

**QUANTITATIVE MEASUREMENT OF INTRACRANIAL DEFORMATION
BY
MAGNETIC RESONANCE IMAGING**

Dongyong Dai

Thesis submitted
for the
Degree of Doctor of Philosophy



**UNIVERSITY
of
GLASGOW**

Department of Neurosurgery

December 1993

ProQuest Number: 11007754

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

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



ProQuest 11007754

Published by ProQuest LLC (2018). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346



Thesis

9894

Copy 1

ACKNOWLEDGMENTS

The project this thesis describes was supported by a Cancer Research Campaign PhD studentship (grant number SP2112/1001).

It is with great pleasure that I wish to express my sincere appreciation to Dr Barrie Condon for suggesting the research problems investigated in this thesis and for his constant encouragement, guidance, support, and patience.

I would like to thank Professor Graham Teasdale for valuable suggestions, advice, and kind support during the course of this work.

I also would like to thank the following people for, amongst other things, their encouragement and support during my stay in the MRI Unit, Institute of Neurological sciences in the Southern General Hospital:

Dr. David Wyper for many useful suggestions and his help throughout the project, Dr. Roy Rampling for providing the clinical assessment reports for all the patients, Dr. Donald Hadley for providing the neuroradiological reports for all the patients and allowing me access to the MRI data, Dr. Jim Patterson for allowing me access to computing facilities in the MRI Unit and Mrs Jean Winning, Mrs Marion Cresswell and Mrs Pat Christie for acquiring the images.

I also wish to express my gratitude for the patience, encouragement, and love of my parents and my wife during the course of the work for this thesis.

ABSTRACT

A new quantitative framework for the objective assessment of the severity of intracranial deformation caused by brain tumours and how they respond to treatment is developed using shape analysis and feature extraction techniques. Conventional criteria for assessing the effects of a brain tumour by medical images are inaccurate and unsatisfactory because of problems defining tumour boundaries. Clinical response is relevant but is influenced by many factors and assessment is very subjective. In order to provide an objective and accurate assessment of the effects of brain tumours, a new and reliable method is introduced which quantifies the distortions and displacements of intracranial structures caused by brain tumours.

The deformations of the hemispheres, the lateral ventricles and the falx cerebri are analysed in terms of size, position and shape in both individual sectional images and three dimensional MRI image data sets. 28 invariant attributes (size, rotation and position invariant), such as compactness, elongation, and central moments, are used to characterise the deformation of the hemispheres.

The shape of the lateral ventricles is specified by Fourier descriptors, the method being used to match ventricular shapes even through they have different sizes and orientations. Using the Fourier descriptors, the shape of the lateral ventricles can be characterised by 6 invariant attributes.

A new technique to describe deformation in the intracranial surfaces is presented. The technique is used specifically to analyse the deformation of the falx cerebri. The deformation of the falx cerebri is described by measuring the Gaussian curvature and the mean curvature. 28 attributes are obtained to quantify changes in the surface of the falx cerebri.

Based on the measurements of the hemispheres, the lateral ventricles and the falx cerebri from 15 volunteers and 28 patients with intracranial deformation and combining all the 62 invariant attributes using data reduction techniques (the Karhunen-Loève transform), three criteria (the decision rules) are established to classify the *normal* and *abnormal* intracranial structures. A scale for assessing the severity of intracranial deformation and a scale for assessing the effects of therapy also are established.

The criteria and scales are applied in a blinded prospective manner to a further group of patients. The results reveal some important relationships between an intracranial deformation and its effects. For example, all studied patients who initially had a larger severity scale ($SSID > 5$), with one exception, died within 8 months no matter how great the decrease of the deformation following treatment. The quantitative assessments are compared with clinical assessments for the same group of patients, and results show that the quantitative approach is more objective and accurate. The quantitative framework provides an objective and accurate index of intracranial deformation and may provide a sensitive and relevant basis for clinical decision making.

Contents

Acknowledgements	ii
Abstract	iii
Chapter 1 Introduction	1
Chapter 2 The Basic Principles of Magnetic Resonance Imaging	10
2.1 The Development of Nuclear Magnetic Resonance	11
2.2 Nuclear Magnetic Resonance	13
2.3 Relaxation Times	15
2.3.1 T_1 Relaxation	15
2.3.2 T_2 Relaxation	17
2.4 Pulse Sequences	19
2.4.1 Spin Echo	19
2.4.2 Inversion Recovery	22
2.5 Image Generation	25
2.5.1 Use of Magnetic Field Gradients	26
2.5.2 Slice Selection	27
2.5.3 Frequency and Phase Encoding	29
2.5.4 Image Reconstruction	33
2.6 MRI Hardware	34
2.6.1 The Magnet	35
2.6.2 Radiofrequency Coils and Gradient Coils	36

2.6.3 Computer System	37
2.7 Safety of MRI	37
2.7.1 Direct Biological Effects	37
2.7.2 Indirect Effects	38
2.8 Other Medical Imaging Modalities	38
2.8.1 X-ray Imaging	39
2.8.2 Ultrasonography	41
2.8.3 Radionuclide Imaging	43
2.8.4 Comparison of MRI with other Modalities	45
2.8.4.1 Advantages of MRI	45
2.8.4.2 Disadvantages of MRI	47
Chapter 3 The Basic Anatomy and Pathology of the Brain	48
3.1 The Brain	48
3.1.1 Meninges	50
3.1.2 The Cerebral Hemispheres	51
3.1.3 The Ventricles	52
3.1.4 The Brainstem	52
3.1.5 The Cerebellum	53
3.2 Intracranial Deformation	53
3.2.1 Intracranial Expanding Lesions	55
3.2.2 The Internal Hernia	56
3.2.3 Brain Swelling	57

3.2.4 The Deformation Caused by Brain Tumour	58
Chapter 4 Image Segmentation	60
4.1 Introduction	60
4.2 Edge Detection Using Canny's Detector	63
4.2.1 Laplace Operators and Zero Crossings	64
4.2.2 Canny's Detector	65
4.3 Splines Segments	69
4.3.1 Curve Representation Using B-splines Approximation	69
4.3.1.1 B-splines	69
4.3.1.2 Interpolation Using B-splines	71
4.3.2 Surface Representation Using Cubic B-splines Approximation	75
4.4 Applications	80
4.4.1 The Hemispheres and the Ventricles in Each Sectional Image	80
4.4.2 The Surface of the Longitudinal Fissure	82
4.5 Conclusions	87
Chapter 5 Measurement of the Hemispheres	90
5.1 Introduction	90
5.2 Basic Relationship Between Pixels	92
5.3 Basic Geometric Parameters	95
5.3.1 Perimeter	95
5.3.2 Area	95
5.3.3 Volume	96

5.3.4 Surface Area	97
5.3.5 Central Moments	97
5.3.6 Compactness	98
5.3.7 Elongation	99
5.4 Measurement of the Deformation of the Hemispheres	99
5.4.1 Image Acquisition and Segmentation	99
5.4.2 Two Dimensional Measurement	100
5.4.3 Three Dimensional Measurement	106
5.4.4 Invariant Attributes	111
5.5 Conclusions	113
Chapter 6 Shape Analysis of the Lateral Ventricles Using Fourier Descriptors	116
6.1 Introduction	116
6.2 Fourier Descriptors and Its Invariant Properties	118
6.2.1 Fourier Descriptors	118
6.2.2 Invariant Properties	119
6.3 Shape Similarity and Symmetry	120
6.3.1 Shape Similarity	121
6.3.2 Shape Symmetry	122
6.4 Applications	123
6.4.1 Image Acquisition and Segmentation	123
6.4.2 The Training Set	126
6.4.3 Results	127

6.5 Conclusions	131
Chapter 7 Shape Analysis of the Falx Cerebri Using Surface Curvatures	133
7.1 Introduction	133
7.2 Surface Differential Geometry	135
7.3 Surface Curvature	137
7.4 Applications	140
7.4.1 Data Acquisition	140
7.4.2 Measurement	141
7.4.3 Results	142
7.5 Conclusions	147
Chapter 8 Classification and Interpretation	150
8.1 Introduction	150
8.2 Basic Pattern Recognition	153
8.3 Feature Extraction Based on Karhunen-Loève Transform	155
8.4 Bayes Decision	160
8.5 The Decision Rules for Intracranial Structures	164
8.5.1 The Decision Rule for the Hemispheres	164
8.5.2 The Decision Rule for the Lateral Ventricles	171
8.5.3 The Decision Rule for the Falx Cerebri	178
8.6 Scales of Intracranial Deformation	185
8.7 Applications	187
8.7.1 Patients	188

8.7.2 Results	188
8.7.3 Clinical Assessments	190
8.7.4 Discussion	192
8.8 Measurement Reliability	197
8.9 Conclusions	201
Chapter 9 Conclusions and Further Developments	203
9.1 Conclusions	203
9.2 Further Developments	209
References	213
Appendix A Notation	228
Appendix B Relevant Data	231
Appendix C The Program for Image Segmentation	241
Appendix D The Program for Basic Geometric Measurement	263
Appendix E The Program for Shape Analysis	278
Appendix F The Program for Surface Measurement	294

Chapter 1 Introduction

Objective assessment of the severity of a brain tumour and its response to therapy plays an important role in cancer management. Unfortunately the criteria available for measuring an intracranial deformation are inaccurate and inadequate. Clinical response is relevant but is influenced by many factors and assessment is very subjective.

The aim of this thesis is to develop a new quantitative method for the objective assessment of the severity of brain tumours and how they respond to treatment. This is based on *in vivo* analysis of intracranial deformation by MRI data, using shape information and feature extraction techniques. The quantitative assessment will be used as an objective index of intracranial deformation which it is hoped will provide a sensitive and relevant basis for clinical decision making, thus leading to more effective treatment.

Gliomas and other malignant brain tumours are the most common intracranial

neoplasm. They provide a unique challenge in oncology in that, in spite of remaining localised to the brain, they are almost never cured by local treatment [Bloom 1982]. Patients treated with surgical resection and radiation therapy die with local relapse after a median survival of only nine months [Bloom 1982]. There has been little improvement in this success rate in the last 20 years. Other modalities and novel treatments, such as interstitial radiation [Gutin 1985], chemotherapy [Hildebrand 1985], or biological response modifiers [Nagai 1984], may have the potential for prolonging survival times but have not been adequately assessed.

The management of brain tumours is particularly difficult because the relationship between the appearance of the tumour using conventional imaging methods and the patient's clinical condition and prognosis is poorly understood. This frustrates clinical decision making and impairs the assessment of response to treatment.

The volumetric measurement of brain tumour on computed tomography (CT) has demonstrated a possibility of quantitative *in vivo* assessment [Kretzschmar 1982]. However this approach appears to be unsatisfactory because of the significant variations in volumetric measurement caused mainly by the definition of the margins of ill-defined tumours [Chisholm 1989].

The effects of an intrinsic brain tumour are often influenced by factors other than the size of the obvious tumour. Typically, "*tumour mass*" is made up of different components, namely a central area composed entirely of tumour tissue and surrounding

this an often greater zone in which tumour cells infiltrate oedematous and even normal brain [Kelly 1987]. Usually the most crucial factor is the total space-occupying effects that results from the mass of the tumour, and the effects of this on intracranial dynamics. Thus, there is a complex combination of local tissue destruction and distortion, intracranial shift, interference with cerebrospinal fluid circulation, raised intracranial pressure, intracranial herniation, and ischaemia [Anderson 1980]. Clinical symptoms and signs are most immediate and important to the patient but they do not alone provide a satisfactory and secure basis for assessing the effects of the tumour or its response to treatment. Typically, patients are often on multiple palliative drug therapies to suppress unpleasant symptoms.

Radiological methods have improved the diagnosis of brain tumours but not the assessment of their intracranial effects. With contrast computed tomography the area within which enhancement occurs is usually equated with the zone composed entirely or predominantly of tumour tissue. However, enhancement reflects increased blood brain barrier permeability and neovascularisation, rather than tumour *per se*, and thus can be affected by a range of factors, independent of the size and extent of the tumour. Moreover, tumour tissue often extends beyond the area of enhancement and even beyond the area of oedema on both contrast computed tomography and magnetic resonance imaging [Brant-Zawadski 1984, Andrew 1990]. Although Riding *et al* advocated that brain tumour volume should be used as a true scientific end point

in brain treatment protocols, it has proved impossible to measure tumour volume from computed tomography with sufficient accuracy and consistency. Wilson *et al* [Wilson 1979] reported that changes in tumour volume of less than 25% could not be reliably detected by computed tomography and concluded that the most practical methods for assessing the progress of a glioma was the clinical neurological examination. The deficiencies of standard computed tomography were re-emphasised by Chisholm *et al* [Chisholm 1989] who found that the inter and intra observer reproducibility of computed tomography measurements were so poor that “most changes in tumour volume following treatment were obscured by the variability of assessments”.

Magnetic resonance imaging has shown an advantage over computed tomography because of its multi-planar capability, high tissue contrast, and absence of ionizing radiation. However measurement of T_1 and T_2 have not proved useful diagnostically in assessing changes in the tumour, for example Houdek *et al* [Houdek 1988] showed that there was no significant change in T_1 and T_2 measurements in gliomas as a result of treatment by radiation.

The assessment of “total” tumour mass effect on magnetic resonance imaging or computed tomography data has not been satisfactorily achieved with current medical image processing approaches. Quantification of “total” tumour mass has been hindered both by the poorly defined nature of the limits of the tumour and also by the complex, irregular geometry of the area of evident oedema (see Fig.1.1).

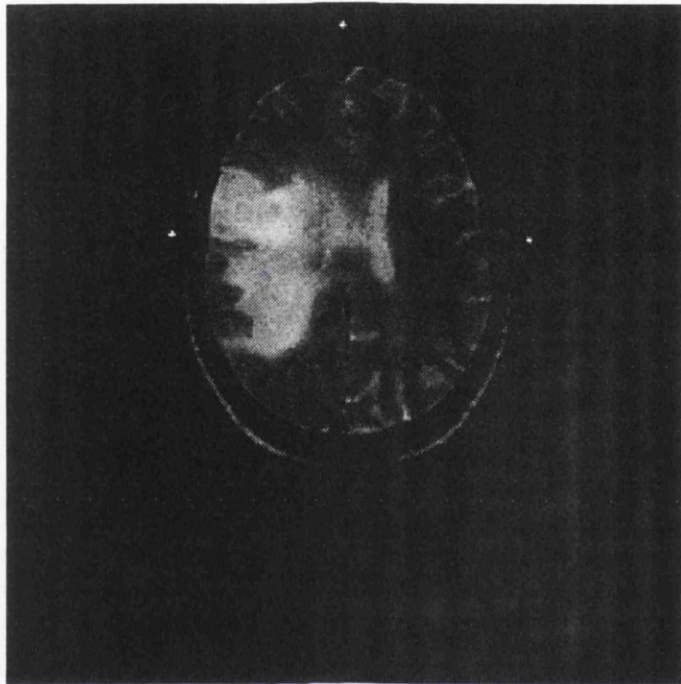


Fig.1.1 An axial T_2 -weighted magnetic resonance image of a patient with a tumour. A large irregular mass lesion is present involving the right mid frontal and parietal lobe surrounded by vasogenic oedema.

Brain tumours cause complex three-dimensional distortion and displacement of major cerebral structures. These take the form of shifting of the longitudinal fissure, displacement of the brain stem, tentorial herniation, and distortion of the ventricular systems. Such deformations are called *secondary space occupying effects* (see Fig.1.2). Usually the distortion and displacement of the brain that results from the mass of the tumour is often of greater significance with regard to the immediate survival of the patient than the nature of the lesion or the amount of cerebral tissue

destroyed by it [Anderson 1980]. The structures themselves can be delineated much more easily than tumours with MR images because they usually have high contrast boundaries between brain tissue and cerebrospinal fluid.

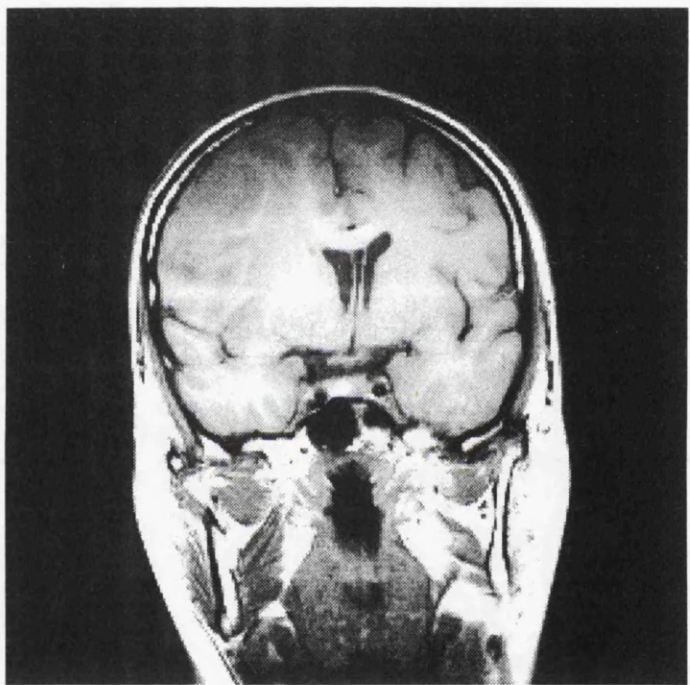


Fig.1.2 A coronal T_1 -weighted MR image of a patient with an intracranial deformation caused by a tumour in the left frontal lobe. It shows a moderate degree of space occupying effect with displacements of the midline and lateral ventricles.

In this thesis we will present a new approach to quantitatively assessing the severity of an intracranial deformation caused by brain tumour and how it responds to

treatment on magnetic resonance images. Instead of trying to determine the ambiguous tumour boundary, we measure these secondary space occupying effects based on the segmentation of intracranial structures, such as the boundary of the cerebral hemisphere, the shapes of the lateral ventricles, and the edge of the longitudinal fissure. In order to comprehensively assess the secondary occupying effects caused by brain lesions, we will use shape analysis and features extraction techniques.

An objective index of intracranial deformation would provide a sensitive and relevant basis for clinical decision making and hence lead to more effective treatment. It would greatly facilitate research into the management of brain tumours. A measure of deformation would provide an objective index of initial severity and hence would be useful in stratifying different groups of patients. Follow-up measurements would be particularly valuable in providing a sensitive index of early response to different modes of management.

This thesis consists of nine chapters including the introduction. In Chapter 2 we give a brief introduction to magnetic resonance imaging and discuss the various modalities and their advantages and disadvantages.

Chapter 3 introduces the basic anatomy and pathology of the brain. In Chapter 4 an effective and practical technique is developed to segment the boundaries and surfaces of intracranial structures in magnetic resonance images. Firstly Canny's edge detector is introduced as this image processing technique provides the optimal

trade off between signal to noise and accuracy in locating edges. Then the B-splines method is developed to represent the boundary or surfaces of brain structures.

In Chapter 5 various methods of measuring the hemispheres are developed. As well as conventional volumetric measurement concepts such as changes in size, we investigate methods to quantify changes in shape and position in terms of shift, symmetry and compactness of the cerebral hemispheres. The results from measurements on normal volunteers and patients are presented.

In Chapter 6 a new shape analysis method (Fourier descriptors) is developed to measure the shapes of the lateral ventricles even when they have different sizes, positions and orientations. Measurements are made in groups of volunteers and patients.

In Chapter 7 a new surface measurement technique is developed and applied in volunteers and patients to characterise the deformation of intracranial surfaces based on novel applications of differential geometry. This method is size, position and rotation independent.

In Chapter 8 a description is presented of an attempt to comprehensively analyse the severity of intracranial deformation and the effects of treatment. Feature extraction techniques (Karhunen-Loève transform) are utilised to reduce the 62 measurement attributes to make them more conceptually accessible to clinicians. Based on the measurements of the hemispheres, the lateral ventricles, and the falx cerebri, three criteria (decision rules) are established to classify the *normal* and *abnormal*

hemispheres, lateral ventricles, or falx cerebri. We introduce a severity scale and variation scale of the intracranial deformation in order to comprehensively assess the severity of intracranial deformation and its response to treatment. Once these have been established from retrospective analysis of normal and patient groups, they are applied in a blinded prospective manner to a further group of patients. The results from this are discussed.

Chapter 9 highlights the original contributions and findings in the thesis and discusses further improvements and further work.

Chapter 2 The Basic Principles of Magnetic Resonance Imaging

Magnetic resonance imaging (MRI) is established as an important modality in medical practice. It can yield a great deal of information because the nuclear magnetic resonance (NMR) signal from which the images are constructed is a function of a number of separate components. The most important of these are: density of the nuclear species (hydrogen in water), two relaxation times (T_1 and T_2), and motion or flow. Other parameters such as chemical shift, diffusion and susceptibility effects resulting from difference in oxygen utilisation, also affect the returned signal and are beginning to find clinical utilisation. The study of other nuclei with net magnetic moments, such as phosphorous-31, as well as hydrogen bound to biomolecules is also possible on some clinical systems though the clinical utility of these techniques has been limited to date.

The clinical applications of MRI are increasing rapidly particularly as it does not

use ionizing radiation and is thus free of the potential hazards of X-rays. All parts of the anatomy, from the head to the extremities, are currently evaluated with MRI, often as the modality of choice. By selecting the proper MRI techniques lesions, areas of oedema, hemorrhage, and flowing blood which are difficult to distinguish using X-ray CT, can often be resolved using MRI.

2.1 The Development of Nuclear Magnetic Resonance

The phenomenon of NMR was first discovered independently by Bloch *et al* [Bloch 1946] and by Purcell *et al* [Purcell 1946]. This was soon followed by the discovery of chemical shift, allowing nuclei in different chemical environments to be identified as a result of the small change in resonant frequency caused by the electron cloud of the molecule.

Although high-resolution NMR has developed as a versatile tool for studying the chemistry and structure of solids and liquids, the major biochemical and medical interest has arisen from the possibilities of making non-invasive measurements in living tissue. Initial measurements of phosphorous-31 in intact blood cells were carried out by Moon and Richards [Moon 1973]. Measurements of frog sartorius muscle followed, but earlier experiments were limited by the small bore of the available

magnets. Developments in magnet technology permitted phosphorus studies to be extended, initially to small animals and then , with the advent of wide-bore high-field magnets, to studies of humans. As *in vivo* NMR spectroscopy of animals and humans has become possible, methods of obtaining spatially localised signals from a well defined region of tissue have been developed.

In parallel with the development of spectroscopic techniques, methods of imaging the distribution of hydrogen protons in tissue water evolved. These techniques again depend on the spatial localisation of the NMR signal, although in this case with a much higher spatial resolution. In 1973 the principle of utilising the shift in resonant frequency resulting from the imposition of a magnetic-field gradient was proposed by Lauterbur [Lauterbur 1973], and by Mansfield and Grannell [Mansfield 1973]. The early images that were formed were limited to small objects, but the first whole-body image was published in 1977 by Damadian *et al* [Damadian 1977]. These early results have been followed rapidly by technical and commercial developments, producing a variety of techniques that allow proton images to be acquired, providing information on spin density and T_1 and T_2 relaxation times. Fast and “real-time” imaging methods have been developed, as well as methods for separating water and fat in proton images, and techniques for measuring blood flow *in vivo*. Despite the rapid rate of development in recent years, NMR is still at a relatively early stage in its development, and many further advances are likely.

2.2 Nuclear Magnetic Resonance

Nuclear magnetic resonance depends on a property called *spin* , which is possessed by some atomic nuclei. Protons in the nucleus have a positive electrical charge. A proton also has a spin, and thus the moving electrical charge produces a magnetic field. The spins experience a torque when subjected to an external magnetic field. As the result, they precess around the axis of the magnetic field at a rate given by the Larmor equation:

$$\omega = \gamma B_0$$

where ω is the resonance frequency in Hz , B_0 is the strength of the static magnetic field, and γ is a constant for the particular nucleus (*gyromagnetic ratio*).

The most abundant nucleus in biologic tissue is hydrogen (one proton, no neutrons). Other atomic nuclei that are theoretically suitable for NMR include carbon-13, sodium-23, and phosphorous-31. Hydrogen in water is used as the signal source in most clinical NMR scans because it is so prevalent.

In the absence of an external magnetic field, the spinning protons are randomly oriented in the body and there is no net magnetisation. If placed in a very strong externally applied magnetic field B_0 , the spinning protons have a tendency to align with or against the field. More will align with the field than against it. This slight preponderance creates a small collective net equilibrium magnetisation M_0 which points along the axis of B_0 (Fig. 2.1). M_0 is also called the *longitudinal magnetisation*.

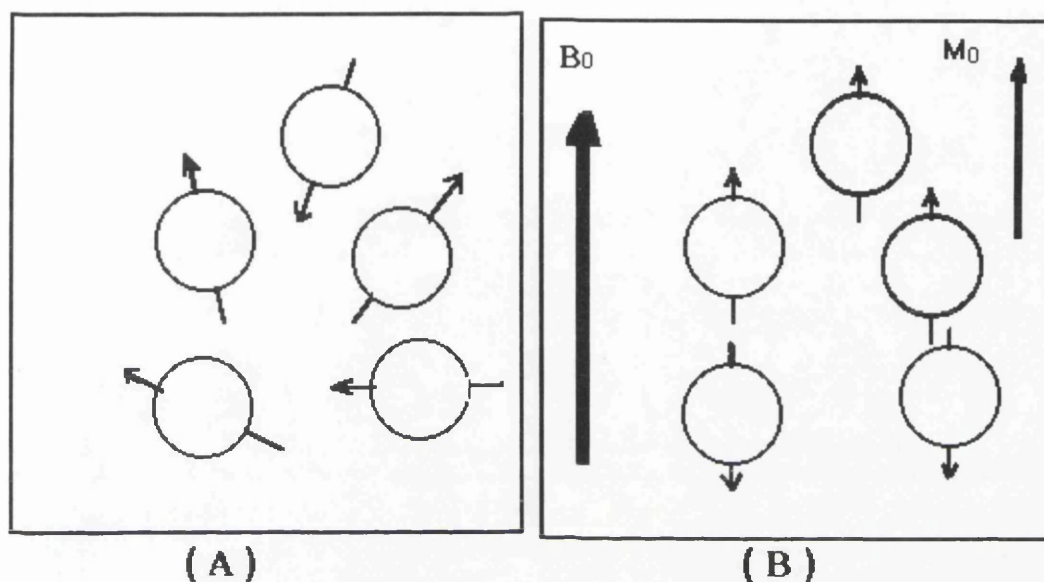


Fig.2.1 Single protons which comprise the nuclei of hydrogen atoms are shown (A) without an external magnetic field applied. The spinning protons are oriented randomly. (B) With an external magnetic field (B_0) applied, protons align either with or against the field. The slight preponderance of protons that are oriented with the field creates a small magnetisation M_0 .

For a given magnetic field strength, different nuclei precess at different resonant frequencies and can thus be distinguished from one another. In order to measure these signals, transverse magnetisation is created. It is generated when a small radiofrequency (RF) field of amplitude B_1 , rotating synchronously with the precessing spins, is applied. When this radiofrequency field acts in a direction perpendicular to the main field, the effect is to rotate the longitudinal magnetisation away from its rest

state. If the duration of the B_1 field is such that the net magnetisation is rotated by an angle of 90 degrees, it will become transverse. The RF which rotates the longitudinal magnetisation by an angle of α is called an α pulse. In most conventional sequences the pulse angle is normally 90° or 180° degrees (though low flip angle gradient echo techniques, such as FLASH, are being increasingly utilised).

2.3 Relaxation Times

As soon as the RF pulse is switched off, the transverse magnetisation will start to decay to zero, while at the same time the longitudinal magnetisation begins to grow back to its equilibrium value. The processes determining the return to equilibrium of both longitudinal and transverse magnetisation are called T_1 relaxation and T_2 relaxation processes. These two processes are the keys to distinguishing tissues and lesions in MR images.

2.3.1 T_1 Relaxation

T_1 relaxation is often called longitudinal relaxation or spin-lattice relaxation. It describes the transfer of energy to or from the spin system as a whole. T_1 is the rate constant of the monoexponential return of the longitudinal magnetisation. As T_1 depends on tissue composition, structure and surroundings, hydrogen protons in different tissue have different T_1 recovery rates. Some tissues (such as fat) have very

rapid recovery rates or “*short*” T_1 values (the order of 150 – 250 milliseconds), while others have much longer T_1 values (*e.g.* about 3000ms for cerebrospinal fluid) [Wehrli 1991]. The return to equilibrium occurs exponentially as [Dixon 1982]

$$M_z(t) = M_0(1 - e^{-t/T_1}) \quad (2.1)$$

where $M_z(t)$ is the longitudinal magnetisation at time t and M_0 is the equilibrium magnetisation. Fig.2.2 shows the T_1 relaxation with time of three types of tissue. At time 0, there is no longitudinal magnetisation at all, and this would be the time immediately after the first 90° pulse.

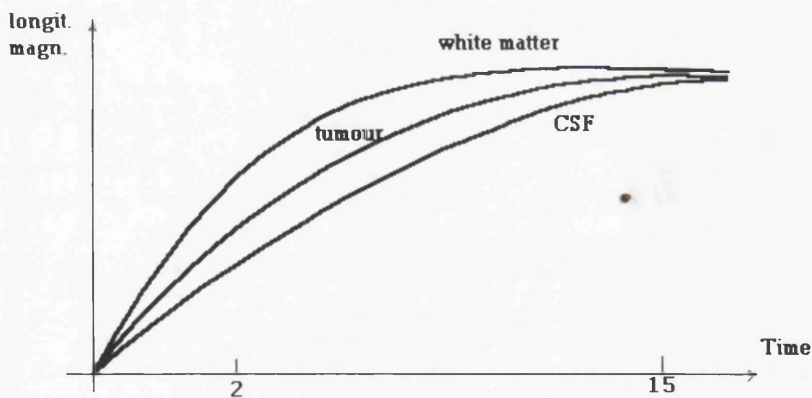


Fig.2.2 Return of the longitudinal magnetisation with time (seconds) for three types of tissue in the body.

If we wait a long time (Say $TR = 15$ seconds, where TR is the repetition time)

before we repeat the 90° pulse, longitudinal magnetisation will have almost completely recovered even for long T_1 substances such as CSF, so there will be only a small difference in signal between white matter, tumour and CSF. If, however, we apply a second pulse after a shorter TR (about 2 seconds in Fig 2.2), the difference in longitudinal magnetisation between the three tissues is much larger, so there will be a better tissue contrast.

2.3.2 T_2 Relaxation

T_2 transverse or spin-spin relaxation describes the redistribution of energy between spins, causing a dephasing of the spins with time. Just as different tissues have different T_1 relaxation times, they also have different T_2 values that are primarily due to different macromolecular environments. This process typically is exponential and can be described as [Edelstein 1983]

$$M_{xy}(t) = M_{xy}^0 e^{-t/T_2} \quad (2.2)$$

where M_{xy}^0 is the initial transverse magnetisation and $M_{xy}(t)$ is the transverse magnetisation at time t .

One spin affects another by slightly altering the magnetic field experienced by the second spin. The resonant frequency of this second spin will change slightly, causing an increase or decrease in phase. In fluids the net difference in field due to tumbling molecules tends to even out, producing little phase difference and so a long

T_2 results. In more solid tissues field differences are more constant, hence there is greater dephasing and so a shorter T_2 .

In tumours, which tend to have more unbound water than normal tissue, spins take longer to dephase and hence have a longer T_2 . T_1 is greater than T_2 in all substances because the longitudinal magnetisation can never be fully re-established until all the transverse magnetisation has decayed away. In most body tissues the dephasing, or decay of transverse magnetisation, occurs much faster than the recovery of longitudinal magnetisation. The T_2 of a given tissue is therefore usually much shorter than its T_1 . T_2 values are typically only 10 – 20% of T_1 values, the main exception being CSF where T_2 is close to T_1 .

Table 2.1 lists relaxation times in different brain tissues at 0.15T (the field strength at which the patients in this project were imaged) [Condon 1986].

Tissue	T_1 (msec)	T_2 (msec)
Grey matter	513 ± 57	118 ± 8
White matter	242 ± 14	86 ± 9
CSF	3302 ± 170	2269 ± 128

Table 2.1 T_1 and T_2 of different brain tissues at 0.15T. Figures are given as mean \pm standard deviation.

2.4 Pulse Sequences

In many parts of the human body, the proton density does not vary greatly from one tissue to another. Thus if the NMR signals displayed in the images depended only on the proton density the contrast would not be very good. However, the NMR signals depend also on the relaxation times T_1 and T_2 , and living tissues vary widely in their relaxation properties. Various NMR pulse sequences can be employed to improve the contrast between the various organs and between normal and pathological tissue. We will discuss two of the most common of these.

2.4.1 Spin Echo

Following a 90° pulse, the signal should decay with a time constant T_2 , the spin-spin relaxation time. However, the transverse magnetisation decays much faster, with a time constant T_2^* , the effective transverse relaxation time. This happens because spins at different locations experience slightly different magnetic fields due to tiny imperfections in the machine's static magnetic field. This results in slightly different precession frequencies for the different spins. Consequently the spins lose their phase coherence and the transverse magnetisation decays faster than it would on the basis of T_2 processes alone. A 180° RF pulse, applied some time τ after the 90° pulse, can be shown [Andrew 1990] to reestablish phase coherence at time τ later (*i.e.*, at the time $t = 2\tau$ following the 90° pulse, see Fig. 2.3). Thus any remaining loss of phase

coherence will be due to the “pure” T_2 effect. The period (2τ) between the initial 90° pulse and the echo is denoted “echo delay” or “echo time” (TE), and the time before the start of the next set of pulses is called the repetition time (TR).

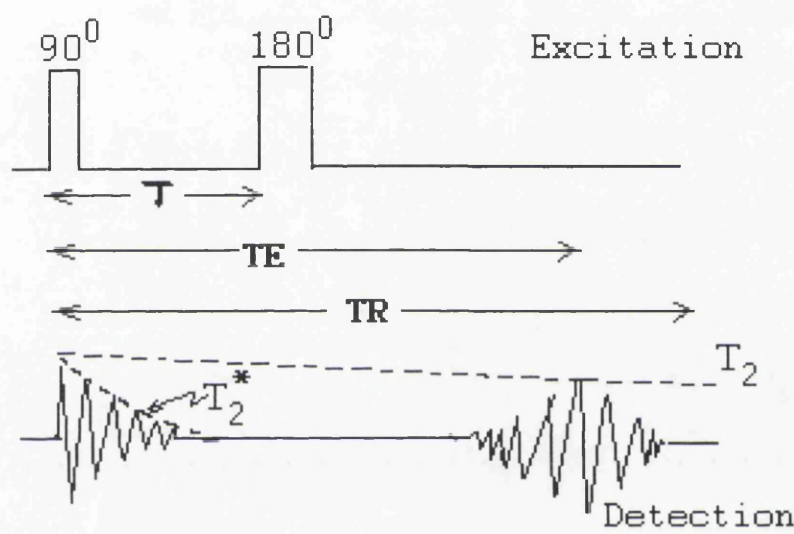


Fig.2.3 The spin-echo pulse sequence. The 180° pulse, applied τ ms after the initial 90° pulse, generates a spin echo at time $t = 2\tau = TE$. The amplitude of the echo is a function of the echo delay and the spin-spin relaxation time T_2 . By contrast, the free induction signal decays with a time constant T_2^* .

A spin echo pulse sequence is therefore composed of two pulses: a 90° pulse and, after time $TE/2$, a 180° pulse.

The echo signal is produced by the refocusing the transverse proton magnetisation,

so that it is decaying exponentially with only the time constant T_2 . Some tissues have much longer values of T_2 than others(*e.g.* tumourous tissue compared to white matter). If TE is made comparable with the longer T_2 values, we will get weaker signals, but greater contrast between tissues with different T_2 's. In this way, a T_2 -*weighted image* is generated, with tissues having long T_2 values appearing brighter in the image (see Fig.2.4). Spin echo sequences can also be used to produce T_1 weighting by reducing TE (and hence minimising the T_2 effect) in conjunction with a reduction in TR (thus increasing the T_1 effects contribution to the signal).

The spin echo signal intensity I is approximately given by the expression [Wehrli 1991]:

$$I \propto N(H)(1 - e^{-TR/T_1})e^{-TE/T_2} \quad (2.3)$$

where $N(H)$ represents the proton density. As indicated by equation (2.3), the signal intensity is related to both T_1 and T_2 . Furthermore it can be inferred that the spin echo signal intensity increases if :

1. the repetition time (TR) between successive 90° pulses is increased.
2. T_1 decreases.
3. the echo delay (TE) is shortened.
4. T_2 increases.

Obviously T_1 and T_2 are tissue specific, while TR and TE are under operator control.

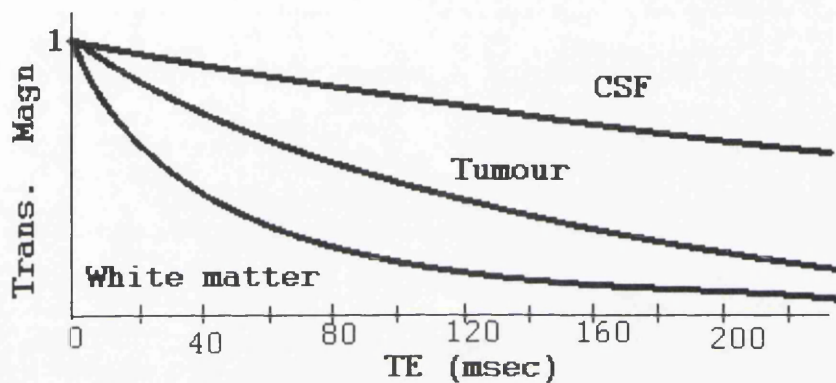


Fig.2.4 Transverse relaxation

2.4.2 Inversion Recovery

An *inversion recovery sequence* uses first a 180° pulse which is then followed by a 90° pulse (Fig.2.5) and is generally used to produce high contrast T_1 weighted images.

The 180° pulse inverts the longitudinal magnetisation. This is illustrated in Fig. 2.6 and 2.7 for two tissues with different T_1 's. To get a measurable signal, some transverse magnetisation is needed. Thus a 90° pulse is applied. The signal thus depends on the time TI (*inversion time*) between the 180° and the 90° pulse, as well as the TR of the sequence.

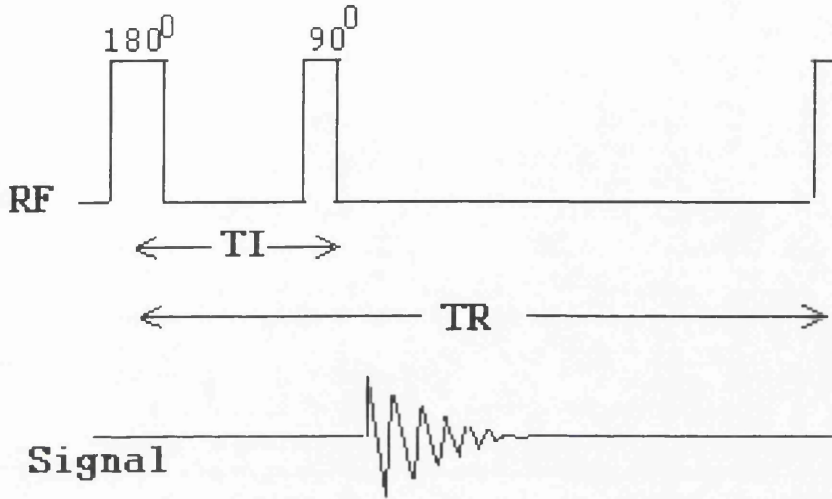


Fig.2.5 The inversion recovery sequence. The amplitude of the free induction signal, generated by the 90° detection pulse, is a function of the interpulse delay (TI), the pulse sequence repetition time (TR) and the spin lattice relaxation time (T_1).

The signal intensity can be shown to be [Wehrli 1991]

$$I \propto N(H)(1 - 2e^{-TI/T_1} + e^{-TR/T_1}) \quad (2.4)$$

If the pulse interval (TI) between the 180° pulse and the subsequent 90° detection pulse is much shorter than tissue T_1 relaxation times, the signal according to equation (2.4), has negative signal values (inverted magnetisation). Equation (2.4) shows that image density is dependent on the proton density $N(H)$, relaxation time (T_1), on

the interpulse interval (TI), and on the repetition time (TR) between successive 180° pulses.

