substantially.

6.7 Conclusions

The theory for the adaptive refinement of FFCM classified MRI data has been presented and encouraging results have been obtained. The method has been demonstrated on the femoral and tibial bone structures which it extracted successfully from 3D MRI data. Furthermore, the process is automatic and extremely fast. With a dedicated processor, near real-time performance of the AR segmentation scheme can be achieved. Further work still remains to be done on developing region splitting methodologies and extending the adaptability of the 3D extraction algorithm for which some ideas are suggested in chapter 8.

In the following chapter, applications of the AR process, when used to extract

3D structures from MRI data, will be demonstrated for the purpose of subsequent 3D rendering.

Chapter 7

Fast 3D Visualisation Techniques: Performance and Applications

This chapter investigates the efficacy and practicality of the techniques presented in chapters 4, 5 and 6, when implemented as part of the complete 3D visualisation process. The performance of different 3D rendering schemes, applied to data which has been acquired and processed using these methods, is examined and discussed. Clinical applications for fast and automatic 3D visualisation methods are then explored, with a particular emphasis on 3D image guided surgery.

7.1 Evaluation of 3D Rendering Methods

This section examines the behaviour of different rendering techniques when applied to classified and/or segmented MRI data obtained using the methods detailed in the previous three chapters. A selection of various rendering schemes for the purpose of this study is first detailed. Results are then presented and discussed.

7.1.1 Methods

Surface Rendering

Several different methods for extracting a surface model from segmented volume data for subsequent rendering are described in section 3.1. We have chosen to use the marching cubes algorithm [LC87] since it outputs a standard triangular tiled geometric model which may be subsequently rendered using standard computer graphics software. We have used the Advanced Visualisation System (AVS) software package for this purpose. Although other approaches exist for surface extraction, the end results of all these techniques are virtually indistinguishable since they all attempt to model the same surface. We may therefore derive almost equivalent conclusions, with regard to the quality of the resulting rendered images, for all existing methodologies.

Volume Rendering

In order to compare the quality of volume rendered results directly with those from surface rendering, we first consider the rendition of binary volumes. This utilises a pre-segmented binarised volume to generate the required opacity and voxel colour arrays for subsequent rendering (see section 3.2.1). These arrays are produced by setting the opacity values of the voxels which lie within the segmented structure to 1, and those outside the structure to 0. The voxel colours are found by directly shading¹ the segmented volume.

We have chosen to adopt the *fuzzy surface model* approach [Lev88], described in section 3.2.1, for the purpose of assessing the performance of volume rendering non-binarised data sets. This method is less computationally involved than its *translucent coloured voxel model* [DCH88] counterpart (see also section 3.2.1). It is

¹The gradient values used for all the volume rendering results presented in this section were computed using equation 3.2.

therefore more likely ultimately to achieve real-time performance. The image data is first classified by the FFCM algorithm (see chapter 5) and is subsequently prepared for rendering. A limitation of the fuzzy surface approach to volume rendering is the way in which it shades the voxel colour array. The method assumes that every entry of the volume array is illuminated by the same coloured light source. If several different tissue types are present in the data, it becomes extremely difficult to distinguish between them. We therefore illuminate each voxel by a light source which is coloured according to the colour blended output of the FFCM algorithm for that specific voxel. This is equivalent to independently illuminating each different tissue within the volume by a different coloured light source and then compositing the resulting renditions into one scene. The actual rendering stage of the overall process is performed by ray casting and front to back compositing as described in section 3.2.2.

To conclude the study, we examine the need for, and the effect of, matting operations (see section 3.2.3) in order to produce meaningful and useful volume renditions.

7.1.2 Results and Discussion

In this section, we demonstrate the performance of the above described methods by applying them to the IMR head data set which was acquired using the imaging protocols described in chapter 4 (figure 4.2(a) shows the middle slice of this data set).

Quality of 3D Rendered Display

Figure 7.1(a) shows a surface rendition of the segmented structure corresponding to the grey matter of the brain. This was extracted by first applying the FFCM algorithm to the data, whilst setting $c = 5$, $m = 2.0$, etc., as detailed in section

5.4.2. The result was then defuzzified and the structure of interest was subsequently extracted by direct 3D seed filling using the algorithm described in appendix C. No adaptive refinement was required during this procedure due to the excellent spatial grey scale homogeneity of the grey matter of the brain and the high resolution of the acquired data. In order to give a more realistic result, the segmented data was smoothed by convolving it with a (3×3) mean averaging filter before being rendered. It should be noted that due to the fairly high inter-slice resolution, using a (3×3) filter completely removes any jagged effect which would result with no filtering. The rendered knee bone data, displayed in the previous chapter in figure 6.25, required a (4×4) kernel in order to produce a smooth output. This was due to the much lower inter-slice resolution and to the inter-slice gap of its original data.

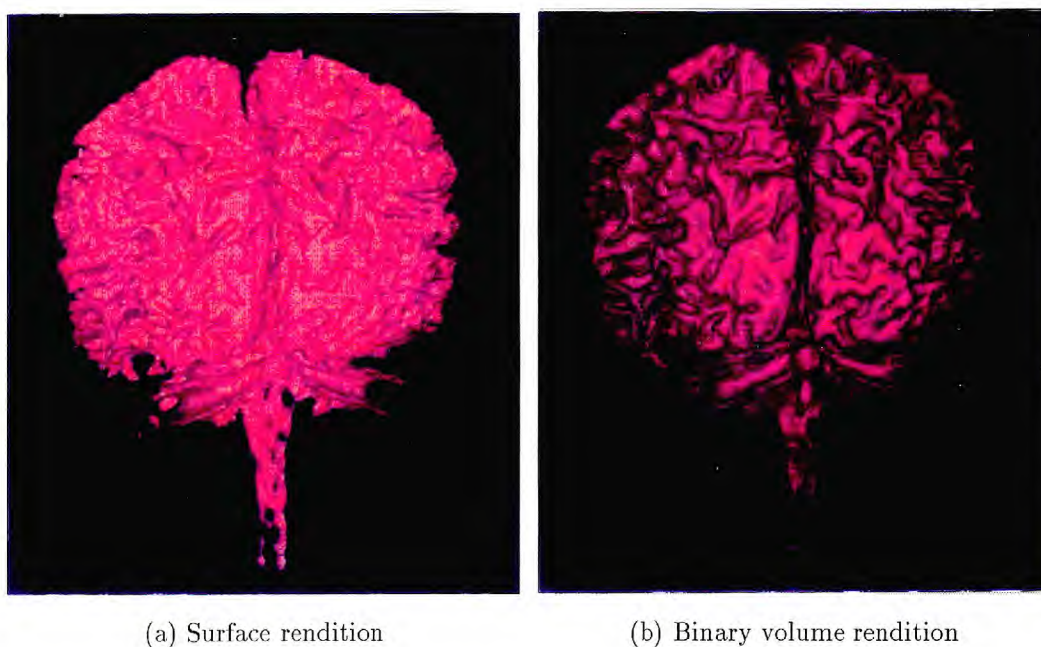


Figure 7.1: Surface and volume renditions of segmented IMR brain data

The result from binary volume rendering the same segmented structure, also previously smoothed by a (3×3) mean averaging filter and viewed from the same direction and perspective, is displayed in figure 7.1(b). It appears from these results

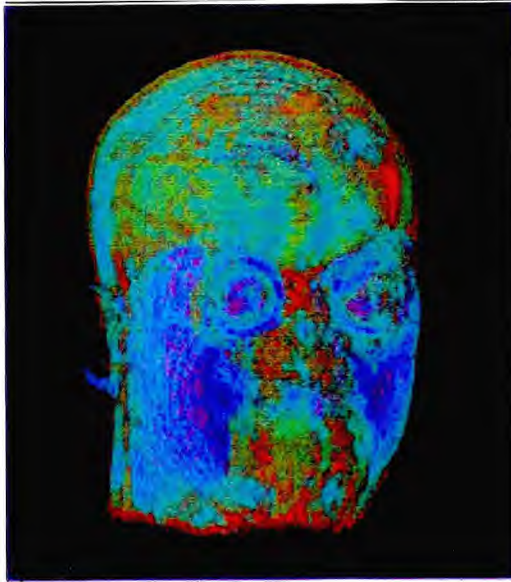
that there is little to choose between the two approaches in terms of image quality. Both outputs provide very life like renditions and show distinctly the structural details of the brain. The only difference worth mentioning is that the surface rendered image looks slightly more synthetic than the volume rendered image due to the unrealistically smooth nature of its surface. However, in terms of providing clinically useful information, it appears that both results are equally good.

Figure 7.2 shows results from volume rendering the entire head volume using the fuzzy surface model approach. The data was initially classified by the FFCM algorithm, as described above. The opacities assigned to the different tissue types as well as their corresponding colour allocations are referenced in table 7.1. Figure

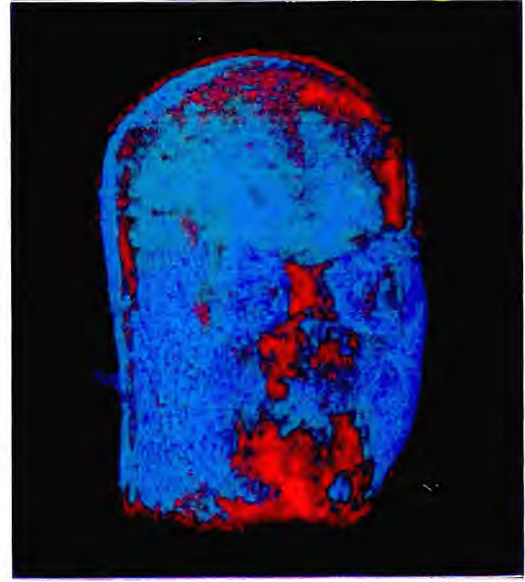
Opactiy	Colour	Corresponding tissue type
α_1	black	air and bone
α_2	red	fluid tissue
α_3	green/yellow	white matter tissue
α_4	cyan	grey matter tissue
α_5	purple	high density fat tissue

Table 7.1: Tissue opacities and corresponding colour allocations and tissue types resulting from FFCM classification.

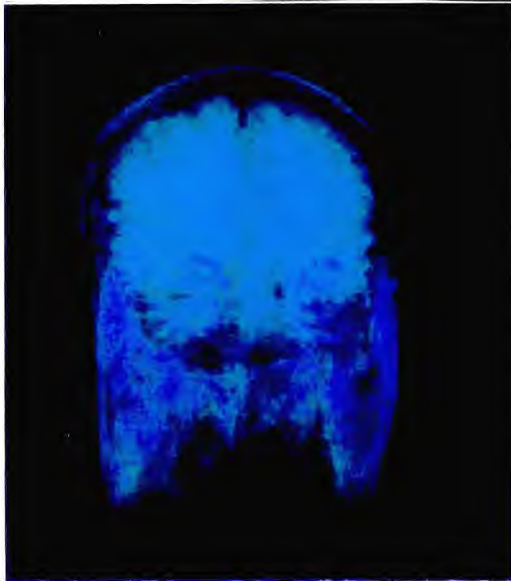
7.2(a) shows the resulting volume rendition when the opacity values have been set as follows: $\alpha_1 = 0$; $\alpha_2 = 0.3$; $\alpha_3 = 0.5$; $\alpha_4 = 1.0$; $\alpha_5 = 0.8$. The external features of the head are quite clear in this image. However, due to the multi-tissue composition of its external skin layers, it is difficult to identify the entire structure clearly. We attempted to dissolve the outer layers, in order to view the interior anatomy, by adjusting the opacity values. We set α_3 and α_5 to zero whilst maintaining the other values as listed above. The resulting rendition is shown in figure 7.2(b). It is now possible to make out the outline of the grey matter of the brain through the semi-transparent fluid tissue. However, the outer skin layer of the head is also partially composed of grey matter type tissue. This blocks the view of the inside of the head



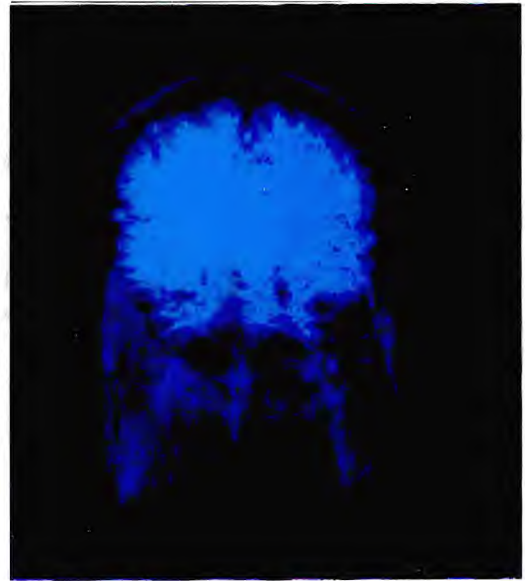
(a) Opacity values: $\alpha_1 = 0$; $\alpha_2 = 0.3$; $\alpha_3 = 0.5$; $\alpha_4 = 1.0$; $\alpha_5 = 0.8$.



(b) Opacity values: $\alpha_1 = 0$; $\alpha_2 = 0.3$; $\alpha_3 = 0$; $\alpha_4 = 1.0$; $\alpha_5 = 0$.



(c) Matte volume corresponding to segmented brain data is applied with opacity values set as for figure 7.2(a).



(d) Matte volume corresponding to segmented brain data is applied with opacity values set as for figure 7.2(b).

Figure 7.2: Volume renditions of FFCM classified IMR head data.

and prevents any internal structures from being visualised distinctly. In fact, we did not succeed in identifying clearly any individual structure by the adjustment of tissue opacity values alone.

In order to demonstrate the effect of matting on volume rendering, the same segmented volume, chosen earlier to demonstrate surface and binary volume rendering, was used. A matte volume was formed whose voxel values corresponding to the segmented brain were set to 1.0 and those outside the brain were set to 0.15. This was then multiplied with the same opacity arrays used to prepare the results of figures 7.2(a) and (b). New renditions were consequently generated, the results of which are displayed in figures 7.2(c) and (d). The grey matter structure within the brain may now be clearly delineated from its surrounding structures which have been made more transparent. The degree to which the external surrounding is made transparent is easily controlled by the corresponding values of the matte volume. It may be observed that the effectiveness of 3D shading on the various external structures are reduced as they are made more transparent. Furthermore, the structural details of the brain are not as distinct as those of the surface or binary voxel renditions of figure 7.1. However, it is important to note that these results are fairly preliminary. Suggestions for future research which may improve these results are discussed in chapter 8.

It is evident from the results displayed in figure 7.2 that FFCM classification alone does not provide sufficient demarcation between different anatomical structures to facilitate clinically useful volume renditions. At best, a rough and ready view results. Although the original proponents of volume rendering [DCH88, Lev88] claimed that pattern recognition classification techniques alone provide satisfactory results, they only applied their methods to X-ray CT data. The image contrast of X-ray CT is extremely good for bone but rather poor for other tissues. This is not the case for MRI. The wide ranging tissue contrast which is imaged makes it impossible to classify the anatomy uniquely according to only its feature space information. We

may therefore conclude that volume rendering is only capable of producing meaningful renditions in conjunction with appropriate segmentation techniques and matting operations.

Computational Efficiency of Rendering Techniques

For surface rendering, the processes of smoothing and surface model formation lasted about 15 seconds on a DEC Alpha AXP/200 workstation. The subsequent rendering stage took about five seconds. These processes could be accelerated to give real-time performances by using a dedicated graphics engine with built in rendering hardware.

The computational overheads of the volume rendering implementations were greater than for surface rendering. However, we did not attempt to optimise the necessary processes. As discussed in section 3.3.1, various algorithmic approaches [Lev90b, UO93, LL94] have been proposed which can significantly reduce the required processing times. By combining these software improvements with specific hardware, real-time performances can be achieved.

7.2 3D Image Guided Surgery

In this section, we explore the feasibility of applying the fast and automatic 3D visualisation techniques, developed in this thesis, to guide and enhance surgical procedures. As discussed in chapter 1, in order to initiate conventional 3D image guided surgery, it is necessary to obtain pre-operative image data within the context of a fixed reference. This would not be required if the surgery was performed, whilst 3D image data was simultaneously acquired, within an IMR unit. However, since real-time 3D IMR imaging is not yet realisable, we will focus specifically on conventional image guided routines.

In order to demonstrate the potential of 3D image guided surgery, as well as to

bring to light the necessary considerations for its implementation, we conducted an experiment on a phantom knee model. In the following sections, the procedure for this experiment is detailed. This is followed by a discussion on the registration of two co-ordinate systems within the context of 3D image guided surgery. Experimental results are then presented and discussed.

7.2.1 Experimental Methods

We constructed a perspex model of a simplified knee. This comprises two hollow “bones” which are filled with copper sulphate solution. These “bones” are held in place inside a tube, the “leg”, with a number of holes drilled at specific points around its circumference.

In order to simulate the steps performed during 3D image guided surgery, the following procedure was performed. The knee model was imaged by a conventional MR scanner. During the scan, the holes in the “leg” were filled with MR sensitive markers in order to provide a fixed reference between image-space and world-space. Following the imaging sequence, the knee model was positioned on an accurately marked grid according to a fixed reference point. An arthroscope² was inserted through one of the pre-drilled holes in the “leg” and held in position by a clamp. The world-space co-ordinates of the centres of the holes, which had been filled with markers during imaging, were then measured. In addition, the co-ordinates of two points along the length of the arthroscope were measured and the tip co-ordinate was thereby computed. The x and y co-ordinates of all the above mentioned locations were measured by reading their values straight off the grid, and their z co-ordinates were found using a metre rule.

Finally, the world co-ordinate system was registered with its image-space counterpart by solving the equations described in the next section. The measured and computed co-ordinates of the probe were hence converted to image-space in which a

²An arthroscope is an endoscope used specifically for orthopaedic surgery.

graphical model of the probe was subsequently generated. The synthetic probe and the phantom image data were then rendered together to produce a 3D global view of the operative scene.

7.2.2 The Registration of World-Space with Image-Space

Given the 3D co-ordinates of at least three fixed reference markers in both world-space and image-space, the values for suitable rotation and translation matrices, which convert the co-ordinates from one system to the other, may be computed as follows:

Let the two 3D point sets be referred to as, $\{p_i\}$ and $\{p'_i\}$, where $\{p'_i\}$ is $\{p_i\}$'s corresponding point set in a different co-ordinate system. $i = 1, 2, \dots, N$ corresponds to N different points in each co-ordinate system and each member of the 3D point sets is a 3×1 column matrix. The two co-ordinate systems may be registered with each other by finding suitable values for R and T such that,

$$p'_i = Rp_i + T + N_i \quad (7.1)$$

where, R is a 3×3 rotation matrix³, T is a translation vector (3×1 column matrix), and N_i is a noise vector. Optimal values of R and T may be found by minimising the following equation:

$$\sum^2 = \sum_{i=1}^N \|p'_i - (Rp_i + T)\|^2 \quad (7.2)$$

where $\|\cdot\|$ is the Euclidean vector norm. A solution for this minimisation problem may be found in appendix D, and is presented in full detail in [AHB87].

³In the event that one co-ordinate system needs to be scaled with the other, the degree of scaling will be incorporated within the values of the rotation matrix.

7.2.3 Results and Discussion

The Registration of World-Space with Image-Space

A proton density weighted data set of the knee model was obtained from an Elscint 1.9 Tesla MRI scanner. The volume comprised 70 slices, each slice of 256×256 resolution with each pixel of size $(1 \text{ mm})^2$. The slice thickness was 3 mm with no inter-slice gaps.

The measured values, in both image and world space, of the co-ordinates of the markers attached to fixed points on the “leg” of the knee model, are displayed in table 7.2. The dimensions used, in both co-ordinate systems, are in mm. The

Co-ordinates of Markers (x, y, z) (measured in mm)			
Measured in image-space	Measured in real world-space	Registered from world to image-space	Errors in registration
96, 100, 107.5	0.0, 0.0, 181.2	97.1, 102.2, 108.7	+1.1, +2.2, -0.8
132, 87, 109.5	34.7, 13.2, 181.6	131.1, 86.4, 109.2	-0.9, -0.6, -0.5
171, 102, 109.5	52.0, 51.3, 182.3	170.8, 100.7, 109.4	-0.2, -1.3, -0.1
186, 139, 106.5	36.1, 89.4, 180.2	187.1, 138.9, 106.7	+1.1, -0.1, +0.2
99, 177, 109.5	-51.9, 49.4, 181.8	96.4, 174.5, 108.3	-2.6, -2.5, -1.2
80, 138, 107.3	-37.6, 16.1, 182.3	82.4, 140.7, 109.4	+2.4, +2.7, +2.1

Table 7.2: Measured and registered co-ordinates of markers positioned around the circumference of the “leg” of the knee model.

top left hand vertex of the image volume is considered as the origin of image-space. Thus, the x - y co-ordinates in image-space are simply the x - y voxel co-ordinates. Since each slice is 3 mm thick, the z co-ordinates in image-space are scaled by three. The co-ordinates of centres of each marker in image space were measured by the direct 3D seed filling of each marker whilst simultaneously computing its centroid.

It should be noted that three fixed points are sufficient to calculate the appropriate registration transformations. However, the accuracy of the results improves as more points are used. This is due to the consequent averaging of any introduced noise, N_i . We have therefore considered the co-ordinate values of six different

markers.

The measured values of the markers in world-space were assigned to $\{p_i\}$ and the corresponding values in image-space to $\{p'_i\}$, where $i = \{1, \dots, 6\}$. The optimal transformation matrices, R and T , which result from minimising equation 7.2, were then calculated using the algorithm described in appendix D. Their values were found to be:

$$R = \begin{bmatrix} 0.70 & 0.72 & 0.01 \\ -0.72 & 0.70 & 0.02 \\ 0.01 & -0.01 & 1.00 \end{bmatrix} \quad \text{and} \quad T = \begin{bmatrix} 96.07 \\ 99.42 \\ -73.95 \end{bmatrix}$$

Table 7.2 shows the results of applying these transformation matrices to the co-ordinates of the markers when measured in world space. The errors resulting, from registering the marker co-ordinates, are also displayed in table 7.2. Although the mean error in (x, y, z) co-ordinates is $(0.8, 0.4, -0.3)$, the worst case errors approach 3 mm in all three dimensions. Although such errors are not enormous, within the context of surgery they may be quite significant. It should be noted however that the accuracy of this experiment is limited by the 3 mm slice thickness as well as by the very crude approach used to measure the co-ordinates of the markers in world-space. Higher resolution acquisitions and more accurate measurement techniques should reduce any registration errors dramatically. It should be mentioned that for IMR image guided interventions, image resolution does not limit the registration accuracy since no reference marker system is required as both imaging and tracking are conducted within the same environment.

3D Rendered display of Image Data and Surgical Instruments

In order to best represent the results of the phantom knee experiment, we have chosen to do so within a surface rendered environment. Figure 7.3(a) shows a surface rendition of the knee model, the fixed reference markers and a geometric model of the arthroscopic probe. Due to the poor inter-slice resolution of the image

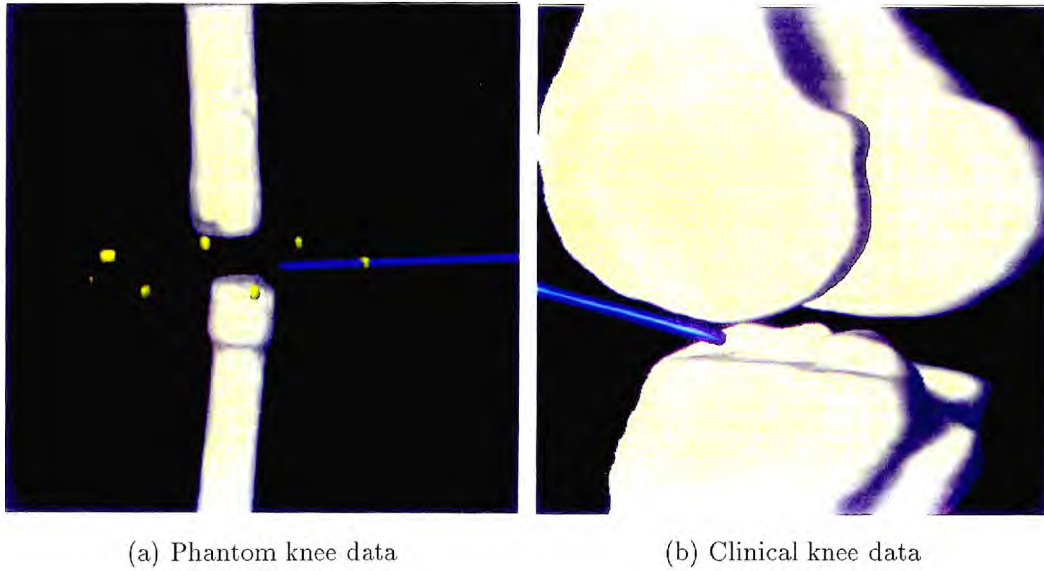


Figure 7.3: (a) Surface rendition of knee phantom, localisation markers (yellow) and registered geometric probe model (blue). (b) Magnified surface rendered view of segmented knee bone data with simulated probe positioned inside the joint.

data, we have smoothed it with a (3×3) mean averaging filter before rendering it using the methods described in section 7.1.1. The probe model was generated by transforming the measured co-ordinates of two points along its axis, as well as the calculated co-ordinate of the tip, into image-space by using the computed R and T matrices. Note how the probe intersects one of the markers. As mentioned in section 7.2.1, the arthroscope was inserted through one of the drilled holes in the “leg” which previously housed a marker during imaging. Thus it would appear that the registered synthetic probe provides an accurate rendition of the real probe.

The experiment which we have conducted on the knee model demonstrates the benefits of combining 3D imaging with video endoscopy during minimally invasive procedures. In practice, we envisage that two separate displays will be positioned in parallel, one with the video image captured by the endoscope and the other with the 3D rendered global view. This will allow the surgeon to look at either display during the operation.

It should be noted that arthroscopic probes do not view ahead but rather at an angle (typically 30°) off the probe axis. In order to perceive the axis viewed by such a probe within the rendered scene, it will be necessary to track additional points on the arthroscope which lie off the actual axis of the probe. These must then be transformed to image-space in which the viewing direction of the probe may then be determined. A virtual arrow pointing to this direction may then be generated at the end of the probe in the rendered image.