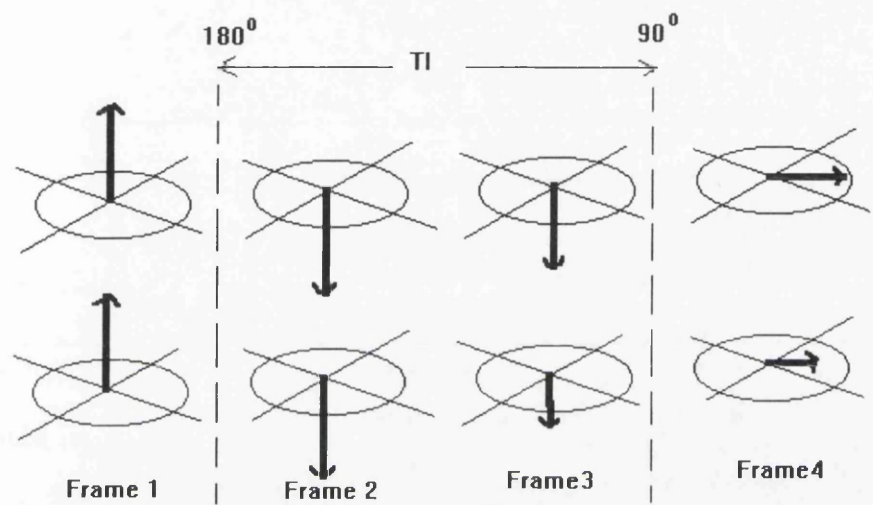


Fig.2.6 The inversion recovery sequence uses a 180° pulse that inverts the longitudinal magnetisation, followed by a 90° pulse after time TI. The tissue in the bottom row has the shorter T_1 due to the faster recovery.

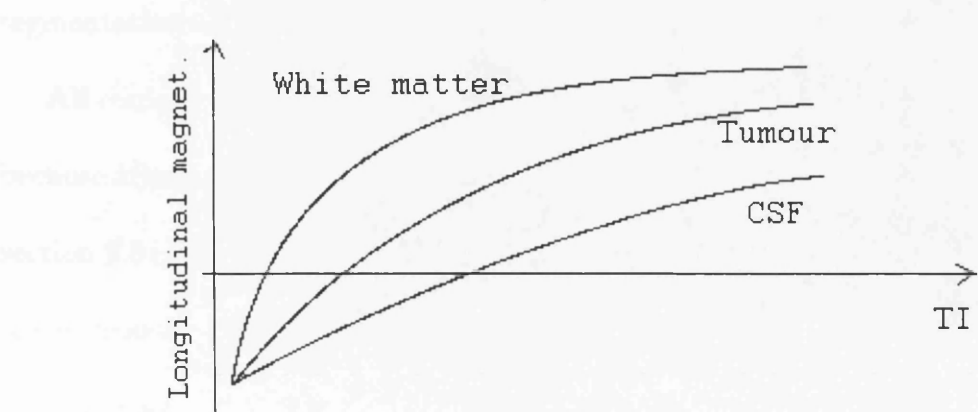


Fig.2.7 Effect of an inversion recovery pulse sequence on contrast.

Generally inversion recovery sequences are not used on modern imagers. Though they provide very good T_1 contrast they tend to have lower signal to noise ratios than the spin echo sequences. The spin echo sequences are generally used to provide T_1 contrast by decreasing TE and TR , and T_2 contrast by increasing TE and TR [Atlas 1991].

Field (gradient echo) sequences [Frahm 1985] are being increasingly utilised on modern systems because of their relative speed. However the imager used for this project could not perform such sequences.

In our project, we use an IRSE sequence to produce T_1 -weighted images. The full sequence used on our Picker MRI system (0.15T) is an IRSE 3400/500/32, namely an IR sequence with a delay time of 500 *ms*, a TE of 32 *ms* and a TR of 3400 *ms*. The reason we used this sequence is that it gives the best interface definition (which is necessary for determining accurate brain structure boundaries during the image segmentation) in the least time on the Picker imaging system.

All commercial systems typically include an SE read in their IR sequences. This is because after the 90° pulse time is required to apply the various imaging gradients (see section 2.5), during which T_2^* effects can rapidly reduce the transverse magnetisation. An additional 180° pulse is therefore applied to rephase the transverse magnetisation and increase the acquired signal. The echo time was kept as short as possible to minimise “pure” T_2 effects, as it was T_1 contrast that we were interested in. T_1

contrast was optimally achieved (at this field strength) using a TI of $500ms$, thus giving best interface definition. The long TR was necessary because all 24 slices were acquired on an interleaved basis in one acquisition, thus minimising acquisition time and patient discomfort (14.5 minutes).

2.5 Image Generation

2.5.1 Use of Magnetic Field Gradients

A gradient occurs when the magnetic field is varied linearly along, for example, the z axis. The resonance frequency becomes dependent on the location of the volume element of interest with respect to z .

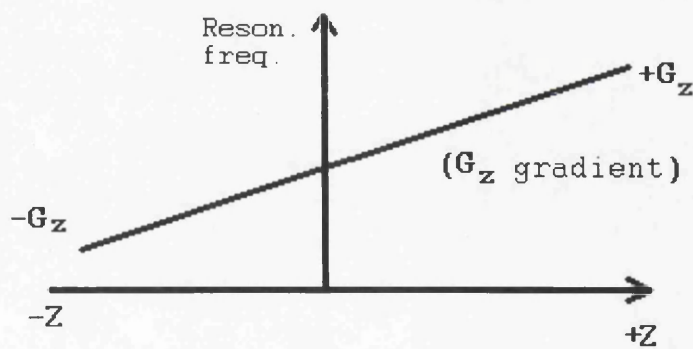


Fig.2.8 Schematic illustration of the effect of a z gradient.

This variation can be generated along either of the three directions, x , y , or z [Lai 1981]. In this case, the x , y , and z coordinates refer to the fixed, “laboratory” frame

of reference. The three gradients are called G_x , G_y and G_z , respectively (see Fig2.8).

The mathematical statement of their effect is :

$$B = B_0 + xG_x \quad \text{for the x gradient}$$

$$B = B_0 + yG_y \quad \text{for the y gradient}$$

$$B = B_0 + zG_z \quad \text{for the z gradient}$$

2.5.2 Slice Selection

In all two-dimensional MR imaging techniques the protons in the imaging slice are selectively excited by means of a combination of a frequency selective RF pulse and a magnetic field gradient perpendicular to the imaging plane (See Fig. 2.9). We shall assume that the slice is axial, so the z direction is the slice selection direction. The z gradient is turned on by the computer's pulse program at the same time as a specially shaped RF pulse is transmitted. In this situation, as can be appreciated in Fig. 2.9, only those protons with a certain z position will have (local) Larmor frequencies matching the transmitted RF frequency, and thus only they will be excited. Protons located at other positions will experience a mismatch between the Larmor and RF frequencies and will remain unexcited.

For a specified slice there are two ways to dictate the slice thickness [Stark 1988].

- We send in not only one specific frequency but an RF pulse that covers a range of frequencies; the wider the range of frequencies, the thicker the slice in which

protons will be excited. This has been illustrated in Fig.2.9. If we use a RF pulse with frequencies from 64 to 65 MHz, we will get a slice thickness S_1 (Fig.2.9 (a)). If, however, we only use frequencies from 64 to 64.5 MHz, only the protons in a smaller slice, S_2 , will show resonance (Fig.2.9 (b)) .

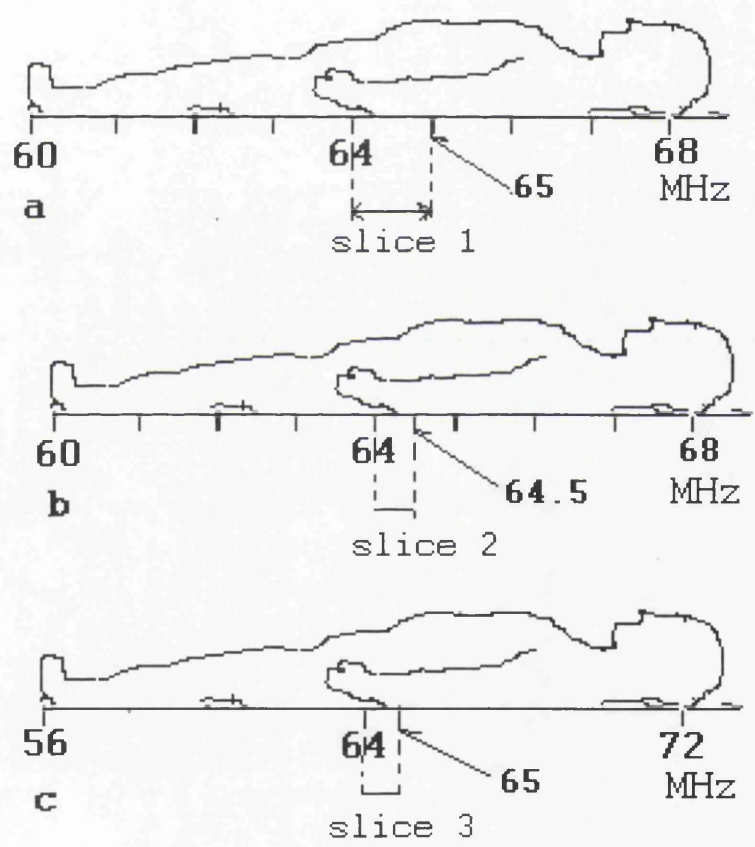


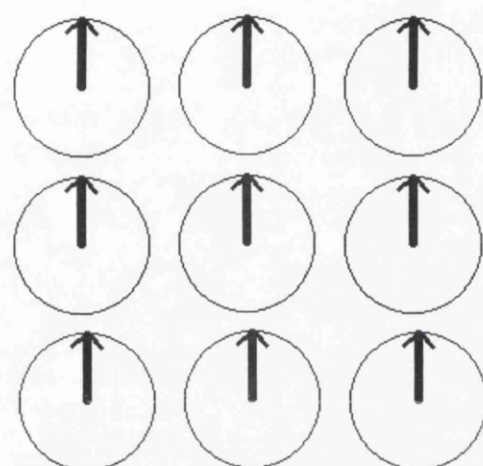
Fig.2.9 The selection of slice thickness

- If we use the same range of radio frequencies (from 64 to 65), slice thickness can be modified by the slope of the gradient field, as is illustrated in Fig.2.9 (c).

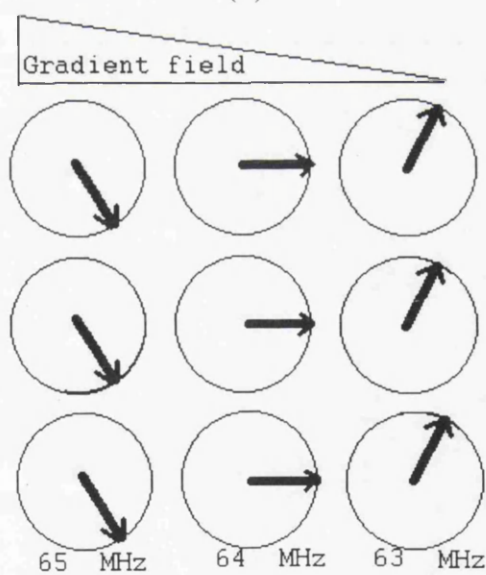
2.5.3 Frequency and Phase Encoding

After slice selection, we need to localise each point in the slice according to its proton signal. To achieve this a magnetic gradient, called *frequency encoding gradient* [Valk 1985] is applied after the slice selection gradient. We will consider it being applied in the direction of the x -axis. (in modern machines slice select, frequency and phase encoding gradients are actually interchangeable in terms of x , y and z directions). This results in different precession frequencies along the x -axis, and thus different frequencies of the returned signals.

This is illustrated in Fig.2.10, which shows the situation of protons in the slice selected. In Fig 2.10 (a) nine protons in the same slice are depicted. They precess in phase with the same frequency after the RF pulse is applied. A magnetic field gradient is then superimposed on the external field, which in (b) decreases in strength from left to right. The protons in the three rows now experience different magnetic fields, and thus emit signals with different frequencies (*e.g.* 65, 64, and 63 MHz). The gradient applied is thus called the *frequency encoding gradient*. By applying a Fourier Transform to the conglomerate returned signal we can separate into frequency, which because of the applied x gradient is now directly related to position along the x axis. However all protons in one column will still have signals with the same frequency. We can thus tell from which row (x direction) a signal comes from, but cannot determine which column (y direction) a signal comes from.



(a)



(b)

Fig.2.10 The effect of applying a frequency encoding gradient

In order to encode the y position, another gradient, called *phase encoding gradient* [Valk 1985], is turned on for a short time after the RF pulse along the y -axis. During this short time, the protons along the y -axis precess with different frequencies and so develop a spread of phases. When this gradient is switched off, they go back to their former precession frequency, which was the same for all of them but they will now have different phases. The application of a second Fourier Transform will then allow us to establish the y position.

Fig.2.11 illustrates the phase encoding technique. In Fig.2.11 we have the protons of one column from Fig. 2.11, the 65 MHz column. The protons are in phase after the RF pulse (Fig.2.11(a)). Then an additional gradient is applied along this column for a short time. This causes the protons to speed up their precession according to the strength of the magnetic field to which they are being exposed. In the example (Fig 2.11 (b)) the increase in speed is less from top to bottom in the column. When this short gradient is switched off, all the protons of the column experience the same magnetic field again, and thus have the same precession frequency. However, there is an important difference. Formerly the protons (and their signals) were in phase. Now the protons and their signals still have the same frequency, but they are out of phase.

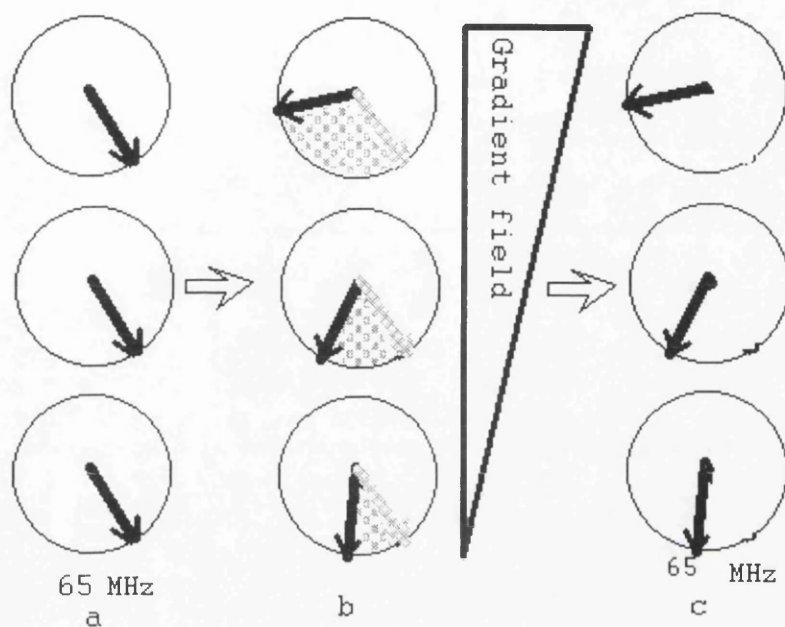


Fig.2.11 Phase encoding

The phase encoding gradient is turned on for a brief period during each imaging cycle of duration TR . The strength of the gradient is changed slightly from one cycle to another. This is necessary to distinguish its effects from those of the frequency encoding gradient. In a very general sense each gradient setting creates one *temporal view* that will be used in image reconstruction. However, a phase encoded *temporal view* is distinctly different from the directional views of computed tomography. The number of image elements (voxels) to be created in phase-encoded direction

determines the minimum number of *temporal views* or repetitions of the imaging cycle required to create a complete image. If the image matrix is to contain 256 rows (or y-lines), then the imaging cycles must be repeated at least 256 times with the phase-encoding gradient strength changed from one cycle to another. This is the primary factor that causes the acquisition of an MR image to be a relatively time-consuming process. Examination time is proportional to spatial resolution along the phase-encoded dimension (y), as resolution of N rows requires N imaging cycles (TR).

2.5.4 Image Reconstruction

Image reconstruction is the mathematical process of converting the composite signals obtained during the acquisition phase into an image. It is performed by an array processor or computer. Several methods can be used to reconstruct MR images [Morris 1986]. Two dimensional Fourier transformation is the method most frequently used.

The basic concept of a Fourier transform is illustrated in Fig.2.12. The primary function of the Fourier transform is to convert a signal from the time domain into the frequency domain. In our example we have a signal that has two distinct frequency components. In principle, the Fourier transform determines the frequency values of each component. In the illustration the two components are shown on a frequency scale.

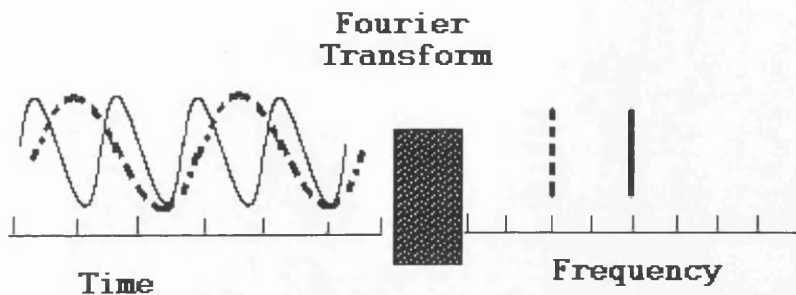


Fig. 2.12 Fourier transform is a mathematical operation that breaks down a signal into its frequency components.

In the reconstruction of an MR image the Fourier transform is used to “*decode*” both the frequency and phase encoded signal components [Kumar 1975a,b]. Each composite signal is passed through a Fourier transformation, which breaks it down into a series of individual frequency or phase components that correspond to the voxel columns within the selected slice.

2.6 MRI Hardware

The scanner used for magnetic resonance imaging consists of a large magnet, radiofrequency coils, gradient coils and a computer system. These will be briefly described.

2.6.1 The Magnet

Magnets used for imaging mostly have field strengths between 0.04 to 1.5 Tesla. The magnetic field has to be very homogeneous, as this directly determines the precession frequency. The magnet can be of three types, *permanent*, *resistive*, or *superconducting*.

- Permanent magnets. These do not require electricity or very low temperatures to operate but they suffer from thermal instability and limited field strength. They also cannot be turned off in an emergency.
- Resistive magnets. In a resistive magnet, an electrical current is passed through coiled wire to generate the magnetic field. Compared with permanent magnets they can achieve a higher field strength but cannot generate field as high as superconducting systems because of the vast amount of heat the current produces. Resistive magnets can however be turned off quickly in an emergency without revenue consequences.
- Superconducting magnets. Superconducting magnets are the ones most widely used in MR machines at the present time. They also use electrical currents to generate the field but the current carrying conductor is cooled down to superconducting temperatures (about $4^{\circ}K$ or $-269^{\circ}C$). At this temperature, the

current conducting material loses its electrical resistance. Once applied an electrical current circulates permanently, creating a constant magnetic field. So called *cryogens* (liquid helium and nitrogen) are used for cooling these magnets and require regular re-fills. The advantages of superconducting magnets are high magnetic field strength and excellent magnetic field homogeneity. Disadvantages of superconducting magnets include high initial costs, and the use of rather expensive cryogens. The magnet can be turned off in an emergency but this involves making the superconductor resistive. Heat is generated and this turns several thousand pounds worth of cryogens to gas, which has to be vented to the atmosphere.

2.6.2 Radiofrequency Coils and Gradient Coils

The radiofrequency (RF) coils are embedded behind curved panels which surround the patient during scanning. They act as generators for RF excitation and as receivers for detecting the returned signal. RF coils take many forms ranging from simple circular single turn coils to “birdcage” arrangements for higher fields systems.

Gradient coils are used to systematically vary the magnetic field by producing additional linear electromagnetic fields, thus making slice selection and spatial information possible. There is a set of gradient coils for each of the three dimensions in space. Coils to produce a gradient along the patient’s long axis (z direction) tend to

be solenoidal, whilst those producing x and y gradients are usually saddle shaped.

2.6.3 Computer System

A computer system is required to control gradients and RF pulses, digitise the returned analogue signal, perform the Fourier Transforms and allow image processing on display terminals. It also has hard copy facilities and a large storage capability for image archiving.

2.7 Safety of MRI

There are two areas of safety to consider in operating an MR facility: direct and indirect biological effects.

2.7.1 Direct Biological Effects

MRI involves exposure to three different types of magnetic and electromagnetic field. There is the main, or static magnetic field, the alternating magnetic fields produced by the gradients, and the radiofrequency field produced by the RF coils. Both patients and staff are exposed to these three different fields. At the exposure levels encountered in current MR systems, no evidence of any deleterious long-term or short-term effects have been reported [Department of Health, HMSO 1993]. However, at much higher values of RF and alternating magnetic fields, there are possible mechanisms that may

lead to damage (*e.g.* the possibility of muscle and nerve activation by currents induced by rapid changes in magnetic field gradient). Limits of exposure have been produced by the National Radiological Protection Board [NRPB 1983] in the UK and by the Food and Drug Administration [FDA, 1982] in the USA.

2.7.2 Indirect Effects

Care must be taken with loose metal objects in the vicinity of the magnets. Objects can become lethal projectiles, being drawn towards and into the magnet. Surgical clips in the patient may be dislodged, and the magnetic field could also affect electrical implant such as cardiac pacemaker used for cardiac disorders. Sensitive instruments such as cameras and watches may be damaged in the area of the magnetic field and any magnetic recording (tape, disk) could be erased. Department of Health Guidelines were adhered to for all imaging performed in this project. All exposures were well below NRPB limits.

2.8 Other Medical Imaging Modalities

In addition to magnetic resonance imaging there are three other modalities with tomographic capabilities. They are:

- X-ray Computed Tomography (CT)
- Ultrasound.

- Single Photon Emission Computed Tomography (SPECT) and Positron Emission Tomography (PET).

The first two of these imaging techniques provide essentially anatomical information (as opposed to SPECT and PET which produce indices of function). The principles of the techniques are discussed in the following sub-sections. In addition, the advantages and disadvantages of employing these methods are examined.

2.8.1 X-ray Imaging

X-rays are radiation that appear in the electromagnetic spectrum. Diagnostic x-rays are not radiation of single frequency but cover a limited spectrum of wavelengths from 0.01\AA to 0.05\AA . This region in the frequency band has been found to be suitable for imaging the interior parts of the human body. X-ray based systems represent the oldest and probably the most widely utilized diagnostic imaging techniques.

The selection of the useful frequency spectrum is governed by resolution and attenuation factors. To obtain a worthwhile image, the radiation must have a wavelength under 1.0cm in the body for resolution considerations. Also, there should be some degree of attenuation in the radiation when passing through the body. If it is too highly attenuated, noise dominates and results in a poor quality image. Such attenuation can be due to absorption or multiple scatter. Almost complete transmission without attenuation implies that the measurement accuracy will be too small to extract

meaningful information.

The traditional approach in x-ray imaging is to record the received x-ray intensity on a film or to intensify the image for presentation on a TV monitor. Thus a conventional x-ray shows the projection of a 3-dimensional volume (*e.g.* the patient's chest) on to a 2-dimensional surface (*e.g.* a chest x-ray film).

Computed tomography (CT) is based on the same principle of differential x-ray absorption by tissues of varying atomic numbers. However in CT a large number of beam transmissions can be used from different angular positions to extract data relevant to the internal structure [Scudder, 1978]. Using a computer and the series of angular projections acquired, the data can be back projected and filtered to obtain a cross-sectional image. Unlike MR where image planes can be in any axis, CT is usually restricted to axial or off-axial planes. With a sufficient number of 2-dimensional slices a complete 3-dimensional data set can be produced.

A conventional 2-D projection x-ray view cannot differentiate between organs or structures that overlap. Moreover, they can only resolve tissues with significantly different densities. For example, bone and soft tissues can easily be isolated from each other in x-ray imaging. The reconstruction approach, which is an integral part of the CT technique, essentially eliminates the problem of structure overlap and greatly improves the resolution.

Although CT is used primarily for brain scanning, its ability to resolve small

differences in density has allowed it to be used to monitor lesions of the kidney, pelvis, liver and the pancreas.

2.8.2 Ultrasonography

This imaging technique uses ultrasound for the production of images of organs within the human body [Brinkley 1983]. Ultrasound is an acoustic radiation that occupies a frequency range above that of audio frequency, *i.e.* greater than 30kHz.

The velocity of propagation of sound in water and in most body tissues is about 1.5×10^3 m/sec. This gives rise to a useful frequency spectrum well above 0.15MHz. The attenuation coefficient in body tissues varies approximately proportionally to the acoustic frequency at about 1.5db/cm/MHz. The result is that at high frequencies excessive attenuation becomes a problem. This means that in imaging the various parts of the body different frequencies are used. For example, for the thick parts of the body such as in the imaging of the abdomen, the frequency range is 1.0 to 1.3MHz. For imaging the shorter pathlengths, such as the eye, the frequency used can be as high as 20MHz.

In ultrasound reflection images are produced using the known velocity of propagation to calculate the depth. At the frequency band at which soft tissue imaging is feasible, air produces considerable attenuation of the radiation. This means that certain parts of the anatomy, particularly the lungs, are very difficult to study using

ultrasound imaging procedures. Ultrasonic studies of the heart would be difficult but for a window called the cardiac notch. This is an opening in the front part of the left lungs that allows access to the heart.

Digital two-dimensional echocardiography is an ultrasonic imaging technique that is used as an important noninvasive technique in the comprehensive characterization of the left ventricular structure and function [Chu 1988]. In this method, a pulse is sent along a ray from a transducer towards the organ that is being imaged. The pulse is attenuated and reflected when it hits a medium with an acoustic impedance different from that of the medium in which the pulse is traveling. The time it takes in transit is a measure of the distance of the boundary from the transducer, and the amount of energy reflected is a measure of the difference of the acoustic impedance across the boundary.

In practice, since the energy of the pulse diminishes as it travels, the postprocessing of the reflected signal includes time gain control that compensates for the attenuation of the signal over time. Assuming the pulse travels at a single speed in the body, and by taking different rays across a plane, a two dimensional record of the received energy in spatial coordinates represents a cross-sectional view of the organ.

The advantage of ultrasound imaging is that it is quick and simple to perform, and harmless to the patient. The main disadvantage as far as imaging of the brain is concerned is that ultrasound is absorbed readily by bone, so bony structures act as

a barrier to the technique. Ultrasound is therefore limited to intraoperative investigations of the brain, or blood flow studies of some cerebral vessels *via* the temporal bone (transcranial doppler) [Bamber 1988].

2.8.3 Radionuclide Imaging

The radioactive isotopes used in diagnostic imaging emit gamma rays as they decay. Gamma rays are electromagnetic radiation, similar to x-rays, produced by radioactive decay of the nucleus. Radionuclide imaging depends on the fact that certain substances concentrate selectively in different parts of the body. Radionuclides can be chemically tagged to these substances. Occasionally, the radionuclide in its ionic form will selectively concentrate in an organ (*e.g.* Iodine-131 in the thyroid), so there is no need to attach it to another compound. The radionuclide most commonly used is technetium-99m. It is readily prepared, has a convenient half life of six hours and emits gamma radiation of a suitable energy for easy detection. Many other radionuclides are used in medicine, including gallium-67, iodine-123, indium-113m and thallium-201 [Srivastava 1983].

Technetium-99m can be used in ionic form (as pertechnetate) for brain, thyroid and vascular imaging, or ^{99m}Tc can be tagged to other substances [Neirinckx 1987]. If a sulphur colloid is labelled with technetium-99m and injected intravenously it will be taken up by the reticuloendothelial system and can be used to visualise the liver

and spleen. Larger particles are used in lung perfusion images ; macroaggregates of albumin with a particle size of 10-75 μm when injected intravenously are trapped in the pulmonary capillaries. If the macroaggregates are labelled with $^{99\text{m}}\text{Tc}$ then the blood flow to the lungs can be visualised. It is also possible to label the patient's own red blood cells with $^{99\text{m}}\text{Tc}$ to assess cardiac function or the white cells with indium-111 for abscess detection. Small quantities of radioactive gases, such as Xe-33, xenon-127 or krypton-81m can be inhaled to assess ventilation of the lungs.

The gamma rays emitted by the isotope are detected by a gamma camera enabling an image to be produced. A gamma camera consists of a circular sodium iodide crystal. Light is produced when the gamma rays strike and activate the sodium iodide crystal, and the light is then electronically amplified and converted to an electrical pulse. The electrical pulse is further amplified and analysed by a processing unit so that a recording can be made. Usually, some form of computer is linked to the gamma camera to enable rapid serial images to be taken and to perform computer enhancement of the images if necessary.

In the case of conventional emission tomography , the gamma camera moves around the patient. A computer can analyse the information and produce sectional images similar to computed tomography. Emission tomography can detect lesions not visible on the standard views. Because only one usable photon for each disintegration is emitted this technique is also known as single photon emission computed

tomography (SPECT). Positron emission tomography (PET) is limited in its clinical application by the need to have a dedicated cyclotron close by, and by the highly sophisticated imaging and computing system which is employed . PET has exciting possibilities for research into biochemical processes *in vivo* because of its ability to utilise short -lived tracers of biologically important atoms such as carbon, oxygen and nitrogen.

Nuclear medicine techniques are mainly used to measure and image function and not anatomy. Even the bone scan depends on bone turn-over. The images produced are also limited by the relatively poor spatial resolution compared to MR, conventional radiography, ultrasound, computed tomography.

2.8.4 Comparison of MRI with other Modalities

2.8.4.1 Advantages of MRI

Soft tissue contrast:

Probably the most important factor in favour of using magnetic resonance imaging instead of x-ray based techniques is that MRI has a high level of sensitivity over a wide range of disease. Unlike CT, MRI is not restricted to the axial plane. Anatomical images can be generated in coronal, sagittal and oblique planes as well as in axial sections.

A major advantage of MRI in assessment of intracranial tumours is in its ability to

accurately demonstrate the positions, and relationships of lesions using multiplanar scanning. Low-grade astrocytomas not seen on CT may be detected by MRI. Pathological change associated with an increased T_2 is highlighted with *spin echo* scans. Sagittal and coronal images are readily obtained, and can be useful to demonstrate and localize certain lesions, particularly when they are in the mid-line or deep-seated. Longitudinal demonstration of the spine and spinal cord by MRI is another advantage over CT.

The resolution of nuclear medicine images (such as SPECT or PET) is inherently limited and although technological advances will give some improvements, the degree of anatomical detail possible will always be less than that achieved by conventional radiology, CT and MRI.

Although ultrasound is useful in the detection of neonatal haemorrhage and hydrocephalus, it gives no information about diffuse cerebral pathology.

Safety:

Unlike Nuclear Medicine and CT, MR and ultrasound imaging do not involve X-rays and so avoid the potentially harmful effects of ionizing radiation. The absorption of radiofrequency and ultrasound waves in the body do however result in the transfer of energy with a rise in local temperature. With very high energy radiation, mechanical disruption can occur at cellular level. However, the energies used for MRI and ultrasound are several orders of magnitude lower than the levels needed to produce

any biologically measurable effects, so MRI and ultrasound are generally regarded as being free of risk as far as direct biological effects are concerned (indirect effects due to the action of the MRI fields on metal were discussed in section 2.7).

2.8.4.2 Disadvantages of MRI

Disadvantages of MRI include the high cost of the sophisticated machinery, the long imaging times at present required and its inability to image calcification. Compared to SPECT or PET, MRI is insensitive to information on tissue perfusion and organ functions (though new techniques are being investigated which may change this situation). It is also unsuitable for patients with cardiac pacemakers as these can be adversely affected by the magnetic fields. Similarly patients harbouring metallic clips or implants should not be subjected to MRI.

Chapter 3 The Basic Anatomy and Pathology of the Brain

3.1 The Brain

The brain of an average adult is one of the largest organs in the body, weighing typically between 1,200 g and 1,600 g [Henry 1985]. The brain lies within the skull and is separated from the bone by three membranes or meninges, called the *dura mater*, the *arachnoid* and the *pia mater* [Williams 1980]. As shown in Fig. 3.1, the brain is divided into three principal parts:

1. The *forebrain* lies above the tentorium cerebelli, and comprises the right and left cerebral hemispheres and the diencephalon, which lies between the two hemispheres.
2. The *midbrain* is the part of the brain lying in the opening in the tentorium cerebelli.

3. The *hindbrain* comprises the pons, medulla oblongata, and the cerebellum, and lies below the tentorium cerebelli.

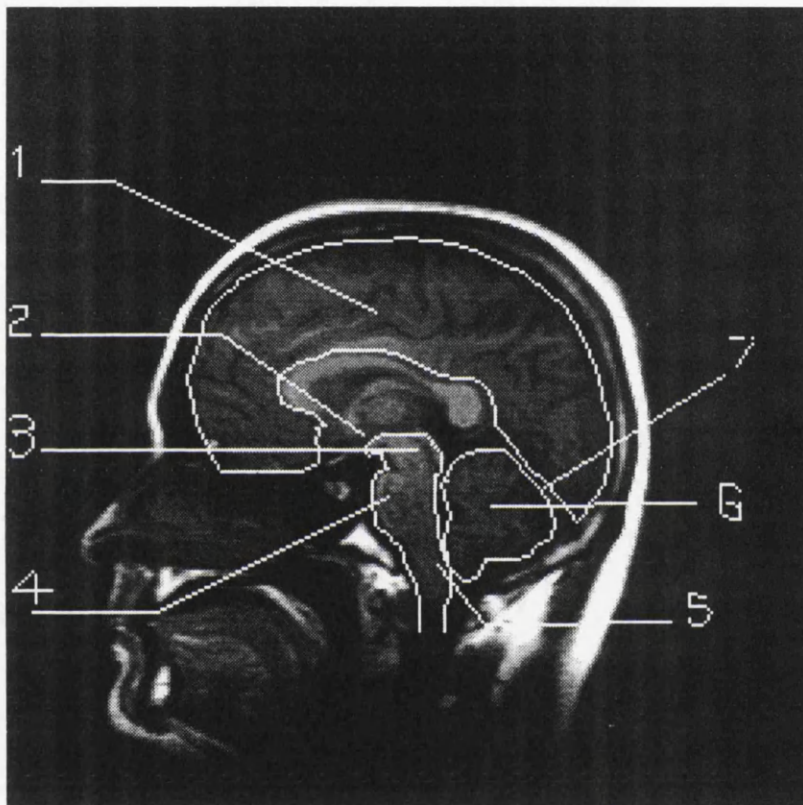


Fig. 3.1 Median section of head

- | | |
|-----------------------|----------------|
| 1 hemisphere | 2 diencephalon |
| 3 midbrain | 4 pons |
| 5 medulla oblongata | 6 cerebellum |
| 7 tentorium cerebelli | |

Note that all parts of the central nervous system are made up of what are called *grey matter* and *white matter*. The grey matter consists of aggregations of cell-bodies of neurones. The white matter, which is preponderant in the central nervous system, consists of nerve fibres. The surface of the cerebral hemispheres and of the cerebellum comprises a thin layer of grey matter.

3.1.1 Meninges

The outermost membrane is the dura mater, the middle one the arachnoid mater, and the inner one is the pia mater. These layers have a protective function in that they enclose the central nervous system and anchor it against sudden movements. They also enclose the cerebrospinal fluid (CSF), which forms a fluid cushion to protect the brain from trauma and is an intermediary in the exchange of substances between the brain and the rest of the body. The cranial dura mater is a double layer of tough connective tissue. Its outer layer adheres to the bones of the skull and forms the periosteum. Its inner layer, the true dura mater, lines the skull and forms sheets of tissue which dip between the cerebral hemispheres (falx cerebri), between the cerebellar hemispheres (falx cerebelli) and between the cerebellum and the cerebrum (tentorium cerebelli).

The arachnoid mater is composed of connective tissue with flat interdigitating cells on its surface. A narrow potential space, the subdural space, lies between the arachnoid and the dura mater. It contains only a little serous lubricating fluid. A

wider space, the subarachnoid space, separates the arachnoid from the pia mater.

The pia mater is very thin and rich in capillaries. It is attached to the brain, closely following the contours of its folds (gyri) and fissures (sulci). It is also closely bound to the spinal cord.

3.1.2 The Cerebral Hemispheres

The two cerebral hemispheres are the largest parts of the brain and contain many other structures. The hemispheres are incompletely separated by a deep median cleft, named the *longitudinal fissure*, and each possesses a central cavity, termed the *lateral ventricle*.

Each cerebral hemisphere is divided into four lobes, named according to their relation to the four main bones that form the calvaria (the frontal, parietal, occipital, and temporal bones). The anterior convexity of the hemisphere forms most of the *frontal lobe*. The posterior part of the hemisphere is the *occipital lobe*, which lies on the upper surface of the tentorium cerebelli. The *parietal lobe* lies between the frontal and occipital lobes, from which it is demarcated by somewhat arbitrary boundaries. The *temporal lobe* lies on the lateral aspect of the brain, in front of the occipital lobe, projecting forwards below the parietal and frontal lobes.

The longitudinal fissure of the cerebrum contains a sickle-shaped process of dura mater named the *falx cerebri*, and the anterior cerebral vessels. In front and behind,

the fissure completely separates the cerebral hemispheres from each other; in the middle, however, it only extends down to a great central white commissure, named the *corpus callosum*, which connects the hemispheres across the median plane.

3.1.3 The Ventricles

Deep inside the forebrain, midbrain and hindbrain are a series of connection chambers (*ventricles*) lined with an epithelium called ependyma. There are two large lateral ventricles inside the cerebral hemispheres(forebrain), each of which connects in the midline through the interventricular foramen (of Monro) which leads into the midline third ventricle. This connects through the narrow cerebral aqueduct (of Sylvius) in the midbrain to the midline fourth ventricle in the pons and medulla oblongata (hindbrain).

3.1.4 The Brainstem

The brainstem consist of the *midbrain*, the *pons* and the *medulla oblongata*. All of these are midline structures which are overgrown by the cerebral hemispheres during development. Rostrally the brainstem is continuous with the diencephalon. Caudally it bends with the first cervical segment of the spinal cord. Dorsally it is connected to the cerebellum by the superior, middle and inferior cerebellar peduncles.

The brainstem lies on the floor of the cranial cavity. The medulla rests on the

basi-occiput, the pons on the sphenoid bone as far forward as the dorsum sellae. The midbrain passes through the tentorial notch of the tentorium cerebelli.

3.1.5 The Cerebellum

The cerebellum consists of two cerebellar hemispheres united by a central, median vermis. The surface of the cerebellum is deeply folded. Major folds, the fissures, subdivide the cerebellum into superior and inferior halves, and demarcate subdivisions, the anterior, posterior, and flocculonodular lobes within each hemisphere.

3.2 Intracranial Deformation

The intracranial contents consisting of the brain, cerebrospinal fluid, and blood are enclosed in a rigid bony container (the skull). Any increase in the volume of one of these components will, to a certain extent, cause intracranial deformation or distortion, such as lateral shift of the midline structures, internal herniae and displacement of the brain stem (see Fig.3.2). These deformations are called *secondary space occupying effects*. The increase in the volume of the intracranial contents can be caused by various pathological processes within the brain such as tumour, haematoma, or a massive recent cerebral infarct. The increase in the volume of the intracranial contents will ultimately cause an increase in intracranial pressure. Although it has been observed that the distortion and displacement of the brain and the associated

increase in intracranial pressure are significant with regard to the immediate survival of the patient [MacSween 1992], there are no available criteria in current clinical and radiological diagnosis for objective assessment of the severity of such intracranial deformation and its response to treatment. This lack formed the impetus for this Ph.D project.

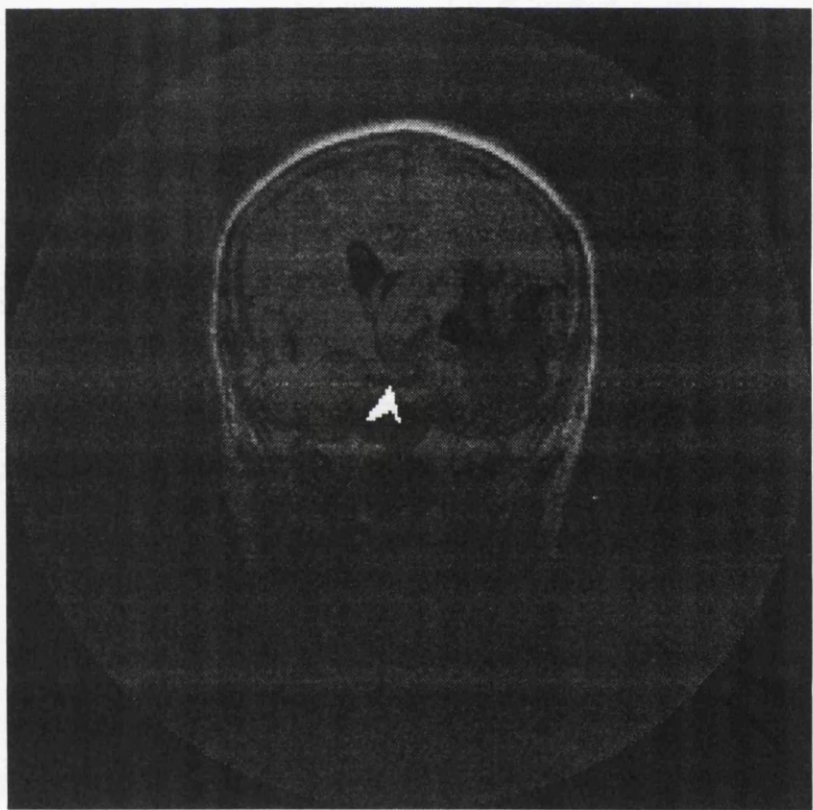


Fig.3.2 The distortion and herniation of the brain caused by a tumour in a cerebral hemisphere. The midline and lateral ventricles are displaced, tentorial herniation (arrow) is produced.

It is well known that different parts of the brain perform different functions. This localisation of function means that the changes in size, position, and shape within the brain will characterise the effects of the intracranial lesions.

3.2.1 Intracranial Expanding Lesions

When a massive lesion in a cerebral hemisphere, usually a tumour, a haematoma, a recent infarct or an abscess, starts to expand, the first deformation is distortion of the adjacent brain. Because of its viscoelastic properties, brain tissue adjacent to the mass will flow away from it. This will cause an axial movement of the brain in addition to the more conventional displacements.

As the lesions expands, so also does the hemisphere. The surface of the brain is pressed against the unyielding dura, gyri are flattened, sulci are progressively narrowed. Cerebrospinal fluid is displaced from the ventricular system with the result that the lateral ventricle on the same side as the lesion becomes smaller while the contralateral ventricle may become larger. Further expansion of the affected hemisphere leads to a shift of the midline structures, *i.e.* the interventricular septum, the anterior cerebral arteries and the third ventricle.

An expanding lesion in the frontal lobe will produce displacement of the free margin of the anterior part of the falx; the posterior part of the falx however is rarely displaced laterally because it is firmly tethered at this level. Furthermore a lesion

in a temporal lobe will produce selectively severe shift of the third ventricle and will displace upwards the Sylvian fissure and the branches of the middle cerebral artery which it contains.

3.2.2 The Internal Hernia

As the result of an intracranial expanding lesion, the *internal hernia*, i.e. displacement of brain tissue from one intracranial compartment into another, then develops. Usually there are four types of hernia associated with an expanding lesion: *tentorial hernia*, *supracallosal hernia*, *central transtentorial hernia*, and *tonsillar hernia*.

1 Tentorial hernia This occurs when the medial part of the ipsilateral temporal lobe is squeezed through the tentorial opening (Fig. 3.2). The herniated brain tissue compresses and displaces the midbrain which is pushed against the contralateral rigid edge of the tentorium. This pressure is often sufficient to produce a distinct groove on the surface of the midbrain.

2 Supracallosal hernia When a supratentorial lesion causes downwards displacements of the roof of the ipsilateral ventricle, the ipsilateral cingulate gyrus will herniate under the free edge of the falx. As a result there may be a displacement of the pericallosal arteries away from the midline (Fig 3.3).

3 Central transtentorial hernia This is brought about by caudal displacement of the diencephalon and the rostral brain stem. It be preceded by a lateral transtentorial hernia, and occurs particularly in response to frontal and parietal lesions or to bilateral expanding lesions.

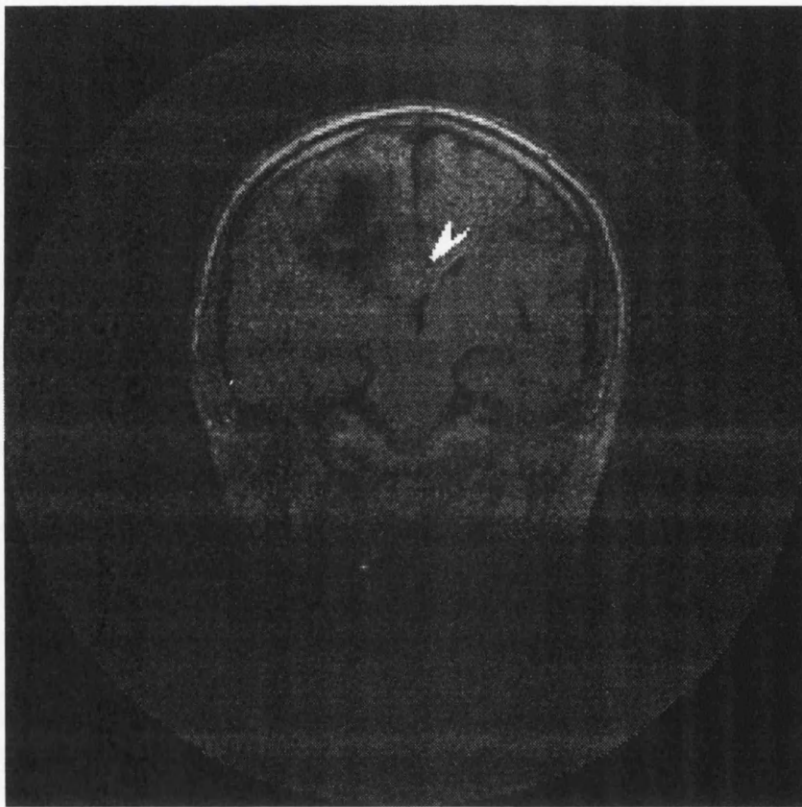


Fig. 3.3 Supracallosal hernia (arrow)

4 Tonsillar hernia The cerebellum and brain stem are forced downwards towards and through the foramen magnum. The medulla becomes compressed and the perfusion of the whole brain stem is in peril. The end result of unrelieved brain compression is respiratory arrest.

3.2.3 Brain Swelling

An increase in cerebral blood volume or in the water content of brain tissue will cause the expansion of brain tissue. This process is called *brain swelling*. The oedema fluid is mainly interstitial and the cut surface of the brain appears pale and swollen.

3.2.4 The Deformation Caused by Brain Tumour

Intracranial tumours may produce local effects which will depend upon their site, *e.g.* focal epilepsy, hemianopia, and they also behave as expanding intracranial lesions leading to a raised intracranial pressure. The effective size of the tumour is frequently contributed to by oedema in adjacent brain. This usually responds dramatically to steroid therapy.

A slowly growing tumour such as a meningioma may be accommodated by demyelination of central white matter, which thus loses bulk while keeping most of its axons intact. A fast growing tumour, on the other hand (such as glioblastoma, or secondary carcinoma) may enhance its deadly potential by causing an oedematous reaction in the surrounding brain tissue. With some tumours, notably with gliomas of the astrocytic series, space occupation is as much due to the formation of an associated cyst as it is to expansion of the solid tumour tissue.

The tumours most commonly met with are *gliomas*. The commonest of these is the fast growing malignant *glioblastoma*. It occurs mainly in adults in the cerebral

hemispheres and is rapidly growing with extensive necrosis and haemorrhage. This produces considerable distortion of the brain and often a rapid increase of intracranial pressure.

Chapter 4 Image Segmentation

4.1 Introduction

Before measuring a cerebral structure (two-dimensional, or three-dimensional) in an MR image or a sequence of MR images, we have to segment the object from the background. In other words we have to define the object or the object region. The reliability of subsequent measurements are obviously dependent upon image segmentation. The segmentation problem for MRI data depends on a large number of factors. Image contrast, signal-to-noise ratio, slice thickness, complexity of the scene, data set size, and radio frequency coil uniformity are just a few of the important ones. Based on some or all of these factors, an approach must be selected for the segmentation task.

A range of approaches has been proposed for semi-automatic or automatic detection of various structures in the head, with varying levels of automation and practical

applicability. MRI's ability to generate different contrasts has been exploited by analysis techniques involving multispectral MRI data [Windham 1988, Gerig 1991]. Hu *et al* [Hu 1990] report on an interactive gradient-guided edge-tracking technique, operating on individual image slices. Kennedy *et al* [Kennedy 1989] describe an interactive algorithm using intensity contours, a variant of thresholding. Pizer *et al* [Pizer 1990] designed an algorithm which divides the image into a hierarchy of elementary regions using intensity extrema and, initially for 2-D images, coupled this algorithm to an interactive region editor. Bomans *et al* [Bomans 1990] extend the Marr-Hildreth edge operator to 3-D, and use this to optimize tissue border locations. Brummer *et al* [Brummer 1993] designed a procedure for automated detection of brain contours from 3-D MRI data. This procedure first detects structures in the head using histogram-based thresholding. This is then followed by a morphological procedure which refines the binary threshold mask images.

All these approaches can be divided into two categories, thresholding and boundary detection. In the thresholding technique the grey value of each pixel is compared to a *threshold* value, and the pixel is assigned to one of two categories depending on whether the threshold is exceeded or not. The selection of the threshold value is usually made from a histogram. The problem is that in MR images of the brain the brightness or grey value at the boundary of an object may vary from point to point. It is thus not always possible to identify the threshold value for the boundary of the

object, so the thresholding approach may produce a false or inaccurate segmentation. This is particularly true of most brain tumours where clear cut-off boundaries are rarely present.

Boundary detection techniques may produce more accurate segmentation than the thresholding method does. This is because boundary detection approaches segment objects based on the local changes of grey level at the boundary. All the boundary detection approaches that we have mentioned above have shown satisfactory results in 3-D rendering of MRI brain data, but not yet in quantitative measurement because of the unsatisfactory accuracy in the segmentation [Brummer 1993].

In fact there are a lot of methods for boundary detection [Jain 1989, Ballard 1982]. Canny's edge detector [Canny 1986] is probably one of the best detectors [Lee, 1990] in the sense of low probability of failing to mask true edge points and minimising the distance between the detected edge and the true edge. Surprisingly, we have not found any application of Canny's method in the current literature about brain structure segmentation.

In this chapter an attempt will be made to develop a successful approach to segmenting the boundaries and surfaces of the brain structures in MR images. The overall approach is outlined in Fig.4.1. The new segmentation technique combines the advantages of the accuracy of Canny's edge detector and the analytical compactness of the B-splines method. It has the advantage over current MR segmentation techniques

in terms of accuracy and speed. We first, in section 4.2, introduce Canny's edge detector which is one of the most popular edge detection techniques in modern image processing and computer vision. In section 4.3 we introduce B-splines segmentation methods and investigate curve and surface representation with B-splines. Then in section 4.4 we apply, for the first time, Canny's detector and B-splines methods to MR image segmentation [Dai 1992], using them to segment the boundaries and surfaces of brain structures in MR images. Finally we will give our conclusions about these segmentation approaches.

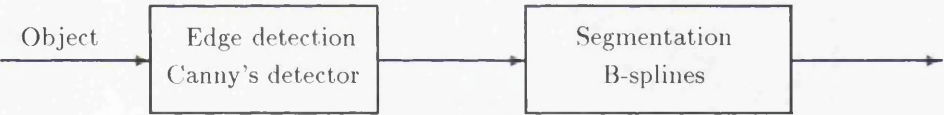


Fig.4.1 The Basic scheme for segmentation

4.2 Edge Detection Using Canny's Detector

An edge is the boundary between two regions with relatively distinct grey level (brightness) properties. Basically, the idea underlying most edge-detection techniques is the computation of a local derivative operator. If we take a cross section of the image grey level along a line at right angles to an edge, we might hope to see a step discontinuity. For MR images the transition will not be abrupt because of the nature

of tissues and the limitations of the imaging system. Image blurring and noise make it very difficult to define the true edge.

There are two common criteria relevant to edge detector performance [Canny 1986]. The first and the most obvious is low error rate. It is important that edges that occur in the image should not be missed. The second criterion is that the edge points be well localised. In other words that the distance between the points marked by the detector and the “centre” of the true edge should be minimized.

4.2.1 Laplace Operators and Zero Crossings

As the transition region gets wider, it is more advantageous to apply the second-order derivatives. One frequently encountered operator is the *Laplacian operator*.

For a given image $f(x, y)$, the Laplacian operator of f is defined as :

$$\nabla^2 f = \partial^2 f / \partial x^2 + \partial^2 f / \partial y^2$$

Because of the second-order derivatives this gradient operator is more sensitive to noise. Also the thresholded magnitude of $\nabla^2 f$ produces double edges. For these reasons, together with its inability to detect the edge direction, the Laplacian as such is not a good edge detection operator. Better utilisation of the Laplacian is to use its *zero-crossing* to detect the edge locations. A generalised Laplacian operator which approximates the Laplacian of Gaussian functions is a useful zero-crossing detector [Marr 1980]. It is defined as

$$h(x, y) = c(1 - \frac{x^2 + y^2}{\sigma^2})exp(-\frac{x^2 + y^2}{2\sigma^2})$$

where σ controls the width of the Gaussian kernel and c normalises the sum of the elements of a given mask to unity. Zero-crossings of a given image convoluted with $h(x, y)$ give its edge locations. On a two-dimensional grid a zero-crossing is said to occur wherever there is a zero-crossing in at least one direction.

4.2.2 Canny's Detector

Canny's edge detector is one of the best detectors in the sense of low probability of failing to mask true edge points and minimising the distance between the detected edge and the true edge. Because of noise there are many zero-crossings from the Marr-Hildreth type filter response which do not correspond to edges. Canny's detector aims to minimise such spurious responses.

Here we define contour points as those points for which the convolution of the initial image with a certain function (*filter*) h has a local maximum (this local maximum is computed in the direction of the grey level gradient). In practice, h is chosen to approximate a grey level gradient computation. Canny has shown that h could be defined using optimality criteria for the detection of contours.

In Canny's approach, boundaries are modelled in one dimension by an amplitude threshold A to which is added zero-mean Gaussian noise n with constant variance n_0^2 ,

giving:

$$I(x) = Au_{-1}(x) + n(x)$$

where

$$u_{-1}(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{otherwise} \end{cases}$$

An image line is thus convoluted by the function $h(x)$, giving a resulting image line

$$I'(x) = I * h(x)$$

The local maximum of $I'(x)$ defines the edge points. The criteria for evaluating boundary quality are the following:

Detection It must be robust to noise, an unavoidable condition to ensure a low probability of failing to detect edge points. Detection quality is measured quantitatively as the ratio of the response obtained at the edge point in the absence of noise to the square root of the average response of the noise squared:

$$C_1 = \frac{A}{n_0} \sum$$

where

$$\sum = \frac{|\int_{-\infty}^0 h(x)dx|}{\sqrt{\int_{-\infty}^{+\infty} h^2(x)dx}}$$

\sum is preferred to C_1 in order to remove the term $\frac{A}{n_0}$ which is image-dependent.

Thus $\sum(h)$ depends only on the filter h .

Localisation It should be as precise as possible in identifying the point to the centre of the true edge. Localisation quality is measured as the inverse of the variance σ^2 in the position of the maximum of $I'(x)$:

$$C_2 = \frac{A}{n_0} \Lambda$$

where

$$\Lambda = \frac{|h'(0)|}{\sqrt{\int_{-\infty}^{+\infty} h'^2(x) dx}}$$

Λ is preferred to C_2 in order to eliminate the term $\frac{A}{n_0}$ which depends on the image. Thus $\Lambda(h)$ depends only on the filter h .

Uniqueness Multiple responses should be avoided in the neighborhood of a single edge. We thus impose a value on the average distance between two maximum of $I'(x)$. This value is expressed as:

$$C_3 = 2\pi \sqrt{\frac{\int_{-\infty}^{+\infty} h'^2(x) dx}{\int_{-\infty}^{+\infty} h''^2(x) dx}}$$

Resolution Once these partial criteria are defined, Canny defines a global criterion which consists of maximising the product $\sum \Lambda$ - invariant to scale changes—subject to the constraint C_3 .

The solution is of the form:

$$\begin{aligned} h(x) = & a_1 e^{\alpha x} \sin(\omega x) + a_2 e^{\alpha x} \cos(\omega x) \\ & - a_3 e^{-\alpha x} \sin(\omega x) - a_4 e^{-\alpha x} \cos(\omega x) + C \end{aligned}$$

where the coefficients a_i are determined by the boundary condition. The function $h(x)$ is subsequently approximated by the first derivative of a Gaussian:

$$h(x) = -\frac{x}{\sigma^2} e^{-x^2/2\sigma^2}$$

In two dimensions, the solution proposed by Canny amounts to convolving the initial image with a symmetric Gaussian impulse response followed by computation of the derivative in x and y of the result.

The technique is extended to two dimensions by utilising the fact that the derivative of $I(x, y)$ in an arbitrary direction can be obtained from derivatives in the x and y directions.

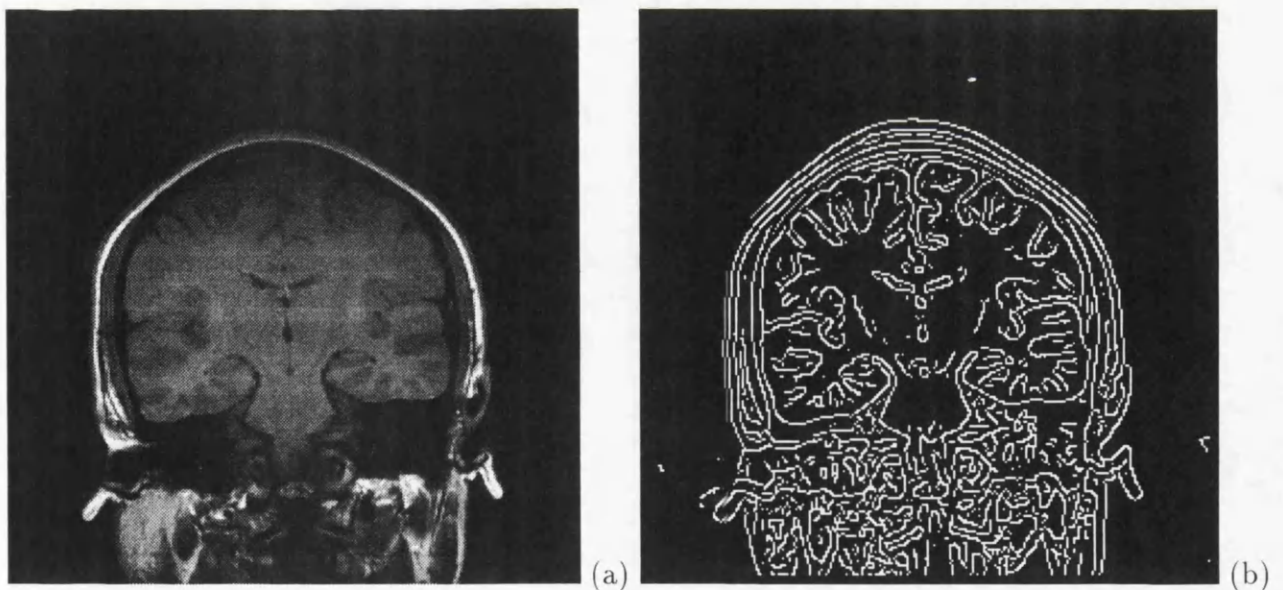


Fig.4.2 (a) A 256×256 coronal section of T_1 weighted image. (b) Edges from (a) at $\sigma = 2.0$.

An example of Canny's detector is given in Fig 4.2. Fig. 4.2 (a) is a 256×256 coronal section of a T_1 weighted image. Fig. 4.2 (b) shows the edges from (a).

4.3 Splines Segments

Splines provide a very useful way of generating smooth curves and surfaces from a small number of control points, thus saving space in the database at the expense of increased computation time. They also provide an effective way of modeling irregular shaped objects.

4.3.1 Curve Representation Using B-splines Approximation

The theory for B-splines was first suggested by Schoenberg [Schoenberg 1946]. A recursive definition useful for numerical computation was independently discovered by Cox [Cox 1971] and by de Boor [de Boor 1972]. Gordon and Riesenfeld [Gordon 1974] applied the B-splines basis to curve definition.

4.3.1.1 B-splines

Let $P(\eta)$ be the position vectors along the curve as a function of the parameter η . A B-spline curve is given by

$$P(\eta) = \sum_{i=0}^n P_i B_{i,k}(\eta)$$

where $\eta_{min} \leq \eta < \eta_{max}$, $1 \leq k \leq n$, the \mathbf{P}_i are the position vectors of the $n + 1$ defining polygon vertices and the $B_{i,k}$ are the normalized B-spline basis function.

For the i th normalized B-spline basis function of order k , the basis functions $B_{i,k}$ are defined by the Cox-deBoor recursion formulas. Specifically,

$$B_{i,1}(\eta) = \begin{cases} 1 & \text{if } x_i \leq \eta < x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

and

$$B_{i,k}(\eta) = \frac{(\eta - x_i)B_{i,k-1}(\eta)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - \eta)B_{i+1,k-1}(\eta)}{x_{i+k} - x_{i+1}}$$

The values of x_i are elements of a knot vector satisfying the relation $x_i \leq x_{i+1}$. The parameter t varies from η_{min} to η_{max} along the curve $P(\eta)$. The convention $\frac{0}{0} = 0$ is adopted.

Formally a B-spline curve is defined as a polynomial spline function of order k (degree $k - 1$) since it satisfies the following two conditions :

- The function $P(\eta)$ is a polynomial of degree $k - 1$ on each interval $x_i \leq \eta < x_{i+1}$.
- $P(\eta)$ and its derivatives of order 1, 2, ..., $k - 2$ are all continuous over the entire curve.

Thus, for $-2 \leq \eta < 2$, $k = 4$, and $x_{i+1} - x_i = 1$, we have

$$B(\eta) \equiv B_{0,4}(\eta) = \begin{cases} 1/6(2 + \eta)^3 & -2 \leq \eta \leq -1 \\ 1/6(4 - 6\eta^2 - 3\eta^3) & -1 \leq \eta \leq 0 \\ 1/6(4 - 6\eta^2 + 3\eta^3) & 0 \leq \eta \leq 1 \\ 1/6(2 - \eta)^3 & 1 \leq \eta \leq 2 \\ 0 & 2 \leq |\eta| \end{cases} \quad (4.1)$$

where $B(\eta)$ is called the *spline blending function*. Alternatively, a cubic B-spline curve may be defined as

$$\mathbf{P}(\mu) = \sum_{i=0}^n \mathbf{P}_i B(n\mu - i)$$

As the parameter μ takes on values in a specified range, usually 0 to 1, the function \mathbf{P} traces out the location of the curve.

4.3.1.2 Interpolation Using B-splines

B-splines may be used to interpolate data points instead of simply passing close to them.

Here we consider the case where $n + 1$ geometric knots are to be interpolated. They will be denoted \mathbf{P}_0 to \mathbf{P}_n .

Instead of using the actual knots, \mathbf{P}_i , in the spline equation $n + 3$ *phantom* knots will be used, denoted \mathbf{V}_{-1} to \mathbf{V}_{n+1} . The extra two knots are to allow flexibility in setting the gradient at the ends of the spline.

The equation of the interpolating spline is

$$\mathbf{P}(\mu) = \sum_{i=-1}^{n+1} B(n\mu - i)\mathbf{V}_i \quad (4.2)$$

Let $\mu = 0, 1/n, 2/n, \dots, 1$ and substitute the geometric knots on the right-hand side of this equation with the corresponding μ in the left-hand side, a set of equations is obtained.

$$\mathbf{P}(i/n) = \mathbf{P}_i = 1/6\mathbf{V}_{i-1} + 2/3\mathbf{V}_i + \mathbf{V}_{i+1}$$

This produces $n + 1$ equations. However, there are $n + 3$ unknown values of \mathbf{V}_i . The other two equations come from placing a further constraint on the ends of the curve, which in this case will be to fix the gradient at the ends. The gradients at the ends will be denoted by the direction vectors \mathbf{g}_0 and \mathbf{g}_n . To compute them, the spline equation 4.2 needs to be differentiated with respect to μ . This yields:

$$\mathbf{P}'(\mu) = n \sum_{i=-1}^{n+1} B'(n\mu - i)\mathbf{V}_i \quad (4.3)$$

By substituting the values $\mu = 0$ and $\mu = 1$ into equation 4.3,

$$\mathbf{P}'(0) = \mathbf{g}_0 = \frac{n}{2}(\mathbf{V}_1 - \mathbf{V}_{-1})$$

and similarly:

$$\mathbf{P}'(1) = \mathbf{g}_n = \frac{n}{2}(\mathbf{V}_{n+1} - \mathbf{V}_{n-1})$$

since the gradient at the end is an input, or alternatively it can be set to some default value. The equations may be neatly summarized by writing them in matrix

form

$$MV = A$$

where

$$M = \begin{pmatrix} -n/2 & 0 & n/2 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1/6 & 2/3 & 1/6 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1/6 & 2/3 & 1/6 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1/6 & 2/3 & 1/6 & 0 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 2/3 & 1/6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1/6 & 2/3 & 1/6 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & -n/2 & 0 & n/2 \end{pmatrix}$$

$$\mathbf{V} = \begin{pmatrix} \mathbf{A}_{-1} \\ \mathbf{A}_o \\ \cdots \\ \mathbf{A}_n \\ \mathbf{A}_{n+1} \end{pmatrix}$$

$$\mathbf{A} = \begin{pmatrix} \mathbf{g}_o \\ \mathbf{P}_o \\ \dots \\ \mathbf{P}_n \\ \mathbf{g}_n \end{pmatrix}$$

where the vector \mathbf{V} contains the phantom knots, \mathbf{A} contains the geometric knots and the gradients at the ends, and \mathbf{M} is a matrix that depends only on the number of knots, n .

The solution can be found, for example, by Gaussian elimination. Since the matrix does not depends on the values of the knots, a more direct method is to invert the matrix, which can be precalculated for a number of n values. If the inverted matrix for a given n is available, then the values of the phantom knots may be calculated by matrix and vector multiplication:

$$\mathbf{V} = \mathbf{M}^{-1} \mathbf{A}$$

4.3.2 Surface Representation Using Cubic B-splines Approximation

There are two ways in which spline surfaces can be constructed. The first is called *interpolation* in which the surface passes through each knot. In the second method, called *approximation*, the surface does not necessarily pass through the knots but instead passes close to or through them. Interpolation is more accurate whilst approximation is easier to compute.

If $(n + 1)(m + 1)$ control points are given,

$$\begin{array}{ccccccccc} P_{0,0} & P_{0,1} & \cdots & P_{0,m-1} & P_{0,m} \\ P_{1,0} & P_{1,1} & \cdots & P_{1,m-1} & P_{1,m} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ P_{n-1,0} & P_{n-1,1} & \cdots & P_{n-1,m-1} & P_{n-1,m} \\ P_{n,0} & P_{n,1} & \cdots & P_{n,m-1} & P_{n,m} \end{array}$$

the B-spline surface $\mathbf{P}(\mu, v) = \mathbf{P}(x(\mu, v), y(\mu, v), z(\mu, v))$ can be written as :

$$\mathbf{P}(\mu, v) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{P}_{i,j} B(n\mu - i)B(mv - j) \quad (4.4)$$

As the parameters μ and v take on values in a specified range, usually 0 to 1, the parametric functions x , y and z trace out the location of the surface. In Equation 4.4, $B(\eta)$ is the B-spline blending function defined in Equation 4.1.

In Equation 4.4 the B-spline surface does not necessarily pass through the knots. This way of computing the spline produces good results in the centre of the surface,

but unfortunately does not produce good results at the edge points. This can be seen by substituting the values $\mu = 0$ and $v = 0$ into Equation 4.4, yielding

$$\mathbf{P}(0,0) = 4/9\mathbf{P}_{0,0} + 1/9\mathbf{P}_{0,1} + 1/9\mathbf{P}_{1,0} + 1/36\mathbf{P}_{1,1} \neq \mathbf{P}_{0,0}$$

Furthermore, B-splines may be used to interpolate the knot points fully instead of simply passing close to them. In order to achieve this a set of phantom ,or *parametric knots* is introduced. Suppose the equation of the interpolating spline is

$$\mathbf{P}(\mu, v) = \sum_{i=-1}^{n+1} \sum_{j=-1}^{m+1} \mathbf{V}_{i,j} B(n\mu - i)B(mv - j) \quad (4.5)$$

Note that there are two extra rows and columns in the parametric knots. The extra knots are to allow flexibility in setting the gradient at the edges of the splines. They may be determined by the following steps.

If the surface is to pass through all the parametric knots the following equation is obtained:

$$\mathbf{P}\left(\frac{i}{n}, \frac{j}{m}\right) = \begin{pmatrix} \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{i-1,j-1} & \mathbf{V}_{i-1,j} & \mathbf{V}_{i-1,j+1} \\ \mathbf{V}_{i,j-1} & \mathbf{V}_{i,j} & \mathbf{V}_{i,j+1} \\ \mathbf{V}_{i+1,j-1} & \mathbf{V}_{i+1,j} & \mathbf{V}_{i+1,j+1} \end{pmatrix} \begin{pmatrix} \frac{1}{6} \\ \frac{2}{3} \\ \frac{1}{6} \end{pmatrix} = \mathbf{P}_{i,j} \quad (4.6)$$

for $i = 0, 1, \dots, n$ and $j = 0, 1, \dots, m$. There are in total $(m+1)(n+1)$ linear equations from Equation 4.6. However, there are $(m+3)(n+3)$ unknown values of $\mathbf{V}_{i,j}$. In order to solve the equations extra constants are introduced which in this case

will fix the gradient at the edge points. This is known as a *clamped end condition* and can be used to determine the shape at the ends of a spline or to piece together two splines without introducing a discontinuity in the gradient. The gradient of the surface must be specified with respect to μ at the top and bottom edges, where top and bottom refer to the way the tensor has been arranged in Equation 4.5, and at the left and right edges with respect to v . In order to find the gradient with respect to μ , the partial derivative of the tensor, Equation 4.5 with respect to μ must be taken, namely:

$$\frac{\partial \mathbf{P}}{\partial \mu} = n \sum_{i=-1}^{n+1} \sum_{j=-1}^{m+1} \mathbf{V}_{i,j} B'(n\mu - i) B(mv - j)$$

Thus at the edge points where $\mu = 0$, for example $\mathbf{P}_{0,j}$, the gradient at $\mu = 0$, $v = \frac{j}{m}$, is:

$$\frac{\partial \mathbf{P}}{\partial \mu} = \begin{pmatrix} -\frac{n}{2} & 0 & \frac{n}{2} \end{pmatrix} \begin{pmatrix} \mathbf{V}_{-1,j-1} & \mathbf{V}_{-1,j} & \mathbf{V}_{-1,j+1} \\ \mathbf{V}_{0,j-1} & \mathbf{V}_{0,j} & \mathbf{V}_{0,j+1} \\ \mathbf{V}_{1,j-1} & \mathbf{V}_{1,j} & \mathbf{V}_{1,j+1} \end{pmatrix} \begin{pmatrix} \frac{1}{6} \\ \frac{2}{3} \\ \frac{1}{6} \end{pmatrix} \quad (4.7)$$

The other partial derivatives are evaluated in the same way, giving a set of $2(m+1)$ equations at the top and bottom edges and $2(n+1)$ equations at the left and right

edges. Let

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_{-1,-1} & \mathbf{V}_{-1,0} & \cdots & \mathbf{V}_{-1,m} & \mathbf{V}_{-1,m+1} \\ \mathbf{V}_{0,-1} & \mathbf{V}_{0,0} & \cdots & \mathbf{V}_{0,m} & \mathbf{V}_{0,m+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{V}_{n,-1} & \mathbf{V}_{n,0} & \cdots & \mathbf{V}_{n,m} & \mathbf{V}_{n,m+1} \\ \mathbf{V}_{n+1,-1} & \mathbf{V}_{n+1,0} & \cdots & \mathbf{V}_{n+1,m} & \mathbf{V}_{n+1,m+1} \end{pmatrix}$$

$$\mathbf{B}_n = \begin{pmatrix} -n/2 & 0 & n/2 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1/6 & 2/3 & 1/6 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1/6 & 2/3 & 1/6 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1/6 & 2/3 & 1/6 & 0 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 2/3 & 1/6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1/6 & 2/3 & 1/6 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & -n/2 & 0 & n/2 \end{pmatrix}$$

$$B_m = \begin{pmatrix} -m/2 & 0 & m/2 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1/6 & 2/3 & 1/6 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1/6 & 2/3 & 1/6 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1/6 & 2/3 & 1/6 & 0 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 2/3 & 1/6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1/6 & 2/3 & 1/6 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & -m/2 & 0 & m/2 \end{pmatrix}$$

$$A = \begin{pmatrix} D_{-1,-1} & D_{-1,0} & \cdots & D_{-1,m} & D_{-1,m+1} \\ D_{0,-1} & P_{0,0} & \cdots & P_{0,m} & D_{0,m+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ D_{n,-1} & P_{n,0} & \cdots & P_{n,m} & D_{n,m+1} \\ D_{n+1,-1} & D_{n+1,0} & \cdots & D_{n+1,m} & D_{n+1,m+1} \end{pmatrix}$$

where $D_{i,j}$ is the edge gradient which may be set to a default such as the forward difference. Then from Equation 4.6 and 4.7, the following equation can be obtained:

$$A = B_n V B_m$$

So the parametric knots $V_{i,j}$ can be found:

$$V = B_n^{-1} A B_m^{-1} \quad (4.8)$$

Equation 4.8 may be solved by means of Gaussian elimination [Press 1992, Ralson 1978].

4.4 Applications

The brain structures which we are interested in are the hemispheres, the longitudinal fissure and the ventricles. In this section we will show how to segment these structures.

4.4.1 The Hemispheres and the Ventricles in each Sectional Image

First the Canny's edge detector is applied to each sectional image, producing the boundary profiles of these structures, as shown in Fig. 4.2.



Fig.4.3 31 control points are chosen in each of the hemisphere.

Using the profiles as guidelines to the boundary of each hemisphere, 31 control points are interactively chosen along the boundary profiles of the hemisphere (Fig. 4.3).

Based on the control points, a B-splines representation of the hemisphere contour is generated (Fig. 4.4). Both the left and right closed curves in Fig. 4.4 pass exactly through the control points in Fig. 4.3 and are generated by using Eq. 4.8. The number of control points dominates the precision of the approximating splines. From the experience of applying this to more than 80 different hemispheres, we found that 31 control points was a reasonable compromise between precision and operator time.



Fig.4.4 The B-splines representation of each hemispheric contour.

Similarly, the B-splines representation of the ventricular contour can be produced with 15 control points (Fig.4.5).

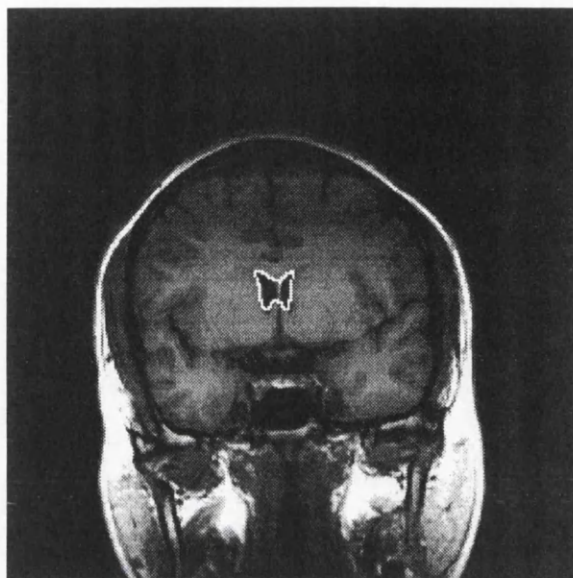


Fig.4.5 The B-splines representation of the ventricular contour.

4.4.2 The Surface of the Falx Cerebri

In order to study the falx cerebri or the longitudinal, we examined the slices between the genu and splenium of the corpus callosum (see Fig 4.6, typically 8 to 11 slices).

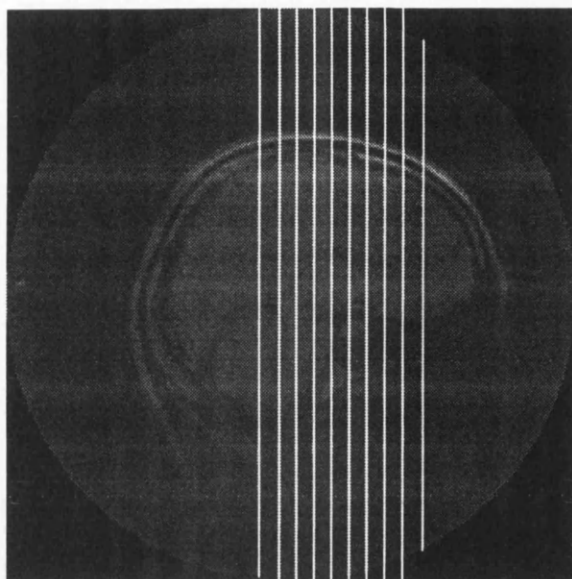


Fig.4.6 An example of chosen slices between the genu and splenium of the corpus callosum.

In order to obtain the B-splines surface of the longitudinal fissure, 8 control points are picked up along the profiles of the longitudinal fissure in each slice (see Fig. 4.7).

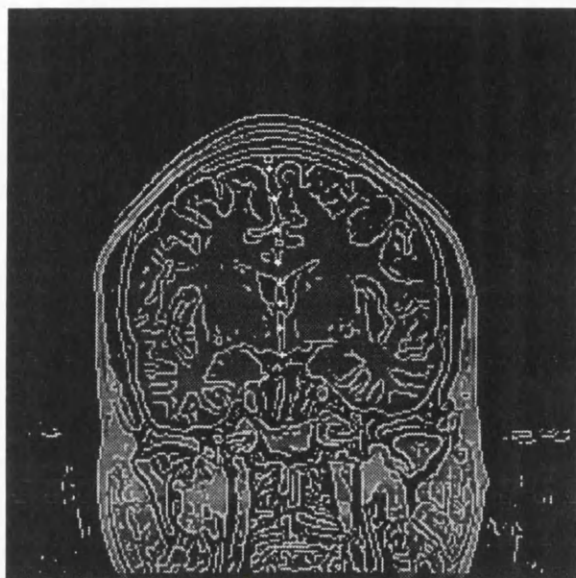


Fig.4.7 8 control points are chosen along the profiles of the longitudinal fissure in each sectional image.

Fig. 4.8 shows all the control points from 11 contiguous slices between the genu and splenium of the corpus callosum.

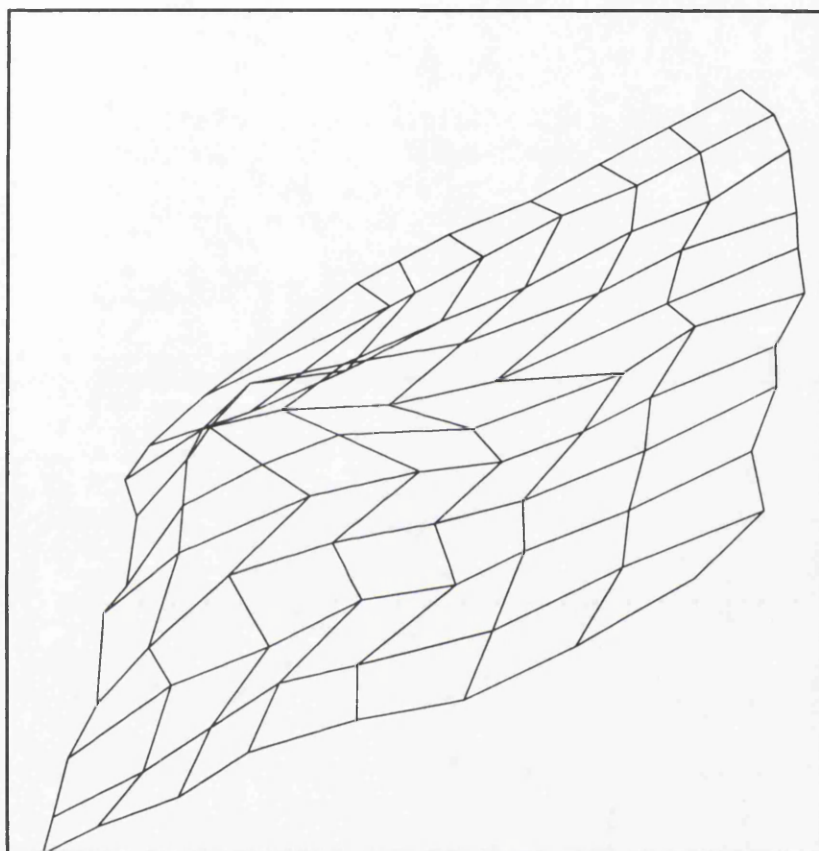


Fig.4.8 A surface representation of a falx cerebri. The 8 knots along the bottom oblique line are obtained from the first of the 11 contiguous slices. The knots along the top oblique line correspond to the points in the last slice.