The results discussed so far, from the knee model, have highlighted the registration and display aspects of 3D image guided surgery. Since the phantom knee is homogeneous, the entire region which corresponds to the phantom in the acquired image data is of almost uniform intensity. Thus, no segmentation is required in order to render the data successfully. However, this is clearly not the case when acquiring real clinical data. Figure 7.3(b) shows the magnified view of a surface rendition of segmented, conventional MRI knee bone data which was obtained by the AR process as described in chapter 6. A synthetic probe model was generated and positioned inside the joint in order to simulate the output that might be obtained during an actual MIS procedure.

We have only illustrated combined 3D views of segmented data and endoscopic data by adopting a surface rendering scheme. The pursuit of a volume rendering approach might provide more anatomical information due to its ability to incorporate transparency effects and accommodate fuzzy surfaces. This would allow the peeling away or dissolving of structures through which the probe traverses. However, this requires further work on both segmentation and volume rendering processes. Although by use of the AR process, described in chapter 6, it is possible to extract the main anatomical structures automatically and accurately, it cannot yet be applied to isolate some of the more complicated and poorly defined regions successfully. Further work is likely to improve its versatility. With regard to improving the capability of volume rendering, the reader is referred to the following chapter. It

should be noted that if a volume rendering scheme is adopted, the smoothness of the rendered probe would be limited by the resolution of the data. This could result in producing quite a jagged effect. A potential solution would be to surface render the probe within the volume rendered environment using a hybrid method such as described in [Lev90a]. Further research is required in order to determine which rendering strategy is optimal for image guided surgery. However, it is likely that the eventual method adopted will depend on the particular region of the anatomy being imaged and the type of MIS procedure to be performed.

The results in this section have only been presented for conventional 3D image guided surgery where the image data is acquired prior to surgery. We have demonstrated in section 7.1 how the fast and automatic techniques developed in chapters 5 and 6 are able to identify structures within IMR data accurately. It is therefore fair to assume that as the duration of IMR imaging times is steadily being reduced by improved manufacturing developments, it will eventually be possible to acquire and visualise 3D IMR data simultaneously in real-time during interventional routines.

7.3 Other Applications

This chapter has specifically focused on the application of fast and automatic 3D visualisation techniques in image guided surgery. However, these methods are beneficial to all the other applications described section 1.3. The ability to process the data automatically and quickly is likely to encourage and enable more clinicians to take advantage of 3D visualisation methods. Furthermore, the potential realisation of simultaneous real-time 3D acquisition and visualisation of IMR data will enable a far wider range of data to be viewed and analysed. This may include kinematic studies, e.g. patellar tracking, the effects of loading on different structures, etc..

7.4 Conclusions

We have demonstrated the efficacy of different rendering schemes applied to FFCM classified and AR segmented MRI data. Surface rendering and binary volume rendering produced excellent highly defined 3D renditions of specific structures of interest. It was shown that volume rendering non-binarised FFCM classified data yields a confusing display in which, due to the over portrayal of anatomical information, it is not possible to clearly identify any specific structure. The application of appropriate matting operations was shown to alleviate this problem. However, further research is required to enable the contents of the resulting display to be of comparable clarity to their surface rendered counterparts. It should be noted that the results of the evaluation study presented in this chapter are from only IMR data. Equivalent, if not better, renditions would result from conventional MR image data.

The fast visualisation techniques developed in this thesis were considered for various clinical applications. In particular, their applicability to 3D image guided surgery was examined. A phantom knee experiment was described which demonstrated how 3D visualisation methods applied to global MRI data could be used to guide MIS procedures. The FFCM classification and AR segmentation methods favoured the adoption of a surface rendered viewing scheme. However, future work is required to determine the use of a volume rendering approach in this context. Assuming that it will eventually be possible to acquire 3D IMR image data in real-time, the newly developed classification and segmentation techniques should allow real-time visualisation of anatomical structures during both surgery and other procedures.

Chapter 8

Final Discussion, Conclusions and Future Work

This chapter concludes the work presented in this thesis. A summary of the main results of the thesis is first presented. Final conclusions are then made in the light of these results. Finally, ideas for future work are detailed.

8.1 Summary of Main Results

The main results of this thesis may be summarised as follows:

- T_1 weighted anatomical image data, suitable for subsequent 3D visualisation, were obtained from both conventional MRI as well as IMR scanners.
- The new FFCM algorithm was shown to classify MRI data accurately. As well as producing better quality results to alternative existing methods, the FFCM method reduces the computation time of its original FCM counterpart from several hours to less than one second.
- The AR process was developed for the purpose of refining the classified output of the FFCM algorithm adaptively. This was accomplished in order to identify accurately individual structures which were either misclassified or composed

of more than one tissue type. The method was applied to MRI knee data and by this means it was possible to extract 3D bone data accurately and quickly.

- 3D rendering methods were applied to FFCM classified, and AR segmented, data. Excellent renditions were obtained when applying both surface rendering and binary volume rendering schemes to segmented volumes. It was found that volume rendering non-binarised FFCM classified data yielded poor results. These were improved by performing appropriate matting operations.

8.2 Final Conclusions

We may make the following conclusions in light of the results presented in this thesis:

The FFCM method produces fast and accurate classifications. However it has been shown to be insufficient to enable meaningful renditions to result. This is due to the multiple tissue consistency of anatomical structures present in MRI data. It is therefore necessary to identify actual structures through a rigorous segmentation procedure. The AR process has been shown to extract the most prominent anatomical structures within volume data accurately. However, it is not yet possible to isolate complex or poorly defined structures through applying the AR process alone.

The application of FFCM classification and AR segmentation methods to the 3D visualisation process has been explored. The results from a preliminary study suggest that a more meaningful final display is obtained by pursuing a surface rendering rather than a volume rendering approach. However, further research is necessary before final conclusions may be made.

The importance of fast and automatic 3D visualisation methods for clinical applications has been demonstrated for 3D image guided surgery. It was shown how the global anatomical view provided may be used to guide the surgeon.

The work presented in this thesis provides a step forwards towards fast and automatic visualisation of both 3D MRI and IMR data. Although we have found that the image acquisition stage of the process takes several minutes, it is envisaged that with further development, real-time or near real-time imaging will be realisable. This will allow clinicians to explore and manipulate the internal anatomy of a patient, in real-time, through a virtual 3D rendered world.

8.3 Future Work

In order to develop the research presented in this thesis further, we recommend the following areas for future consideration.

8.3.1 Image Acquisition

In chapter 4, a preliminary study to investigate the capability of conventional and interventional MR imagers, for subsequent 3D visualisation, was presented. It was found that due to the weaker magnetic field in the IMR scanner it was necessary to increase the NEX in order to obtain high quality images. This resulted in an extended acquisition process and the risk of motion artefacts.

In order to investigate ways of reducing the required acquisition times for IMR imaging, we suggest that the following avenues of research be pursued:

1. The same IMR imaging sequences as described in section 4.1.2 should be adopted but with varying combinations of lower NEX and/or lower resolutions. The performance of the processing and rendering techniques described in chapters 5 to 7 should then be assessed on the newly acquired data to see if useful results can be produced from them.
2. Different IMR imaging sequences should be investigated. It is worth attempting to acquire spin echo head data and compare it with its SPGR counterpart.

This may result in lower imaging times. Furthermore, the performance of the fast SSFSE and FSPGR sequences especially designed for IMR imaging (see chapter 4) should be examined.

It is also worth investigating image processing tasks such as noise removal and restoration algorithms to see whether they may improve the quality of noisy data obtained through fast acquisitions so that it can be subsequently visualised in 3D. However, it is important that such techniques should not take longer to process than the time saved by using a faster imaging sequence.

8.3.2 Image Segmentation

New approaches for fast and automatic classification and segmentation have been described in chapters 5 and 6. The AR process, described in chapter 6, is a technique which requires further development if its performance is to be optimised. The following areas are worth pursuing:

Region Merging

The hierarchical approach to region merging was described in section 6.4.2. Although very good results have been obtained when applying this technique to MRI data, the following aspects may further improve the consistency of its performance. First of all, it is worth investigating the relative efficacy of other processing techniques for simple homogeneity testing. Secondly, the performance of MRF based homogeneity testing may be improved if a model can be chosen for the low level process which fits the MR image grey scale distribution better than the currently adopted Gaussian model.

Region Splitting

The need for incorporating region splitting into the AR process has been discussed in chapter 6. Future research is necessary in order to formalise methods which are able to split under-segmented regions accurately. We suggest the following ideas:

1. Morphological opening operations (see section 2.3.2) should be performed along the boundary contour of a selected region. If at any section of the boundary contour the morphological operation suggests that the region should be split, various homogeneity tests should be performed within its vicinity in order to verify the suspicion.
2. When the AR process is applied to 3D data, it is necessary to extract a 3D region from the volume as described in section 6.5. During this procedure, if the shape of the region extracted from one slice differs greatly from either of its adjacent counterparts, this suggests that the region may need to be split.
3. The incorporation of a knowledge based system may be useful in guiding the region splitting stage of the AR process. This may involve comparing the data with pre-segmented data previously acquired from the same patient, or with a 3D anatomical “atlas”.

3D Region Extraction

As mentioned above, when applying the AR process to 3D image data, it is necessary to perform 3D region extraction. In section 6.5, algorithm 6.6 was detailed for this purpose. Although this algorithm is capable of extracting 3D regions such as knee bones, it is unable to cope with complex branching structures. Future work on extending the algorithm to such situations should be carried out. This may be best achieved by using a binary tree data structure to represent a complex 3D region. The consideration of knowledge based systems, mentioned above for the purpose of region

splitting, may also help guide the 3D region extraction process when attempting to isolate very complex structures. Furthermore, knowledge based systems should be considered for the purpose of the automatic identification of specific structures which have already been segmented.

8.3.3 3D Rendering Methodologies

A preliminary study on the performance of different 3D rendering schemes was presented in chapter 7. It was found that volume rendering methods need further development if they are to be practically adopted. We suggest that the following areas be investigated:

1. Instead of adopting the colour blending approach, described in chapter 5, different colours, which may provide a clearer contrast, should be allocated to each tissue type.
2. The edge information used for the volume rendering performed in this thesis has been computed from equation 3.2. It is worth investigating the effect of considering the density characteristics defined by equation 3.4 whilst extracting the surface information.
3. It is worth investigating the behaviour of different types of matting volumes applied during rendering. The effects of applying a depth cue to the matte voxel values according to their distances from the segmented structure should be examined. Furthermore, the way matting functions are formed and applied should be considered for the purpose of peeling away different structures from within the data.
4. The colouring of voxels according to fuzzy structures as opposed to tissue types should be considered. This may involve research into fuzzy object extraction.

It is worth investigating whether performing smoothing operations prior to classification and segmentation yields better or worse results than applying them afterwards. This should be checked for both surface and volume rendering methods.

8.3.4 3D Image Guided Surgery

After further research has been conducted on improving 3D rendering methods, a new study should be performed which examines the relative merits of the different techniques within the context of 3D image guided surgery. Ways in which 3D rendered MRI data may be best displayed alongside endoscopic video data should also be explored. Finally, a prototype system should be built and tested for both conventional image guided routines and for IMR image guided procedures. With regard to the latter, it is important to pay special attention to the limited space available within the IMR unit.

Appendix A

Fundamentals of MRI

In this appendix, the basic fundamentals of MRI are presented. First, elementary principals of Nuclear Magnetic Resonance (NMR) are discussed. This is followed by a brief description of MR image formation and properties. For a more detailed and thorough presentation of this subject, the reader is referred to [EKWA90, Fos84, FH87, TIB88].

A.1 Basic NMR Theory

A.1.1 Spins in a Magnetic Field

During the development of quantum mechanics, it became apparent that most nuclei possess a property called spin momentum. Since atomic nuclei are charged, the spinning motion generates a magnetic moment along the spin axis. The Hydrogen (^1H) nucleus possesses the strongest magnetic moment. This, together with its biological abundance, makes ^1H the nucleus of choice for MR imaging.

Let us consider a collection of protons (i.e. ^1H nuclei). In the absence of an external magnetic field, each individual nucleus spins at a random orientation. However, under the influence of an external magnetic field, $\mathbf{B}_0$, each proton takes up one of two possible orientations: parallel or anti-parallel to $\mathbf{B}_0$. A slightly larger

number of protons adopt the parallel alignment, due to its lower energy state, and a net magnetic moment, M , parallel to B_0 , results, as shown in figure A.1(a). Although the net moment, M , is parallel to B_0 , each individual proton precesses, at a slight tilt off the B_0 axis, at the Larmor frequency, f_L . This is defined by the equation

$$f_L = \gamma B_0 \quad (\text{A.1})$$

where γ is the gyromagnetic ratio and B_0 is the magnitude of the external field.