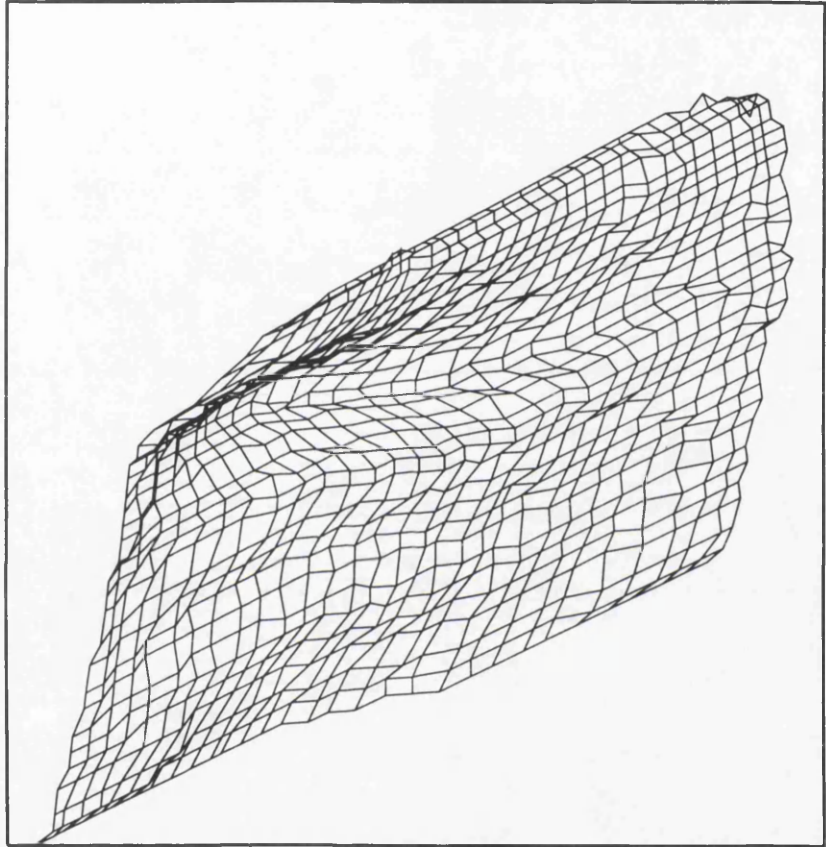


Fig.4.9 B-splines surface through the 8 control points in Fig. 4.8.

The surface shown in Fig. 4.9 is the B-splines surface generated by the control points from Fig. 4.8. The B-splines surface passes exactly through the original control points in Fig. 4.8 and is generated by Eq. (4.8), where the phantom edge gradients

D_{ij} are set equal to 0.

4.5 Conclusions

In this chapter an effective and practical technique has been developed to segment the boundaries and surfaces of brain structures in MR images. This technique is designed specially for quantitative measurement of brain structures. Canny's edge detector was employed to segment brain structure. It provides the optimal trade off between signal to noise ratio and accuracy in locating edges. The boundary profiles provided by Canny's detector are, however, disconnected. A further procedure is therefore needed in order to determine the exact measurement region and so we interactively pick up boundary points in each slice (31 for the hemisphere, 15 for the lateral ventricles and 8 for the falx cerebri) using the boundary profiles generated by Canny's edge detector as a guide. Using these points as the control points, we can produce a parametric representation of the boundary of an intracranial object. Comparing this with surfaces and interfaces displayed in the original raw images, our parametric representation of boundaries has the following advantages:

- The parametric representation provides meaningful information about object shape, in term of such things as curvature and symmetry. We will see in the following chapters that this sort of information is essential in shape analysis. The raw data by itself does not provide this information.

- The parametric representation is economical in terms of space. For example, the parametric representation of the left hemisphere contours in a stack of 24 images with an image matrix of 256×256 takes less than 750 bytes, compared with the raw data representation which takes 1,572,864 bytes.

Interactively choosing the control points from the boundary profiles is however time consuming. In this analysis, it takes a total of about 15, 5 and 4 minutes respectively to choose the control points for the hemispheric contours (24 slice images), ventricular contours (3 or 4 slice images) and the falx cerebri (typically 10 slice images). We believe this is acceptable for quantitative analysis. Due to the limitations in image quality in current MR imaging systems, it is unlikely that a fully automatic segmentation method to segment brain structures with satisfactory accuracy and feasibility could be designed. Brummer *et al* [Brummer 1993] recently developed a fully automatic segmentation technique for 3-D MRI data to aid in the visualisation and volumetric analysis of structures. However, as they said, “our automatic segmentation procedure satisfies neither requirement”. We believe that any practical system for segmentation of MRI data will be incomplete without some interactive capability because of such things as inherent signal non-uniformity and partial volume effects.

We have computerised our new segmentation procedure based on the techniques described in this chapter. The new software package implements both image processing (Canny’s detector) and image segmentation (B-splines). On a Sparc platform, the

processing time for Canny's edge detection is typically 3.5 min for a 24-slice image set and the processing time for curve fitting for the hemisphere contours is about 3 min for a 24-slice image set. It takes about 35 sec and 15 sec for curve fitting for the falx cerebri and the lateral ventricles respectively.

Chapter 5 Measurement of the Hemispheres

5.1 Introduction

Conventional methods for quantitative analysis of MRI scans usually consider changes in the size of brain structures or brain tissues, such as the volume of brain [Condon 1988], cerebrospinal fluid (CSF), and the ventricles [Condon 1986]. A semi-automated computerised method of *in vivo* quantitative analysis with MRI scans has been developed by Filipek *et al.* This method provides volume measurement of the cerebral hemispheres, ventricular system and cerebellum [Filipek 1989]. Kohn *et al* [Kohn 1991, Tanna 1991] developed a computerised system to process standard spin-echo MRI data for the estimation of brain parenchyma and CSF volumes. In their study, the quantitative index of brain and CSF volumes (total, extra-ventricular, ventricular, and third ventricular) associated with ageing and Alzheimer dementia was studied.

These works have furthered our understanding of the pathological processes of aging and dementia.

However, there has been no further progress in the quantitative assessment of the effects of brain tumours since the introduction of MRI. As we mentioned in Chapter 1, measurement of brain tumours with CT does not provide a reliable index of the effects of brain tumours because of the poorly defined region of the tumour. MRI images provide higher contrast between brain structures, such as grey and white matter than CT images, but still cannot reliably define tumour boundaries.

In this chapter we will develop, for the first time, techniques to quantitatively assess the severity of brain tumours and the effectiveness of treatment. Instead of considering error prone estimates of tumour mass, we measure hemisphere deformations caused by brain tumours, the so called the *secondary space occupying effects*. To our knowledge such a quantitative approach has not been attempted before. The measurements in our study are based on the information from the boundaries of the hemispheres. In MR images hemispheric boundary can be delineated much more easily than tumour boundary. Besides the conventional size measurement methods, we introduce new size independent parameters to describe the hemispheric deformation, such as the symmetry and compactness of the hemispheres. These size independent indices are very useful because they overcome the problems caused by wide variations in brain size within the population.

In section 5.2 we first discuss basic relationships between pixels, because many geometric concepts that are well defined for analogue pictures have no counterparts in discrete images (sets of pixels). Therefore, it is necessary to redefine these terms for discrete images. In section 5.3 we will define the basic geometric parameters, such as area, perimeter, volume and the central moments, for such images. Then in section 5.4, we will use these parameters to characterise the deformation of the hemispheres. The results from techniques described in this chapter form the basis of a paper which is in preparation.

5.2 Basic Relationships Between Pixels

Let D be a function which maps pairs of points into non-negative numbers. D is called a *distance* function if for all points p, q, r :

1.

$$D(p, q) = 0 \text{ if and only if } p = q,$$

2.

$$D(p, q) = D(q, p),$$

3.

$$D(p, r) \leq D(p, q) + D(q, r).$$

For example, the following functions

$$E((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2},$$

$$T((x_1, y_1), (x_2, y_2)) = |x_1 - x_2| + |y_1 - y_2|,$$

and

$$C((x_1, y_1), (x_2, y_2)) = \max(|x_1 - x_2|, |y_1 - y_2|)$$

are *distance functions*. E is called the *Euclidean distance*, T is called *City block distance*, and C is called *Chess-board distance* [Gonzalez 1990].

Neighbour Two pixels are said to be *direct neighbours* (D-neighbours) if the respective cells share a side, and *indirect neighbours* (I-neighbours) if these cells touch only at a corner. The name *neighbour* denotes either type. A pixel p at coordinates (x, y) has four D-neighbours:

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1),$$

and four I-neighbours:

$$(x + 1, y + 1), (x - 1, y + 1), (x + 1, y - 1), (x - 1, y - 1).$$

The set of all the neighbours of p is denoted $N(p)$.

Path A *path* is a sequence of distinct pixels p_1, p_2, \dots, p_n such that for $k > 1$, p_{k-1} is a neighbour of p_k and for $k < n$, p_{k+1} is a neighbour of p_k . A *closed path* is one where the first and last pixels coincide.

Connectivity A set of pixels S is connected if for every pair of pixels p and q in S , there is a *path* whose first and last elements are p and q respectively and all its other pixels belong to S . The boundary of S is a closed path of a connected pixel set S' , with each boundary point having at least one I-neighbour.

The boundary chain code A boundary chain code starts by specifying the x, y coordinates of an arbitrarily selected starting point p on the boundary. The pixel p has eight neighbours as shown in Fig 5.1. The numbers in these neighbours are called *direction code*. The chain code assigns the direction according to the direction codes in which a step must be taken to go from the present to the next boundary points. Therefore, the boundary chain code consists of the starting point, followed by the sequences of direction codes that specify the path around the boundary.

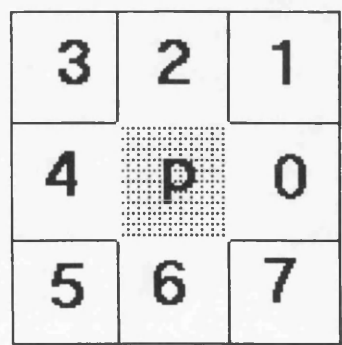


Figure 5.1 The boundary direction code

5.3 Basic Geometric Parameters

5.3.1 Perimeter

Perimeter is the circumferential distance around the boundary. The perimeter is easily obtained from the boundary chain code. It is also simple to compute from the object segment file, provided one is careful to compute accurately the centre-to-centre distance between adjacent pixels on the boundary. Image noise usually produces artificial jaggedness in the object boundary. This generally combines with sampling grid effects to make perimeter measurements artificially large. However, the boundary obtained from the B-splines segmentation method is free of the noise problem. In our case, the perimeter of an object is given by

$$\mathcal{P} = N_e + \sqrt{2}N_o$$

where N_e is the number of even and N_o is the number of odd steps in the boundary chain code.

5.3.2 Area

The area of a region is defined as the number of pixels contained within its boundary. Area is a convenient measure of overall size. It is dependent only on the boundary of the object.

$$\mathcal{A} = \sum_{p \in \mathcal{R}} X(p)$$

where \mathcal{R} denote the object region and

$$X(p) = \begin{cases} 1 & \text{if } p \in \mathcal{R} \\ 0 & \text{otherwise} \end{cases}$$

5.3.3 Volume

For a three-dimensional object in a digital image, its volume is the number of points (*voxels*) in the region of the subject. For an MR image, a three-dimensional object \mathcal{O} may be represented by a stack of two-dimensional objects S_1, S_2, \dots, S_n which are the tomographic images of \mathcal{O} . If the thickness of each S_i equals one pixel, then the volume of \mathcal{O} may be defined as

$$\mathcal{V} = \sum_{i=1}^n \mathcal{A}_i$$

where \mathcal{A}_i is the area of S_i .

If the thickness of each S_i is h and larger than one pixel, then the volume of \mathcal{O} may be approximated by interpolation.

$$\mathcal{V} \simeq \sum_{i=1}^{n-1} \mathcal{V}_i$$

where

$$\mathcal{V}_i = \frac{h}{3}(\mathcal{A}_i + \mathcal{A}_{i+1} + \sqrt{\mathcal{A}_i \mathcal{A}_{i+1}})$$

which is the volume of a frustrum of a pyramid. It is obvious that the thinner the slice thickness, the more accurate the volume of the object.

5.3.4 Surface Area

The surface area of a three-dimensional object is defined as the number of points on the surface of the object. For an MR image, the area of surface may be calculated by the perimeter. If a three-dimensional object \mathcal{O} is represented by a stack of two-dimensional objects S_1, S_2, \dots, S_n which are the tomographic images of \mathcal{O} and the thickness of S_i equals one pixel, then the surface area of \mathcal{O} is defined as

$$\mathcal{S} = \sum_{i=1}^n \mathcal{P}_i + \mathcal{A}_1 + \mathcal{A}_n$$

where \mathcal{P}_i is the perimeter of S_i , \mathcal{A}_1 and \mathcal{A}_n are the area of S_1 and S_n respectively.

If the thickness of each of S_i is h and larger than one pixel, then the surface area of \mathcal{O} may be approximated by

$$\mathcal{S} \simeq \sum_{i=1}^{n-1} T_i + \mathcal{A}_1 + \mathcal{A}_n$$

where

$$T_i = \frac{h}{2}(\mathcal{P}_i + \mathcal{P}_{i+1})$$

which is the area of a trapezium. The accuracy of the approximating surface area is dependent on the slice thickness h .

5.3.5 Central Moments

Some shape feature can be conveniently represented in terms of moments. For a shape represented by a region \mathcal{O} containing N pixels, the centre of mass is given as:

$$\begin{aligned}\bar{x} &= \frac{1}{N} \sum_{(x,y,z) \in \mathcal{O}} x \\ \bar{y} &= \frac{1}{N} \sum_{(x,y,z) \in \mathcal{O}} y \\ \bar{z} &= \frac{1}{N} \sum_{(x,y,z) \in \mathcal{O}} z\end{aligned}$$

The (p, q, r) order central moments become

$$\mu_{p,q,r} = \sum_{(x,y,z) \in \mathcal{O}} (x - \bar{x})^p (y - \bar{y})^q (z - \bar{z})^r$$

The central moments are position invariant.

5.3.6 Compactness

Compactness tends to reflect the complexity of the boundary of a two-dimensional object or the surface of a three dimensional object. For a two dimensional object \mathcal{O} with perimeter \mathcal{P} and area \mathcal{A} , the following magnitude is used to describe the feature shape:

$$\gamma_2 = \frac{\mathcal{P}^2}{4\pi\mathcal{A}}.$$

For a disc, γ_2 is a minimum and equals 1.

In the case of three dimensions, we have

$$\gamma_3 = \frac{\mathcal{S}^3}{36\pi\mathcal{V}^2}$$

For a sphere, γ_3 is a minimum and equals 1.

5.3.7 Elongation

Let r_{min} and r_{max} be the minimum and maximum distances, respectively, to the boundary from the center of mass. The ratio

$$\zeta = \frac{r_{min}}{r_{max}}$$

is used as a measure of eccentricity or elongation of the object.

5.4 Measurement of the Deformation of the Hemispheres

Now we will discuss how to measure the deformation of the hemispheres based on the parameters we have introduced above.

5.4.1 Image acquisition and segmentation

Each subject (patient or volunteer) has been scanned in the coronal section with the same T_1 sequence (IRSE 3400/500/32), 256 phase encoding steps, one signal average the same 7mm slice thickness, and the same 24 contiguous slices encompassing the whole head. Using the segmentation method described in Chapter 4, the contours of the left and right hemisphere of each slice were obtained (Fig 4.4). Then using the contour filling algorithm [Shani 1980] for each hemisphere, a uniform region of each

hemisphere is obtained (Fig. 5.2). Based on these uniform regions in the image we can now calculate all the parameters discussed in the last section.

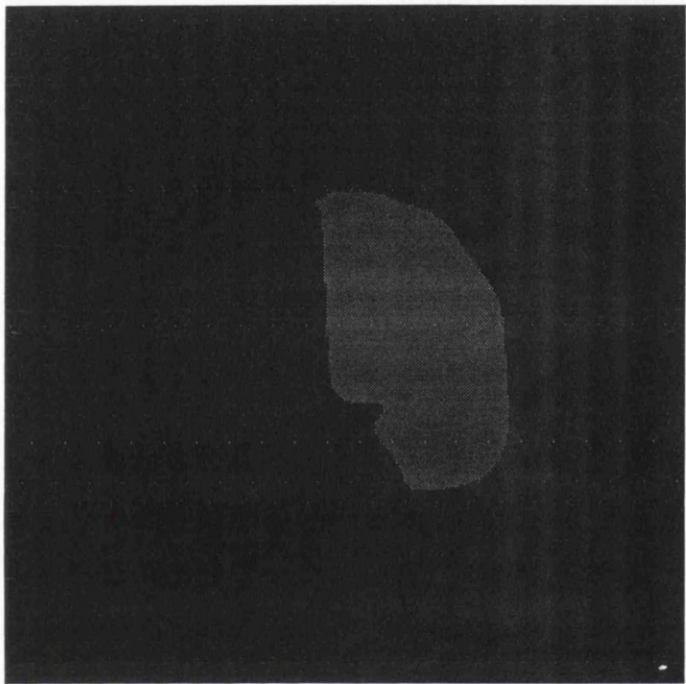


Figure 5.2 The regions of hemispheres obtained from Fig 4.4 by using a contour filling algorithm.

5.4.2 Two Dimensional Measurement

For both the left and right hemispheres on each sectional image, we first measure the following basic geometric features:

- The *area* of the sliced hemisphere, \mathcal{A} .
- The *perimeter* of the sliced hemisphere, \mathcal{P} .

- The *compactness* of the sliced hemisphere, $\gamma_2 = \frac{p^2}{4\pi A}$.
- The *elongation* of the sliced hemisphere, ζ .
- The *centre of gravity* of the sliced hemisphere, (\bar{m}, \bar{n}) .

In order to measure the change in the centre of gravities of the hemispheres on a sectional image, we consider the following equation

$$\delta_2 = \frac{(a - \bar{a}) - (\bar{m} - m)}{(a - \bar{a}) + (\bar{m} - m)}$$

where (\bar{a}, \bar{b}) and (\bar{m}, \bar{n}) are the centre of gravities of the left and right sliced hemispheres respectively. $a > \bar{a}$ and (a, \bar{b}) is on the boundary of the left sliced hemisphere. $m < \bar{m}$ and (m, \bar{n}) is on the boundary of the right sliced hemisphere.

The symmetry of the hemispheres, δ_2 is expected to approach 0 in the case of normal subjects though there is in reality almost always some degree of asymmetry. A large positive value of δ_2 indicates an expansion of the left hemisphere or a diminution of the right hemisphere. A large negative value of δ_2 indicates diminution of the left hemisphere or an expansion of the right hemisphere.

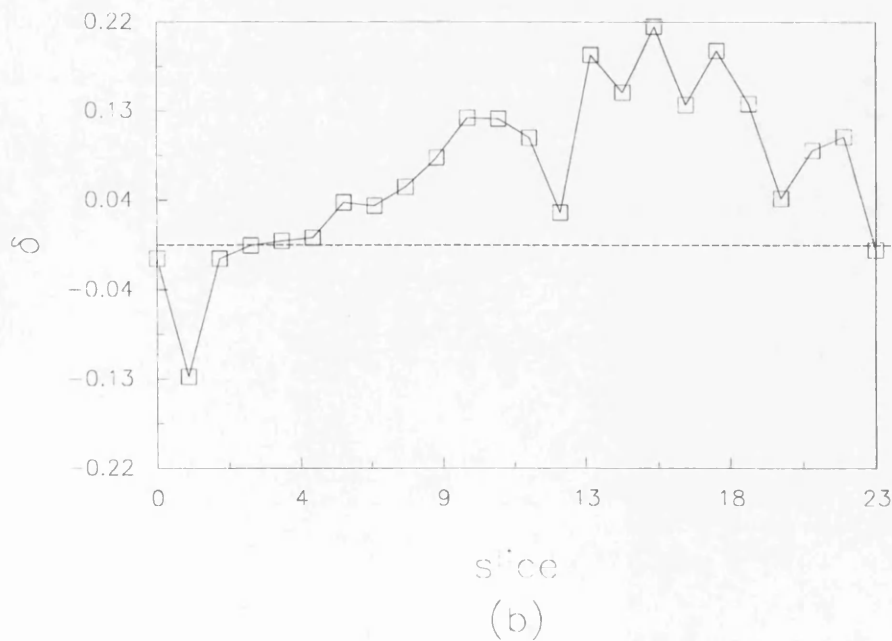
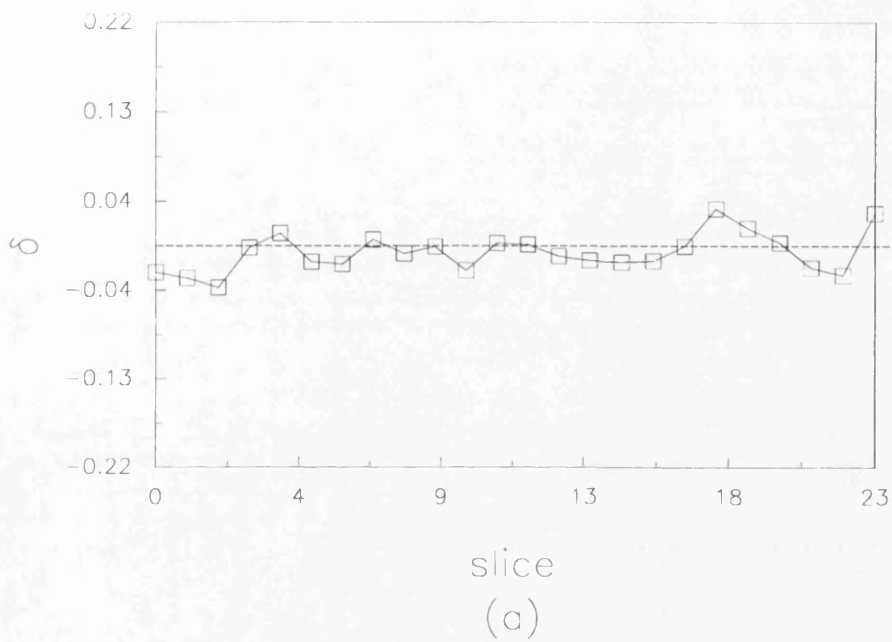
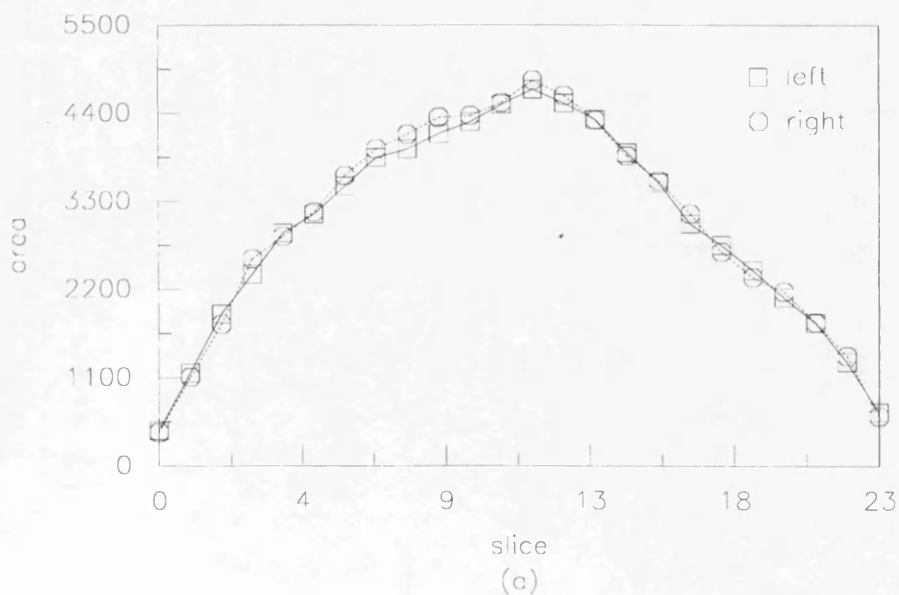


Fig.5.3 Shift of centre of gravity of the hemispheres. The horizontal axis indicates the number of the image slice and the vertical axis indicates the shift ratio. The data in (a) is from a volunteer and (b) from a patient with a malignant neoplasm.

It should be noted that δ_2 is position, rotation and size invariant. Fig.5.3 shows an example of the shift of centre of gravity of the hemispheres. δ_2 was calculated on each of the 24 contiguous slices. The values of δ_2 were plotted along the horizontal axis. The data in Fig.5.3 (a) is from a male volunteer 41 years old and the data in Fig.5.3 (b) from a 61 years old patient who suffered a malignant neoplasm . The curve in (a) is obviously more smooth and nearer zero than the curve in (b) because of the symmetry of the hemispheres. In Fig.5.3. (b), $\delta_2 > 0$ on most of sectional images which implies expansion of the left hemisphere.



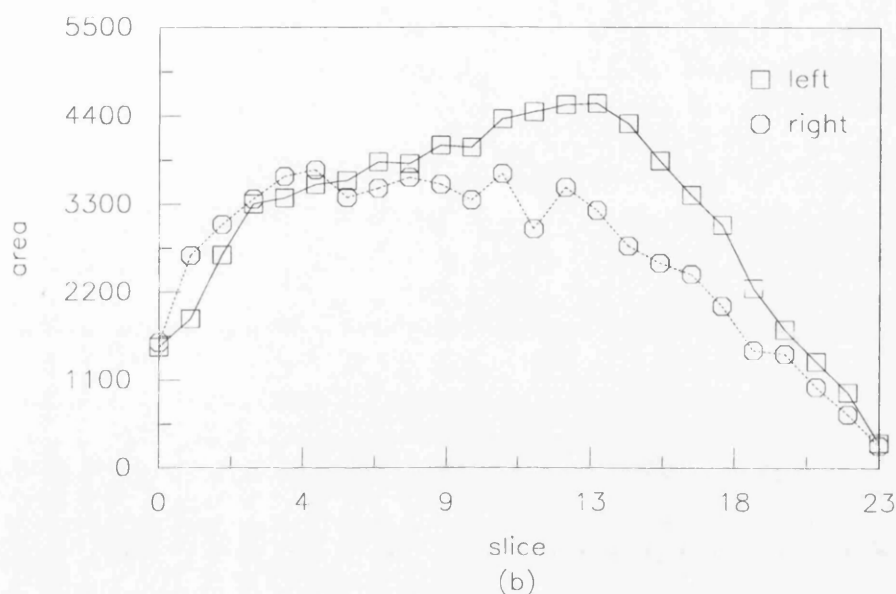
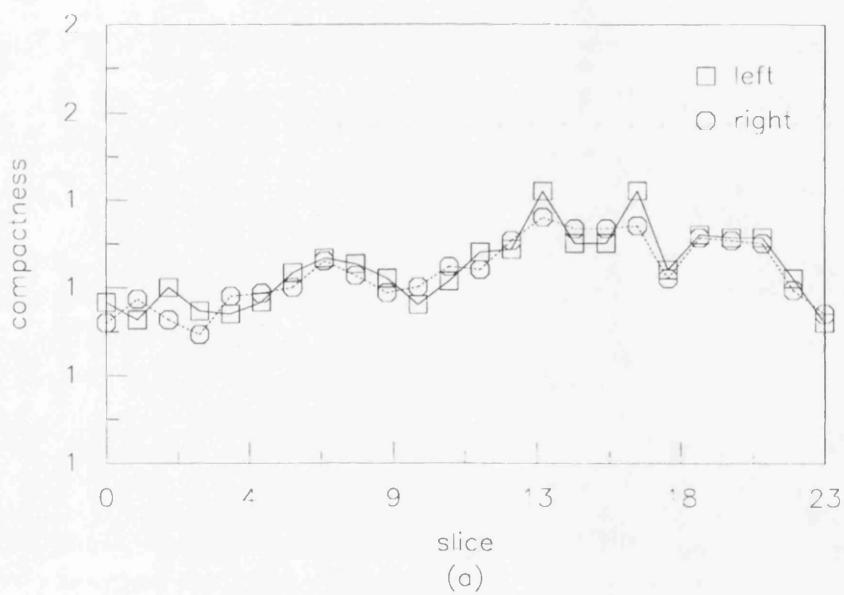


Fig.5.4 Areas of the sliced hemispheres. The horizontal and vertical axis indicate the number of the image slice and the area size (in pixels) of the sliced hemisphere respectively. (a) and (b) are from the same volunteer and patient as Fig.5.3.

Fig.5.4 shows the area change in each sliced hemisphere. The data in Fig.5.4 were from the same volunteer and patient as Fig.5.3. The areas of left and right sliced hemisphere were plotted along the horizontal axis. Comparing Fig 5.4 (a) and (b), we come to the same conclusion that the left hemisphere of the patient has suffered a significant degree of expansion.

Fig.5.5 shows the compactness change in each sliced hemisphere. The overlapping of two curves in Fig.5.5 (a) indicates the symmetry of the hemispheres. A significant difference in the compactness between the left and right hemisphere can be observed in Fig.5.5 (b).



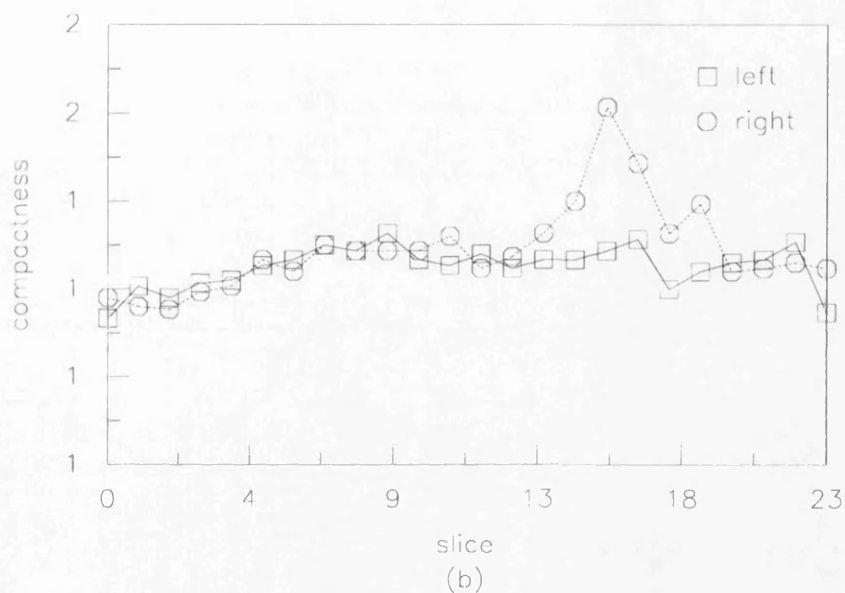


Fig.5.5 Compactness of the sliced hemispheres. The horizontal and vertical axes indicate the number of the image slice and the compactness of the sliced hemisphere respectively. (a) and (b) are from the same volunteer and patient as Fig.5.3.

5.4.3 Three Dimensional Measurement

In three dimensional measurement, we first consider the following attributes for each of the hemispheres.

- *volume* of the hemisphere, \mathcal{V}

- *surface area* of the hemisphere, \mathcal{S} .
- *compactness* of the hemisphere, $\gamma_3 = \frac{\mathcal{S}^3}{36\pi\mathcal{V}^2}$.
- *elongation* of the hemisphere, ζ .
- *centre of the gravity* of the hemisphere, $(\bar{p}, \bar{q}, \bar{r})$.

We will also consider the shift of the centre of gravity of the hemispheres in the three dimensional case. Similarly to the two dimensional case :

$$\text{central shift ratio: } \delta_3 = \frac{(a - \bar{a}) - (\bar{p} - p)}{(a - \bar{a}) + (\bar{p} - p)}$$

where $(\bar{a}, \bar{b}, \bar{c})$ and $(\bar{p}, \bar{q}, \bar{r})$ are the centre of gravities of the left and right hemispheres respectively. $a > \bar{a}$ and (a, \bar{b}, \bar{c}) is on the surface of the left hemisphere. $p < \bar{p}$ and (p, \bar{q}, \bar{r}) is on the surface of the right hemisphere. δ_3 has the same invariant properties as δ_2 .

Based on the two dimensional measurement, we can produce some three dimensional features of the brain structures. There are

$$\begin{aligned} \text{area variation: } r_1 &= \frac{\sum_{i=1}^n (A_i - B_i)}{\sum_{i=1}^n (A_i + B_i)} \\ \text{absolute area variation: } r_2 &= \frac{\sum_{i=1}^n |A_i - B_i|}{\sum_{i=1}^n (A_i + B_i)} \end{aligned}$$

where A_i and B_i are the areas of the left and right hemispheres in the slice i , respectively.

$$\text{perimeter variation: } r_3 = \frac{\sum_{i=1}^n (P_i - Q_i)}{\sum_{i=1}^n (P_i + Q_i)}$$

$$\text{absolute perimeter variation: } r_4 = \frac{\sum_{i=1}^n |P_i - Q_i|}{\sum_{i=1}^n (P_i + Q_i)}$$

where P_i and Q_i are the perimeters of the left and right hemispheres in the slice i , respectively.

$$\text{central shift variation: } r_5 = \frac{\sum_{i=1}^n \delta_i}{n}$$

$$\text{absolute central shift variation: } r_6 = \frac{\sum_{i=1}^n |\delta_i|}{n}$$

where δ_i is the two-dimensional central shift ratio δ in the slice i .

It is obvious that r_1, r_2, \dots, r_6 are size, translation, and rotation invariant. They may therefore be regarded as indices of the shape of the hemisphere. A large positive value of r_1 or r_3 indicates an expansion of the left hemisphere; while a large negative value of r_1 or r_3 indicates an expansion of the right hemisphere. A large positive value of r_5 indicates a diminution of the right hemisphere, and a large negative value of r_5 indicates a diminution of the left hemisphere. If all the r approach zero, this indicates symmetry of the hemispheres.

Although there is in reality some degree of cerebral asymmetry in normal subjects [Chui 1980], we would expect a more significant difference in the cerebral asymmetry between the normal population and patients with intracranial deformation. Our technique allows us to test this by analysing the variation of the left and right hemispheric

volumes between normal volunteers and patients with intracranial deformation. A parameter to describe the variation is given by

$$\varrho = \frac{|\mathcal{V}_l - \mathcal{V}_r|}{\mathcal{V}_l + \mathcal{V}_r}$$

which is called *hemispheres variation ratio*. Based on analysis of a total of 15 volunteers from age 23 to 44 and 28 patients with intracranial deformation from age 19 to 55, the mean, standard deviation (SD), maximum and minimum of ϱ are obtained (Table 5.1).

	mean	SD	maximum	minimum
volunteers	0.00530	0.00635	0.02459	0.00012
patient	0.03556	0.02539	0.10093	0.00060

Table 5.1

Fig. 5.6 shows the histogram of the hemisphere variation ratio ϱ of the volunteers (a) and the patients (b). The variations between the ratios (ϱ) of the volunteers and the patients were assessed using t-test [Hines 1990] at the level of significance $\alpha = 0.001$. From Table 5.1, we can calculate the t-test statistic as $t = 4.5173$. From the t-table [Neave 1979] using 41 $(15 + 28 - 2)$ degrees of freedom, we have the critical value $t_{.001} = 3.307$ which is less than the test statistic.

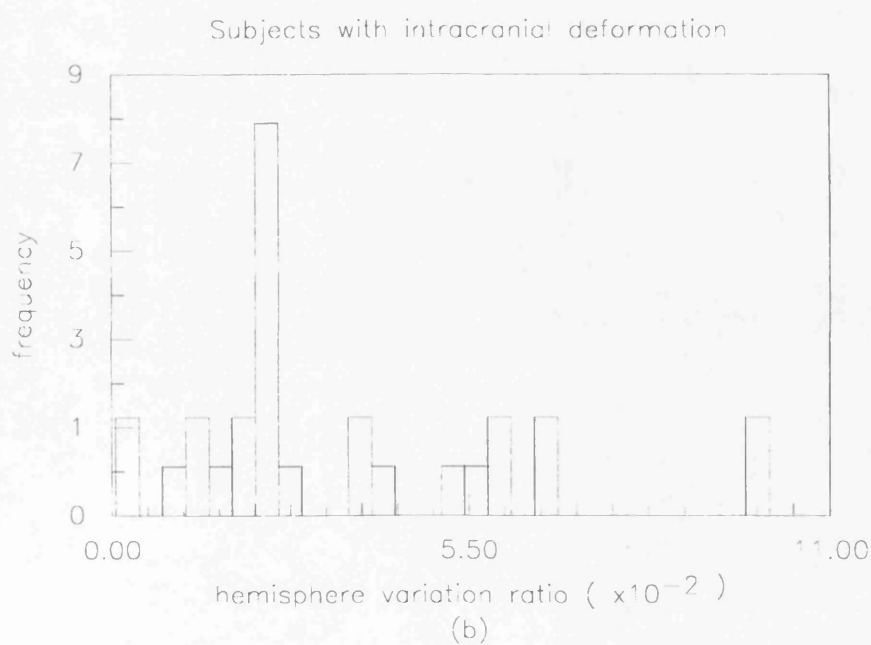
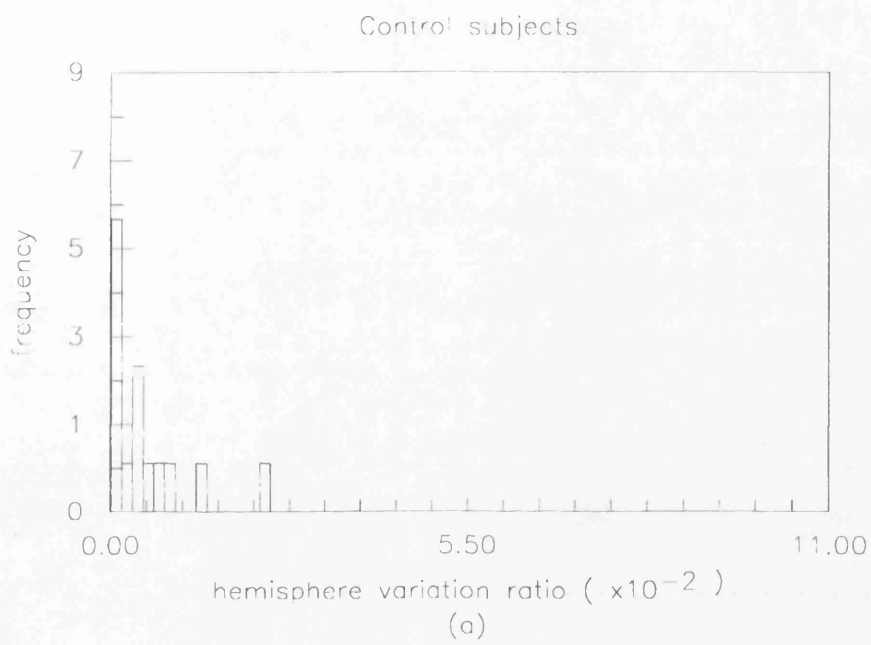


Fig.5.6 Histogram of the hemisphere variation ratio (ϱ), (a) and (b) are based on the measurement from 15 volunteers and 28 intracranial deformation patients, respectively.

5.4.4 Invariant Attributes

Among all the basic geometric attributes, we are more interested in those which are size, rotation and position invariant. The invariance properties are very important in our study, because there exists wide variations in brain size within the normal population and differences in patient orientation during image acquisition (*i.e.* differences from the acquisition of one patient to the next, and in the same patient if scanned serially over weeks or months). The following are all invariant attributes arising from our methods which we can use to measure the deformation of the hemispheres.

1. area variation r_1 .
2. absolute area variation r_2 .
3. perimeter variation r_3 .
4. absolute perimeter variation r_4 .
5. central shift variation r_5 .
6. absolute central shift variation r_6 .

7. $\mathcal{V}_l/(\mathcal{V}_l + \mathcal{V}_r)$, where \mathcal{V}_l and \mathcal{V}_r are the left and right hemisphere volumes, respectively.
8. the maximum of γ_2 among the 24 contours of the left hemisphere.
9. the minimum of γ_2 among the 24 contours of the left hemisphere.
10. the mean of γ_2 among the 24 contours of the left hemisphere.
11. the standard deviation of γ_2 among the 24 contours of the left hemisphere.
12. the maximum of ζ among the 24 contours of the left hemisphere.
13. the minimum of ζ among the 24 contours of the left hemisphere.
14. the mean of ζ among the 24 contours of the left hemisphere.
15. the standard deviation of ζ among the 24 contours of the left hemisphere.
16. the maximum of γ_2 among the 24 contours of the right hemisphere.
17. the minimum of γ_2 among the 24 contours of the right hemisphere.
18. the mean of γ_2 among the 24 contours of the right hemisphere.
19. the standard deviation of γ_2 among the 24 contours of the right hemisphere.
20. the maximum of ζ among the 24 contours of the right hemisphere.
21. the minimum of ζ among the 24 contours of the right hemisphere.