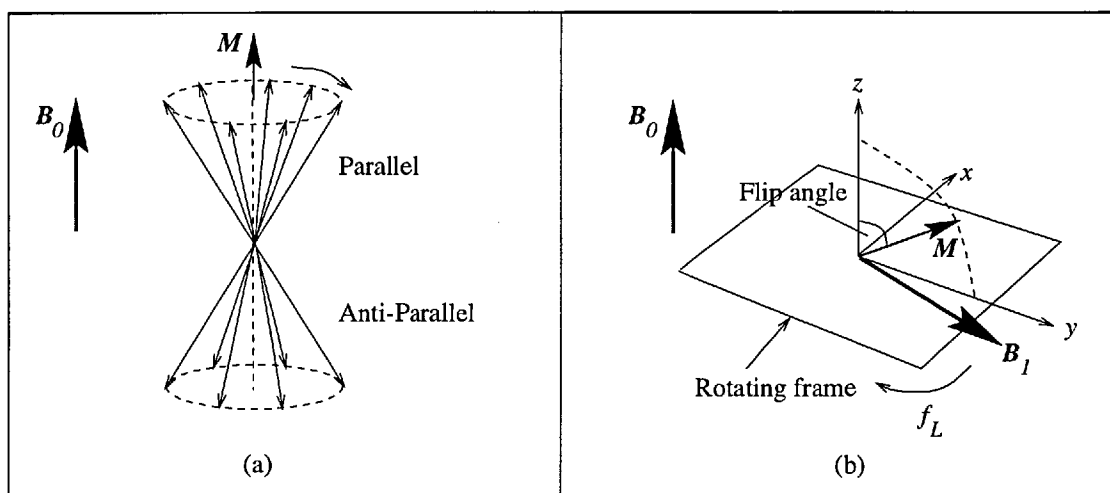


Figure A.1: (a) Protons under the influence of an external field, B_0 . (b) An RF pulse is applied, which acts as a second field, B_1 , in the plane perpendicular to B_0 . This is portrayed within the *rotating frame* system of reference

A.1.2 The Effect of Radiofrequency (RF) Pulses

In the presence of only B_0 , since the magnetic moment, M , is parallel to B_0 , no MR signal may be detected. In order for an MR signal to be induced in receiver coils, it is necessary for M to vary with time, outside its equilibrium position. This condition is referred to as *resonance*. Resonance may be achieved if the protons are irradiated by RF energy at the Larmor frequency. This RF energy may be considered as a second magnetic field, B_1 , which is perpendicular to B_0 and is rotating at f_L . This causes M to spiral away from the B_0 field, also at the Larmor frequency, f_L . In order to simplify our perspective of the motion of M , we adopt the *rotating frame*

system of reference, as shown in figure A.1(b). This considers $\mathbf{M}$ and $\mathbf{B}_1$ as seen from an observer sitting on an orthogonal plane which is rotating around $\mathbf{B}_0$ at the Larmor frequency. Within this reference system, the vector $\mathbf{M}$ is simply tilted away from the z axis. As the strength and/or duration of the RF pulse increases, $\mathbf{M}$ moves further from its equilibrium position. The total angle through which $\mathbf{M}$ traverses is called the *flip angle*, as shown in figure A.1(b).

Let us consider the situation immediately after the irradiation of a 90° pulse. The net magnetisation, $\mathbf{M}$, now lies in the transverse plane and precesses around the z axis. This induces an AC current in a suitably positioned receiver coil. The induced signal is referred to a *free induction signal* (FIS). This signal decays with time due to a process known as relaxation.

A.1.3 Relaxation: The Return to Equilibrium

After the RF pulse has been switched off, the excited magnetisation vector, $\mathbf{M}$, begins to return to its original equilibrium position. This is caused by two independent relaxation processes which take place simultaneously. These are known as *longitudinal* and *transverse* relaxation.

Longitudinal Relaxation

Immediately after a 90° pulse, the net magnetisation vector lies in the transverse plane. Therefore, the amount of longitudinal magnetisation is zero. Furthermore, the two energy levels, which correspond to parallel and anti-parallel spins, are equal. Longitudinal relaxation restores the original equilibrium population difference by returning some spins to the lower (parallel) state, giving out energy to the environment (lattice) in the process. With time, the longitudinal magnetisation, M_L , returns to its equilibrium value, M_L^0 , according to the exponential process $M_L = M_L^0(1 - e^{-t/T_1})$. T_1 is the amount of time taken for M_L to return to 63% of

its original value, M_L^0 . The process is therefore referred to as T_1 , or spin-lattice, relaxation.

Transverse Relaxation

The net magnetisation in the transverse plane, M_T , decays, from its initial value, M_T^0 , to its equilibrium zero, through the exponential process, $M_T = M_T^0 e^{-t/T_2^*}$. T_2^* is the amount of time taken for M_T to decay by 37%. The physical cause of this relaxation is the *dephasing* of the different components of magnetisation which precess at slightly different rates. The main cause of this dephasing is the inhomogeneity in the external field, B_0 . However, even if B_0 were completely homogeneous, a slower dephasing, and accompanying signal decay, would take place due to the physical properties of the system under investigation. This is referred to as T_2 , or *spin-spin*, relaxation.

In order to measure the T_2 relaxation time only, without any influence from B_0 inhomogeneities, the Carr-Purcell-Meiboom-Gill (CPMG), or spin echo, sequence may be implemented, as illustrated in figure A.2. A 90° pulse flips M into the

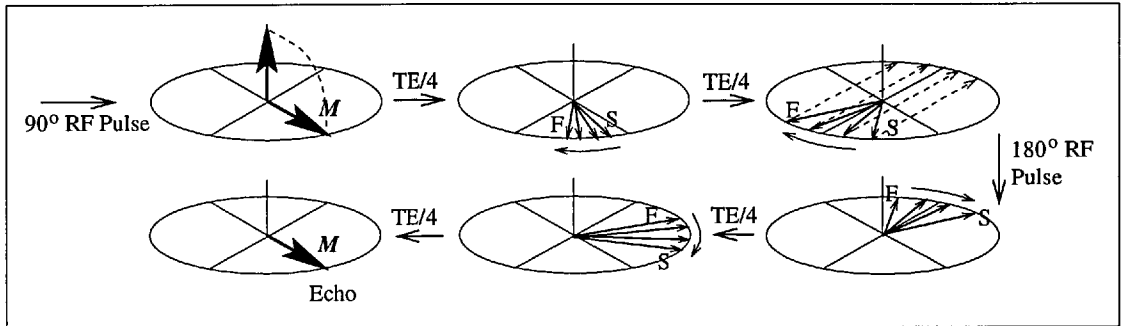


Figure A.2: The CPMG (Spin Echo) Sequence.

transverse plane. As M precesses, its components begin to dephase so that the fast (F) and the slow (S) components may be identified. After time $TE/2$, a 180° RF pulse rotates all the components of M to the opposite side of the transverse plane. Precession now continues with the fast components behind their slower counterparts.

After a full period of TE , M has fully re-phased and the resulting *echo* signal reaches a maximum value. It therefore follows, that the amplitude of this signal decays as a function of T_2 rather than T_2^* .

The values of T_1 and T_2 vary from one tissue to another. However, in all instances, $T_1 \gg T_2$. It is these values which are used as a contrast measure during MR image formation.

A.2 MR Image Formation

In section A.1, the physics behind the NMR signal was described. To facilitate the eventual extraction of image data from this signal, it is necessary to spatially encode the signal. Two main approaches exist which enable either 2D slice or 3D volume acquisition to be achieved.

A.2.1 2D Slice Acquisition

The key to MR image formation is the correct coding of the NMR signal. The most important component of this process is the magnetic field gradient. Figure A.3 demonstrates how imposing a field gradient along the z axis affects the overall external magnetic field. A 2D MR image may be formed by activating a series of

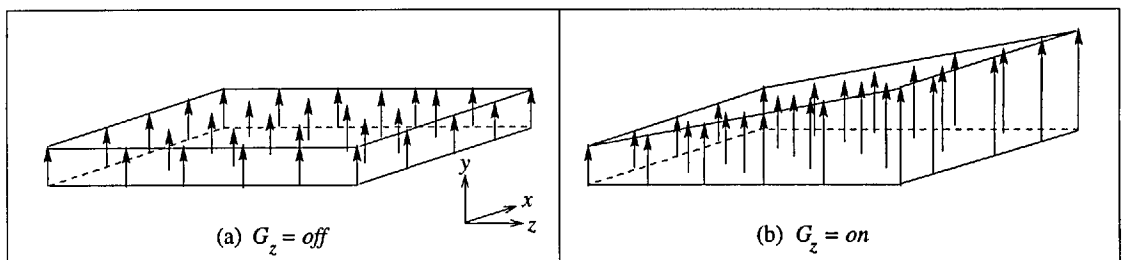


Figure A.3: The external magnetic field with (a) no imposed gradient (b) a gradient introduced along a specific axis.

magnetic field gradients along different axes which change the magnitude of the

Larmor frequencies linearly along the gradients. Three main steps, as summarised in figure A.4(a), are involved in the process. These are:

1. *Slice Excitation:* A gradient, G_{slice} , is imposed along the axis orthogonal to the slice plane of interest. By choosing an appropriate bandwidth for the RF pulse, whilst the gradient is applied, only the protons which lie within the desired slice are excited. The thickness of this slice depends on the gradient magnitude and the RF bandwidth.
2. *Frequency Encoding:* The locations of coordinates along the x axis are encoded by applying a gradient along the x axis (adopting the convention of figure A.4(a)). This is activated while the echo is present, i.e after a period TE . An additional gradient is applied earlier in the sequence to act as a dephaser.
3. *Phase Encoding:* In order to encode information along the y axis of the selected slice, phase encoding is performed. For each individual point along the y axis, a specific gradient is induced which affects the phase of the spins in the slice. The protocol is repeated, after a period TR , with a slightly different gradient value until every point along the y axis has been encoded.

The Spin Echo protocol, as depicted in figure A.5, is the most popular slice acquisition pulse sequence for clinical applications. This takes advantage of the CPMG sequence (see figure A.2) to obtain high quality image data.

In order to extract the image data from the encoded MR signal, the Fast Fourier Transform (FFT) is employed. For each phase encoded signal, a 2D FFT is able to extract the data occupying the corresponding line of the image along the x axis, until a complete image results.

It is possible to acquire more than one slice at once, whilst only marginally increasing the acquisition time, by employing a process called *multi-slice acquisition*. This takes advantage of the fact that the period between each individual pulse sequence, TR , is very much greater than TE . By fitting in sequences, which excite

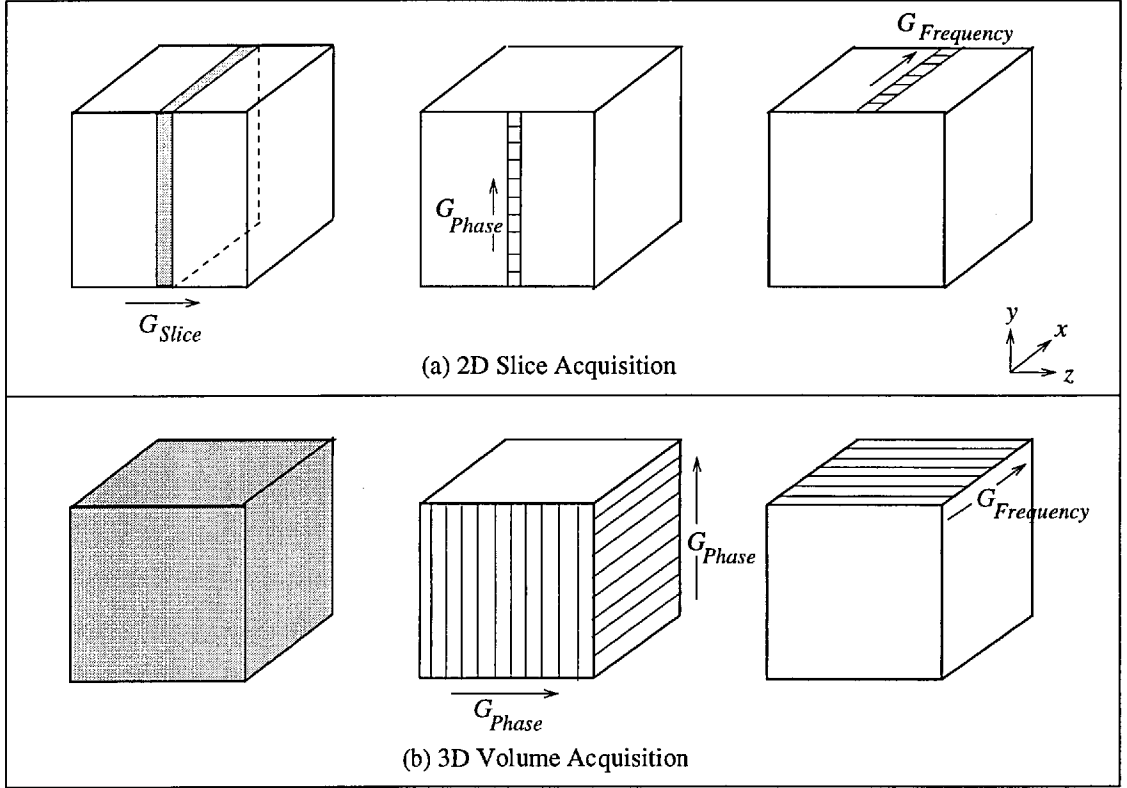


Figure A.4: A summary of how gradient sequences are applied to generate (a) 2D slice images, and (b) 3D volumes.

neighbouring slices, between TE and TR , several image slices may be acquired at once.

A.2.2 3D Volume Acquisition

The process of 3D volume acquisition enables an entire volume to be imaged in one go, rather than via the sequential collection of a stack of individual 2D image slices. Figures A.4(b) shows how this may be achieved. Instead of selecting an individual slice, the entire volume is excited. This requires an additional phase encoding of the spatial information along the z direction. Images are subsequently reconstructed using the 3D FFT.