22. the mean of ζ among the 24 contours of the right hemisphere.
23. the standard deviation of ζ among the 24 contours of the right hemisphere.
24. compactness of left hemisphere γ_{3l} .
25. compactness of right hemisphere γ_{3r} .
26. elongation of left hemisphere ζ_l .
27. elongation of right hemisphere ζ_r .
28. 3-dimensional central shift ratio δ_3 .

43 subjects (28 patient and 15 volunteers) have been analysed with these parameters. All patients were diagnosed as having malignant brain tumours (specifically gliomas), usually from their CT appearances. Provided an MR acquisition slot became available before treatment was commenced, then they were included in the study (this proved a very stringent criteria as steroid treatment was usually commenced soon after diagnosis and the MR was usually unavailable for patients without appointments). No other form of selection was applied to this group of patients, or to the group used for the prospective analysis in Chapter 8.

For each subject, we obtained a 28-dimensional vector to characterise the hemispheres. Obviously such a representation would not be readily comprehensible. This is particularly important because the ultimate aim of this project is to produce data

which will be conceptually accessible to clinicians. In Chapter 8 we therefore will use pattern recognition techniques to select the “*best*” features from these parameters and others (see the chapters 6 and 7) to reduce the multi-dimensional data format into a more comprehensible two dimensional display.

5.5 Conclusions

In this chapter we have developed a quantitative framework to measure the deformation of the hemispheres. We have considered not only the change in size, for example conventional volumetric measurements, but also the changes in shape and position, in symmetry and compactness of the hemispheres, and the shift of the centre of gravity of the hemispheres. This technique provides a way to objectively assess the intracranial deformation caused by brain lesions, especially brain tumours. The example in section 5.4 shows that the changes in size (Fig. 5.4), position (Fig. 5.3) and shape (Fig. 5.5) can be described by our technique. In the last part of section 5.4 we used a 28-dimensional vector to represent the deformation of the hemispheres. This measurement vector includes quantitative information about the changes in size, position and shape.

Brain lesions cause complex three dimensional deformation and displacement. Deformation is manifested by relative displacement of interfaces between intracranial

structures. Besides the cerebral hemispheres, the lateral ventricle and the longitudinal fissure also are obvious and easily delineated structures. Methods for analysing deformation of these structures will be developed in the next two chapters.

Chapter 6 Shape Analysis of the Lateral Ventricles Using Fourier Descriptors

6.1 Introduction

Intracranial structures have a shape characteristic that may be expressed independently of their volumes or positions. In this chapter a shape analysis method is developed to analyse this shape characteristic. The new technique is totally independent of size and position of brain structures which is advantageous in the clinical situation because of the wide variations in normal brain anatomy. In this chapter, we concentrate on measuring the shape of the lateral ventricles. The ventricles are located deep inside the brain from forebrain to hindbrain (Pons and Medulla Oblongata). Various pathological processes within the brain such as tumours, hematoma, or intracranial pressure can cause distortion and displacement of the ventricles, especially the lateral ventricles. However, volumetric measurement of ventricular size is

inaccurate and unsatisfactory because of the *partial volumes problem* in MR scanning which is caused by the complex and irregular shape of the ventricles [Planter 1991].

Fourier descriptors is one method of describing the shape of a closed, planar object [Zahn 1972]. This method has shown its advantage in pattern recognition because of its invariant properties. Several researchers have made useful application of various types of Fourier descriptors, such as in character recognition [Granlund 1972, Persoon 1974], machine parts recognition [Persoon 1977], and identification of three dimensional objects [Wallace 1980]. To our knowledge, such an approach has never been applied to the quantitative analysis of MR data.

In this chapter we will describe the properties of Fourier descriptors and their application to intracranial deformation. The method involves the decomposition of a shape into its constituent spatial frequency components which can then be expressed in the form of Fourier descriptors. Once the Fourier descriptor has been computed, the operations of rotation, scaling and moving are easily implemented in the frequency domain by simple arithmetic on frequency domain coefficients. Therefore a complex shape and its change in spatial domain can be easily described in the frequency domain. In the third section of this chapter we develop a new technique to analyse shape similarity and symmetry based on the invariant properties of Fourier descriptors. The fourth section is an application of Fourier descriptors, showing how it is used to describe the deformation of the lateral ventricle. Example analyses are

provided for groups of 12 volunteers and patients. An example of how the technique is applied serially in the same patient to assess the effects of therapy is provided. This work forms the basis of a paper in preparation.

6.2 Fourier Descriptors and Its Invariant Properties

6.2.1 Fourier Descriptors

Once the boundary trace is known (see Chapter 4), we can consider it as a pair of waveforms $x(t)$, $y(t)$. Hence any of the traditional one-dimensional signal representation techniques can be used. For any sampled boundary we can define

$$u(n) = x(n) + jy(n), n = 0, 1, \dots, N - 1 \quad (6.1)$$

which, for a closed boundary, would be periodic with period N . Its discrete Fourier transform is

$$a(k) = \sum_{n=0}^{N-1} u(n) \exp\left(\frac{-j2\pi kn}{N}\right), 0 \leq k \leq N - 1 \quad (6.2)$$

And the inverse transform is

$$u(n) = 1/N \sum_{k=0}^{N-1} a(k) \exp\left(\frac{j2\pi kn}{N}\right), 0 \leq n \leq N - 1 \quad (6.3)$$

The complex coefficients $a(k)$ are called the *Fourier descriptors* (FDs) of the boundary. For a continuous boundary function $u(n)$, defined in a similar manner, the FDs are its (infinite) Fourier series coefficients.

6.2.2 Invariant Properties

Several geometrical transformations of a boundary or shape can be related to simple operations on the FDs (Table 6.1) [Granlund 1972, Zahn 1972]. If the boundary is translated by

$$u_0 = x_0 + jy_0$$

then the new FDs remain the same except at $k = 0$. The effect of scaling, that is, shrinking or expanding of the boundary results in scaling of $a(k)$. Changing the starting point in tracing the boundary results in a modulation of $a(k)$. Rotation of the boundary by an angle θ_0 causes a constant phase shift of θ_0 in the FDs. Reflection of the boundary (or shape) about a straight line inclined at an angle θ

$$Ax + By + C = 0$$

gives the new boundary $\tilde{x}(n), \tilde{y}(n)$ as

$$\tilde{u}(n) = u^* e^{j2\theta} + 2\gamma$$

Where u^* denote the conjugate complex number of u , and

$$\gamma = \frac{-(A + jB)C}{A^2 + B^2},$$

$$\exp(j2\theta) = \frac{-(A + jB)^2}{A^2 + B^2}$$

From Table 6.1 it can be observed that the FD magnitudes have some invariant properties. For example $\tilde{a}(k)$, $k = 1, 2, \dots, N - 1$ are invariant to starting point, rotation, and position. The features $\tilde{a}(k)/|\tilde{a}(k)|$ are invariant to scaling. These properties can be used in characterising shapes regardless of their size and orientation. The property of FD's reflection can be used to analyse symmetry of shapes.

Transformation	Boundary	Fourier Descriptors
Identity	$u(u)$	$a(k)$
Translation	$\tilde{u}(n) = u(n) + u_0$	$\tilde{a}(k) = a(k) + u_0\delta(k)$
Scaling	$\tilde{u}(n) = \alpha u(n)$	$\tilde{a}(k) = \alpha a(k)$
Starting Point	$\tilde{u}(n) = u(n - n_0)$	$\tilde{a}(k) = a(k)e^{-j2\pi n_0 k/N}$
Rotation	$\tilde{u}(n) = u(n)e^{j\phi}$	$\tilde{a}(k) = a(k)e^{j\phi}$
Reflection	$\tilde{u}(n) = u^*(n)e^{j2\theta} + 2\gamma$	$\tilde{a}(k) = a^*(-k)e^{j2\theta} + 2\gamma\delta(k)$

Table 6.1

6.3 Shape Similarity and Symmetry

Analysis of shape similarity and symmetry can be based on the invariant properties of Fourier descriptors. This makes them very useful for our study.

6.3.1 Shape Similarity

The Fourier descriptors can be used to match similar shapes even when they have different sizes and orientations. If $a(k)$ and $b(k)$ are the FDs of two boundaries $u(n)$ and $v(n)$, respectively, the *distance between u and v* is defined as

$$d(u, v) = \sqrt{\sum_{k=1}^{N-1} |a(k) - b(k)|^2} \quad (6.4)$$

Note that the differences in position are taken out by setting $a(0) = b(0)$. In order to minimise the distance $d(u, v)$, v will be scaled by α , rotated by ϕ and adjusted to the starting point n_0 . We have then to determine α , ϕ and n_0 such that

$$D = \sum_{k=1}^{N-1} |a(k) - \alpha e^{j(k\theta_0 + \phi)} b(k)|^2 \quad (6.5)$$

is minimised, where $\theta_0 = 2\pi n_0/N$.

From Equation (6.5), we will have

$$\begin{aligned} D &= \sum_k (a(k) - \alpha e^{j(k\theta_0 + \phi)} b(k))(a^*(k) - \alpha e^{-j(k\theta_0 + \phi)} b^*(k)) \\ &= \sum_k a(k)a^*(k) + \alpha^2 \sum_k b(k)b^*(k) - 2\alpha \sum_k \text{Re}(a^*(k)b(k)e^{j(k\theta_0 + \phi)}) \end{aligned}$$

Where $\text{Re}(c)$ denotes the real components of c . Let $a^*(k)b(k) = \rho(k)e^{j\psi(k)}$, then we have equivalently,

$$D = \sum_k a(k)a^*(k) + \alpha^2 \sum_k b(k)b^*(k) - 2\alpha \sum_k \rho(k)\cos(\psi(k) + k\theta_0 + \phi) \quad (6.6)$$

In order to minimize equation (6.6) with respect to α , θ_0 and ϕ we will compute

the partial derivatives

$$\begin{aligned}\frac{\partial D(\alpha, \theta_0, \phi)}{\partial \alpha} &= 2\alpha \sum_k b(k)b^*(k) - 2 \sum_k \rho(k) \cos(\psi(k) + k\theta_0 + \phi) \\ \frac{\partial D(\alpha, \theta_0, \psi)}{\partial \theta_0} &= 2\alpha \sum_k \rho(k) k \sin(\psi(k) + k\theta_0 + \phi) \\ \frac{\partial D(\alpha, \theta_0, \phi)}{\partial \phi} &= 2\alpha \sum_k \rho(k) \sin(\psi(k) + k\theta_0 + \phi)\end{aligned}$$

By setting those derivatives equal to zero we obtain

$$\begin{aligned}\alpha &= \frac{\sum_k \rho(k) \cos(\psi(k) + k\theta_0 + \phi)}{\sum_k b(k)b^*(k)} \\ \tan \phi &= -\frac{\sum_k \rho(k) \sin(\psi(k) + k\theta_0)}{\sum_k \rho(k) \cos(\psi(k) + k\theta_0)}\end{aligned}$$

Since both $a(k)$ and $b(k)$ only contain a finite number (N) of harmonics, then the minimum value of D can be obtained by :

$$\mathcal{D} \equiv \min(D) = \min_{\theta_0} \left\{ \sum_k |a(k) - \alpha e^{(k\theta_0 + \phi)} b(k)|^2 \right\} \quad (6.7)$$

where θ_0 is evaluated for each $\theta_0 = 2\pi n_0/N$, $n_0 = 0, 1, \dots, N-1$. \mathcal{D} is called the *shape distance* between object u and v . When $\mathcal{D} = 0$, we say u and v have the same shape.

6.3.2 Shape Symmetry

For a given FD's $a(k)$ of a closed curve $u(n)$, if $u(n)$ is symmetric about the centre line $x = 0$, from section 5.1 we can get the reflection of $u(n)$ about the vertical axis $x = 0$

$$\tilde{a}(k) = -a^*(-k)$$

Since axial symmetry means invariance under a reflection, then we use

$$\mathcal{Y} = \sum_{k=1}^{N-1} (a(k) - \tilde{a}(k)) \quad (6.8)$$

for symmetry measurement. A curve with perfect vertical axial symmetry has $\mathcal{Y} = 0$, and we expect that \mathcal{Y} near zero will generally indicate a shape that would be judged nearly axially symmetric.

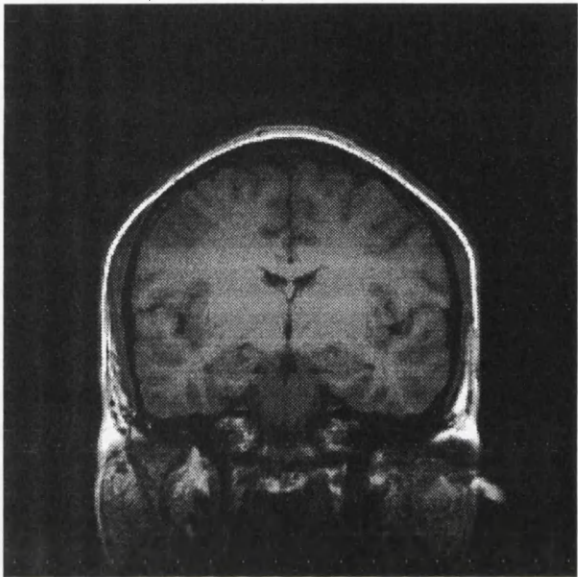
6.4 Applications

In this section we will show how to apply the Fourier descriptors technique to the quantitative analysis of intracranial deformation. We consider the deformation of the lateral ventricle. In coronal section scanning, the boundary of the lateral ventricle is a closed curve, so the deformation of the ventricle can be described by means of Fourier descriptors.

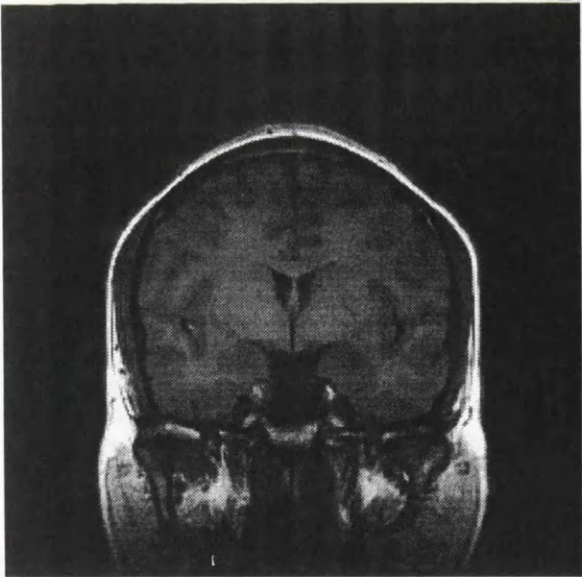
6.4.1 Image acquisition and segmentation

For each subject, three slices which lie within the body of lateral ventricle are chosen from the stack of slices obtained using the T_1 -weighted sequence (IRSE 3400/500/32) in the coronal plane (Fig. 6.1). These chosen slices lie between the mamillary body and the anterior horn of lateral ventricle. There are two reasons why we choose these slices. Firstly the boundaries of the left and right lateral ventricles in these chosen

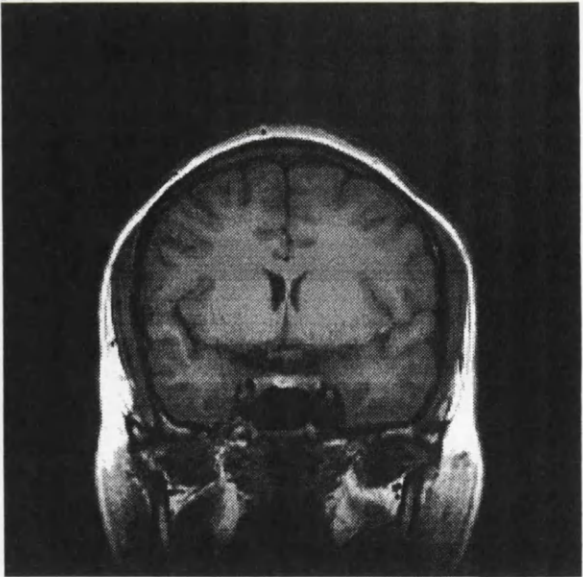
slices are almost connected (Fig. 6.1).



slice 1



slice 2



slice 3

Fig. 6.1 The slices chosen between the mamillary body and the anterior horn of lateral ventricle.

Thus the contours of left and right lateral ventricles can be considered as one closed curve which is available for Fourier descriptor analysis. In the other slices the left and right lateral ventricles are separated by the corpus callosum (Fig. 6.2) making it difficult to use Fourier descriptors. Secondly, among the slices in which the left and right lateral ventricles are closed, the contours of the lateral ventricles in these three chosen slices are larger than in the other slices. The larger region will provide more information about the shape of the lateral ventricles.

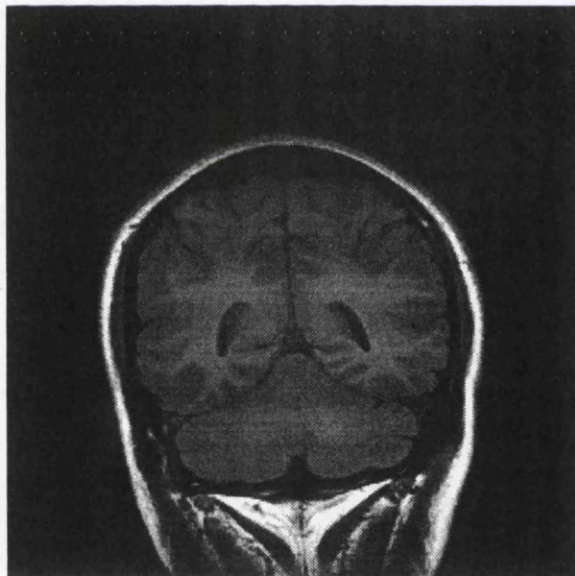
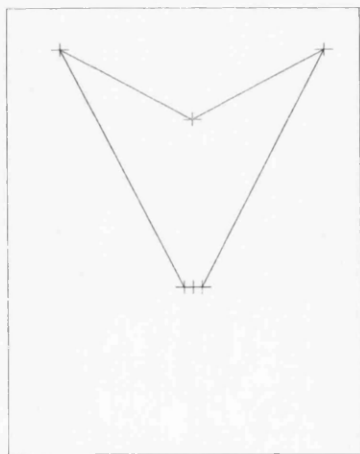


Fig.6.2 The left and right lateral ventricles are separated in some slices.

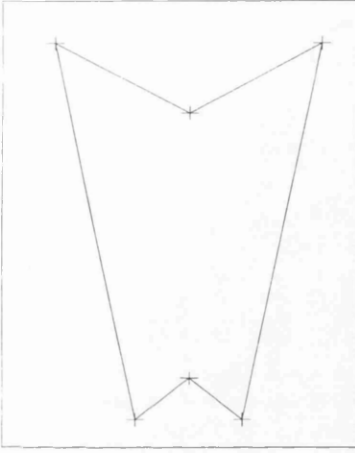
Using the segmentation method described in Chapter 4, the boundaries of the lateral ventricles in the chosen slices are obtained. As each of the boundaries is a closed curve we can use Fourier descriptors to analyse the object shape.

6.4.2 The Training Set

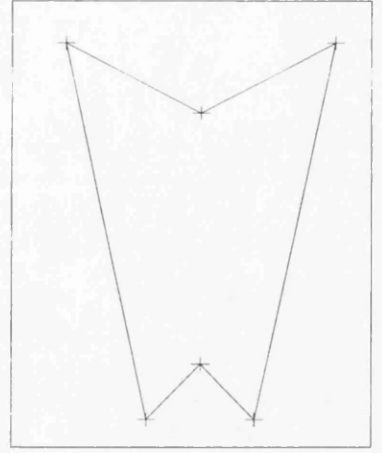
Using Equation (6.5), we can match similar shapes even if they have different sizes and orientations. For each of the three slices, a closed curve is given as a reference with which the closed lateral ventricles can be compared. This reference is called a *training set*. It can be regarded as an ideal contour of the lateral ventricles (Fig. 6.3). Note that the reference is symmetric about the centerline line. The design of the training set is based on the shape of lateral ventricles of normal volunteers at the same position.



Slice 1



Slice 2



Slice 3

Fig.6.3 The references (training data sets) for matching lateral ventricles.

6.4.3 Results

For the similarity measurement, the three contours of lateral ventricles from 15 volunteers (8 men and 7 women, between the ages of 21 and 43) and 28 patients with brain tumours (16 men and 12 women, between the ages of 17 and 65) were compared with the three references respectively by using Equation (6.7). For each patient or volunteer, therefore, three shape distances ($\mathcal{D}_1, \mathcal{D}_2$ and \mathcal{D}_3) were obtained to characterise the shape of the lateral ventricles. We then calculated the mean of the three shape distances by

$$M_{\mathcal{D}} = \frac{\mathcal{D}_1 + \mathcal{D}_2 + \mathcal{D}_3}{3}.$$

We call $M_{\mathcal{D}}$ a *shape value of an object*. Table 6.2 summarises the measurement results. It shows the mean, standard deviation, maximum and minimum of the shape values ($M_{\mathcal{D}}$). The mean and the standard deviation of the *shape value* in the patients with brain tumours are significantly larger than those in the volunteer group. The smaller mean and standard deviation of the *shape value* in the volunteer group indicates shape similarity and stability.

	mean	SD	maximum	minimum
volunteers	2.3216	0.4097	3.4017	1.7900
patients	4.6027	0.7990	5.2857	2.9434

Table 6.2

The variations between the *shape values* of volunteer and patient groups are assessed using t-tests. From Table 6.2, we can calculate the test statistic as $t = 10.3144$ with 41 ($15 + 28 - 2$) degrees of freedom which is significant at the $\alpha = 0.001$ level (the critical value $t_{.001} = 3.307$).

Using Equation (6.8), we measure the symmetry of the contours of the lateral ventricles within the three slices. For each patient or volunteer, therefore, three symmetry parameters (\mathcal{Y}_1 , \mathcal{Y}_2 and \mathcal{Y}_3) were obtained to characterise the symmetry of the lateral ventricles. We then calculated the mean of the three symmetry parameters

by

$$M_y = \frac{y_1 + y_2 + y_3}{3}.$$

We call M_y a *symmetry value of an object*. Table 6.3 summaries the measurement results after measuring all the 15 volunteers and 28 patients. It shows the mean, standard deviation, maximum and minimum of the symmetry values (M_y).

	mean	SD	maximum	minimum
volunteers	1.4049	0.4658	2.3435	0.5002
patients	3.6676	0.9897	6.0489	2.4411

Table 6.3

The variations between the *symmetry values* of volunteer and patient groups are assessed using t-tests. From Table 6.3, we can calculate the test statistic as $t = 8.314$ with 41 ($15 + 28 - 2$) degrees of freedom which is significant at the $\alpha = 0.001$ level (the critical value $t_{.001} = 3.307$).

As an illustrative example of how this technique can be used to serially assess therapy we consider a 19 year old female patient with a tumour in the left temporal lobe. Table 6.4 shows the measurement results (the *shape distance* and the *symmetry parameter*) both before and after six months radiotherapy.

	shape			symmetry		
	1	2	3	1	2	3
pre	4.090	2.940	2.347	3.540	4.830	4.532
post	3.293	2.583	2.028	3.531	1.592	2.410

Table 6.4 Quantitative assessment of tumour response to treatment. The second and third rows show the results before and after six months radiotherapy respectively.

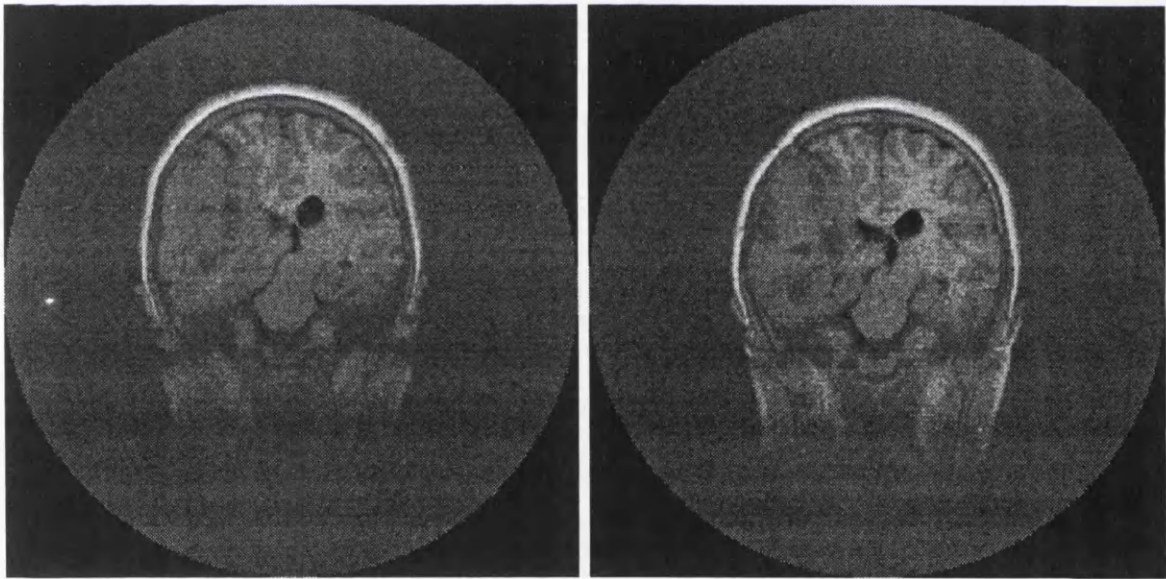


Fig. 6.4 A change of the ventricular shape. (a) before treatment; (b) after 6 months treatment.

Fig.6.4 (a) and (b) show the change of the ventricular shape before and after treatment. From both Table 6.4 and Fig.6.4, we can see that the ventricular shape of the patient is changed.

It is interesting that a clinical assessment of the patient also showed an improvement after treatment. The clinical assessment shown in Table 6.5 was made independently by Dr R. Rampling, a neurooncologist in the Beatson Oncology Centre of the Western Infirmary, Glasgow.

	headache	seizure	orientation	memory	motor	speech	vision
pre	no deficit	multiple daily	no deficit	no deficit	resistance	fragmentary only	no deficit
post	responds to treatment	controlled with drug	no deficit	no deficit	resistance	no deficit	no deficit

Table 6.5 Clinical assessment of the patient's response to treatment. The second and third rows show the results before and after six months of radiotherapy respectively.

6.5 Conclusions

In this chapter a quantitative technique for characterising the shapes of intracranial structures has been developed. The technique can be used to match similar shapes even when they have different sizes and orientations. It also provides a way to describe the symmetry of an object. Its application to the quantitative assessment of ventricular deformation has demonstrated advantages which may not be achieved by conventional quantitative techniques.

The new technique has been applied to groups of 15 volunteers and 28 patients presenting with brain tumours. The results have shown a remarkable difference in

the shape of the lateral ventricle between volunteers and the patients. The *shape values* of volunteers (mean 2.3216) are general smaller than that of patients (mean 4.6027) (Table 6.2). The measurement of symmetry of the lateral ventricles also shows a significant difference between the volunteers and patients, with the volunteers generally having smaller *symmetry values* (mean 1.4049) than the patients (mean 3.6676) (Table 6.3).

6 invariant attributes have been obtained by applying the Fourier descriptors. This 6-dimensional measurement vector is used to characterise the shape and symmetry of the lateral ventricles.

Unfortunately the technique is currently limited to two-dimensional images. In section 6.4 the shape analysis of the lateral ventricle was based on the two-dimensional measurement of selected slices which cut the lateral ventricles. Only three of these slices were chosen in our application for the reasons mentioned in section 6.4. In the case of the cerebral hemispheres, it is unlikely that we could gain information about the global deformation by measuring only a few cross sections. However, if we were to measure all slices which contain the hemispheres, it would cause difficulties in both computing time and feature extraction from the wealth of parameters generated. For this reason we feel the technique is not suitable for the hemispheres.

Chapter 7 Shape Analysis of the Falx Cerebri Using Surface Curvatures

7.1 Introduction

In three-dimensional MR brain images, secondary space occupying effects may be manifested by changes in the convexity and concavity of the cerebral lobe surfaces between one side of the brain and the contralateral side . Displacement of the midline structures can be described as changes in the curvature of the falx cerebri, and internal herniation can be regarded as variations in the surface curvature of the cerebral lobes.

The aim of this chapter is to develop a novel shape analysis technique for characterising the deformations in the surfaces of the midline structures. These structures do not form closed contours in cross section so the techniques described in the previous sectors cannot be applied. We therefore develop a technique based on differential

geometry to classify surfaces into certain character types. The deformation of a surface then can be described by measuring the curvature changes in these types. Like the Fourier descriptors approach, this method possesses several invariant properties, namely size, translation, and rotation invariance. As far as we are aware this is the first time differential geometry has been applied to any aspect of medical imaging.

Classical differential geometry provides a complete local description of smooth surfaces and guides our selection of surface characteristics. The *Gaussian curvature* and the *mean curvature* are identified as the second-order surface characteristics [de Carmo 1976] that capture the characteristics of a surface. Using the Gaussian curvature and the mean curvature, any surface in three-dimensional Euclidean space can be described as the combination of basic and simple surfaces called *primary surfaces*. Intuitively speaking a surface is composed of *elliptic*, *hyperbolic*, *saddle* and *flat* surfaces. The deformation of a surface can be interpreted by the curvatures of these primary surfaces.

A stack of MR images provides three-dimensional images of internal structures of the body. After a local boundary segmentation on each sectional image, a set of edge elements are obtained. These edges can be considered as the trace of a certain number of surfaces. Based on the surface trace information, B-splines surfaces can be obtained which approximate the original image surfaces. In this way we can characterise the image surfaces.

In the next section, we first introduce surface differential geometry. Section 7.3 will discuss *the mean curvature* and *the Gaussian curvature* and how to characterise surfaces by curvatures. Then in Section 7.4 we apply the technique in a group of 15 normal volunteers and 28 patients. Two examples of patients whose condition changed after treatment are provided to demonstrate how the technique can be used to serially assess the effects of therapy. The 28 new size independent attributes this technique generates are listed.

The technique and results in this chapter have been published recently as a paper in the *IEEE Transactions on Medical Imaging* [Dai 1993].

7.2 Surface Differential Geometry

The explicit parametric form of a general surface $\mathbf{r}(\mu, v)$ with respect to a known coordinate system is :

$$\mathbf{r}(\mu, v) = \{x(\mu, v), y(\mu, v), z(\mu, v)\} \quad (7.1)$$

The parametric derivatives \mathbf{r}_μ , \mathbf{r}_v , $\mathbf{r}_{\mu v}$... depend on the specific parameterization adopted but from them the *intrinsic* differential characteristics of the surface can be derived, such as the unit normal and the Gaussian curvature and principal directions, which are independent of parameterization. The unit surface normal is defined in terms of the parametric derivatives \mathbf{r}_μ and \mathbf{r}_v by

$$\mathbf{n} = \frac{\mathbf{r}_\mu \times \mathbf{r}_v}{|\mathbf{r}_\mu \times \mathbf{r}_v|} \quad (7.2)$$

On a curved surface, the infinitesimal distance element between two neighbouring points (μ, v) and $(\mu + d\mu, v + dv)$ is given by

$$ds^2 = \mathbf{r}_\mu \bullet \mathbf{r}_\mu d\mu^2 + 2\mathbf{r}_\mu \bullet \mathbf{r}_v d\mu dv + \mathbf{r}_v \bullet \mathbf{r}_v dv^2 \quad (7.3)$$

Equation (7.3) is often called the *first fundamental quadratic form* of a surface and is written in the form $E d\mu^2 + 2F d\mu dv + G dv^2$. The quantities

$$\begin{aligned} E(\mu, v) &= \mathbf{r}_\mu \bullet \mathbf{r}_\mu, \\ F(\mu, v) &= \mathbf{r}_\mu \bullet \mathbf{r}_v \\ G(\mu, v) &= \mathbf{r}_v \bullet \mathbf{r}_v \end{aligned} \quad (7.4)$$

are the first fundamental, or *metric*, coefficients of the surface. The metric coefficients provide the basis for the measurement of the lengths and areas and the specification of directions and angles on a surface.

The first fundamental form gives the distance ds between neighbouring points (μ, v) and $(\mu + d\mu, v + dv)$ on a surface to first order in $d\mu$ and dv . The distance element ds lies in the *tangent plane* of the surface at point (μ, v) and therefore yields no information on how the surface curves away from the tangent plane at the point.

To investigate surface curvature, we must examine the vector distance between neighbouring points (μ, v) and $(\mu + d\mu, v + dv)$ to second order in $d\mu$ and dv . The

component of this vector distance *perpendicular* to the tangent plane at (μ, v) is one half of dh where d^2h is defined by

$$d^2h = \mathbf{n} \bullet \mathbf{r}_{\mu\mu} d\mu^2 + 2\mathbf{n} \bullet \mathbf{r}_{\mu\nu} d\mu dv + \mathbf{n} \bullet \mathbf{r}_{\nu\nu} dv^2 \quad (7.5)$$

Equation (7.5) is often called the *second fundamental quadratic form* of a surface and is written in the form $Ld\mu^2 + 2Md\mu dv + Ndv^2$. The quantities

$$\begin{aligned} L(\mu, v) &= \mathbf{n} \bullet \mathbf{r}_{\mu\mu}, \\ M(\mu, v) &= \mathbf{n} \bullet \mathbf{r}_{\mu\nu}, \\ N(\mu, v) &= \mathbf{n} \bullet \mathbf{r}_{\nu\nu} \end{aligned} \quad (7.6)$$

are called the second fundamental coefficients of the surface and form the basis for defining and analyzing the curvature of a surface.

7.3 Surface Curvature

It has been established that surfaces are uniquely characterized by the first and the second fundamental coefficients which completely determine surface shape and intrinsic surface geometry [de Carmo 1976]. The *Gaussian curvature* K and *mean curvature* H are defined by

$$K = \frac{LN - M^2}{EG - F^2} \quad (7.7)$$

$$H = \frac{1}{2} \frac{EN - 2FM + GL}{EG - F^2} \quad (7.8)$$

The Gaussian curvature K and the mean curvature H have the following properties [de Carmo 1976] which are very important to surface characterization:

- Gaussian and mean curvature are invariant to arbitrary transformations of the (μ, v) -parameters of a surface.
- Gaussian and mean curvature are invariant to arbitrary rotations and translations of a surface.
- Gaussian and mean curvature are local surface properties because K and H do not depend on global properties of a surface.
- Gaussian and mean curvature indicate *surface shape* at individual surface points.
 1. $K > 0$, the surface is locally ellipsoidal, or peak shaped.
 2. if $K < 0$, the surface is locally hyperbolic, or saddle shaped.
 3. if $K = 0$ and $H \neq 0$, the surface is locally parabolic, or ridge shaped.
 4. if $K = 0$ and $H = 0$, the surface is locally flat or planar.

Therefore according to the signs of the Gaussian curvature and mean curvature, a surface can be classified into four basic types (Fig.7.1).

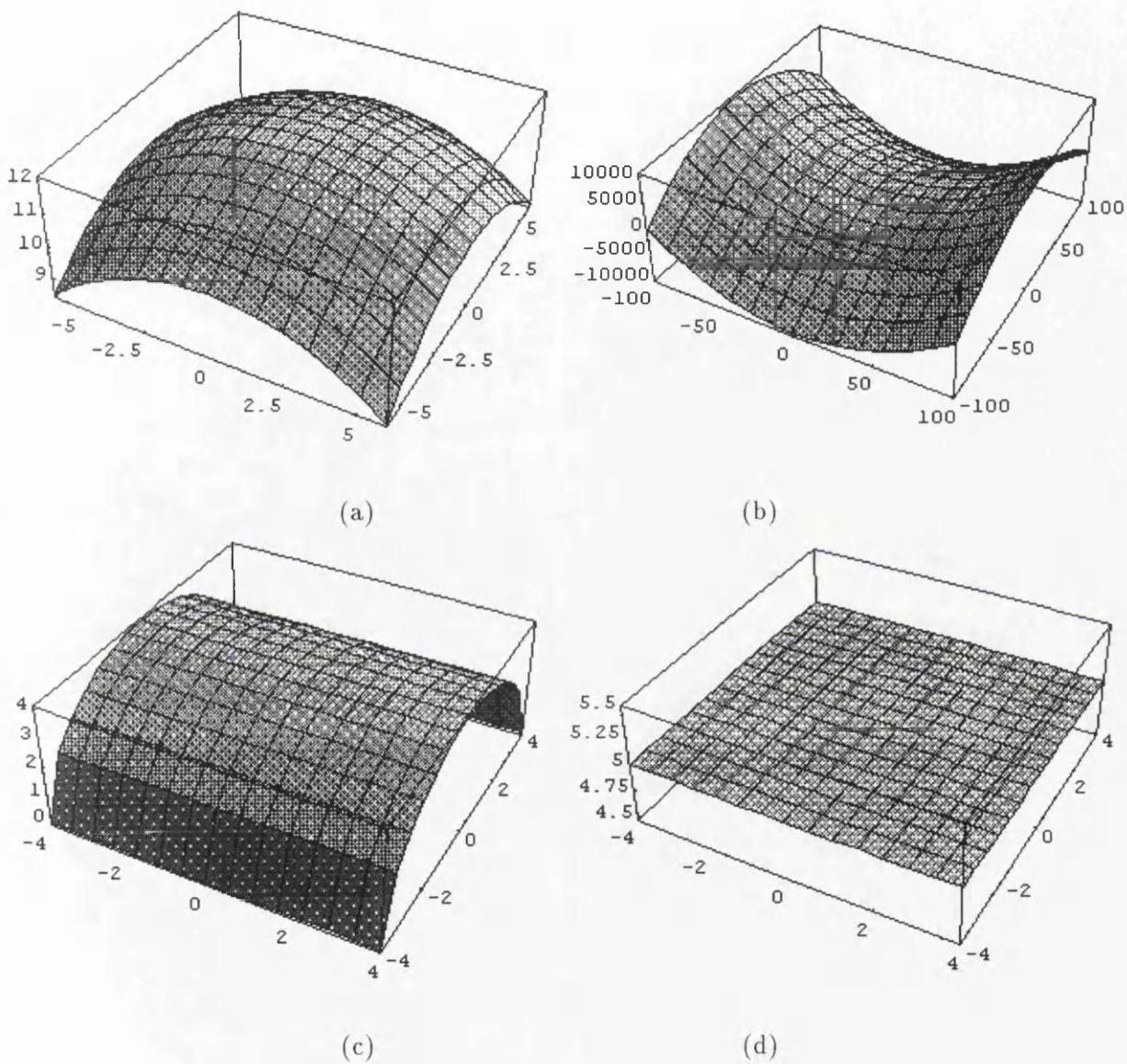


Fig.7.1 Surface type labels from surface curvature signs. (a) ellipsoidal $K > 0$; (b) hyperbolic $K < 0$; (c) parabolic $K = 0$ and $H \neq 0$; (d) planar $K = 0$ and $H = 0$.

7.4 Applications

Now we will apply the surface measurement technique to clinical MRI data. We chose the *falx cerebri* as the object of our study in this instance. The falx cerebri is a sickle-shaped fold of dura mater that lies in the midline between the two cerebral hemispheres. From the geometric point of view the deformation of this can be described in terms of concavity and convexity of a surface. Through the applications, we attempt to show the difference in the curvature of the falx cerebri between normal subjects and subjects with intracranial deformation. From the repeated scans of patients, we will show how surface deformation can change with treatment.

7.4.1 Data acquisition

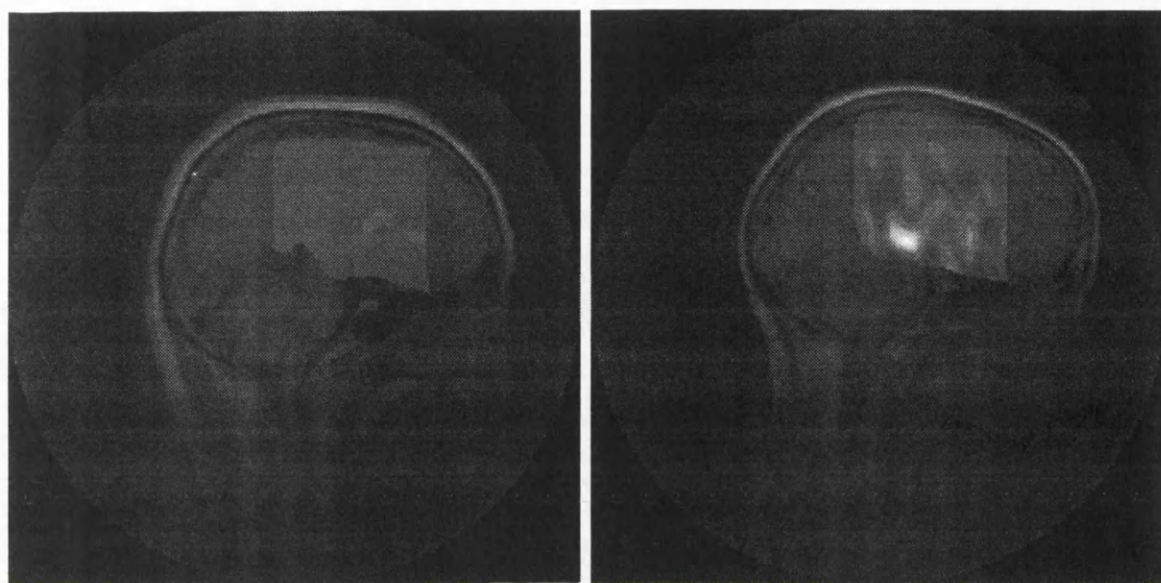
Both the normal volunteers and patients have been scanned in the coronal section with the same MR sequence (T_1 -weighted, IRSE 3400/500/32) and the same 7mm slice thickness. We chose for the curvature analysis slices between the *genu of the corpus callosum* and the *splenium of the corpus callosum* (Fig. 4.6) We chose this because most intracranial deformation tends to involve this region. Also the falx cerebri outside this region may not be accurately defined on coronal MR images because of the partial volume problem. At each coronal slice, 8 trace points along the falx were interactively obtained within the region from *superior sagittal sinus* to *supracellar cistern*, *interpeduncular cistern*, or *supracerebellar cistern*(Fig. 4.7). For

each subject, 8×11 surface trace points in total were obtained from 11 contiguous slices.

7.4.2 Measurement

Using the B-splines method described in Chapter 4, a splines surface was obtained which was a differentiable continuous function $\mathbf{P}(\mu, v)$. As described in section 7.3, the mean curvature H and the Gaussian curvature K were evaluated at the corresponding discrete points on the fit surface. Fig. 7.2 shows the values of the Gaussian curvature for each point of the fit surface approximating the falx cerebri. Fig. 7.2 (a) is from a volunteer and (b) from a patient.

Based on the Gaussian curvature and the mean curvature at each point on the fit surface, the falx cerebri is classified into the four characterisation types. The ratio of area of each curve type to the total area (*i.e.* the area of splines surface) is used as a index to characterise the surface deformation. Each ratio is called a surface character ratio.



(a)

(b)

Fig.7.2 Gaussian curvature map. The Gaussian curvature map is superposed on the falx cerebri. Each point in the map indicates the value of the Gaussian curvature at the corresponding point on the B-splines surface which fit the falx cerebri in the sagittal plane . The brighter the grey level of the pixel, the higher the value of the Gaussian curvature. (a) is from a normal volunteer and (b) is from a patient.

7.4.3 Results

The results from 15 volunteers (8 men and 7 women, from age 21 to 43) and 28 patients (16 men and 12 women, from age 17 to 65) with intracranial deformation were obtained.

Table 7.1 summarises the measurement results. It shows the mean, maximum, minimum and standard deviation of each surface character ratio for the volunteers and the patients.

the mean surface character ratio				
	planar	parabolic	elliptic	hyperbolic
volunteers	62.42%	26.12%	5.26%	6.42%
patients	27.79%	25.55%	18.94%	27.73%
the SD surface character ratio				
	planar	parabolic	elliptic	hyperbolic
volunteers	14.00%	9.30%	3.71%	5.09%
patients	10.03%	5.51%	5.44%	9.71%
the maximum surface character ratio				
	planar	parabolic	elliptic	hyperbolic
volunteers	83.11%	45.90%	12.70%	17.29%
patients	49.27%	33.59%	27.15%	46.48%
the minimum surface character ratio				
	planar	parabolic	elliptic	hyperbolic
volunteers	39.94%	7.76%	0.42%	1.46%
patients	10.89%	15.48%	8.52%	9.91%

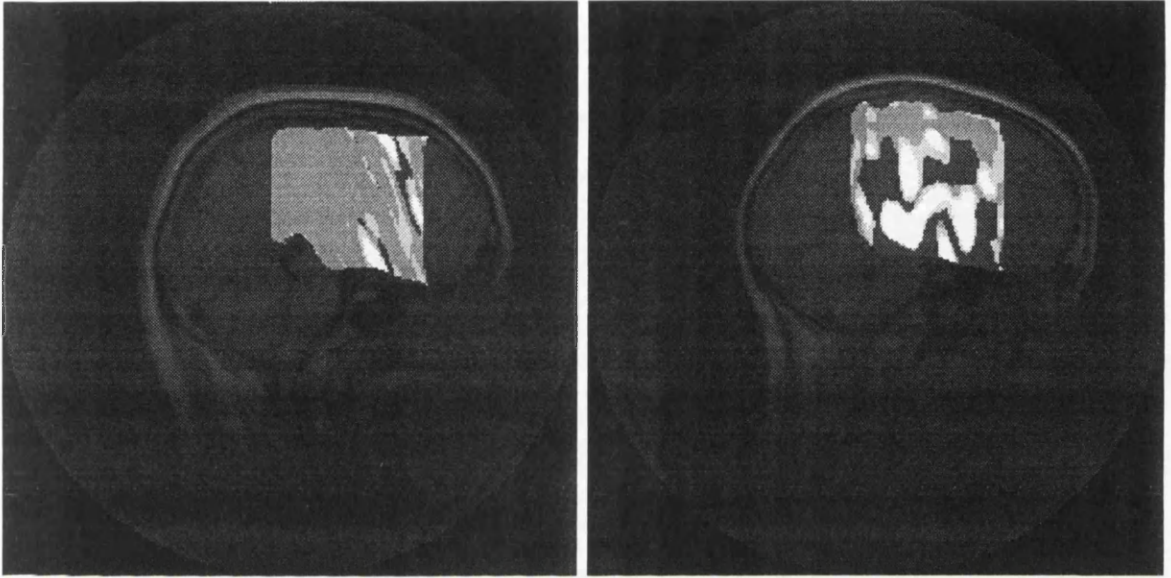
Table 7.1

From Table 7.1 we can see that the majority of the area of the falx cerebri in the volunteer group is classified as being of the planar or parabolic type. The mean ratio of planar and parabolic area to the area of falx cerebri is 88.54% for the normal group and only 53.34% for the patient group. The differences in the basic surfaces between the volunteers and patients were assessed using paired t -tests (see Table 7.2).

	planar	parabolic	elliptic	hyperbolic
t	9.3924	0.2533	8.7495	7.9127
$\alpha = .01$	3.307			
$\alpha = .005$	2.704			
$\alpha = .001$	2.423			

Table 7.2 t -tests for the variations in the four characterised surfaces between the volunteers and the patients with 41 (15+28-2) degrees of freedom. (In the table t and α indicate the test statistic and the level of significance respectively.

Fig 7.3 gives an illustrative example. In this figure the four characterised surfaces in the falx cerebri are marked according to the Gaussian curvature and the mean curvature at each point, where grey indicates the planar (flat) region, darker grey the parabolic (ridge) region, black the elliptical (peak) region, and white indicates the hyperbolic (saddle) region. (a) and (b) are from a normal volunteer and a patient with a brain tumour respectively.



(a)

(b)

Fig.7.3 Classified curvature map. The falx cerebri is classified into four basic types according to the mean curvature and the Gaussian curvature at each discrete point on the surface, namely, planar (grey), parabolic (light grey), elliptic (black) and hyperbolic (white). (a) and (b) are from a normal volunteer and patient , respectively.

In Fig. 7.3 (a) 82% of the measured region is flat, while in the case of the patient (Fig. 7.3 (b)), only 22% of the region is flat, and 30% of the region is concave toward the left hemisphere (hyperbolic); 45% of the region is convex toward the right hemisphere (ellipsoidal).

In order to demonstrate how the deformation of the falx cerebri can change with treatment, we consider MR images from two patients before and after a period of

treatment (7 days and 6 months respectively).

Table 7.3 shows the measurement results from these two patients. The data in the table are the surface character ratios. The first and second rows are the results from a patient before and then 7 days after surgical treatment respectively. The third and fourth rows are the results from a patient before and after six months of radiotherapy. From the table we can see there is a significant changes in the shape of the falx cerebri. After treatment, the falx cerebri has more of a tendency towards flatness in both patients.

	planar	parabolic	elliptic	hyperbolic
pat1 (pre)	22.35%	21.80%	20.32%	35.53%
pat1 (post)	43.79%	25.29%	11.81%	19.12%
pat2 (pre)	18.60%	21.87%	24.95%	34.59%
pat2 (post)	49.12%	36.81%	7.60%	6.47%

Table 7.3

This findings are matched by changes in clinical assessment of neurological status of the patients which shows improvement for both patients. Table 7.4 show the assessment results for the patients' treatments.

	headache	seizure	orientation	memory	intellect	motor	speech
pat1(pre)	no deficit	no deficit	no deficit	no deficit	mild confusion	no deficit	no deficit
pat1(post)	no deficit	no deficit	no deficit	no deficit	no deficit	no deficit	no deficit
pat2(pre)	no deficit	multiple daily	no deficit	no deficit	no deficit	resistance	fragmentary only
pat2(post)	responds to treatment	controlled with drug	no deficit	no deficit	no deficit	resistance	no deficit

Table 7.4

7.5 Conclusions

The main contribution of this chapter is in the development of a size-independent general purpose method for measuring the deformation of intracranial surfaces. It provides an objective, precise and practicable way to assess the intracranial deformation due to brain lesions. In the described applications we investigated only the deformation of the falx cerebri. Obviously, this method is also applicable to other anatomical structures, such as the tentorium, corpus callosum and the whole of the brain.

Section 7.4 is intended to show the potential of our analysis approach which of course could be applied to other conditions which involve surface deformation of brain

structures. Examples are given for two patients whose clinical condition has changed pre and post treatment and this has been matched by a commensurate change in our curvature parameters. The clinical assessment was based on a broad range of tests of neurological status. Through the clinical assessment in both this chapter and Chapter 6 was administered and scored by an experienced neurooncologist, these “*conventional*” tests must be considered, by their very nature, to contain an element of subjectivity. They may also be effected by drug therapies to suppress individual symptoms. Unfortunately, other than survival times, these are the only clinical means of assessing change (this problem will be discussed in more detail in the final chapter).

The quantitative parameters used to describe surface characters in the application are the ratios of the planar, parabolic, ellipsoid, and hyperbolic areas to the whole measurement surface (the falx cerebri in our application). In fact, based on the Gaussian curvature and mean curvature, we can get other surface feature parameters. Table 7.5 lists 28 such size independent attributes.

1	minimum K in the whole surface
2	maximum K in the whole surface
3	mean K in the whole surface
4	standard deviation of K in the whole surface
5	minimum H in the whole surface
6	maximum H in the whole surface
7	mean H in the whole surface
8	standard deviation of H in the whole surface
9	minimum K in the ellipsoid region
10	maximum K in the ellipsoid region
11	mean K in the ellipsoid region
12	standard deviation of K in ellipsoid region
13	ratio of <i>ellipsoid</i> – <i>area/whole</i> – <i>area</i>
14	minimum H in the parabolic region
15	maximum H in the parabolic region
16	mean H in the parabolic region
17	standard deviation of K in parabolic region
18	ratio of <i>parabolic</i> – <i>area/whole</i> – <i>area</i>
19	minimum H in the planar region
20	maximum H in the planar region
21	mean H in the planar region
22	standard deviation of K in planar region
23	ratio of <i>planar</i> – <i>area/whole</i> – <i>area</i>
24	minimum K in the hyperbolic region
25	maximum K in the hyperbolic region
26	mean K in the hyperbolic region
27	standard deviation of K in hyperbolic region
28	ratio of <i>hyperbolic</i> – <i>area/whole</i> – <i>area</i>

Table 7.5

Presentation and interpretation of data in such a format would not be comprehensible as it stands. In addition, many attributions may be equally uniformly sensitive to change (*i.e.* redundant). In Chapter 8 we will use pattern recognition techniques to select the “*best*” features from these parameters, and others from previous chapters, to reduce the multi-dimensional data into a more comprehensible two dimensional format.

Chapter 8 Classification and Interpretation

8.1 Introduction

As we know space-occupying lesions usually cause complex three-dimensional deformations and displacements. In order to characterise the global deformation, we need to consider all possible abnormalities. In the previous chapters we have developed techniques to analyse the changes in the geometric features of the hemispheres, the shape of the lateral ventricles, and the displacement of the midline structures. The aim of this chapter is to provide a comprehensive interpretation of the global deformation based on our quantitative measurements. This will allow us to objectively assess the effects of an intracranial deformation and its response to treatment. There are three stages in achieving this:

- Feature extraction. Each of our methods yields a multi-dimensional vector which describes a deformation in some cerebral structures (such as the hemispheres, the lateral ventricles, and the falx cerebri). This multi-dimensional mathematical representation of intracranial deformation is not going to be conceptually accessible to clinicians who are the ones who will ultimately have to make use of it. In addition, many attributes may be equally uniformly sensitive to change (*i.e.* highly correlated). Thus we need to optimally extract deformation *features* among the measurement attributes and reduce a large number of measurement attributes into a smaller number of features.
- Classification. In order to make a quantitative assessment, we need to set a *decision function* (or *decision rule*) which classifies a feature vector into either a *normal* or *abnormal* group. In other words, we need to set a threshold between the normal and abnormal brain based on the distribution of feature vectors from both volunteers and patients.
- Interpretation. Finally, based on the decision rule we will attempt to quantitatively assess the severity of an intracranial deformation and how it responds to treatment.

There are few studies on the quantitative measurement of intracranial deformation in current quantitative analysis of MRI data. The only related study is the investigation of intracranial herniation and the level of consciousness [Ropper 1989, Ross

1989, Reich 1993]. All the techniques used in these studies are based on simple two-dimensional linear shift so are neither accurate nor reliable in characterising a three dimensional deformation. To our knowledge, our new techniques form the most comprehensive analysis of intracranial deformation ever attempted.

In the next section, we first introduce the basic scheme for pattern recognition. Then in section 8.3 we will describe a feature extraction technique, the *Karhunen-Loève* transform, which will reduce the multi-dimensional measurement space into a smaller dimensional (two-dimension, in our case) feature space that retains most of the information needed for classification. The Karhunen-Loève transform, or the *method of principal components*, was derived originally for random processes [Karhunen 1947, Loève 1948]. The application of the Karhunen-Loève transform to feature extraction is due to Chien [Chien 1968] and Watanabe [Watanabe 1965]. The Karhunen-Loève transform developed in this section is based on statistical properties of vector representations. Section 8.4 discusses a decision rule, *Bayes decision*. The decision rule described here is optimal in the sense that it minimises either the probability of error or another quantity closely related to it. The Bayes decision developed in this section is derived originally from the Bayes classification of pattern vectors established by Marill [Marill 1960], Kanal *etal* [Kanal 1962] and Chu [Chu 1965]. Based on the measurement of the hemispheres, the lateral ventricles and the falx cerebri, in section 8.5 we set the decision rules for each of these brain structures. The decision

rules can classify a measurement object (the hemisphere, the lateral ventricle and the falx cerebri) into either the *normal category* ω_1 or the *abnormal category* ω_2 . However the decision rules do not assess how serious the deformation is. In order to assess the severity of the intracranial deformation and the effects of treatment, in section 8.6 we propose the severity scale and variation scale to characterise intracranial deformation comprehensively. Then in section 8.7 we will apply the decision rules and the deformation scales established from the analysis of our first group of 28 patients to prospectively analyse the clinical MRI data from a further group of patients to assess intracranial deformation and the effects of treatment. The significance of our approach will be demonstrated. In section 8.8 intra and inter observer reproducibility of our techniques are assessed.

8.2 Basic Pattern Recognition

The goal of pattern recognition is to classify objects of interest into one of a number of categories or *classes*. The objects of interest are generally called *patterns*. Our concern here will be a two-class pattern recognition problem, *i.e.* *normal* (ω_1) and *abnormal* (ω_2) classes. Each patient or non-patient volunteer measurement will yield a set of attributes which define a multidimensional vector \mathbf{x} . The vector \mathbf{x} is called the *measurement vector* and the multidimensional space $\Omega_{\mathbf{x}}$ in which it resides is called the *measurement space*. A *feature extractor* is a transformation \mathfrak{S} (linear or

non-linear) that maps an N -dimensional measurement vector \mathbf{x} in the measurement space $\Omega_{\mathbf{x}}$ into an M -dimensional *feature* vector \mathbf{y} in the *feature space* $\Omega_{\mathbf{y}}$.

$$\mathbf{y} = \mathfrak{F}(\mathbf{x})$$

It should be noted that since typically $M < N$, the mapping is not one-to-one. The selection of measurements is based on our prior knowledge or experience on the particular pattern recognition problem. Feature extraction is , on the other hand, essentially a scheme that reduces the dimensionality from N to M . Fig.8.1 shows the basic scheme for pattern recognition.

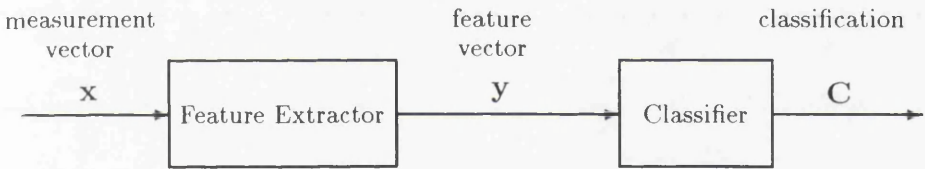


Fig.8.1 The Basic scheme for pattern recognition.

The classification at the output depends on the input feature vector \mathbf{y} , hence we write

$$C = \delta(\mathbf{y})$$

$\delta(\mathbf{y})$ is called a *decision function*, or *classifier*. The classifier essentially induces a partitioning of the feature space into a number of disjointed regions.

In our study, the *Karhunen-Loève* transform is employed as the feature extractor. It maps the multidimensional measurement space into a two-dimensional feature space

in which certain algebraic and geometric criteria are optimized.

There are two basic approaches to classification, *statistical* or *distribution free*. Distribution-free methods do not require knowledge of any a priori probability distribution functions and are based on reasoning and heuristics. Statistical classification is based on probability distribution models, such as Gaussian distributions. In our study we can assume the measurement vector \mathbf{x} has a Gaussian distribution, and also that the *normal* and *abnormal* classes are mutually complementary events. Thus we will use statistical techniques for classification.

8.3 Feature Extraction Based on Karhunen-Loève Transform

Consider a population of random measurement vectors \mathbf{x} of the form

$$\mathbf{x} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}$$

the *mean* vector of the measurement space is defined as

$$\mathbf{m}_x = E\{\mathbf{x}\}. \tag{8.1}$$

Where $E\{\mathbf{x}\}$ is the expected value of the measurement vector \mathbf{x} . The covariance

matrix of the vector population is defined as

$$\mathbf{C}_x = E\{(\mathbf{x} - \mathbf{m}_x)(\mathbf{x} - \mathbf{m}_x)^T\}. \quad (8.2)$$

where T indicates vector transposition. It should be noted that \mathbf{C}_x and $(\mathbf{x} - \mathbf{m}_x)(\mathbf{x} - \mathbf{m}_x)^T$ are matrices of order $m \times m$. Element c_{ii} of \mathbf{C}_x is the variance of x_i , the i th component of the \mathbf{x} vectors in the population, and element c_{ij} of \mathbf{C}_x is the covariance between elements x_i and x_j of these vectors. The matrix \mathbf{C}_x is real and symmetric. If elements x_i and x_j are uncorrelated, their covariance is zero, and therefore, $c_{ij} = c_{ji} = 0$.

For N vector samples from a random population, the mean vector and covariance matrix can be approximated from the samples by

$$\mathbf{m}_x \simeq 1/N \sum_{k=1}^N \mathbf{x}_k \quad (8.3)$$

and

$$\mathbf{C}_x \simeq 1/N \sum_{k=1}^N \mathbf{x}_k \mathbf{x}_k^T - \mathbf{m}_x \mathbf{m}_x^T. \quad (8.4)$$

Because \mathbf{C}_x is real and symmetric, finding a set of m orthonormal eigenvectors is always possible [Nobel 1969]. Let \mathbf{e}_i and λ_i , $i = 1, 2, \dots, m$ be the eigenvectors and corresponding eigenvalues of \mathbf{C}_x , arranged in descending order. That is

$$\mathbf{C}_x \mathbf{e}_i = \lambda_i \mathbf{e}_i, \text{ for } i = 1, 2, \dots, m$$

and

$$\lambda_i \geq \lambda_{i+1}, \text{ for } i = 1, 2, \dots, m-1.$$

Consider a transformation

$$\mathbf{u} = \mathbf{A}^T(\mathbf{x} - \mathbf{m}_x). \quad (8.5)$$

where the transformation matrix \mathbf{A} is given by

$$\mathbf{A} = (\mathbf{e}_1 \mathbf{e}_2 \dots \mathbf{e}_m)$$

whose columns are the eigenvectors of \mathbf{C}_x . It is obvious that the transform in Equation (8.5) is a linear transform which maps an m -dimensional vector to another m -dimensional vector. It does not reduce the dimensionality of the measurement space. Now we consider a transformation which does reduce the dimensionality of the measurement space.

$$\mathbf{y} = \mathbf{A}_k^T(\mathbf{x} - \mathbf{m}_x) = \begin{pmatrix} \mathbf{e}_1^T \\ \vdots \\ \mathbf{e}_k^T \end{pmatrix} (\mathbf{x} - \mathbf{m}_x) = \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_k \end{pmatrix} \quad (8.6)$$

where $k < m$, and the matrix \mathbf{A}_k is from the k eigenvectors corresponding to the k largest eigenvalues. For each component \mathbf{y}_i of \mathbf{y} , we have

$$\mathbf{y}_i = \mathbf{e}_i^T(\mathbf{x} - \mathbf{m}_x) \quad i = 1, \dots, k$$

The random variable \mathbf{y}_i is called the i th principal component of \mathbf{X} . The transformation in Equation (8.6) is called the *Karhunen-Loève (K-L) transform*, and the matrix \mathbf{A}_k is called *K-L transform matrix*. Since $k < m$, there is always a loss of information representing \mathbf{x} by \mathbf{y} . We wish to determine A such that the loss is minimised.

The K-L transform has several important properties. We shall examine some of these.

1. The principal components are uncorrelated and the variance of the i th component is λ_i . Let us first examine the covariance matrix of \mathbf{y} . This matrix is given by

$$\mathbf{C}_y = E\{(\mathbf{y} - \mathbf{m}_y)(\mathbf{y} - \mathbf{m}_y)^T\} \quad (8.7)$$

where \mathbf{m}_y is equal to the zero vector $\mathbf{0}$ as can be shown directly from Equation (8.1) and (8.5):

$$\begin{aligned} \mathbf{m}_y &= E\{\mathbf{y}\} \\ &= E\{\mathbf{A}^T(\mathbf{x} - \mathbf{m}_x)\} \\ &= \mathbf{A}^T E\{\mathbf{x}\} - \mathbf{A}^T \mathbf{m}_x \\ &= \mathbf{0} \end{aligned} \quad (8.8)$$

Substitution of Equation (8.5) and (8.8) into (8.7) yields the following expressions for \mathbf{C}_y in terms of \mathbf{C}_x :

$$\begin{aligned} \mathbf{C}_y &= E\{(\mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{m}_x)(\mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{m}_x)^T\} \\ &= \mathbf{A}E\{(\mathbf{x} - \mathbf{m}_x)(\mathbf{x} - \mathbf{m}_x)^T\}\mathbf{A}^T \\ &= \mathbf{A}\mathbf{C}_x\mathbf{A}^T \end{aligned} \quad (8.9)$$

where the last step follows from the definition of \mathbf{C}_y given in Equation (8.2).

It can be shown [Lawley 1963] that \mathbf{C}_y is a diagonal matrix with elements equal to the eigenvalues of \mathbf{C}_x ; that is,

$$\mathbf{C}_y = \begin{pmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ & & \ddots \\ 0 & & & \lambda_m \end{pmatrix} \quad (8.10)$$

and the covariance of the \mathbf{y} 's can be obtained in terms of \mathbf{A} and \mathbf{C}_x by

$$\mathbf{C}_y = \mathbf{A}^T \mathbf{C}_x \mathbf{A} \quad (8.11)$$

Since the terms of the main diagonal are 0, the elements of \mathbf{y} are *uncorrelated*. In addition, each eigenvalue λ_i is equal to the variance of the i th element of \mathbf{y} along eigenvector \mathbf{e}_i .

2. Consider Equation (8.5) and (8.6), and let us define a loss function which is the mean square error between \mathbf{x} and \mathbf{y} .

It can be shown [Young 1974] that the loss function can be given by the expression

$$\begin{aligned} \delta &= \sum_{i=1}^m \lambda_i - \sum_{i=1}^k \lambda_i \\ &= \sum_{i=k+1}^n \lambda_i \end{aligned} \quad (8.12)$$

The first part of Equation (8.12) indicates that the error is zero if $k = m$ (that is, if all the eigenvectors are used in the transformation). Because the λ_i 's decrease monotonically, Equation (8.13) also shows that the error can be minimized by selecting the

k eigenvectors associated with the largest eigenvalues. Thus the K-L transform is optimal in the sense that it minimises the mean square error between the measurement vector \mathbf{x} and its approximation (feature vector) \mathbf{y} .

3. Chien [Chien 1968] has shown that the total variance from the k principal components is the largest among all possible choices of k orthonormal vectors. In fact the first component provides the greatest variance (scattering of the points with respect to the mean value), and the second component provides the second greatest variance, and so on.

8.4 Bayes Decision

As mentioned before, we restrict attention only to the two-class (ω_1 and ω_2) case. One measure of the performance of the decision rule is the probability of making an incorrect decision or the *probability of error*. The decision rule described here is optimal in the sense that it minimizes the probability of error.

Before an object has been measured, our knowledge of it consists merely of *a priori* probabilities, $P(\omega_1)$ and $P(\omega_2)$. After measurement, however, we should be able to use the measurement and the conditional probabilities to improve our knowledge of class membership. After measurement, the so-called *a posterior* probability that the object belongs to ω_i is given by Bayes' theorem

$$p(\omega_i|\mathbf{y}) = \frac{p(\mathbf{y}|\omega_i)P(\omega_i)}{\sum_{j=1}^2 p(\mathbf{y}|\omega_j)P(\omega_j)} \quad (8.13)$$

It should be noted that ω_1 and ω_2 are mutually exclusive and collectively exhaustive. Bayes' theorem allows us to combine the *a priori probabilities* of class membership, the conditional density function and the measurement to compute, for each class, the probability that the measured object belongs to that class. Given this information, we might choose to assign each object to its most likely class.

Suppose that we are given a feature vector \mathbf{y} to be classified. In order to classify this feature vector, we might simply evaluate the posterior probability of each class $p(\omega_i|\mathbf{y})$ and choose the class with the largest posterior probability. We should assign the object to class ω_1 if

$$p(\omega_1|\mathbf{y}) \geq p(\omega_2|\mathbf{y})$$

and assign it to class ω_2 otherwise.

From Equation (8.13) we can get

$$\begin{array}{ll} \text{if} & \frac{p(\mathbf{y}|\omega_1)P(\omega_1)}{\sum_{j=1}^2 p(\mathbf{y}|\omega_j)P(\omega_j)} \geq \frac{p(\mathbf{y}|\omega_2)P(\omega_2)}{\sum_{j=1}^2 p(\mathbf{y}|\omega_j)P(\omega_j)}, \text{ choose } \omega_1 \\ \text{else} & \text{choose } \omega_2 \end{array}$$

Equivalently, we can state the decision rule as:

$$\begin{array}{ll} \text{if} & \frac{p(\mathbf{y}|\omega_1)}{p(\mathbf{y}|\omega_2)} \geq \frac{P(\omega_2)}{P(\omega_1)}, \text{ choose } \omega_1 \\ \text{else} & \text{choose } \omega_2 \end{array} \quad (8.14)$$

Let

$$\ell(\mathbf{y}) = \frac{\mathbf{p}(\mathbf{y}|\omega_1)}{\mathbf{p}(\mathbf{y}|\omega_2)},$$

the quantity $\ell(\mathbf{y})$ is called the *likelihood ratio* and the decision rule of the equation is called a *likelihoods ratio test*. The likelihood ratio is simply the ratio of the two-class conditional density functions $\mathbf{p}(\mathbf{y}|\omega_i)$ evaluated at the point \mathbf{y} . Therefore, when the class conditional densities are known, the likelihood ratio is a fixed function of \mathbf{y} .

In our case, the *normal* class ω_1 and the *abnormal* class ω_2 are mutually complementary and the prior probabilities of \mathbf{y} in both ω_1 and ω_2 can be calculated easily.

Thus we have

$$\frac{P(\omega_1)}{P(\omega_2)} = C_0$$

here C_0 is a constant. The decision rule can be written as

$$\begin{array}{ll} \text{if} & \ell(\mathbf{y}) > C_0 \text{ choose } \omega_1, \\ \text{otherwise} & \text{choose } \omega_2 \end{array} \tag{8.15}$$

Suppose that we measure n objects in class ω_i , $i = 1$, or 2 . Using Equation (8.6) and choosing $k = 2$, the measurement vector $\mathbf{x} = (x_1, x_2, \dots, x_m)$ is mapped onto a two-dimensional space with the form

$$\mathbf{y} = (y_1, y_2).$$

The mean vector and covariance matrix of \mathbf{y} are \mathbf{m} and \mathbf{C} , respectively. Using

Equation (8.9), we define

$$\mathbf{A} = \mathbf{A}^T \mathbf{C} \mathbf{A}$$

Then the class conditional density function of the two-dimensional Gaussian random vector \mathbf{y} can be defined as

$$p(\mathbf{y}|\omega_i) = \frac{1}{2\pi|\mathbf{A}|^{1/2}} \exp(-1/2(\mathbf{y} - \mathbf{m})^T \mathbf{A}^{-1}(\mathbf{y} - \mathbf{m})) \quad (8.16)$$

From the properties of the K-L transform (Eqs. (8.10) and (8.11)), we will have

$$p(\mathbf{y}|\omega_i) = \frac{1}{2\pi\sqrt{\lambda_1\lambda_2}} \exp(-1/2\mathbf{r}) \quad (8.17)$$

where $\mathbf{r} = ((y_1 - m_1)^2/\lambda_1, (y_2 - m_2)^2/\lambda_2)$, $\mathbf{y} = (y_1, y_2)$, and $\mathbf{m} = (m_1, m_2)$.

Because \mathbf{y} is obtained from a K-L transform, the elements of the \mathbf{y} vectors are uncorrelated. This means the components of \mathbf{y} are statistically independent. Therefore, from Equation (8.17), we get the Gaussian density function under condition of class ω_i ,

$$\begin{aligned} p(\mathbf{y}|\omega_1) &= \frac{1}{\sqrt{2\pi\lambda_1}} e^{-\frac{(y_1-m_1)^2}{2\lambda_1}} \frac{1}{\sqrt{2\pi\lambda_2}} e^{-\frac{(y_2-m_2)^2}{2\lambda_2}} \\ &= \frac{1}{2\pi\sqrt{\lambda_1\lambda_2}} e^{[-1/2(\frac{(y_1-m_1)^2}{\lambda_1} + \frac{(y_2-m_2)^2}{\lambda_2})]} \end{aligned} \quad (8.18)$$

where $i = 1$, or 2 .

8.5 The Decision Rules for Intracranial Structures

In order to set the decision rules between the normal and abnormal hemispheres, lateral ventricle and falx cerebri, we first consider measurement from a group of volunteers and a group of patients with intracranial deformation.

The MRI data from 15 normal volunteers and 28 patients with intracranial deformation were obtained. All the patients had clinical diagnosis of brain tumour (such as glioma). Table 8.1 gives the information about the volunteers and the patients.

	<i>No</i>	Male <i>No</i>	Female <i>No</i>	Max age	Min age	Mean age	std age
volunteers	15	8	7	43	21	32.9	6.5
patients	28	16	12	65	17	39.4	12.9

Table 8.1

Based on the measurement results from these subjects and the classification theory discussed in previous sections, we can determine the boundaries (or thresholding) between the normal and abnormal hemispheres, lateral ventricles and falx cerebri respectively.

8.5.1 The Decision Rule for the Hemispheres

In section 5.4.4, we used 28 invariant attributes to characterize the deformation of the hemispheres. After measuring all the 15 volunteers and the 28 patients, we

have 43 data sets, each of them is a 28 dimensional measurement vector. Using the K-L transform, each of the 28-dimensional measurement vectors is mapped into a 2-dimensional feature vector (See Appendix B Table B.1).

In Fig. 8.2 , all the feature vectors are plotted on a plane. The horizontal and the vertical axes indicate the first and second features respectively which are obtained by K-L transform. The circles in Fig. 8.2 are the feature vectors from the volunteers, and the triangles are the features vectors from the patients. It is very obvious that the normal feature vectors show a tendency to centralise while the patient feature vectors show a tendency to decentralise.

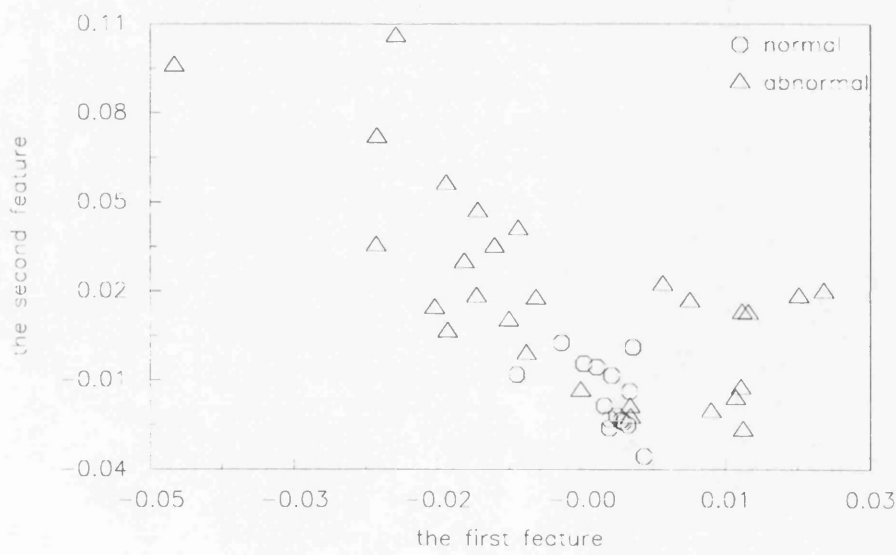


Fig.8.2 A 2-D display of the hemisphere deformation.

Based on the theory of Bayes decision and the measurement results, we can get a decision rule which will set a threshold or decision boundary between the normal and abnormal hemispheres.

Form Table B.1 in Appendix B, we get the mean feature vector and the eigenvalues of the covariance matrix for the volunteers:

$$\mathbf{m} = (m_1, m_2) = (7.247 \times 10^{-4}, -0.0157)$$

$$\lambda_1 = 8.3717 \times 10^{-6}, \quad \lambda_2 = 1.2258 \times 10^{-4}$$

Therefore, from Equation (8.18), we have Gaussian density function for the normal subjects:

$$p(\mathbf{y}|\omega_1) = \frac{1}{2\pi\sqrt{\lambda_1\lambda_2}} e^{[-1/2(\frac{(y_1-m_1)^2}{\lambda_1} + \frac{(y_2-m_2)^2}{\lambda_2})]} \quad (8.19)$$

Similarly, we can get the Gaussian density function for the abnormal subjects:

$$p(\mathbf{y}|\omega_2) = \frac{1}{2\pi\sqrt{t_1t_2}} e^{[-1/2(\frac{(y_1-n_1)^2}{t_1} + \frac{(y_2-n_2)^2}{t_2})]} \quad (8.20)$$

where $n_1 = -0.0035$, $n_2 = 0.0195$, $t_1 = 1.3558 \times 10^{-4}$, and $t_2 = 0.0012$.

From Equation (8.15), the decision rule can be stated as

$$\begin{aligned} \text{if} \quad & p(\mathbf{y}|\omega_1) = \frac{1}{2\pi\sqrt{\lambda_1\lambda_2}} e^{[-1/2(\frac{(y_1-m_1)^2}{\lambda_1} + \frac{(y_2-m_2)^2}{\lambda_2})]} > \\ & p(\mathbf{y}|\omega_2) = \frac{1}{2\pi\sqrt{t_1t_2}} e^{[-1/2(\frac{(y_1-n_1)^2}{t_1} + \frac{(y_2-n_2)^2}{t_2})]} \quad (8.21) \\ & \text{choose } \omega_1, \\ \text{otherwise} \quad & \text{choose } \omega_2. \end{aligned}$$

Equivalently, Equation (8.21) can be reduced to

$$\text{if } \ell(\mathbf{y}) \equiv \frac{(y_1 - 0.0010)^2}{5.6689 \times 10^{-5}} + \frac{(y_2 + 0.0196)^2}{8.6379 \times 10^{-4}} < 1 \quad (8.22)$$

$$\text{if } \ell(\mathbf{y}) \equiv \frac{(y_1 - 0.0010)^2}{5.6689 \times 10^{-5}} + \frac{(y_2 + 0.0196)^2}{8.6379 \times 10^{-4}} < 1 \quad (8.22)$$

otherwise choose abnormal ω_2 .

This decision rule sets a boundary between the normal and abnormal hemispheres.

And for any feature vectors \mathbf{y}_1 and \mathbf{y}_2 , if

$$\ell(\mathbf{y}_1) < \ell(\mathbf{y}_2),$$

it means that \mathbf{y}_1 is closer to the normal class ω_1 than \mathbf{y}_2 .

Applying the decision rule to all the data in Table B.1 in Appendix B, we get the likelihood ratios shown in Table 8.2.

volunteers	patients	patients
7.3341e-002	2.2897e+000	1.5239e+001
1.2038e-001	5.6583e+001	8.7940e+000
5.0266e-002	4.0022e+000	7.3141e+000
2.1769e-002	2.1326e+001	7.8744e+000
1.5247e-002	2.7006e+000	6.3811e+000
2.5457e-001	5.4141e+000	6.2388e+000
3.9310e-001	1.0328e-001	5.4177e+000
6.2074e-001	5.9674e+000	3.1570e+000
1.3634e-001	2.7952e+001	1.2054e-001
1.9765e+000	1.1795e+001	3.5247e+000
1.0402e+000	2.0066e-001	5.0576e+000
1.4239e-001	3.8571e+000	9.4803e+000
5.2803e-002	1.8410e+000	2.6340e+000
5.7574e-001	1.2330e+001	2.9236e+000
5.1455e-003		

Table 8.2 Likelihood ratios for the hemisphere.

From Table 8.2 we can see that 13 normal subjects out of 15 are in the region of *normal class* while 25 out of the 28 patients are in the region of *abnormal class*

according to the decision rule (Equation 8.22). Table 8.3 summarises the statistical results of Table 8.2. Fig.8.3 shows the histogram of the likelihood ratio for the hemisphere. This figure also shows the difference between the normal and abnormal subjects.

	mean	SD	maximum	minimum
volunteers	0.3652	0.5157	1.9765	0.0051
patients	8.5900	11.1196	56.5830	0.1033

Table 8.3 The mean, standard deviation, maximum and minimum of likelihood ratios in Table 8.2.

The variations between the likelihood ratios of volunteer and patient groups were assessed using paired *t*-tests. From Table 8.3, we calculate the test statistic as $t = 2.847$ with 41 degrees of freedom which is significant at the $\alpha = 0.005$ level (the critical value $t_{.005}=2.704$).