The relatively long imaging times demanded by a 3D acquisition preclude the use of conventional spin echo sequences. *Gradient echo* acquisitions are therefore

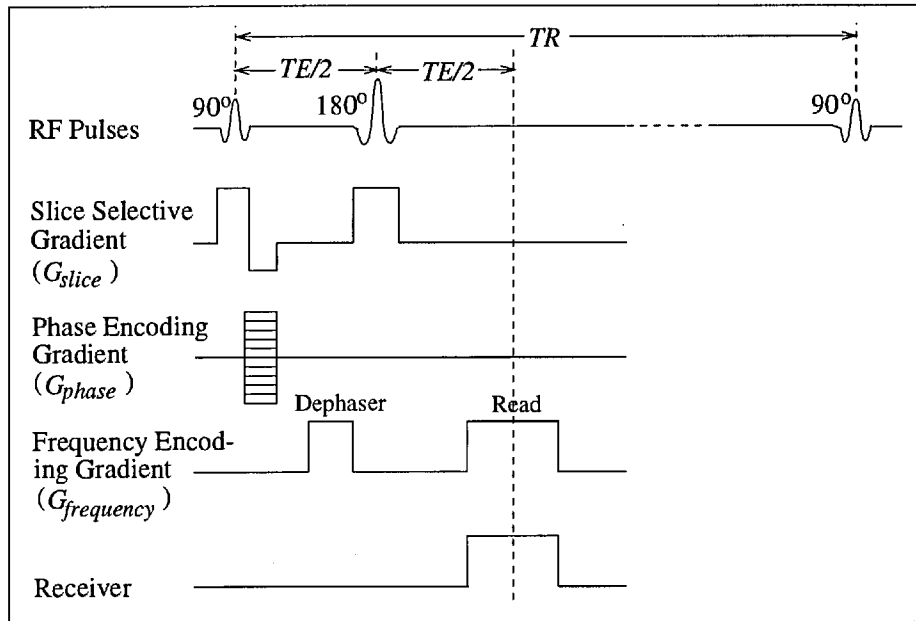


Figure A.5: The Spin Echo (SE) Pulse Sequence

employed. These use RF pulses with flip angles of less than 90° . Gradient pulse sequences enable TR to be substantially reduced, thus allowing volumes to be acquired extremely quickly.

One of the most popular 3D imaging protocols is the “Gradient Recalled Acquisition at Steady State” (GRASS) sequence. This uses very short TR ’s, which results in a steady state condition where some coherent transverse magnetisation is present. By adopting a rewinding gradient pulse in its sequence, the steady state condition is preserved. This contributes some T_2^* weighting to the image. A variation on this sequence, the spoiled GRASS (SPGR), eliminates the T_2^* contribution to the image. This is achieved by adding a phase shift to each successive RF excitation pulse, a process referred to as “spoiling”. This prevents build up of the steady state component.

A.3 MR Image Properties

A.3.1 Image Contrast

Imaging parameters may be chosen such that the image intensity reflects only the proton density, $N(H)$, across the image. However, due to the small variance in $N(H)$ between different tissues, the resulting contrast levels tend to be quite poor. It is therefore desirable to seek other sources of contrast. The relaxation times T_1 and T_2 are good candidates for this. T_1 images tend to give the best contrast for the overall anatomy and T_2 weighted images provide the clearest delineation between different fluid regions.

Table A.1 shows how adjusting the durations of TR and TE affects the contrast of spin echo images. The contrast of GRASS and SPGR images is also dependent

TR	TE	Contrast
$\gg T_1$	$\ll T_2$	$N(H)$ - proton density
$\gg T_1$	$\sim T_2$	T_2
$\sim T_1$	$\ll T_2$	T_1
$\sim T_1$	$\sim T_2$	Mixed

Table A.1: Contrast Extremes for Spin Echo Imaging.

on the values of TR and TE , as shown in table A.1. However, the contrast of these images is also controlled by the magnitude of the flip angle. With GRASS imaging, a low flip angle produces heavily weighted T_2^* image data. As the flip angle is increased, a T_1 and T_2^* mixed weighting results. SPGR imaging, due to the suppression of the steady state condition, is able to eliminate the contribution of T_2^* to the image contrast. Thus, low flip angles yield T_1 and T_2^* mixed weightings and high flip angles produce heavily weighted T_1 images. In general, GRASS sequences are chosen for imaging fluid changes, whereas SPGR protocols are used for anatomical imaging.

A.3.2 Image Quality

All MR images are degraded by noise inherent to the acquisition process. The signal to noise ratio (SNR) and the overall image quality are dependent on the following factors:

1. *Increasing the NEX*: The SNR increases according to the square root of the number of excitations (NEX) per view. Signal averaging, which results from increasing the NEX, can also reduce *ghosting* artefacts which result from patient motion. However, a better remedy is to avoid any motion in the first place.
2. *Image Resolution*: If the image resolution is reduced during acquisition, as a result of the increased voxel size, there are more spins per voxel to contribute to the MR signal. This results in a linear increase of SNR as well as contrast to noise ratio.
3. *External Field Strength*: At higher values of B_0 , the population difference between the two spin energy levels increases, thus improving the SNR.
4. *Receiver Coils*: The electrical resistance of the receiver coils are a source of noise artefact in MR images. By choosing appropriate, high quality, coils, the SNR may improve. In addition, good quality receive and transmit coils improve the spatial homogeneity of the acquired image data.

Appendix B

MRF Based Classification of MRI Data: A Numerical Solution

This appendix describes a numerical solution for the MAP criterion, as defined by equation 2.28 (see chapter 2), for the purpose of MR image data classification. The following sections, describe the models used to represent the low level and high level processes of the hierarchical image model. The optimisation process adopted to solve the MAP criterion is then determined.

B.1 Process Models

The hierarchical image model, as described in section 2.4.2, is adopted as the framework for MRF based classification of MR image data. In this section, we will describe representations for the two processes which are considered by this model.

B.1.1 The Multilevel Logistic (MLL) Model

The MLL model is used to represent the high level process, X , in the hierarchical image model [DC86, DE87]. This model assumes that X is a second order discrete MRF (see definition 2.7). Thus, each element of X is only influenced by a second order neighbourhood, η^2 . The discrete set of possible labels $\Lambda = \{1, 2, \dots, K\}$

is considered throughout. The clique types associated with this configuration are shown in figure 2.5. A further assumption is that the field is homogeneous, i.e. the clique potentials depend on the clique types only, irrespective of their position in the lattice, S . Parameters may therefore be assigned to each clique type, except for the single pixel clique, as follows:

$$[**, \beta_1], \left[\begin{smallmatrix} * \\ * \end{smallmatrix}, \beta_2 \right], \left[\begin{smallmatrix} * \\ * \end{smallmatrix}, \beta_3 \right], \left[\begin{smallmatrix} * \\ * \end{smallmatrix}, \beta_4 \right], \left[\begin{smallmatrix} ** \\ * \end{smallmatrix}, \gamma_1 \right], \left[\begin{smallmatrix} ** \\ * \end{smallmatrix}, \gamma_2 \right], \left[\begin{smallmatrix} * \\ ** \end{smallmatrix}, \gamma_3 \right], \left[\begin{smallmatrix} * \\ ** \end{smallmatrix}, \gamma_4 \right], \left[\begin{smallmatrix} ** \\ ** \end{smallmatrix}, \xi_1 \right]$$

The associated clique potentials are defined as

$$V_c(x) = \begin{cases} -\zeta & \text{if all } x_s, s \in c, \text{ are equal} \\ \zeta & \text{otherwise} \end{cases} \quad (\text{B.1})$$

where ζ is the parameter specified for the type of clique c . For single pixel cliques, the associated clique potential is defined as

$$V_c(x) = \alpha_i \quad \text{if } x_s = i, s \in c, i = 1, \dots, K \quad (\text{B.2})$$

The α_i parameters control the percentage of pixels in each of the K regions types, while the other parameters control the size and direction of clustering.

We limit the extent of this model by setting all $\alpha_i = 0$. In other words we assume that there is no prior knowledge that the frequency of one class is different to that of another. Instead, we rely exclusively on Y , the low level process, described in section B.1.2. In addition, like other researchers [DC86, DE87, LD89, WD92], we only consider the pair clique parameters, β_k . All the other clique parameters are set to zero. The clique potentials may therefore be expressed as

$$V_c(x) = \begin{cases} -\beta_k & \text{if } x_s = x_r, s, r \in c \\ \beta_k & \text{otherwise} \end{cases} \quad (\text{B.3})$$

where $k \in \{1, 2, 3, 4\}$. The value of $P(X = x)$ may thus be defined by equations 2.23, 2.24 and 2.25 as described in section 2.4.2.

B.1.2 Univariate Gaussian Model

We adopt the Gaussian distribution to model the high level process, Y , and assume that it is Gaussian noise which corrupts the ideal label data to produce the resultant

MR image. This is clearly an over-simplification of the actual circumstances. However, due to its simplicity and ease of computation we adopt this model and assume that by considering the ^{level}low process, X , simultaneously, reasonable results will be obtained. It was explained in chapter 5, section 5.2, that optimal results for PR approaches to MR image classification are achieved by considering the distribution of grey scale values only. We therefore choose a univariate Gaussian distribution to model Y as follows:

$$P(Y_s = y_s \mid X_s = i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp \left[-\frac{(y_s - \mu_i)^2}{2\sigma_i^2} \right] \quad (\text{B.4})$$

where Y_s is the field of observed pixel values, modelled by the Gaussian distribution whose parameters μ_i and σ_i^2 , the means and variances of classes $i \in \{1, 2, \dots, K\}$, are computed according to their ML estimates

$$\mu_i = \frac{1}{N_i} \sum_{s \in S: X_s = i} y_s \quad \sigma_i^2 = \frac{1}{N_i} \sum_{s \in S: X_s = i} (y_s - \mu_i)^2 \quad (\text{B.5})$$

B.2 Numerical Solution of the MAP Criterion

This section describes the optimisation techniques adopted to solve the MAP criterion, as defined in equation 2.28, when adopting the hierarchical image model as described in section B.1. To solve equation 2.28 directly is extremely difficult. Won and Derin [WD92] have therefore adopted an alternative maximisation procedure which yields a partial optimal solution (POS) to equation 2.28. This is expressed as

$$x^* = \arg \max_X \left\{ P(Y = y \mid X, \hat{\theta}_Y(X, Y = y)) P(X = x \mid \theta_X^*) \right\} \quad (\text{B.6})$$

$$\theta_X^* = \arg \max_{\theta_X} P(X = x^* \mid \theta_X) \quad (\text{B.7})$$

The POS is obtained by implementing the following adaptive segmentation algorithm (ASA).

Algorithm B.1 (Adaptive Segmentation Algorithm) The following steps are performed:

1. (a) Choose an initial labelling for $x^{(0)}$: This is achieved by performing the FFCM (as described in section 5.2) and assigning to each pixel the label which corresponds to its class of highest membership (see equation 6.1).
 (b) Choose initial parameter set $\theta_X^{(0)}$: To avoid convergence to unwanted local minima, we set the parameters β_k according to a heuristic method, proposed by Matalas [Mat96], as follows:

$$\beta_k^{(0)} = \ln [(K - 1)p_k / (1 - p_k)] \quad (\text{B.8})$$

where $k \in \{1, 2, 3, 4\}$, K is the number of classes (or labels) in the image and $p_k = n_k/N$, is the probability of clique type k . n_k is the number of occurrences of clique type k and N is the number of pixels in the image.

- (c) Compute parameters $\hat{\theta}_Y$ from $x^{(0)}$ labelling according to equation B.5.
 (d) Set $n = 0$.
2. Find $\theta_X^{(n+1)} = \arg \max_{\theta_X} P(X = x^{(n)} | \theta_X^{(n)})$ using SA, as described in section B.2.1. If $P(X^{(n)} | \theta_X^{(n+1)}) = P(X^{(n)} | \theta_X^{(n)})$, stop and set $(x^*, \theta_X^*) = (x^{(n)}, \theta_X^{(n)})$.
3. Find $x^{(n+1)} = \arg \max_X \left\{ P(Y | X^{(n)}, \hat{\theta}_Y) P(X | \theta_X^{(n+1)}) \right\}$ using ICM, as described in section B.2.2. If $P(Y | X^{(n+1)}, \hat{\theta}_Y) P(X^{(n+1)} | \theta_X^{(n+1)}) = P(Y | X^n, \hat{\theta}_Y) P(X^n | \theta_X^{(n+1)})$ stop and set $(x^*, \theta_X^*) = (x^{(n)}, \theta_X^{(n)})$.
4. Let $n = n + 1$ and return to step 2.

Although the ASA yields only a POS to the MAP criterion, which is not necessarily a global or even a local maximum, by adopting initial estimates $(x^{(0)}, \theta_X^{(0)})$ which are close to a local maximum, the POS is guaranteed to be a local maximum [WD92]. It should be noted that the stopping conditions of steps 2 and 3 above are difficult, in practice, to achieve. Instead, it is sufficient to simply stop after a few iterations of the algorithm.

B.2.1 Parameter Estimation

In order to estimate the value of the parameter set θ_X , within the framework of the ASA, it is necessary to solve equation B.7. Although this may be found theoretically by ML estimation, in practice this is very difficult to achieve. However, a near approximation is achieved by maximum pseudo-likelihood (MPL) estimation [Bes86] as follows:

$$\theta_X^* = \arg \max_{\theta_X} \left\{ \prod_{s \in S} P(X_s = x_s \mid X_{\eta_s}, \theta_X) \right\} \quad (\text{B.9})$$

In order to solve equation B.9, we follow the approach proposed by Lakshmanan and Derin [LD89] and apply simulated annealing (SA) via the metropolis algorithm.

Algorithm B.2 (Metropolis Algorithm) The following steps are performed:

1. (a) Set the initial temperature $T_\theta^{(0)} = 4$ (following the experimental values of Geman and Geman [GG84]).
 (b) Set $l = 0$.
2. If $l > 0$, reduce the temperature according to the exponential cooling schedule¹ $T_\theta^{(l)} = cT_\theta^{(l-1)}$. Following [KGV83], we set $c = 0.95$.
3. (a) Choose a clique type $\beta_k^{(l)}$, where k is chosen at random, $k \in \{1, 2, 3, 4\}$.
 (b) Assign an updated candidate value for β_k such that $\beta_k' = \beta_k^{(l-1)} + \gamma$, where γ is chosen randomly from the small interval $[-\gamma_{max}, +\gamma_{max}]$. (N.B: For $l = 0$, the initial parameter value is indexed as $\beta^{(0)}$.)

¹Although it has been shown in [GG84] that convergence can only be guaranteed by the cooling schedule $T_\theta^{(l)} = C/\ln(l+1)$, this converges extremely slowly. We therefore approximate it with an exponential schedule.

4. (a) The following ratio is computed:

$$\begin{aligned} \rho &= \left[\frac{\prod_{s \in S} P(X_s = x_s \mid X_{\eta_s}, \theta'_X)}{\prod_{s \in S} P(X_s = x_s \mid X_{\eta_s}, \theta_X^{(l-1)})} \right]^{1/T_\theta^{(l)}} \\ &= \left[\frac{\prod_{s \in S} \left\{ \left(\sum_{X_s \in \Lambda} \exp \left[- \sum_{c \in \eta_s^+} V'_c(X_s) \right] \right)^{-1} \exp \left[- \sum_{c \in \eta_s^+} V'_c(x_s) \right] \right\}}{\prod_{s \in S} \left\{ \left(\sum_{X_s \in \Lambda} \exp \left[- \sum_{c \in \eta_s^+} V_c^{(l-1)}(X_s) \right] \right)^{-1} \exp \left[- \sum_{c \in \eta_s^+} V_c^{(l-1)}(x_s) \right] \right\}} \right]^{1/T_\theta^{(l)}} \end{aligned} \quad (\text{B.10})$$

where $\eta_s^+ = \eta_s \cup \{s\}$, $V'_c(x_s)$ is the clique potential associated with x_s whose values correspond to those of the candidate parameter set $\{\beta'_m\}$ and the values of $V_c^{(l-1)}(x_s)$ correspond to $\{\beta_m^{(l-1)}\}$, where $m \in \{1, 2, 3, 4\}$.

(b) A new value is assigned to $\beta_k^{(l)}$ as follows:

If $\rho \geq 1$ then set $\beta_k^{(l)} = \beta'_k$

If $\rho < 1$: Randomly choose a value for ξ in the interval $[0, 1]$. If $\xi \leq \rho$, set $\beta_k^{(l)} = \beta'_k$. Otherwise, $\beta_k^{(l)} = \beta_k^{(l-1)}$.

5. Let $l = l + 1$. Go back to step 2 until a suitable stopping criterion is satisfied.

We simply perform a fixed number of iterations.

B.2.2 Iterated Conditional Modes

Besag [Bes86] proposed the iterated conditional modes (ICM) algorithm as an alternative to SA. We adopt this deterministic approach for computing the updated label values, x , in the ASA. In order to solve

$$x = \arg \max_X \left\{ P \left(Y \mid X^{(n)}, \hat{\theta}_Y \right) P(X \mid \theta_X^{(n+1)}) \right\} \quad (\text{B.11})$$

the ICM algorithm maximises the conditional distribution at each site in the image lattice. This is accomplished by considering the following log likelihood for each

site:

$$X_s = \arg \max_{i \in \Lambda} \left\{ \ln P(Y_s = y_s \mid X_s = i, \hat{\theta}_Y) + \ln P(X_s = i \mid X_{\eta_s}, \theta_X) \right\} \quad (\text{B.12})$$

Let us expand the two components of the likelihood function individually:

$$\begin{aligned} \ln P(Y_s = y_s \mid X_s = i, \hat{\theta}_Y) &= \ln \left(\frac{1}{\sqrt{2\pi\sigma_i^2}} \exp \left[-\frac{(y_s - \mu_i)^2}{2\sigma_i^2} \right] \right) \\ &= -\frac{\ln(2\pi\sigma_i^2)}{2} - \frac{(y_s - \mu_i)^2}{2\sigma_i^2} \end{aligned} \quad (\text{B.13})$$

$$\begin{aligned} \ln P(X_s = i \mid X_{\eta_s}, \theta_X) &= \ln \left(\frac{1}{Z_s(\theta_X)} \cdot \exp \left[-\sum_{c \in \eta^+} V_c(X_s = i) \right] \right) \\ &= -\ln Z_s(\theta_X) - \sum_{c \in \eta^+} V_c(X_s = i) \end{aligned} \quad (\text{B.14})$$

where $\eta_s^+ = \eta_s \cup \{s\}$. It should be noted that the term $\ln Z_s(\theta_X)$ in equation B.14 is independent of the value i . This may therefore be disregarded when maximising the RHS of equation B.12 with respect to i . Equation B.12 may now be fully expanded to obtain

$$X_s = \arg \min_{i \in \Lambda} \left\{ \frac{\ln(2\pi\sigma_i^2)}{2} + \frac{(y_s - \mu_i)^2}{2\sigma_i^2} + \sum_{c \in \eta_s^+} V_c(X_s = i) \right\} \quad (\text{B.15})$$

Algorithm B.3 (Iterated Conditional Modes) The following steps are performed:

1. Set $l = 0$.
2. Visit each site in the image according to a random schedule². At each site, the pixel label is updated as follows:

$$X_s^{(l+1)} = \arg \min_{i \in \Lambda} \left\{ \lambda \left(\frac{\ln(2\pi\sigma_i^{2,(l)})}{2} + \frac{(y_s - \mu_i^{(l)})^2}{2\sigma_i^{2,(l)}} \right) + \sum_{c \in \eta_s^+} V_c(X_s = i) \right\} \quad (\text{B.16})$$

²A raster scan visitation schedule often has the effect of biasing the classification in the raster ordering. This is because the prior distribution of X influences adjacent sites to be classified similarly. Hence, a random schedule is adopted.

It should be noted that we have added a λ term to equation B.16. This allows us to control the relative effects of the high level and low level processes on the resultant segmentation. Setting $\lambda = 1$ gives equal weighting to both processes. As $\lambda \rightarrow \infty$, the result depends solely on the global properties of the grey scale distribution. This is equivalent to solving the ML estimates of the Gaussian distribution. Conversely, as $\lambda \rightarrow 0$, the segmentation is influenced by spatial interactions alone ignoring any global properties of the image data.

3. After each site has been updated according to step 2, if the value of $X_s^{(l+1)}$ is different to $X_s^{(l)}$, the parameters $\sigma_i^{2,(l)}$ and $\mu_i^{(l)}$ must be updated. These may be computed from equation B.5. However, if there are many sites in the image, this may involve a very large amount of computation. We therefore simplify the computation by evaluating the new parameters according to their previous values [WD92] as follows:

Let $X_s^{(l)} = i_1$ and $X_s^{(l+1)} = i_2$; $N_{i_1}^{(l+1)}$ = number of pixels assigned label i_1 after pixel update and $N_{i_1}^{(l)}$ = number of pixels assigned label i_1 before pixel update; $N_{i_2}^{(l+1)}$ = number of pixels assigned label i_2 after pixel update and $N_{i_2}^{(l)}$ = number of pixels assigned label i_2 before pixel update. Thus,

$$N_{i_1}^{(l+1)} = N_{i_1}^{(l)} - 1; \quad N_{i_2}^{(l+1)} = N_{i_2}^{(l)} + 1 \quad (\text{B.17})$$

The updated parameters may now be computed as:

$$\mu_{i_1}^{(l+1)} = \frac{1}{N_{i_1}^{(l+1)}} \left(N_{i_1}^{(l)} \mu_{i_1}^{(l)} - y_s \right) \quad (\text{B.18})$$

$$\mu_{i_2}^{(l+1)} = \frac{1}{N_{i_2}^{(l+1)}} \left(N_{i_2}^{(l)} \mu_{i_2}^{(l)} + y_s \right) \quad (\text{B.19})$$

$$\sigma_{i_1}^{2,(l+1)} = \frac{1}{N_{i_1}^{(l+1)}} \left[N_{i_1}^{(l)} \sigma_{i_1}^{2,(l)} - \frac{N_{i_1}^{(l)}}{N_{i_1}^{(l+1)}} \left(y_s - \mu_{i_1}^{(l)} \right)^2 \right] \quad (\text{B.20})$$

$$\sigma_{i_2}^{2,(l+1)} = \frac{1}{N_{i_2}^{(l+1)}} \left[N_{i_2}^{(l)} \sigma_{i_2}^{2,(l)} + \frac{N_{i_2}^{(l)}}{N_{i_2}^{(l+1)}} \left(y_s - \mu_{i_2}^{(l)} \right)^2 \right] \quad (\text{B.21})$$

4. Let $l = l + 1$ and go back to step 2. If $X^{(l+1)} = X^{(l)}$ stop. Alternatively, stop after a fixed number of iterations.

Appendix C

Seed Fill Algorithm

This seed fill algorithm fills 4-connected 2D regions which have clearly defined boundaries. The algorithm is developed using a stack which is simply an array into which values may be sequentially placed or from which they may be sequentially removed. As new values are added, or *pushed*, onto the stack, all previously stored values are pushed down one level. As values are removed, or *popped*, from the stack, previously stored values float or pop up one level. Such a stack is referred to as a First In Last Out (FILO) or push-down stack.

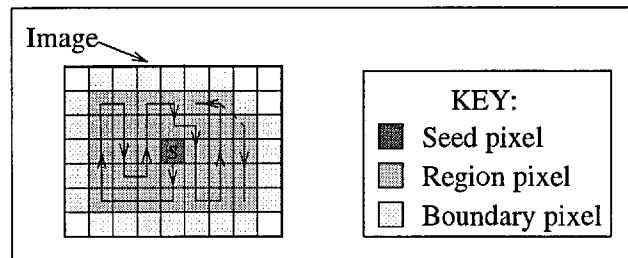


Figure C.1: An example of the seed fill algorithm as it progresses through a 4-connected region.

Algorithm C.1 (Seed Fill) The following terms are first defined:

$\text{Seed}(x, y)$ is the seed pixel; **Push** is a function for placing a pixel on the stack;

Pop is a function for removing a pixel from the stack.

The algorithm may now be described in pseudo-code as follows:

```
Pixel( $x, y$ ) = Seed( $x, y$ )
Initialise stack:
Push Pixel( $x, y$ )
While (stack not empty) {
    Get a pixel from the stack:
    Pop Pixel( $x, y$ )
    If Pixel ( $x, y$ )  $\neq$  New value then
        Pixel( $x, y$ ) = New value
    Else {
        Examine the surrounding pixels to see if they should be placed
        onto the stack:
        If (Pixel( $x + 1, y$ )  $\neq$  New value
        and Pixel( $x + 1, y$ )  $\neq$  Boundary value) then
            Push Pixel( $x + 1, y$ )
        If (Pixel( $x, y + 1$ )  $\neq$  New value
        and Pixel( $x, y + 1$ )  $\neq$  Boundary value) then
            Push Pixel( $x, y + 1$ )
        If (Pixel( $x - 1, y$ )  $\neq$  New value
        and Pixel( $x - 1, y$ )  $\neq$  Boundary value) then
            Push Pixel( $x - 1, y$ )
        If (Pixel( $x, y - 1$ )  $\neq$  New value
        and Pixel( $x, y - 1$ )  $\neq$  Boundary value) then
            Push Pixel( $x, y - 1$ )
    }
}
```

Figure C.1 shows an example of the seed fill algorithm progressing through a 4-connected 2D region. The algorithm may be easily extended to fill 8-connected 2D regions as well as either 6-connected or 24-connected 3D regions.

Appendix D

Least Squares Fitting of two 3D Point Sets

The values of R and T , which minimise equation 7.2, may be found as follows:

Let,

$$q_i = p_i - p \quad (\text{D.1})$$

$$q'_i = p'_i - p' \quad (\text{D.2})$$

where p and p' are the centroids of $\{p_i\}$ and $\{p'_i\}$ respectively, i.e.

$$p = \frac{1}{N} \sum_{i=1}^N p_i \quad (\text{D.3})$$

$$p' = \frac{1}{N} \sum_{i=1}^N p'_i \quad (\text{D.4})$$

It has been shown [AHB87, HBM86] that equation 7.2 may now be rewritten as

$$\sum^2 = \sum_{i=1}^N \|q'_i - Rq_i\|^2 \quad (\text{D.5})$$

The original least-squares problem may now be solved as follows:

(a) Find $\hat{R}$ to minimise $\sum^2$ in Equation D.5

Step 1: Calculate the values of the 3×3 matrix H where,

$$H \stackrel{def}{=} \sum_{i=1}^N q_i q_i^t \quad (\text{D.6})$$

APPENDIX D. LEAST SQUARES FITTING OF TWO 3D POINT SETS

where the superscript t denotes matrix transposition and q_i and q'_i are as defined in equations D.1 to D.4.