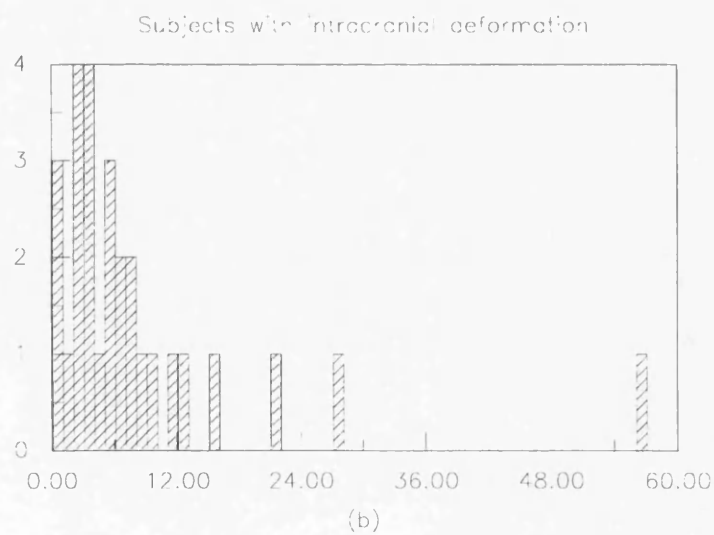
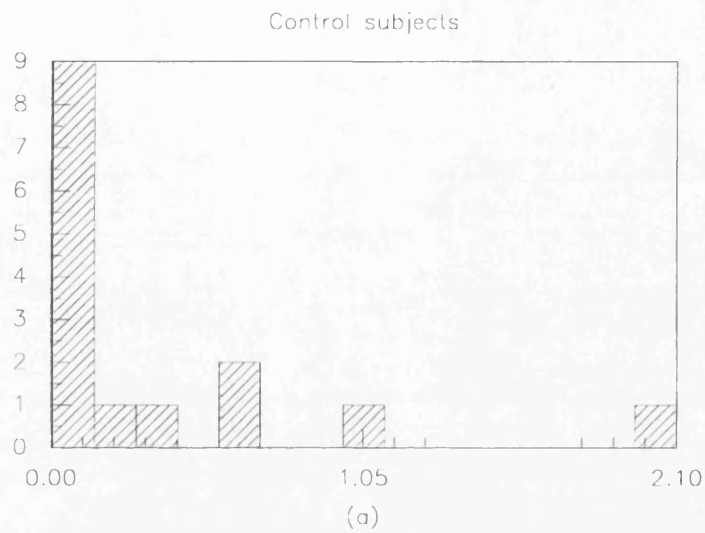


Fig.8.3 A histogram of the likelihood ratio of the hemispheres. Note the considerable change in x-axis scale between (a) and (b).

Although there are two normal subjects which are classified into the abnormal class by the decision rule, their likelihood ratios (1.9765 and 1.0402 respectively) are very close to the decision boundary (1.0000) and significant smaller than the mean likelihood ratio of the patients (8.5900). The mis-classification is due to the hemisphere asymmetry in these two normal subjects. In fact the two subjects have the highest and the second highest *hemispheres variation ratios* (0.02459 and 0.01461 respectively) among all the 15 volunteers. The mean of the hemispheres variation ratio for the volunteers is 0.00525 with standard deviation 0.00635(see Table 5.1) while the mean of the hemispheres variation ratio for the 28 patient data is 0.03556 with standard deviation 0.02539.

8.5.2 The Decision Rule for the Lateral Ventricles

In Chapter 6, we used Fourier descriptors to describe the shape of the lateral ventricles. For each measurement subject, there are 6 parameters to describe the shape of the lateral ventricles.

After measuring all the volunteers and the patients, we have 43 data sets each of them providing a 6 dimensional measurement vector. Using the K-L transform, each of the 6-dimensional measurement vectors is mapped into a 2-dimensional feature vector (See Appendix B Table B.2).

In Fig. 8.4, all the feature vectors are plotted on a plane. The circles in Fig. 8.4

are the feature vectors from the volunteers, and the triangles are the features vectors from the patients. It is very obvious that all the feature vectors shows a strong tendency towards separation between the normal and abnormal lateral ventricles.

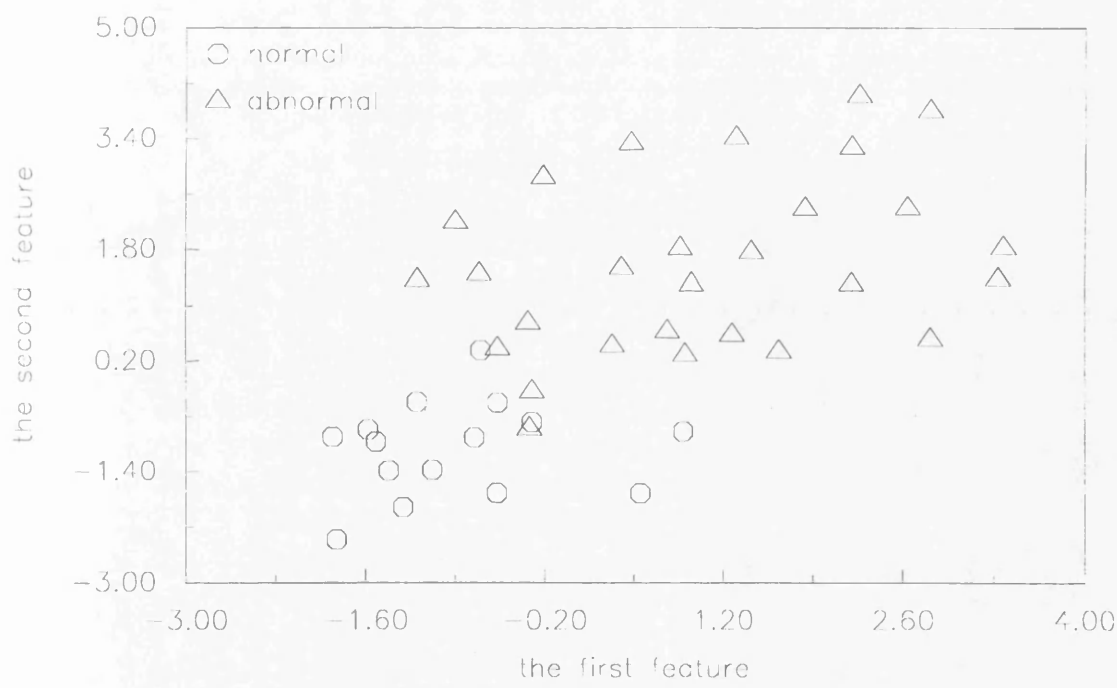


Fig.8.4 A 2-D display of the lateral ventricular deformation.

Form Table B.2, we get the mean feature vector and the eigenvalues of the covariance matrix for the normal subjects:

$$\mathbf{m} = (m_1, m_2) = (-0.8836, -1.0718)$$

$$\lambda_1 = 0.6411, \quad \lambda_2 = 0.4140$$

Therefore, from Equation (1.18), we have Gaussian density function for the normal subjects:

$$p(\mathbf{y}|\omega_1) = \frac{1}{2\pi\sqrt{\lambda_1\lambda_2}} e^{[-1/2(\frac{(y_1-m_1)^2}{\lambda_1} + \frac{(y_2-m_2)^2}{\lambda_2})]}$$

Similarly, we can get the Gaussian density function for the abnormal subjects:

$$p(\mathbf{y}|\omega_2) = \frac{1}{2\pi\sqrt{t_1t_2}} e^{[-1/2(\frac{(y_1-n_1)^2}{t_1} + \frac{(y_2-n_2)^2}{t_2})]} \quad (8.23)$$

where $n_1 = 1.0431$, $n_2 = 1.5767$, $t_1 = 2.1703$, and $t_2 = 1.0662$.

From Equation (8.15), the decision rule can be stated as

$$\begin{aligned} \text{if} \quad p(\mathbf{y}|\omega_1) &= \frac{1}{2\pi\sqrt{\lambda_1\lambda_2}} e^{[-1/2(\frac{(y_1-m_1)^2}{\lambda_1} + \frac{(y_2-m_2)^2}{\lambda_2})]} > \\ p(\mathbf{y}|\omega_2) &= \frac{1}{2\pi\sqrt{t_1t_2}} e^{[-1/2(\frac{(y_1-n_1)^2}{t_1} + \frac{(y_2-n_2)^2}{t_2})]} \end{aligned} \quad (8.24)$$

choose ω_1 ,

otherwise choose ω_2 .

Equivalently, Equation (8.24) can be reduced to

$$\begin{aligned} \text{if} \quad \frac{(y_1+1.6913)^2}{13.9695} + \frac{(y_2+2.7530)^2}{10.3870} &< 1 \\ &\text{choose normal } \omega_1, \\ \text{otherwise} \quad &\text{choose abnormal } \omega_2. \end{aligned} \quad (8.25)$$

For any feature vectors \mathbf{y}_1 and \mathbf{y}_2 , if

$$\ell(\mathbf{y}_1) < \ell(\mathbf{y}_2),$$

then \mathbf{y}_1 is closer to the normal class ω_1 than \mathbf{y}_2 . In other words, the lateral ventricles with the feature vector \mathbf{y}_1 are more likely to approach the shape of lateral ventricles from the normal volunteers than the lateral ventricles with the feature vector \mathbf{y}_2 .

volunteers	patients	patients
3.0711e-001	3.4374e+000	1.7665e+000
3.6875e-001	5.0590e-001	1.2556e+000
1.4583e-002	1.6458e+000	3.8571e+000
3.3157e-001	1.3671e+000	2.4686e+000
1.9243e-001	1.7022e+000	2.6654e+000
6.1753e-001	3.4491e+000	3.8946e+000
9.9637e-001	5.5343e+000	4.5781e+000
4.6267e-001	9.8047e+000	2.6566e+000
3.8935e-001	1.3086e+000	4.2973e+000
8.3435e-001	2.4751e+000	3.9011e+000
1.8373e-001	1.5252e+000	3.1685e+000
2.0776e-001	1.6873e+000	2.4073e+000
7.7311e-002	7.4042e-001	2.0799e+000
5.4730e-001	5.5830e+000	2.0828e+000
5.5031e-001		

Table 8.4 Likelihood ratios for the lateral ventricles.

Applying the decision rule (Equation 8.25) to all the feature vectors in Table B.2 in Appendix B, the likelihood ratios of the feature vectors are obtained (Table 8.4)

From Table 8.4 we can see that all the normal subjects are cataloged into the *normal class* ω_1 , and 26 out of 28 patients are in the *abnormal class* ω_2 according to the decision rule (Equation 8.25). The patient classified into the normal class had a small rounded area lesion lying in the medial left parietal lobe. There is no any space occupying effects on the lateral ventricles (see Fig 8.5).

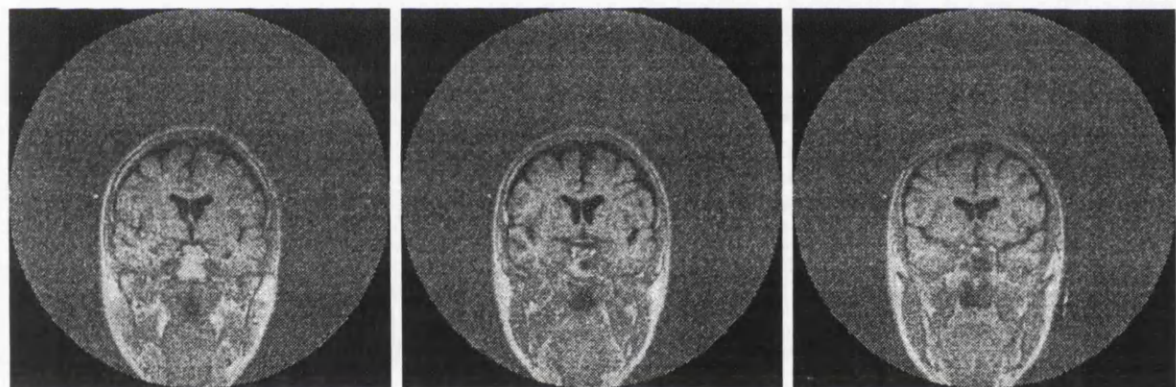


Fig. 8.5 Images of the lateral ventricles for the patient classified into the normal category.

Table 8.5 shows the mean, standard deviation, maximum and minimum of likelihood ratio of the lateral ventricle for both the volunteers and the patients. Fig. 8.6 shows the histogram of the likelihood ratio and the decision boundary.

The variations between the likelihood ratios of volunteer and patient groups were assessed using paired *t*-tests. From Table 8.5, we can calculate the test statistic as

$t = 6.2102$, using 41 ($15 + 28 - 2$) degrees of freedom, which is significant at the $\alpha = 0.001$ level (the critical value $t_{.001} = 3.307$).

	mean	SD	maximum	minimum
volunteers	0.4054	0.2625	0.9964	0.0146
patients	2.6096	1.3538	5.5830	0.5059

Table 8.5

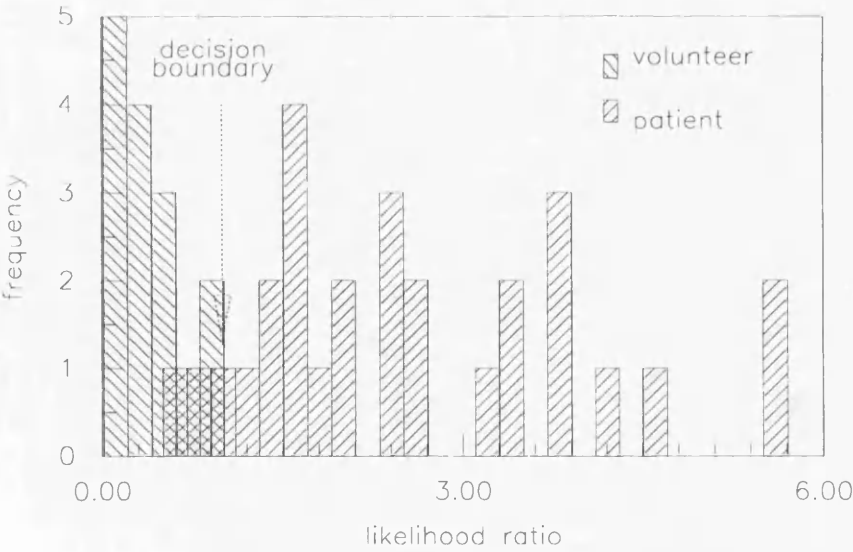


Fig.8.6 A histogram of likelihood ratio for lateral ventricles.

In order to aid in the intuitive understanding of the likelihood ratio, we consider the case of the patient who had the maximum value of $\ell(\mathbf{y})$ among all the patients. The patient suffered a malignant neoplasm in the right hemisphere. Fig. 8.7 shows a

significant displacement of the lateral ventricles and midline shift.

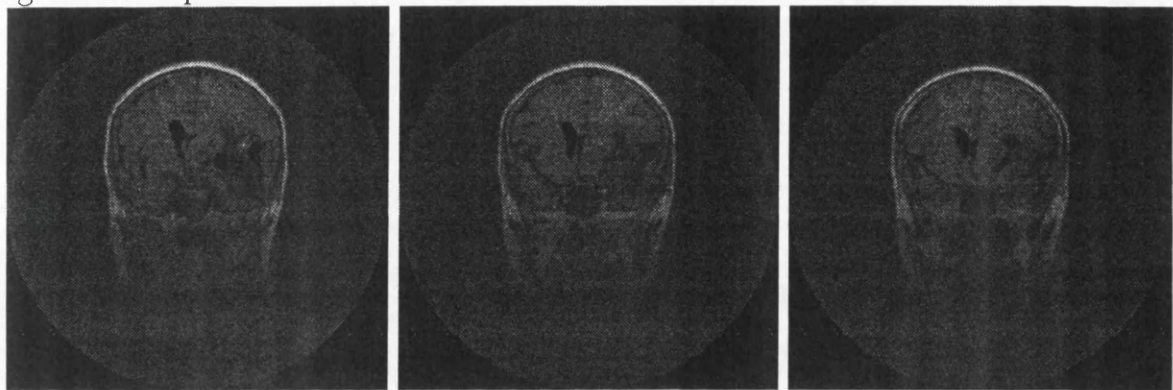


Fig.8.7 Three coronal section images from the patient who has the maximum value of $\ell(\mathbf{y})$ among all the patient.

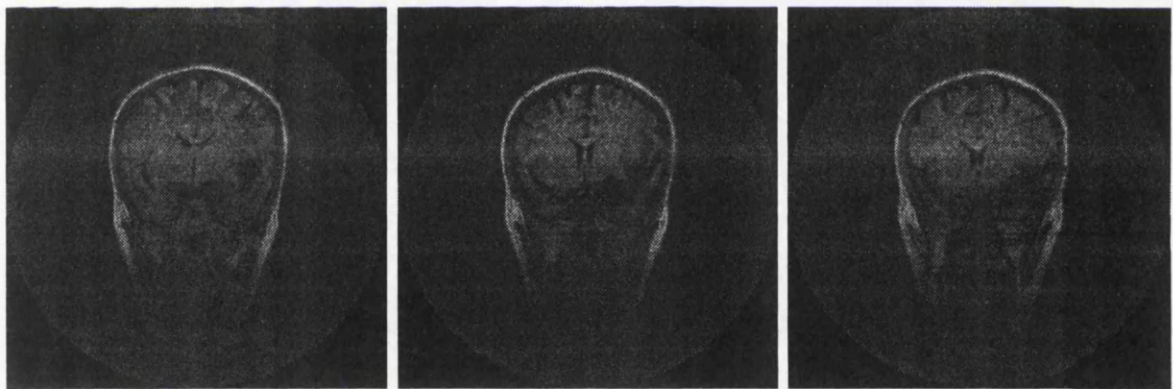


Fig.8.8 Three coronal section images of the patient who has the minimum value of $\ell(\mathbf{y})$ among all the patient.

Comparing the shape of the lateral ventricles of this patient with the patient who has the minimum value of $\ell(\mathbf{y})$ among all the patients (see Fig. 8.8) and the volunteer shown in Fig 6.1, we can see the difference in the shape deformation.

8.5.3 The Decision Rule for the Falx Cerebri

The deformation of the falx cerebri has been measured by means of Gaussian curvature and the mean curvature as described in Chapter 7. For each measurement subject, a 28-dimensional measurement vector is obtained to describe the shape of the falx cerebri. After measuring all the volunteers and the patients, we have 43 data sets each of them making a 28 dimensional measurement vector. Using the K-L transform, each of the 28-dimensional measurement vectors is mapped into a 2-dimensional feature vector (See Appendix B Table B.3).

In Fig. 8.9 , all the feature vectors are plotted on a plane. The circles in Fig. 8.9 are the feature vectors from the volunteers, and the triangles are the features vectors from the patients. It is obvious that all the feature vectors show a tendency towards separation between the normal and abnormal falx cerebri.

Form Table B.3 in Appendix B, we get the mean feature vector and the eigenvalues of the covariance matrix for the normal subjects:

$$\mathbf{m} = (m_1, m_2) = (-0.0989, -9.7576 \times 10^{-5})$$

$$\lambda_1 = 0.0038, \quad \lambda_2 = 8.8353 \times 10^{-9}$$

Therefore, from Equation (1.18), we have Gaussian density function for the normal subjects:

$$p(\mathbf{y}|\omega_1) = \frac{1}{2\pi\sqrt{\lambda_1\lambda_2}} e^{[-1/2(\frac{(y_1-m_1)^2}{\lambda_1} + \frac{(y_2-m_2)^2}{\lambda_2})]}$$

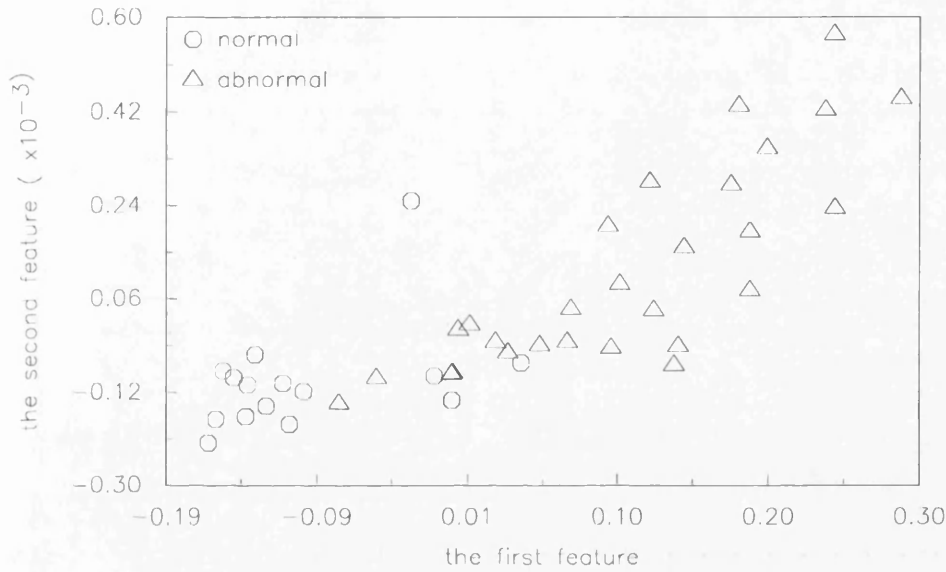


Fig.8.9 A 2-D display of the falx cerebri deformation.

Similarly, we can get the Gaussian density function for the abnormal subjects:

$$p(\mathbf{y}|\omega_2) = \frac{1}{2\pi\sqrt{t_1 t_2}} e^{[-1/2(\frac{(y_1 - n_1)^2}{t_1} + \frac{(y_2 - n_2)^2}{t_2})]} \quad (8.26)$$

where $n_1 = 0.1096$, $n_2 = 1.1277 \times 10^{-4}$, $t_1 = 0.0089$, and $t_2 = 1.1997 \times 10^{-8}$.

From Equation (8.15), the decision rule can be stated as

$$\begin{aligned} \text{if } p(\mathbf{y}|\omega_1) &= \frac{1}{2\pi\sqrt{\lambda_1 \lambda_2}} e^{[-1/2(\frac{(y_1 - m_1)^2}{\lambda_1} + \frac{(y_2 - m_2)^2}{\lambda_2})]} > \\ p(\mathbf{y}|\omega_2) &= \frac{1}{2\pi\sqrt{t_1 t_2}} e^{[-1/2(\frac{(y_1 - n_1)^2}{t_1} + \frac{(y_2 - n_2)^2}{t_2})]} \end{aligned} \quad (8.27)$$

choose ω_1 ,

otherwise choose ω_2 .

Equivalently, Equation (8.27) can be reduced to

$$\begin{aligned}
 &\text{if } \frac{(y_1+0.2569)^2}{0.1603} + \frac{(y_2+6.8535 \times 10^{-4})^2}{7.9444 \times 10^{-7}} < 1 \\
 &\quad \text{choose normal } \omega_1, \\
 &\text{otherwise choose abnormal } \omega_2. \tag{8.28}
 \end{aligned}$$

For any feature vectors \mathbf{y}_1 and \mathbf{y}_2 , if

$$\ell(\mathbf{y}_1) < \ell(\mathbf{y}_2),$$

then \mathbf{y}_1 is closer to the normal class ω_1 than \mathbf{y}_2 . In other words, the falx cerebri with the feature vector \mathbf{y}_1 is more flat than the falx cerebri with the feature vector \mathbf{y}_2 .

Table 8.6 shows the likelihood ratios which are obtained by applying the decision rule (Equation 8.28) to all the feature vectors for the falx cerebri.

volunteers	patients	patients
8.1221e-001	5.7277e-001	2.3507e+000
6.0945e-001	2.6384e+000	2.2089e+000
5.3158e-001	8.5668e-001	1.9746e+000
5.1307e-001	8.5134e-001	1.9167e+000
5.2111e-001	3.4594e+000	2.0957e+000
1.0394e+000	1.2221e+000	1.5395e+000
3.9445e-001	2.7736e+000	1.7715e+000
7.7973e-001	3.5431e+000	1.3263e+000
1.4175e+000	1.5719e+000	1.3424e+000
5.5693e-001	1.5813e+000	1.1391e+000
4.2722e-001	7.0180e-001	1.0433e+000
5.5505e-001	1.4684e+000	1.0388e+000
4.5490e-001	3.0808e+000	1.0026e+000
3.3166e-001	2.6585e+000	1.0441e+000
4.7449e-001		

Table 8.6 Likelihood ratios for the falx cerebri.

As shown in Table 8.6, 13 feature vectors out of 15 from the volunteers belong to the *normal class* ω_1 , and 24 feature vectors out of 28 from the patients are in

the *abnormal class* ω_2 . The mean, standard deviation, maximum and minimum of likelihood ratio of the falx cerebri for both the volunteers and the patients are shown in Table 8.7. Fig. 8.10 shows the histogram of the likelihood ratio and the decision boundary.

	mean	SD	maximum	minimum
volunteers	0.6279	0.2745	1.4175	0.3317
patients	1.7419	0.8166	3.5431	0.5728

Table 8.7

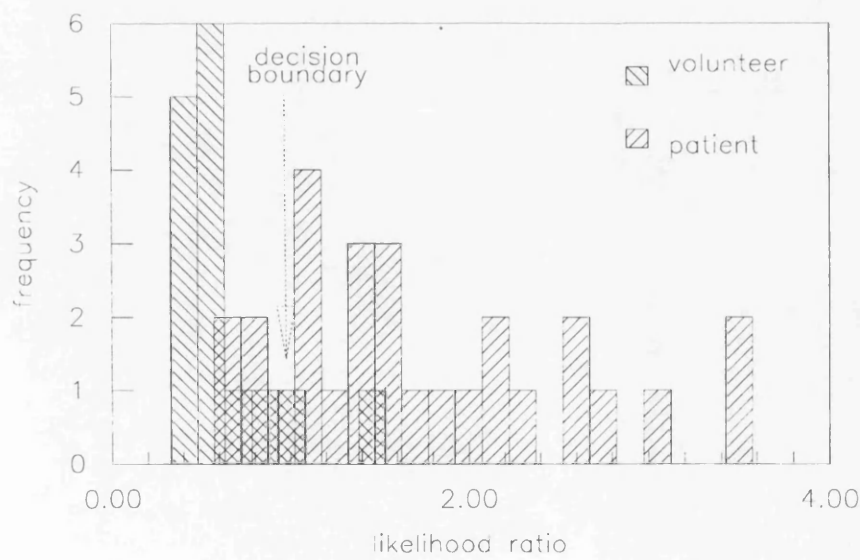


Fig.8.10 A histogram of likelihood ratio for the falx cerebri.

The variations between the likelihood ratios of volunteer and patient groups were assessed using paired t -tests. From Table 8.5, we can calculate the test statistic as $t = 5.0147$, using 41 ($15 + 28 - 2$) degrees of freedom, which is significant at the $\alpha = 0.001$ level (the critical value $t_{.001} = 3.307$).

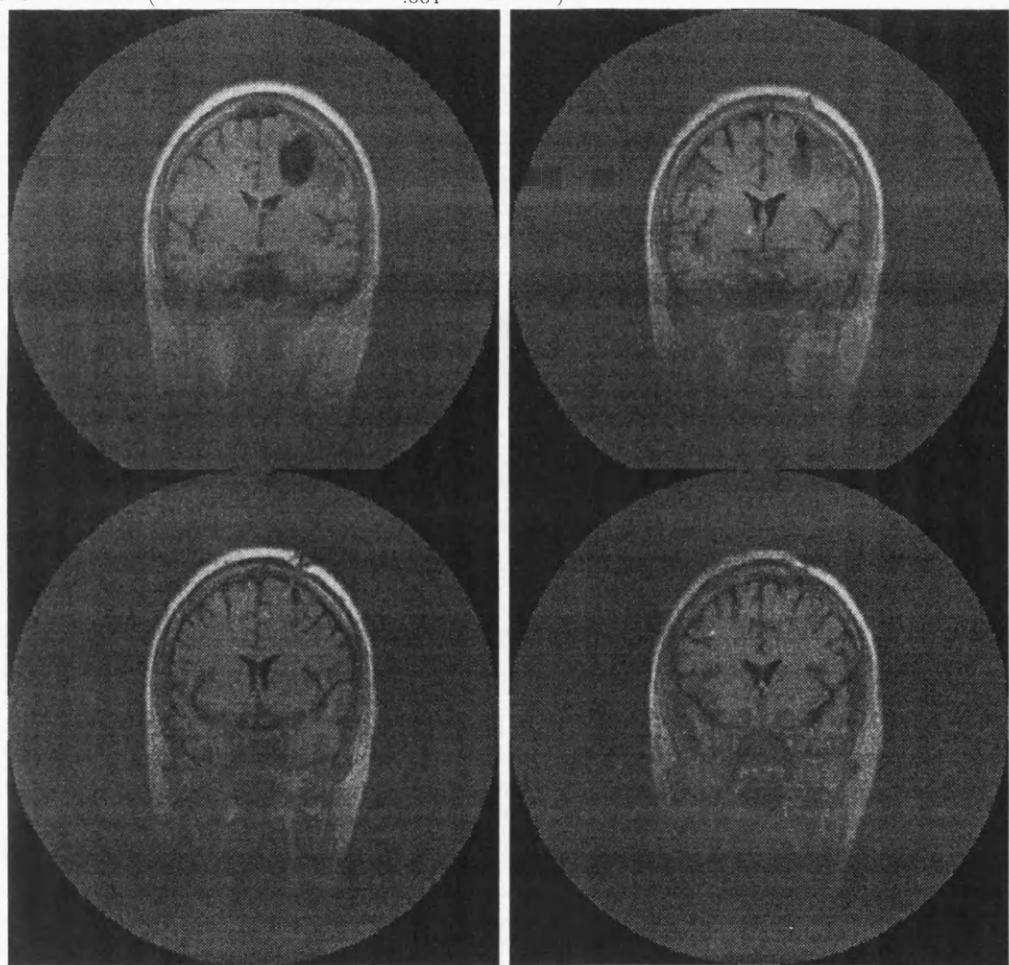


Fig.8.11 Four coronal images from the patient who has the minimum value of $\ell(\mathbf{y})$ among all the patient in the measurement of the falx cerebri.

Shift of the midline did not occur for all the patients. In the images of the patients

who have the feature vectors classified into the normal class ω_1 , there is no visible shift of the midline. Fig. 8.11 shows four coronal images from the patient with the minimum value of $\ell(\mathbf{y})$. Comparing this with the images from the patient who has the maximum value of $\ell(\mathbf{y})$ shows a significant shift of the midline (Fig. 8.12)

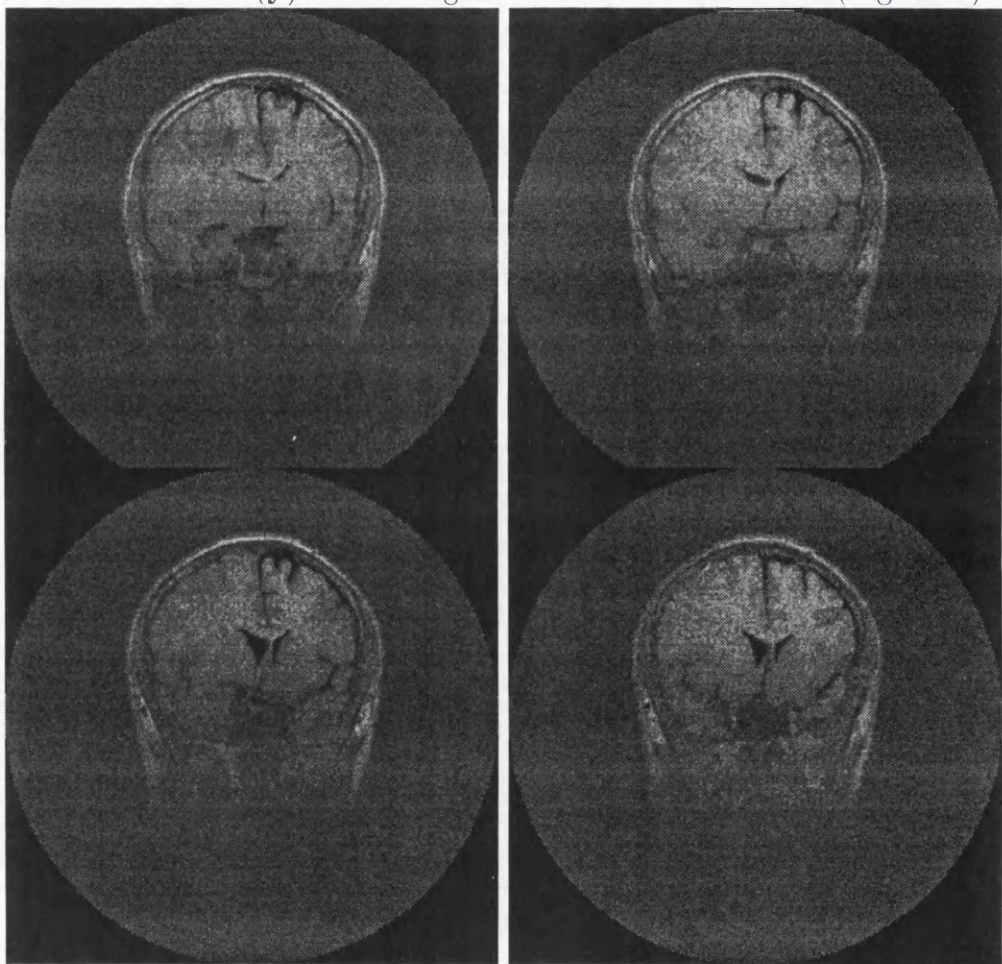


Fig.8.12 Four coronal images from the patient who has the maximum value of $\ell(\mathbf{y})$ among all the patient in the measurement of the falx cerebri.

8.6 Scales of Intracranial Deformation

Our measurement objects (the hemisphere, the lateral ventricle, or the falx cerebri) can now be classified either into the normal category (ω_1) or into the abnormal category (ω_2) by the decision rules. However the decision rules do not assess how serious any distortion is. In order to assess the degree of intracranial deformation, we need to analyse the likelihood ratio further.

From the definition of the likelihood ratio $\ell(\mathbf{y})$ we know that the likelihood ratio provides information about intracranial deformation. The larger the likelihood ratio is, the larger the deformation. In this section we describe how to characterise the degree of the intracranial deformation and use this to assess the response of the deformation to treatment.

The severity scale of intracranial deformation

Suppose that \mathbf{x} is a measurement vector for some brain structure \mathcal{S} (the hemisphere, the lateral ventricle, or the falx cerebri) and \mathbf{y} is the feature vector obtained by K-L transform through x . The mean of the likelihood ratio for each of the three brain structures has been calculated for both volunteer and patient groups in previous subsections (Table 8.3, 8.5, and 8.7). Let \bar{a} and \bar{b} be the mean of likelihood ratio of the brain structure \mathcal{S} for volunteers and patients respectively. Then if $\ell(\mathbf{y}) > \bar{b}$, we say that the deformation of \mathcal{S} is *worse*; if $1 \leq \ell(\mathbf{y}) \leq \bar{b}$, we say that the deformation

of \mathcal{S} is *median*; if $\bar{a} \leq \ell(\mathbf{y}) < 1$, we say that the deformation of \mathcal{S} is *small* ; if $\ell(\mathbf{y}) < \bar{a}$, we say that the deformation of \mathcal{S} is *none* (See Fig. 8.13).

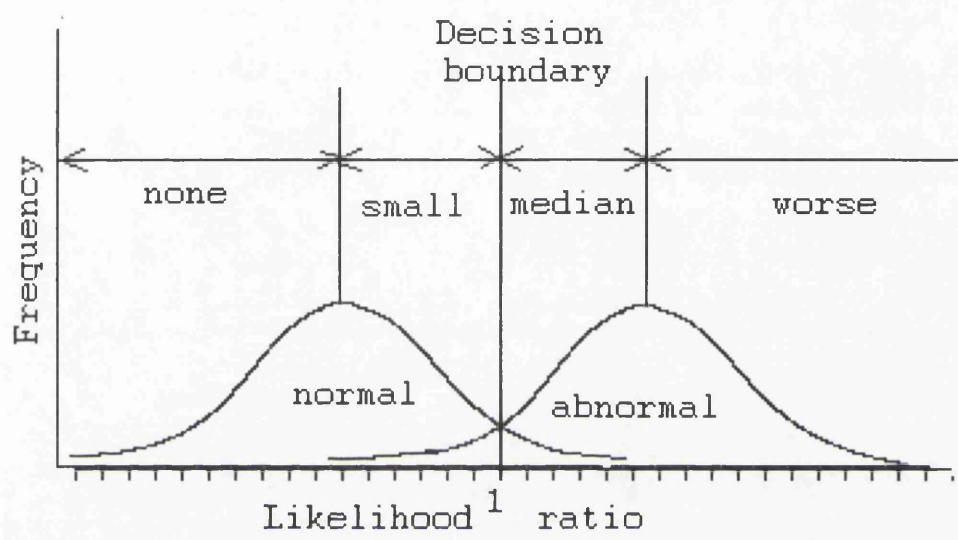


Fig.8.13 The severity scale of intracranial deformation.

This description for the degree of the intracranial deformation is called the *severity scale of intracranial deformation*. (SSID) Alternatively we can use a digital representation for the severity scale of intracranial deformation:

SSID	none	small	median	worse
Score	0	1	2	3

The variation scale of intracranial deformation

Suppose we consider a brain structure \mathcal{S} (the hemisphere, the falx cerebri, or the lateral ventricle). Let $\ell(\mathbf{y}_1)$ and $\ell(\mathbf{y}_2)$ be the likelihood ratios before and after treatment respectively. Let σ be the standard deviation of the likelihood ratio for volunteers as shown in Table 8.3, 8.5, or 8.7. If $\ell(\mathbf{y}_1) - \ell(\mathbf{y}_2) > \sigma$, we say that the deformation shows a tendency of decrease; if $\ell(\mathbf{y}_1) - \ell(\mathbf{y}_2) < -\sigma$, we say that the deformation shows a tendency of increase; otherwise we say that the deformation is stable. This description of the change of intracranial deformation is called the *variation scale of intracranial deformation* (VSID). Similarly we also use digital representation for the change scale of intracranial deformation:

VSID	increase	stable	decrease
Score	1	0	-1

8.7 Applications

Based on the quantitative measurements from 15 volunteers and 28 patients, we have established the decision rules to characterise intracranial deformation by measuring

the hemispheres, the lateral ventricle and the falx cerebri. Now we will apply these decision rules prospectively to a further group of patients to try to quantitatively assess the effects of intracranial deformation and its response to treatment.

8.7.1 Patients

11 patients (not part of the previous 28 patient group) with brain tumours were analysed in the application. Thus the MRI data sets from the 11 patients are totally independent from the previous analyses. The group include five men and six women from age 17 to 57 (with mean of age 39.82 and standard deviation of 12.13). The patients have been scanned before and after treatment (radiotherapy, chemotherapy, surgery or combination of these), in order to assess lesion response. As described previously the subjects were scanned in coronal section with the same T_1 sequence (IRSE 3400/500/32), the same 7mm slice thickness, and the same 24 contiguous slices encompassing the whole head. The quantitative analysis were performed in a blinded manner, then compared with the clinical assessment of change.

8.7.2 Results

For each subject we measure the deformations in the hemispheres, the lateral ventricle, and the falx cerebri with the techniques described in chapter 5, 6 and 7. Three measurement vectors are obtained for each of the patients, a 28-dimensional vector for

the hemisphere, a 6-dimensional vector for the lateral ventricle, and a 28-dimensional vector for the falx cerebri.

After measurement of all the patients, each measurement vector is reduced into a two-dimensional feature space by applying the K-L transform. Using the transform matrix \mathbf{Y} given in Table B.2 in Appendix B, for example, each of the 6-dimensional measurement vectors for the lateral ventricle is mapped into a 2-dimensional feature vector which provides most information about the deformation of the lateral ventricle.

Finally applying each of the decision rules (Equation 8.22, 8.25, and 8.28) to each of the feature spaces (the hemisphere, the lateral ventricle, and the falx cerebri) respectively, we will get the likelihood ratio for each of the three brain structures. Table 8.8 shows all the likelihood ratios for the hemisphere, the lateral ventricle, and the falx cerebri.

	<i>before treatment</i>						<i>after treatment</i>		
patient	sex	age	hemisp	falx c	ventri	interval (d)	hemisp	falx c	ventri
P_1	F	29	2.4100	0.4371	2.2420	223	1.0937	0.5154	0.8138
P_2	F	17	65.355	3.6470	1.4268	99	62.996	3.2010	1.2849
P_3	F	33	4.3105	0.8320	1.4630	192	2.6185	0.8244	1.3750
P_4	M	55	23.882	5.1505	3.0680	7	3.2036	1.3715	1.1230
P_5	F	50	5.8333	4.0649	1.2587	8	0.0672	1.8931	1.1948
P_6	M	34	13.104	1.9230	1.0385	67	12.149	1.6826	0.7077
P_7	M	42	0.3723	1.9226	0.8660	288	0.6337	2.0098	0.9603
P_8	M	57	2.4913	1.7463	1.1607	268	1.8874	1.8024	1.0188
P_9	F	29	0.2649	2.2478	0.8454	240	0.5002	3.7978	0.9923
P_{10}	F	40	0.1370	1.4031	1.1607	57	0.4964	1.3982	1.0188
P_{11}	M	52	29.229	2.3858	0.9920	14	14.005	2.0102	1.0100

Table 8.8 Likelihood ratios of the hemisphere, the falx cerebri and the ventricle.