Step 2: Find the Singular Value Decomposition [GL89] of H ,

$$H = U\Lambda V^t \quad (\text{D.7})$$

Step 3: Calculate

$$X = VU^t \quad (\text{D.8})$$

Step 4: Calculate, $\det(X)$, the determinant of X .

If $\det(X)=+1$, then $\hat{R} = X$.

If $\det(X)=-1$, then $\hat{R} = V'U^t$ where, $V' = [v_1, v_2, -v_3]$.

(b) Compute the Translation matrix

The Translation matrix T is found by,

$$\hat{T} = p' - \hat{R}p \quad (\text{D.9})$$

Bibliography

- [AB94] R. Adams and L. Bischof. Seeded Region Growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, June 1994.
- [AHB87] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-Squares Fitting of Two 3D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, September 1987.
- [AW80] M. Aitkin and G. T. Wilson. Mixture Models, Outliers, and the EM Algorithm. *Technometrics*, 22(3):325–331, August 1980.
- [Bat96] J. Batista. *Techniques for Computer-based Anatomical Segmentation of the brain using MRI*. PhD thesis, Imperial College, London, March 1996.
- [BBH⁺92] A. M. Bensaid, J. C. Bezdek, L. O. Hall, et al. A Partially Supervised Fuzzy *c*-Means Algorithm for Segmentation of MR Images. In *Proceedings of SPIE: Science of Artificial Neural Networks*, volume 1710, pages 522–528, 1992.
- [Bes74] J. Besag. Spatial Interaction and the Statistical Analysis of Lattice Systems. *Journal of the Royal Statistical Society, Series B*, 36(2):192–236, 1974.
- [Bes86] J. Besag. On the Statistical Analysis of Dirty Pictures. *Journal of the Royal Statistical Society, Series B*, 48(3):259–302, 1986.
- [Bez81] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.
- [BG89] P. Burger and D. Gillies. *Interactive Computer Graphics*. Addison Wesley, 1989.

- [BHBC96] A. M. Bensaid, L. O. Hall, J. C. Bezdek, and L. P. Clarke. Partially Supervised Clustering for Image Segmentation. *Pattern Recognition*, 29(5):859–871, 1996.
- [BHC93] J. C. Bezdek, L. O. Hall, and L. P. Clarke. Review of MR image segmentation techniques using pattern recognition. *Medical Physics*, 20(4):1033–1048, July/August 1993.
- [BHP46] F. Bloch, W. W. Hansen, and M. Packard. The Nuclear Induction Experiment. *Phys. Rev.*, 69:474–485, 1946.
- [BK89] R. Bajcsy and S. Kovacic. Multiresolution Elastic Matching. *Computer Vision, Graphics, and Image Processing*, 46:1–21, 1989.
- [Bli82] J. F. Blinn. Light Reflection Functions for Simulation of Clouds and Dusty Surfaces. *Computer Graphics*, 16(3):21–29, July 1982.
- [Blo93] I. Bloch. Fuzzy connectivity and mathematical morphology. *Pattern Recognition Letters*, 14:483–488, 1993.
- [BP92] J. C. Bezdek and S. K. Pal, editors. *Fuzzy Models For Pattern Recognition: Methods That Search for Structures in Data*. IEEE Press, 1992.
- [BTU84] L. J. Brewster, S. S. Trivedi, and J. K. Udupa. Interactive Surgical Planning. *IEEE Computer Graphics and Applications*, 4:31–40, 1984.
- [CA79] G. B. Coleman and H. C. Andrews. Image Segmentation by Clustering. *Proceedings of the IEEE*, 67(5):773–785, May 1979.
- [Cat80] E. Catmull. Computer Display of Curved Surfaces. In H. Freedman, editor, *Tutorial and Selected Readings in Interactive Computer Graphics*, pages 309–315. IEEE, New York, 1980.
- [CC85] R. Chellappa and S. Chatterjee. Classification of Textures Using Gaussian Markov Random Fields. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-33(4):959–963, August 1985.
- [CDB86] R. L. Cannon, J. V. Dave, and J. C. Bezdek. Efficient implementation of the Fuzzy c-Means Clustering Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(2):248–255, March 1986.
- [CHRU85] L. Chen, G. T. Herman, R. A. Reynolds, and J. K. Udupa. Surface Shading in the Cuberille Environment. *IEEE Computer Graphics and Applications*, pages 33–43, December 1985.

- [CL91] S. Chen and W. Lin. Split-and-Merge Image Segmentation Based on Localized Feature Analysis and Statistical Tests. *CVGIP: Graphical Models and Image Processing*, 53(5):457–475, 1991.
- [CL94] Y-L. Chang and X. Li. Adaptive Region Growing. *IEEE Transactions on Image Processing*, 3(6):868–872, November 1994.
- [CLL⁺88] H. E. Cline, W. E. Lorensen, S. Ludke, C. R. Crawford, and B. C. Teeter. Two Algorithms for Three Dimensional Reconstruction of Tomograms. *Medical Physics*, 15(3), May/June 1988.
- [Coh91] L. D. Cohen. On Active Contour Models and Balloons. *CVGIP: Image Understanding*, 53(2):211–218, March 1991.
- [CP80] P. C. Chen and T. Pavlidis. Image Segmentation as an Estimation Problem. *Computer Graphics and Image Processing*, 12:153–172, 1980.
- [CT82] R. L. Cook and K. E. Torrance. A Reflection Model for Computer Graphics. *ACM Transactions on Graphics*, 1(1):7–24, 1982.
- [CVC⁺95] L. P. Clarke, R. P. Velthuizen, M. A. Camacho, et al. MRI Segmentation: Methods and Applications. *Magnetic Resonance Imaging*, 13(3):343–368, 1995.
- [Dam71] R. Damadian. Tumor Detection by Nuclear Magnetic Resonance. *Science*, 171:1151, 1971.
- [Dav80] R. Davis. Meta Rules: Reasoning about control. *Artificial Intelligence*, 5:179–222, 1980.
- [DC86] H. Derin and W. S. Cole. Segmentation of Textured Images Using Gibbs Random Fields. *Computer Vision, Graphics, and Image Processing*, 35:72–98, 1986.
- [DCB93] R. Deklerck, J. Cornelis, and M. Bister. Segmentation of medical images. *Image and Vision Computing*, 11(8):486–503, October 1993.
- [DCH88] R. A. Drebin, L. Carpenter, and P. Hanrahan. Volume Rendering. *Computer Graphics*, 22(4):65–74, August 1988.
- [DE87] H. Derin and H. Elliot. Modelling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(1):39–55, January 1987.

- [DGM77] R. Damadian, M. Goldsmith, and L. Minkoff. Fonar Image of the Live Human Body. *Physiol. Chem. Phys. Med. NMR*, 9:97, 1977.
- [DH73] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.
- [DKS90] J. M. H. Du Buf, M. Kardan, and M. Spann. Texture Feature Performance for Image Segmentation. *Pattern Recognition*, 23(3/4):291–309, 1990.
- [DLR77] A. M. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, series B*, 39(1):1–38, 1977.
- [DSD93] C. L. Dumoulin, S. P. Souza, and R. D. Darrow. Real-Time Position Monitoring of Invasive Devices Using Magnetic Resonance. *Magnetic Resonance in Medicine*, 29:411–415, 1993.
- [DW87] H. Derin and C. S. Won. A parallel image segmentation algorithm using relaxation with varying neighbourhoods and its mapping to array processors. *Computer Vision, Graphics, and Image Processing*, 40:54–78, 1987.
- [EKWA90] R. R. Edelman, J. Kleefield, K. U. Wentz, and D. J. Atkinson. Basic Principles of Magnetic Resonance Imaging. In R. R. Edelman and J. R. Hesselink, editors, *Clinical MRI*, pages 3–38. W. B. Saunders Company, Philadelphia, 1990.
- [Eva93] R. Evans. Surgical Vision. *I.P Magazine*, 5(3), 1993.
- [FC77] W. Frei and C. C. Chen. Fast Boundary Detection: A Generalization and a New Algorithm. *IEEE Transactions on Computers*, C-26:556–566, 1977.
- [FCT⁺97] E. Fisher, R. M. Cothren, J. A. Tkach, et al. Knowledge-based 3D segmentation of the brain in MR images for quantitative multiple sclerosis lesion tracking. In *Proceedings of SPIE: Medical Imaging*, volume 3034(1), pages 19–25, Newport Beach, CA, February 1997.
- [FH87] M. A. Foster and J. M. S. Hutchison. *Practical NMR Imaging*. IRL Press, 1987.

- [FKU77] H. Fuchs, M. Kedem, and S. P. Uselton. Optimal Surface Reconstruction for Planar Contours. *Communications of the ACM*, 20(10):693–702, October 1977.
- [Fos84] M. A. Foster. *Magnetic Resonance in Medicine and Biology*. Pergamon Press, 1984.
- [FU97] A. X. Falcão and J. K. Udupa. Segmentation of 3D objects using Live Wire. In *Proceedings of SPIE: Medical Imaging*, volume 3034(1), pages 228–235, Newport Beach, CA, February 1997.
- [Fuk90] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, London, second edition, 1990.
- [FUSH96] A. X. Falcão, J. K. Udupa, S. Samarasekera, and B. E. Hirsch. User-steered image boundary segmentation. In *Proceedings of SPIE: Medical Imaging*, volume 2710, pages 278–288, Newport Beach, CA, February 1996.
- [FV82] J. D. Foley and A. Van Dam. *Fundamentals of Interactive Computer Graphics*. Addison Wesley, Reading, MA, 1982.
- [Gal75] M. M. Galloway. Texture Analysis Using Gray Level Run Lengths. *Computer Graphics and Image Processing*, 4:172–179, 1975.
- [GELM93] R. L. Galloway, C. A. Edwards II, J. T. Lewis, and R. J. Maciunas. Image display and surgical visualization in interactive image-guided neurosurgery. *Optical Engineering*, 32(8):1955–1961, August 1993.
- [GG84] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Restoration, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, November 1984.
- [GG89] I. Gath and A. B. Geva. Unsupervised Optimal Fuzzy Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):773–781, July 1989.
- [GL89] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Addison-Wesley, second edition, 1989.
- [GME92] R. L. Galloway, R. J. Maciunas, and C. A. Edwards II. Interactive Image-Guided Neurosurgery. *IEEE Transactions on Biomedical Engineering*, 39(12):1226–1231, December 1992.

- [GRB93] J. C. Gee, M. Revich, and R. Bajcsy. Elastically Deforming 3D Atlases to Match Anatomical Brain Images. *Journal of Computer Assisted Tomography*, 17(2):225–246, 1993.
- [GW92] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, 1992.
- [Har84] R. M. Haralick. Digital step edges from zero crossings of second directional derivatives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6:58–68, 1984.
- [HB97] K. P. Hinshaw and J. F. Brinkley. Shape-Based Interactive Three-Dimensional Medical Image Segmentation. In *Proceedings of SPIE: Medical Imaging*, volume 3034(1), pages 236–242, Newport Beach, CA, February 1997.
- [HBC⁺92] L. O. Hall, A. M. Bensaid, L. P. Clarke, et al. A Comparison of Neural Network and Fuzzy Clustering Techniques in Segmenting Magnetic Resonance Images of the Brain. *IEEE Transactions on Neural Networks*, 3(5):672–682, September 1992.
- [HBM86] T. S. Huang, S. D. Blostein, and E. A. Margerum. Least-squares estimation of motion parameters from 3-D point correspondences. In *Computer Vision and Pattern Recognition*, Miami Beach, FL, June 24-26 1986. IEEE.
- [HH92] K. H. Höhne and W. A. Hanson. Interactive 3D Segmentation of MRI and CT Volumes using Morphological Operations. *Journal of Computer Assisted Tomography*, 16(2):285–294, March/April 1992.
- [HL79] G. T. Herman and H. K. Liu. Three-Dimensional Display of Human Organs from Computed Tomography. *Computer Graphics and Image Processing*, 9:1–21, 1979.
- [HM86] A. Huertas and G. Medioni. Detection of Intensity Changes with Sub-pixel Accuracy Using Laplacian-Gaussian Masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8:651–664, 1986.
- [HP76] S. L. Horowitz and T. Pavlidis. Picture segmentation by a tree traversal algorithm. *Journal of the Association for Computing Machinery*, 23:368–388, April 1976.

- [HR94] G. A. Horstmann and H. F. Reinhardt. Micro-Stereometry: A Frameless Computerized Navigating System for open Microsurgery. *Computerized Medical Imaging and Graphics*, 18(4):229–233, July-August 1994.
- [HS85] R. M. Haralick and L. G. Shapiro. Image Segmentation Techniques. *Computer Vision, Graphics, and Image Processing*, 29:100–132, 1985.
- [HS92] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*, volume 1. Addison-Wesley, 1992.
- [HSD73] R. M. Haralick, K. Shanmugan, and I. Dinstein. Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, November 1973.
- [HSZ87] R. M. Haralick, S. R. Sternberg, and X. Zhuang. Image Analysis Using Mathematical Morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(4):532–550, July 1987.
- [HTU88] G. T. Herman, S. S. Trivedi, and J. K. Udupa. Manipulation of 3D imagery. In V. L. Newhouse, editor, *Progress in Medical Imaging*, pages 123–157. Springer-Verlag, New York, 1988.
- [Kan82] A. Kandel. *Fuzzy Techniques in Pattern Recognition*. John Wiley and Sons, New York, 1982.
- [KD89] R. W. Klein and R. C. Dubes. Experiments in Projection and Clustering by Simulated Annealing. *Science*, 22(2):213–220, 1989.
- [Kep75] E. Keppel. Approximation of Complex Surfaces by Triangulation of Contour Lines. *IBM Journal of Research and Development*, 19:2–11, 1975.
- [KGV83] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, May 1983.
- [Kir71] R. Kirsch. Computer Determination of the Constituent Structure of Biologic Image. *Computer Biomedical Research*, 4:315–328, 1971.
- [KWT87] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1(4):321–331, 1987.
- [Lau73] P. C. Lauterbur. Image Formation by Reduced Local Interactions: Examples Employing Nuclear Magnetic Resonance. *Nature*, 242:190, 1973.

- [LC87] W. E. Lorensen and H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics*, 21(4):163–169, July 1987.
- [LD89] S. Lakshmanan and H. Derin. Simultaneous Parameter Estimation and Segmentation of Gibbs Random Fields Using Simulated Annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8):799–813, August 1989.
- [Lev88] M. Levoy. Display of Surfaces from Volume Data. *IEEE Computer Graphics and Applications*, pages 29–37, May 1988.
- [Lev90a] M. Levoy. A Hybrid Ray Tracer for Rendering Polygon and Volume Data. *IEEE Computer Graphics and Applications*, pages 33–40, March 1990.
- [Lev90b] M. Levoy. Efficient Ray Tracing of Volume Data. *ACM Transactions on Graphics*, 9(2):245–261, April 1990.
- [Lev90c] M. Levoy. Volume Rendering by Adaptive Refinement. *The Visual Computer*, 6:2–7, 1990.
- [LGH93] C. Li, D. B. Goldgof, and L. O. Hall. Knowledge-Based Classification and Tissue Labeling of MR Images of Human Brain. *IEEE Transactions on Medical Imaging*, 12(4):740–750, December 1993.
- [Li95] S. Z. Li. *Markov Random Field Modelling in Computer Vision*. Springer-Verlag, 1995.
- [LL94] P. Lacroute and M. Levoy. Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation. *Computer Graphics*, 28(4):451–458, July 1994.
- [LN85] M. D. Levine and A. M. Nazif. Rule-Based Image Segmentation: A Dynamic Control Strategy Approach. *Computer Vision, Graphics, and Image Processing*, 32:104–126, 1985.
- [LT95] J. Le Moigne and J. C. Tilton. Refining Image Segmentation by Integration of Edge and Region Data. *IEEE Transactions on Geoscience and Remote Sensing*, 33(3):605–615, May 1995.
- [LTG⁺92] S. Lavalée, J. Troccaz, L. Gaborit, et al. Image guided operating robot: a clinical application in stereotactic neurosurgery. In *International*

Conference in Robotics and Automation, pages 618–624, Nice, France, May 1992. IEEE.

- [LW93] E-H Liang and E. K. Wong. Hierarchical Algorithms for Morphological Image Processing. *Pattern Recognition*, 26(4):511–529, 1993.
- [Mat96] I. Matalas. *Segmentation Techniques Suitable for Medical Images*. PhD thesis, Imperial College, London, March 1996.
- [MBB⁺88] R. Mohan, G. Barest, L. J. Brewster, et al. A Comprehensive Three-Dimensional Radiation Treatment Planning System. *International Journal of Radiation Oncology, Biology, and Physics*, 15(2):481–495, 1988.
- [Mea82] D. J. Meagher. Efficient Synthetic Image Generation of Arbitrary 3-D objects. In *Proceedings of the IEEE Computer Society Conference on Pattern Recognition and Image Processing*, pages 473–478, June 1982.
- [MH80] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London, series B*, 207:187–217, 1980.
- [MM77] P. Mansfield and A. A. Maudsley. Planar Spin Imaging by NMR. *Journal of Magnetic Resonance*, 27:101, 1977.
- [MM94] G. K. Matsopoulos and S. Marshall. Use of Morphological Image Processing Techniques for the Measurement of a Fetal Head from Ultrasound Images. *Pattern Recognition*, 27(10):1317–1324, 1994.
- [MSC90] B. S. Manjunath, T. Simchony, and R. Chellappa. Stochastic and Deterministic Networks for Texture Segmentation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(6):1039–1049, June 1990.
- [NB80] R. Nevatia and K. R. Babu. Linear Feature Extraction and Description. *Computer Graphics and Image Processing*, 13:257–269, 1980.
- [NB86] V. S. Nalwa and T. O. Binford. On Detecting Edges. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-12:551–553, 1986.
- [NL84] A. M. Nazif and M. D. Levine. Low Level Image Segmentation: An expert system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(5):555–577, September 1984.
- [OU95] D. Odhner and J. K. Udupa. Shell Manipulation: Interactive Alteration of Multiple-Material Fuzzy Structures. In *Proceedings of SPIE: Medical Imaging*, volume 2431, pages 35–42, San Diego, CA, February 1995.

- [Pao89] Y. H. Pao. *Adaptive Pattern Recognition and Neural Networks*. Addison-Wesley, 1989.
- [PB95] N. R. Pal and J. C. Bezdek. On Cluster Validity for the Fuzzy *c*-Means Model. *IEEE Transactions on Fuzzy Systems*, 3(3):370–379, August 1995.
- [PD84] T. Porter and T. Duff. Compositing Digital Images. *Computer Graphics*, 18(3):253–259, July 1984.
- [Pho75] B-T. Phong. Illumination for computer-generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.
- [PL90] T. Pavlidis and Y. Liow. Integrating Region Growing and Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):225–233, March 1990.
- [PP93] N. R. Pal and S. K. Pal. A Review on Image Segmentation Techniques. *Pattern Recognition*, 26(9):1277–1293, 1993.
- [Pre70] J. Prewitt. Object Enhancement and Extraction. In B. Lipkin and A. Rosenfeld, editors, *Picture Processing and Psychopictorics*, pages 75–149. Academic Press, New York, 1970.
- [PTP46] E. M. Purcell, H. C. Torrey, and R. V. Pound. Resonance Absorption by Nuclear Magnetic Resonance Experiments. *Phys. Rev.*, 69:37–38, 1946.
- [Ray90] S. P. Raya. Low-Level Segmentation of 3-D Magnetic Resonance Brain Images - A Rule-Based System. *IEEE Transactions on Medical Imaging*, 9(3):327–337, Septmeber 1990.
- [RD93] T. R. Reed and J. M. H. Du Buf. A Review of Recent Texture Segmentation and Feature Extraction Techniques. *CVGIP: Image Understanding*, 57(3):359–372, 1993.
- [RHC96] R. A. Robb, D. P. Hanson, and J. J. Camp. Computer-Aided Surgery Planning and Rehearsal at Mayo Clinic. *Computer*, pages 311–319, January 1996.
- [RK82] A. Rosenfeld and A. C. Kak. *Digital Picture Processing*, volume 2. Academic Press, 1982.
- [RNM⁺91] H. Rusinek, N. E. Noz, G. Q. Maguire Jr., et al. Quantitative and Qualitative Comparison of Volumetric and Surface Rendering Techniques. *IEEE Transactions on Nuclear Science*, 38(2):659–662, April 1991.

- [Rob65] L. G. Roberts. Machine Perception of Three Dimensional Solids. In J. T. Tippet et al., editors, *Optical and Electro-optical Information Processing*, pages 159–197. MIT Press, Cambridge, MA, 1965.
- [Rob95] S. J. Roberts. Parametric and Non-parametric Unsupervised Cluster Analysis. Technical report, nsg-7-95, Imperial College, September 1995.
- [Ros79] A. Rosenfeld. Fuzzy Digital Topology. *Information and Control*, 40:76–87, 1979.
- [Ros91] A. Rosenfeld. The fuzzy geometry of image subsets. *Pattern Recognition Letters*, 2:311–317, 1991.
- [RR77] G. S. Robinson and J. J. Reis. A Real-Time Edge Processing Unit. In *Proceedings of the SPIE: Advances in Image Transmission Techniques*, pages 155–164, San Diego, CA, 1977.
- [RSF⁺89] J. Rosenman, G. W. Sherouse, H. Fuchs, et al. Three-Dimensional Display Techniques in Radiation Therapy Treatment Planning. *International Journal of Radiation Oncology, Biology, and Physics*, 16(1):263–269, 1989.
- [RW84] R. A. Redner and H. F. Walker. Mixture Densities, Maximum Likelihood and the EM Algorithm. *SIAM Review*, 26(2):195–239, April 1984.
- [Ser82] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.
- [SHCG96] A. J. Strong, D. J. Hawkes, A. C. F. Colchester, and M. J. Gleeson. The use of Computer Graphics for Planning and Navigation in Surgery. In *Eurographics UK Chapter. 14th Annual Conference*, volume 2, pages 255–260, Imperial College, London, March 1996. Eurographics UK.
- [SK97] N. R. Smith and R. I. Kitney. Fast Fuzzy Segmentation of Magnetic Resonance Images: A Prerequisite for Real-Time Rendering. In *Proceedings of SPIE: Medical Imaging*, volume 3034(2), pages 1124–1135, Newport Beach, CA, February 1997.
- [Sob70] I. E. Sobel. *Camera Models and Machine Perception*. PhD thesis, Electrical Engineering Department, Stanford University, Stanford, CA, 1970.
- [Sta86] S. A. Stansfield. ANGY: A Rule-Based Expert System for Automatic Segmentation of Coronary Vessels From Digital Subtracted Angiograms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(2):188–199, March 1986.

- [Sto94] G. Storvik. A Bayesian Approach to Dynamic Contours through Stochastic Sampling and Simulated Annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):976–986, October 1994.
- [TB77] J. M. Tenenbaum and H. G. Barrow. Experiments in Interpretation-Guided Segmentation. *Artificial Intelligence*, 8:241–276, 1977.
- [The89] C. W. Therrien. *Decision, Estimation and Classification: An Introduction to Pattern Recognition and Related Topics*. Wiley, 1989.
- [TIB88] D. G. Taylor, R. Inamudar, and M. C. Bushell. NMR imaging in theory and practice. *Phys. Med. Biol.*, 53:635–666, 1988.
- [Trå91] H. G. C. Tråvén. A Neural Network Approach to Statistical Pattern Classification by “Semiparametric” Estimation of Probability Density Functions. *IEEE Transactions on Neural Networks*, 2(3):366–377, May 1991.
- [Tri86] S. S. Trivedi. Interactive manipulation of three-dimensional binary scenes. *The Visual Computer*, 2:209–218, 1986.
- [TT96] P. Thompson and A. W. Toga. A Surface-Based Technique for Warping Three Dimensional Images of the Brain. *IEEE Transactions on Medical Imaging*, 15(4):402–417, 1996.
- [Udu91] J. K. Udupa. Computer Aspects of 3D Imaging in Medicine: A Tutorial. In J. K. Udupa and G. T. Herman, editors, *3D Imaging in Medicine*, pages 1–69. CRC Press, 1991.
- [UH89] J. K. Udupa and G. T. Herman. Volume Rendering versus Surface Rendering. *Communications of the ACM*, 32(11):1364–1366, November 1989.
- [UHC91] J. K. Udupa, H-M. Hung, and K-S. Chuang. Surface and Volume Rendering in Three-Dimensional Imaging: A Comparison. *Journal of Digital Imaging*, 4(3):159–168, August 1991.
- [UK88] C. Upson and M. Keeler. VBUFFER: Visible Volume Rendering. *Computer Graphics*, 22(4):59–64, August 1988.
- [UO93] J. K. Udupa and D. Odhner. Shell Rendering. *IEEE Computer Graphics and Applications*, pages 58–67, November 1993.

- [US96] J. K. Udupa and S. Samarasekera. Fuzzy Connectedness and Object Definition: Theory, Algorithms, and Applications in Image Segmentation. *Graphical Models and Image Processing*, 58(3):246–261, May 1996.
- [VDO85] L. Van Gool, P. Dewaele, and A. Oosterlinck. Survey: Texture Analysis Anno 1983. *Computer Vision, Graphics, and Image Processing*, 29:336–357, 1985.
- [Wat93] A. Watt. *3D Computer Graphics*. Addison-Wesley, 2nd edition, 1993.
- [WD92] C. S. Won and H. Derin. Unsupervised Segmentation of Noisy and Textured Images Using Markov Random Fields. *CVGIP: Graphical Models and Image Processing*, 54(4):308–328, July 1992.
- [Wes90] L. Westover. Footprint Evaluation for Volume Rendering. *Computer Graphics*, 24(4), August 1990.
- [WGKJ95] W. M. Wells III, W. E. L. Grimson, R. Kikinis, and F. A. Jolesz. Adaptive Segmentation of MRI Data. In *CVRMED '95: Computer Vision, virtual reality and robotics in medicine*, pages 59–69, Nice, 1995.
- [Wol94] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, 1994.
- [WV91] J. Wilhelms and A. Van Gelder. A Coherent Projection Approach. *Computer Graphics*, 25(4):275–284, July 1991.
- [Zad65] L. A. Zadeh. Fuzzy sets. *Information Control*, 8:338–353, 1965.
- [ZJK94] L. Zamarano, Z. Jiang, and Z. M. Kadi. Computer-Assisted Neurosurgery System: Wayne State University Hardware and Software Configuration. *Computerized Medical Imaging and Graphics*, 18(4):257–271, July-August 1994.
- [Zuc76] S. W. Zucker. Region Growing: Childhood and Adolescence. *Computer Graphics and Image Processings*, 5:382–399, 1976.