8.7.3 Clinical Assessments

Separately and in a blinded manner from the MRI quantitative measurements, clinical assessment of neurological condition of the patients was performed. The clinical symptoms assessed included *headache, seizure, orientation, memory, intellect,*

motor, speech, vision, and papilloedema. The overall assessment used here is the *Modified MRC neurological status scale* (MMRC scale). It is described as:

0	No deficit.
1	Mild deficit but function adequate for useful work.
2	Moderate functional impairment(able to care for self most of the time).
3	Substantial impairment (requiring help with self care most of the time, severe dysphasia).
4	Major impairment(confined to bed or chair whole time, total dysphasia).
5	No useful function.

Table 8.9 shows the overall assessment of MMRC and the status of the patients. This was scored by an experienced neurooncologist Dr Rampling of the Beatson Oncology Centre, Western Infirmary, Glasgow.

	<i>P</i> ₁	<i>P</i> ₂	<i>P</i> ₃	<i>P</i> ₄	<i>P</i> ₅	<i>P</i> ₆
before	1	2	2	0	1	2
after	2 (worse)	1 (better)	2 (worse)	0 (stable)	0 (better)	1 (better)
status	alive	dead	alive	dead	alive	dead
	<i>P</i> ₇	<i>P</i> ₈	<i>P</i> ₉	<i>P</i> ₁₀	<i>P</i> ₁₁	
before	1	1	1	1	2	
after	1 (stable)	3 (worse)	0 (better)	1 (stable)	3 (worse)	
status	alive	dead	alive	alive	dead	

Table 8.9 MMRC assessment and patients’ status.

Assessment of the severity of intracranial deformation

Using the severity scale of intracranial deformation and the information from Table 8.3, 8.5, and 8.7, we can represent the results in Table 8.8 more simply (Table 8.10).

	<i>before treatment</i>				<i>after treatment</i>			
patient	hemisp	falx c	ventri	TS	hemisp	falx c	ventri	TS
P_1	m (2)	n (0)	m (2)	4	m (2)	s (1)	s (1)	4
P_2	w (3)	w (3)	m (2)	8	w (3)	w (3)	m (2)	8
P_3	m (2)	s (1)	m (2)	5	m (2)	s (1)	m (2)	5
P_4	w (3)	w (3)	w (3)	9	m (2)	m (2)	m (2)	6
P_5	m (2)	w (3)	m (2)	7	n (0)	w (3)	m (2)	5
P_6	w (3)	w (3)	m (2)	8	w (3)	m (2)	s (1)	6
P_7	s (1)	w (3)	s (1)	5	s (1)	w (3)	s (1)	5
P_8	m (2)	w (3)	m (2)	7	m (2)	w (3)	m (2)	7
P_9	n (0)	w (3)	s (1)	4	s (1)	w (3)	s (1)	5
P_{10}	n (0)	m (2)	m (2)	4	s (1)	m (2)	m (2)	5
P_{11}	w (3)	w (3)	s (1)	7	w(3)	w(3)	m(2)	8

Table 8.10 The deformations of the hemisphere, the falx cerebri and the ventricle described by SSID (w , m , s , n and TS are the abbreviations for *worse*, *median*, *small*, *none* and *total scores* respectively).

From Table 8.9 and 8.10 we can see that, with one exception, all the patients (P_2 , P_4 , P_5 , P_6 , P_8 and P_{11}) with total SSID scores (either before or after treatment) larger than 5, did not survive. The only exception was P_5 whose total SSID scores dropped rapidly from 7 to 5 seven days after surgery. Patient P_2 had the longest survival time of eight months after the last MR scan. The others (P_4 , P_6 , P_8 and P_{11}) died less than six months after the last MR scan. The rest of the patients (P_1 , P_3 , P_7 , P_9 and P_{10}) with total SSID scores (either before or after treatment) less than 6 still survive at the time of the writing of this thesis (*i.e.* all have survival longer than 8 months).

The results indicate that the total score of SSID does indeed characterise the severity of intracranial deformation. Higher SSID scores, in this group at least, are associated with lower chances of survival.

The limitations of the clinical assessment are illustrated in the cases of the non-surviving patients (Table 8.9). This showed that only two patients (P_8 and P_{11}) were getting clinically worse, two patients (P_2 and P_6) were getting better, and one patient (P_4) was stable. This would suggest that our approach is more objective and accurate than the clinical assessment.

Patient P_1 has the smallest total SSID scores (both before and after treatment) among all the patients (Table 8.10). The 30 years old female patient was diagnosed with a neoplasm in her right hemisphere in early 1991. Fig.8-14 shows a sequence of

coronal MR images of her brain.

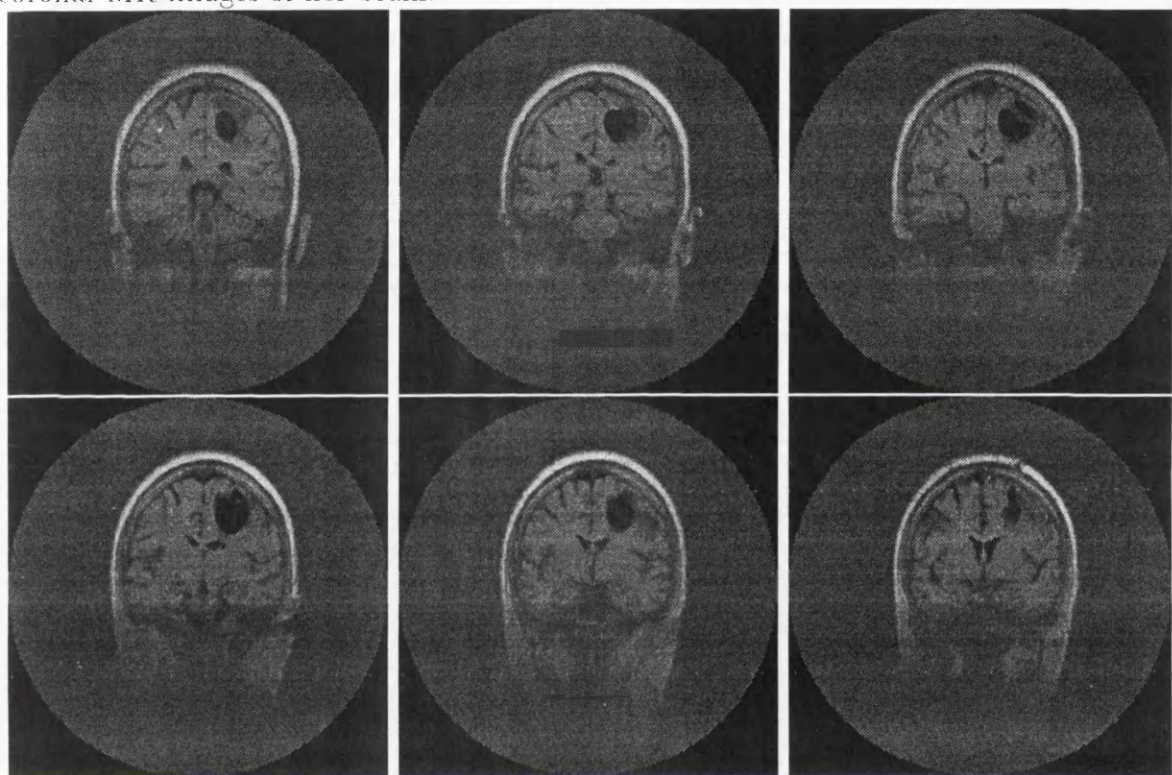


Fig.8.14 Six coronal section images from patient P_1 .

Although these 6 slices demonstrate a large lesion in the right hemisphere, the degree of intracranial deformation, as is shown in Fig.8-14, is actually quite small, compared with, for example, patient P_{11} (see Fig.8.15). It is interesting to note that the lesion areas of the two patients from Fig.8.14 and 8.15 are similar. However patient P_{11} died quickly one month after the final scan, while patient P_1 still survives nearly three years after the first scan. This would appear to indicate, in this instance at least, that the deformations of brain structures are of greater significance with regard to the survival of the patient than the size of the lesion itself.

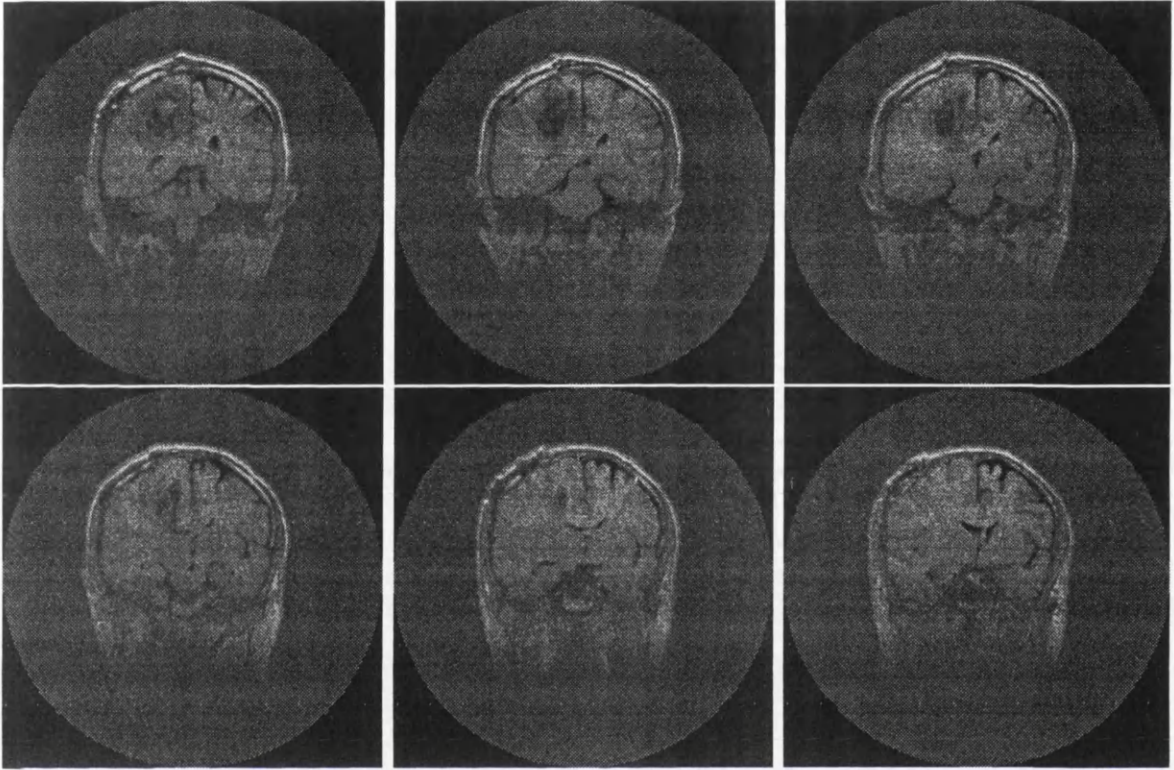


Fig.8.15 Six coronal section images from patient P_{11} .

Assessment of the response of intracranial deformation to treatment

Table 8.8 also provides the variation scale of the intracranial deformation for the 11 patients who have been scanned before and after treatment. Table 8.11 shows the VSID representation.

patient	hemisp	falx c	ventri	T.S
P_1	dec (-1)	stb (0)	dec (-1)	-2
P_2	dec (-1)	dec (-1)	stb (0)	-2
P_3	dec (-1)	stb (0)	stb (0)	-1
P_4	dec (-1)	dec (-1)	dec (-1)	-3
P_5	dec (-1)	dec (-1)	stb (0)	-2
P_6	dec (-1)	stb (0)	dec (-1)	-2
P_7	stb (0)	stb (0)	stb (0)	0
P_8	dec (-1)	stb (0)	stb (0)	-1
P_9	stb (0)	inc (1)	stb (0)	1
P_{10}	stb (0)	stb (0)	stb (0)	0
P_{11}	dec (-1)	dec (-1)	stb (0)	-2

Table 8.11 The deformations of the hemisphere, the falx cerebri and the ventricle described by VSID scores (*dec*, *stb*, *inc* and *T.S* are the abbreviations for *decrease*, *stable*, *increase* and *total scores* respectively).

Table 8.11 shows that for most of patients (8 out of 11), the intracranial deformations have decreased, to some degree, after treatment. Although there are significant decreases of intracranial deformation for the patients who have higher SSID scores (patients P_2 , P_4 , P_6 , P_8 and P_{11} have scores of SSID larger than 5 after treatment),

the patients still do not survive. This implies that the initial degree of severity of intracranial deformation is more crucial than the degree of the change in the intracranial deformation following treatment. If this can be replicated in further patient groups (see final chapter) then this may have profound implications for patient management.

Consider now only the patients who survive (P_1 , P_3 , P_5 , P_7 , P_9 and P_{10}). The total VSID scores show a decrease of deformation for P_1 , P_3 and P_5 . The assessment of the neurological status of the patients by MMRC scale shows that only patient P_5 apparently improved (See Table 8.9).

The variation of intracranial deformation for patients P_7 and P_{10} was too small to be detected by our methods. Thus the deformations are described as *stable*. The clinical assessment for these two patients also shows their conditions to be stable (Table 8.9).

In the case of patient P_9 , our assessment shows that the intracranial deformation has increased (Table 8.11), while the clinical assessment shows the patient condition is better (Table 8.9).

8.8 Measurement Reliability

The accuracy and stability of our measurement are determined by the quality of the original MRI data and the image segmentation.

The sources of error from image acquisition include inhomogeneities of the transmitted and received RF signals and the effect of partial volumes [Condon 1986, Plante 1991].

In order to minimise such sources of error, all the MRI scans in this study were performed with the same T_1 weighted sequence (IRSE 3400/500/32)

In order to assess the reliability of the image acquisition, we analysed the MRI data from six scans of one volunteer’s brain with the same imaging procedure described above. The interval between each of the repeated scans was about 10 minutes. In order to evaluate the error caused by variations in brain orientation between acquisitions, the volunteer was removed from the imager and the scanning room after each scan. Full analyses were performed on each of these image sets. Table 8.12 shows the mean and the standard deviation of the likelihood ratios for the hemispheres, the lateral ventricles and the falx cerebri from the six repeated scans.

	hemisp	ventr	falx c.
mean	0.2374	0.7284	0.7065
SD	0.0909	0.0340	0.0596

Table 8.12 The likelihood ratios from the six repeated scans

Comparing these with Table 8.3, 8.5 and 8.7 which show the mean and the standard deviation of the likelihood ratios from 15 volunteers, the measurement for the six

repeated scans has a much smaller standard deviation.

In our study, the segmentation of intracranial structures is based on Canny’s edge detector and the interactively chosen control points from the boundary profiles. Errors could potentially occur because of inhomogeneities in images causing a false segmentation by edge detector. The control points may also not be exactly assigned to the boundary profile because of observer error. In order to assess these errors, a single MRI data set from another volunteer has been analysed five times by repeating the segmentation procedure. Table 8.13 shows the statistical results of the measurement.

	hemisp	ventr	falx c.
mean	0.3959	0.5637	0.3725
SD	0.0423	0.0226	0.0109

Table 8.13 The likelihood ratios from the six repeated segmentations.

The standard deviations of the hemispheres, the ventricles and the falx cerebri in Table 8.13 are even smaller then those in Table 8.12. In other words the segmenting error is smaller than the imaging error.

The results from Table 8.12 and 8.13 show that the measurement deviation caused by the imaging and segmentation is significantly smaller than the deviation due to the wide variations in the brain shape within the normal population (Table 8.3, 8.5 and 8.7).

To assess the interobserver error, five sets of MRI data from five volunteers were measured by two observers independently. Each observer performed the image segmentation and deformation analysis for every set of the MRI data from the five volunteers. Table 8.14 shows the statistical results of the measurements by the two observers.

	<i>observer 1</i>			<i>observer 2</i>		
	hemisp	ventri	falx c.	hemisp	ventri	falx c
mean	0.2840	0.5317	0.7307	0.2898	0.4440	0.7257
SD	0.2346	0.3014	0.2659	0.2170	0.1927	0.2663

Table 8.14 interobserver reproducibility

The interobserver variation were assessed using standard F -tests [Hoel 1976] on the standard deviation. The test statistics (F) of the hemispheres, the lateral ventricles and the falx cerebri are 1.1688, 2.4463 and 1.0030, respectively. From the F -table [Neave 1979] it was found that the 5 percent critical value of F corresponding to $n_1=4$, and $n_2=4$, is 6.39. Since all the test statistics are less then the critical value, the null hypothesis cannot be rejected. Therefore, we can conclude that the measurements from the two observers are not significantly different.

8.9 Conclusions

A procedure for quantitative assessment of intracranial deformation is given in this chapter. This procedure includes feature extraction, classification and interpretation.

A feature extraction technique, K-L transform, has been introduced to select the *best* features which give most information about the deformation. The transform also reduced dimensionalities of the measurement space.

Using the Bayes decision we established three decision rules which set boundaries between the *normal* and *abnormal* hemispheres, lateral ventricles, and falx cerebri respectively. The decision rules also provide further information about the deformation through the likelihood ratio.

The severity scale and variation scale of intracranial deformation provided a comprehensive interpretation for the intracranial deformations. The total scores of SSID and VSID reflect the deformation in all three brain structures investigated.

From the results in section 8.7, we come to the following conclusions:

1. The total score of SSID is a crucial index which is related to the survival of the patient. Larger values of total SSID score seems to correspond with a lower survival time for the patient. In our application all but one of the patients with a total SSID score (both before and after treatment) larger than 5 did not survive.

2. In most cases, intracranial deformation is decreased, to varying extents, after various treatments (such as surgery, or radiotherapy) in the sense that the shapes of the deformed structures change and become closer to the shapes of the same structures in volunteers. However this decrease, even when quite large, does not correspond to a better chance of survival.
3. Our quantitative assessment of intracranial deformation does not always agree with the conventional clinical assessment. In terms of survival, in this small group at least, our quantitative assessment appears to be more objective than the clinical assessment by *MMRC*. The only truly objective measure we can use to properly assess the reliability of our new analytical approach is a longer term study of patient survival times (see the final chapter). To this end we are still following up a group of 40 patients over a three year period.

Chapter 9 Conclusions and Further Developments

9.1 Conclusions

This thesis has described the development of a new quantitative framework for the objective assessment of the effects of intracranial deformation caused by brain tumour and how it responds to treatment. In order to investigate the secondary space occupying effects caused by brain lesions, we analysed the deformation of the hemispheres, the shape of the lateral ventricles, and the displacement of the falx cerebri. Applications of our new analytical approach have shown its promise as an aid in clinical decision making as it provides an objective and sensitive index of intracranial deformation.

In chapter 4 a new segmentation technique has been developed and applied to

extracting the boundaries and surfaces of brain structures in MR images. The parametric representation of boundary and surface has been demonstrated to show advantages over the conventional raw data representation. Later measurements, such as surface curvature in Chapter 7, have depended upon this parametric representation of surface. As well as allowing such measurements, the parametric representation of boundary and surface has saved a huge amount of computational space during our analysis.

Canny's edge detector can provide accurate location of edges. Using the boundary profiles produced by Canny's detector as a guide, we interactively determined the control points for the parametric representation. This improved the accuracy in locating edges and surfaces.

In Chapter 5 a description is presented of a novel attempt to measure the deformation of the cerebral hemispheres. The hemispheres were analysed in terms of size, position and shape in both individual sectional images (two-dimensional) and as three dimensional MRI data sets. The size of each hemisphere was characterised by its volume, the areas in each sectional image, the perimeter of its contours in each sectional image, and its surface area. The centre of gravity of the hemisphere was used as a landmark in order to measure the shift of the hemisphere. The shape of the hemisphere was quantitatively specified by compactness, elongation, and symmetry with respect to the contralateral hemisphere.

28 invariant attributes (size, rotation and position invariant) were produced to describe the deformation of the hemispheres. This measurement vector included information about the changes in size, position and shape of the the hemispheres. It was used in the comprehensive analysis of the intracranial deformation in Chapter 8.

In Chapter 6 a new quantitative technique to specify the shape of the lateral ventricles is presented. This method was used to match the shapes of the lateral ventricles even though they had different sizes and orientations. The change in shape of the lateral ventricles could then be described after matching it to the lateral ventricle reference (training data set).

Measurement results from 15 volunteers and 28 patients have shown a significant difference in the shape of the lateral ventricle between volunteers and the patients with intracranial deformation (Table 6.2 and 6.3). The mean of shape distances for volunteers is 2.3216 (with standard deviation (SD) 0.4097), while for the patients with intracranial deformation it is 4.6027 (with SD 0.7990). The smaller value of the mean shape distances indicate that the shapes of the volunteers' lateral ventricles are similar to the shape of the reference ventricle. The larger value of the mean shape distances indicates a great difference between the shape of the patients ventricles and the reference ventricles.

The symmetry of the lateral ventricle was then analysed with Fourier descriptors (an original application of this technique). The results show that the lateral ventricles

of the volunteers were more symmetric than those of the patients with intracranial deformation. The mean of the total symmetry parameter is 1.4049 (SD 0.4658) for the volunteers, however it is 3.6676 (SD 0.9897) for the patients with intracranial deformation. A follow up study on a patient with intracranial deformation was used as an example to show that the shape of the lateral ventricles became close to the reference ventricles after treatment, matching the clinical improvement.

6 invariant attributes were obtained by applying the Fourier descriptors. This 6-dimensional measurement vector was used to characterise the shape of the lateral ventricles. The vector was used in the comprehensive analysis of the intracranial deformation in Chapter 8.

In Chapter 7 a new quantitative technique to describe the deformation in the intracranial surfaces is presented. The technique was used to analyse the deformation of the falx cerebri. The method is size, rotation and position independent. The falx cerebri was approximately represented by four *primary surfaces*: *elliptic*, *hyperbolic*, *saddle* and *flat* surfaces. The deformation of the falx cerebri was described by measuring the Gaussian curvature and the mean curvature of these *primary surfaces*.

The MRI data from 15 volunteers and 28 patients with intracranial deformation was analysed and the results showed a marked difference in the falx cerebri between the volunteers and the patients (Table 7.1). The falx cerebri of the volunteers show a tendency towards flatness (62.42% mean planar surface ratio with SD $\pm 14\%$), whilst

those of the patients with intracranial deformation were much more uneven (27.79% mean planar surface ratio with SD $\pm 10\%$). Follow-up studies of two patients (section 7.4.3) demonstrated how the falx cerebri changed from relative unevenness to greater flatness after treatment, matching clinical improvement (Table 7.2 and 7.3).

28 invariant attributes were obtained by measuring the curvatures of the four primary surfaces of the falx cerebri. This multi-dimensional measurement vector was used to characterise surface deformation, and was also applied in the comprehensive analysis of intracranial deformation in Chapter 8.

In Chapter 8 a description is presented of the most comprehensive attempt so far to analyse the severity of intracranial deformation and the effects of treatment. The information about the intracranial deformation was gathered from the hemispheres, the lateral ventricles and the falx cerebri as previously described.

Based on the measurements of the hemispheres, the lateral ventricles and the falx cerebri in the previous chapters from 15 volunteers and 28 patients with intracranial deformation, three criteria (the decision rules) were established to classify the *normal* and *abnormal* hemispheres, lateral ventricles, and falx cerebri respectively. The severity variation scales of the intracranial deformation were introduced to assess the severity of the intracranial deformation and its response to treatment.

Once the decision rules had been established, a further 11 patients were analysed to investigate how their clinical condition assessments and survival times corresponded

with the quantitative analysis. In the application, all but one of the patients (6 in total) with total SSID scores higher than 5 (both before and after treatment) survived at most eight months. The quantitative results show that most deformations were decreased to varying extents after treatment, however, the variation, in this small sample at least, did not appear to correlate with survival time. Clearly much larger samples and longer followings-ups are necessary (see final section) to validate this initial finding. It is important that this should be done, because if deformation at initial presentation is more correlated with survival time than subsequent changes, then this has major implications in terms of therapeutic interventions.

There are four main original contributions in this thesis. The first major development is the brain structure segmentation procedure. A vital part of the procedure is the B-splines representation of intracranial surface and boundary using the guidance provided by Canny's edge detector [Dai 1992]. The second contribution is the measurement of the shape of the lateral ventricles by Fourier descriptors. The method is simple in computation, but it provides rich and meaningful information about ventricular deformation. The third contribution is the measurement of intracranial surface by means of differential geometric techniques [Dai 1993]. The measurement provides important information about the shift of midline structures and the internal herniation. The fourth key contribution is the comprehensive analysis of the intracranial deformation based on the measurement of the hemispheres, the lateral ventricles, and

the falx cerebri. The criteria (the decision rules), the severity scale and the variation scale are new objective indices for assessment of the severity of the intracranial deformation and its response to treatment.

All the measurement techniques and the analysis procedures in this thesis are computerized using the language C++. A software package was developed which performs the hemisphere measurements, the shape analysis of the lateral ventricles, the intracranial surface measurement, and the comprehensive analysis of the intracranial deformation. The package also includes some image processing functions (*e.g.*, the Canny's edge detector and Zero-crossing algorithm), computer graphics routines (image interface routine, visualisation of measurement results) and other useful utilities (*e.g.* image format conversion, image compression).

9.2 Further developments

New techniques in diagnostic medicine are generally assessed with respect to a prevailing “*gold standard*” (*e.g.* MR angiography is compared to conventional X-ray contrast angiography). The problem with assessing the response of brain tumours to therapy is that no such “*gold standard*” exists. For this reason we have had to resort to clinical assessment of the patient's condition as a means of assessing our technique. The main difficulty with this, as was pointed out in the Introduction, is that patients are often on multiple palliative drug therapies to suppress unpleasant symptoms. As

a result assessments of clinical condition (even the elaborate scale system used for the patients examined in this thesis) are coarse and qualitative. The only definitive test of the analytical techniques described in this thesis is that of patient survival time. Despite the high mortality rate of this terrible condition, such an assessment has not been possible over the limited time course of this Ph.D project.

Continuous long term follow-up is therefore vital in this group of patients (39 in total, 24 of whom up to the time of writing have survived). Measurements on a new patient group must also be initiated to validate the finding from our limited data that our initial indices of severity of deformation is much more significant in terms of survival times than post treatment improvements.

We have laid the groundwork for therapy assessment. Studies are necessary to follow-up patients whose therapy has been confined to only one approach (*e.g.* radiotherapies, chemotherapy, or surgery). Clearly such single treatment patients are rare and so again long term studies will be necessary to build up a sufficiently large data set. It would also be interesting and worthwhile to study tumours other than gliomas. Again this would require a long term follow-up of large patient groups.

Many further developments and improvements can be made to the quantitative measurement techniques so as to enhance the reliability and accuracy of the method and expand its application.

In Chapter 8 the establishment of the deformation criteria (decision rules) is based

on the measurements from 15 volunteers and 28 patients with intracranial deformation. Obviously the accuracy and the reliability of the criteria can be enhanced by increasing the number of the measurements from both volunteers and patients. The *variation scale of the intracranial deformation* describes the change in the shapes of the intracranial structures. Further analysis by correlating deformation of specific regions with signs of neurological disorder would be extremely useful for both clinical management and neurological research.

The shape analysis method described in Chapter 6 is a two-dimensional measurement. Although three-dimensional shape analysis can be approached by measuring two-dimensional indicators of shape from a stack of image slices, the analysis become complex and time consuming, especially when larger number of image slices are used. This is the reason why we did not apply the shape analysis technique (Fourier descriptors) to the hemispheres. In theory, it is possible to extend the Fourier descriptors from two dimensions to three-dimensions though this has, as yet, not been attempted in any shape analysis problem because of the massive amount of computation required. If and when such computation becomes practicable then the study of brain tumours would make a worthy application.

In this thesis we have concentrated on the intracranial deformations caused by brain tumours. Obviously our method is applicable to intracranial deformation by other conditions, particularly head injury. There have been two reports about the

relationship between brain shift caused by head injury and the level of consciousness [Ropper 1986, 1989]. Horizontal and vertical brain shifts with altered levels were correlated with consciousness. Ropper concluded that horizontal shift correlated better with clinical status in acutely ill patients. Reich [Reich 1993] investigated the relationship between the clinical symptoms of head injury patients and internal herniation. The weakness of both methods is that they were based only on linear measurement in single MR images. Our surface measurement techniques has great potential in evaluating such intracranial herniation much more comprehensively, and without the need for landmark identification.

References

- [Anderson 1980] J. R. Anderson (Ed), *Muir's Textbook of Pathology*, Eleventh Edition, Edward Arnold Ltd, London, 1980.
- [Andrew 1990] E. R. Andrew, G. Bydder, J. Griffiths, R. Iels and P. Styles, *Clinical Magnetic Resonance Imaging and Spectroscopy*, John Wiley and Sons Ltd, New York, 1990.
- [Atlas 1991] S. W. Atlas, *Magnetic resonance imaging of the Brain and spine*, Raven Press, New York, 1991.
- [Ballard 1982] D. H. Ballard and C. M. Brown, *Computer Vision*, Englewood Cliffs, N.J., Prentice-Hall, 1982
- [Bamber 1988] J. C. Bamber and M. Tristram, Diagnostic Ultrasound, in *The Physics of Medical Imaging*, (ed.), Webb S, IOP Publishing Ltd, 1988.
- [Bloch 1946] F. Bloch, W. W. Hansen and M. E. Pound, Resonance absorption by nuclear magnetic moments in a solid. *Phys. Rev.*, vol.69, p. 127, 1946.
- [Bloom 1982] H. J. Bloom, Intracranial tumours: Response and resistance to therapeutic endeavours 1970-1980. *Int. J. Radiation Oncology Biol. Phys.*, vol.2, pp.1083-1113, 1982.

- [Bomans 1990] M. Bomans, K. H. Hoehne, U. Tiede and M. Riemer, 3-D segmentation of MR images of the head for 3-D display, *IEEE Trans. Med. Imaging*, vol.9, pp.177-183, 1990.
- [Boor 1972] Carl de Boor, On Calculation with B-splines, *J. Approx. Theory*, Vol.6, pp.50-62, 1972.
- [Brant-Zawadski 1984] M. Brant-Zawadski, J. P. Badam, C. M. Mills, D. Norman and T. H. Newton, Primary intracranial tumour imaging: a comparison of magnetic resonance and CT, *Radiology*, vol. 150, pp.434-440, 1984.
- [Brinkley 1983] J. F. Brinkley, Ultrasonic Three-Dimensional Organ Modelling, *Dept. of Comput. Sci.*, Stanford Univ., Yech. Rep. CS-1001, 1983.
- [Brummer 1993] M. E. Brummer, M. M. Russell, L. E. Robert and R. Richard, Automatic detection of brain contours in MRI data sets, *IEEE Trans. Med. Imaging*, vol.12, no.2, pp. 153-166, 1993.
- [Canny 1986] J. F. Canny, "A computational approach to edge detection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11(1), pp. 679-698, 1986.
- [Carmo 1976] M. P. de Carmo, *Differential Geometry of Curves and Surfaces*, Englewood Cliffs, NJ, Prentice-Hall, 1976.

- [Carola 1992] R. Carola, H. Harley and C. R. Noback, *Human anatomy* (2nd Ed), McGraw-Hill, New York, 1992.
- [Chien 1968] Y. T. Chien and K. S. Fu, Selection and ordering of feature observations in a pattern recognition system, *Information and Control*, Vol. 12, pp 395-414, 1968.
- [Chisholm 1989] R. A. Chisholm, S. Stenning and T. D. Hawkings, The Accuracy of Volumetric Measurement of High-grade Gliomas, *Clinical Radiology*, vol.40, pp.17-21, 1989.
- [Chu 1965] J. T. Chu, Optimal decision functions for computer character recognition, *J. Assoc. Computing Machinery*, vol. 12, pp. 213-226, 1965.
- [Chu 1988] C. H. Chu, E. J. Delp, and A. J. Buda, Detecting left ventricular endocardial and Epicardial boundaries by digital two-dimensional echocardiography, *IEEE Trans. Medical Imaging*, vol.7, no.2, pp 81-90, June, 1988.
- [Chui 1980] H. Chui and A. Damasio, Human cerebral asymmetries evaluated by computed tomography, *Journal of Neurology, Neurosurgery and Psychiatry*, Vol.43, pp. 873-878, 1980.
- [Condon 1986] B. R. Condon, J. Patterson, D. Wyper, D. Hadley, G. Teasdale et al, A Quantitative Index of Ventricular and Extraventricular Intracranial

- CSF Volumes Using MR Imaging, *J. Comput. Assist. Tomography*, 10(5), pp.748-792, 1986.
- [Condon 1988] B. R. Condon, R. Grant, et al, Brain and intracranial cavity volumes: in vivo determination by MRI, *Acta Neurol Scand*, vol. 78, pp. 387-393, 1988.
- [Cox 1971] M. G. Cox, The Numerical Evaluation of B-splines, National Physical Laboratory DNAC 4, August 1971.
- [Dai 1992] D. D. Dai, B. Condon and R. Rampling, 3-D MRI Assessment of Intracranial Curvature as a Monitor of Treatment for Brain Tumour, *Eleventh Annual Meeting of Society of Magnetic Resonance in Medicine*, Vol.3, Berlin, 1992.
- [Dai 1993] D. D. Dai, B. Condon, R. Rampling, D. Hadley and G. Teasdale, Intracranial Deformation Caused by Brain Tumours: Assessment of 3-D surface by MRI, *IEEE Medical Imaging*, vol.12, no.4, pp.1-10, 1993.
- [Damadian 1977] R. Damadian, M. Goldsmith and L. Minkoff, NMR in cancer: XVI. FONAR image of the live human body *Physiol. Chem. Phys.*, 9, pp. 97-100, 1977.
- [Damasio 1989] H. Damasio and A. R. Damasio, *Lesion Analysis in Neuropsychology*, New York, Oxford University Press, 1989.

- [Davis 1986] L. Davis, Two-Dimensional Shape Representation, in *Handbook of Pattern Recognition and Image Processing*, edited by King-Sun Fu, Academic Press, 1986.
- [Dixon 1982] R. L. Dixon and K. E. Ekstrand, The physics of proton NMR, *Medical Physics*, vol.9, pp.807-818, 1982.
- [Edelstein 1983] W. A. Edelstein, P. A. Bottomley, H. R. Hart and L. S. Smith, Signal noise, and contrast in nuclear magnetic resonance (NMR) imaging. *J. Comput. Assist. Tomogr.* vol.7, pp. 391-401, 1983.
- [Frahm 1985] J. Frahm, A. Haase, and D. Mattaei, FLASH MR imaging: from images to move, Presented at the 71st Annual Meeting of the Radiological Society of North America, Chicago, Nov, 17-22, 1985.
- [FDA 1982] FDA (Food and Drug Administration) 1982 US Bureau of Radiological Health of the FDA Report BRM (HFX-460) FDA.
- [Filipek 1989] P. A. Filipek, D. N. Kennedy, V. S. Caviness, S. L. Rossnick, T. A. Spraggins and P. M. Starewicz, Magnetic Resonance Imaging-based Brain Morphometry: Development and Application to Normal Subjects, *Annals of Neurology*, vol.25, no.1, pp.61-67, 1989.
- [Frahm 1985] J. Frahm, K. D. Merboldt, W. Hanicke and A. Haase, stimulated echo imaging, *J. Magn. Reson.*, vol.64, pp.81-93, 1985.

- [Gonzalez 1992] . C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, New York, 1992. G. Gerig, J. Martin, R. Kikinis, O. Kuebler, M. Shenton, and F. A. Jolesz, Automating segmentation of dual-echo MR head data, in *Information Processing in Medical Imaging*, A.C.F.Colchester and D.J.Hawkes, eds. Berlin, Springer-Verlag, 1991.
- [Goodman 1986] T. Goodman and K. Unsworth, “ Manipulating Shape and Producing Geometric Continuity in beta-spline Curves ”, *IEEE Computer Graphics and Application* , Feb. 1986, pp.50-56.
- [Gordon 1974] W. J. Gordon and R. F. Riesenfeld, Bernstein-Bézier Methods for the Computer Aided Design of Free-form Curves and Surfaces, *J. ACM*, Vol.21, pp.293-310, 1974.
- [Granlund 1972] G. H. Granlund, Fourier preprocessing for hand print character recognition , *IEEE Trans. Computers*, Vol. C-21 (March 1972), pp 195-201.
- [Gutin 1985] P. H. Gutin and S. A. Leibel, Stereotaxic Interstitial Irradiation of Malignant Brain Tumours in *Neurological Clinics*, vol.3, no.4, pp.883-893, 1985.

- [Henry 1985] H. W. Henry and R. Cornelius, *Textbook of anatomy* (4th ed), Philadelphia Harper & Row, London, 1985.
- [Hildebrand 1985] J. Hildebrand, Current Status of Chemotherapy of Brain Tumours, *Prog. Exp. Tumour Res.*, vol.29, pp.152-166, 1985.
- [Hines 1990] W. W. Hines and D. C. Montgomery, *Probability and Statistics in Engineering and Management Science* (third Ed), John Wiley and Sons, New York, 1990.
- [Hoel 1976] P. G. Hoel, *Elementary statistics* (4th ed), John Wiley and Sons, Inc., New York, 1976.
- [Houdek 1988] P. V. Houdek *et al*, MR characterisation of brain and brain tumour response to radiotherapy, *Int. J. Radiation. Oncol. Biol. Phys.*, vol.15, pp.213-218, 1988.
- [Hu 1990] X. Hu, K. K. Tan, D. L. Levin, R. E. Pelizzari and G. Chen, A volume rendering technique for integrated three-dimensional display of MR and PET data, in *3D Imaging in Medicine: Algorithm, Systems, Applications*, Hoehne KH *et al* , eds, Berlin, Springer-Verlag, 1990.
- [Jain 1989] A. K. Jain, *Fundamentals of digital image processing*, Englewood Cliffs, N.J., 1989.

- [Joseph 1984] P. M. Joseph, L. Axel, and M. O'Donnel, Potential problems with selective pulses in NMR imaging systems, *Med Phys*, vol.11, pp. 772-777, 1984.
- [Kanal 1962] L. Kanal, F. Slaymaker, D. Smith and W. Walker, Basic principles of some pattern recognition systems, *Proc. NEC*, vol.18, pp. 279-295, 1962.
- [Karhunen 1947] K. Karhunen, Über Lineare Methoden in der Wahrscheinlichkeitsrechnung, *Ann. Acad. Sci. Fennicae*, Ser. A137. 1947.
- [Kelly 1987] P. J. Kelly, C. Daumas-Duport, *et al*, Imaging-based stereotaxic serial biopsies in untreated intracranial glial neoplasms, *J. Neurosurgery*, vol.66, pp. 865-874, 1987.
- [Kennedy 1989] D. N. Kennedy, P. A. Filipek and R. Caviness, Automatic segmentation and volumetric calculations in nuclear magnetic resonance imaging, *IEEE Trans. Med. Imaging*, vol.8, pp.1-7, 1989.
- [Kohn 1991] M. I. Kohn, N. K. Tanna, G. T. Herman, *et al*, Analysis of Brain and Cerebrospinal Fluid Volumes with MR Imaging: Part I, Methods, Reliability, and Validation, *Radiology*, vol.178, no.1, pp.115-122, 1991.
- [Kretschmar 1982] K. Kretschmar and K. H. Schickelanz, Measurements of the volume and density of intracerebral tumours by CT following therapy, *Neuroradiology*, vol.23, pp.175-184, 1982.

- [Kumar 1975a] A. Kumar, D. Welti and R. R. Ernst, NMR Fourier zeugmatography, *J. Magn Reson.*, vol.18, pp.69-83, 1975.
- [Kumar 1975b] A. Kumar, D. Welti and R. R. Ernst, Imaging of macroscopic object NMR Fourier zeugmatography, *Naturwissenschaften*, vol.62, p.34, 1975.
- [Lai 1981] C. M. Lai and P. C. Lauterbur, True three-dimensional image reconstruction by nuclear magnetic resonance zeugmatography *Phys. Med Biol*, vol.26, pp. 851-856, 1981.
- [Lauterbur 1973] P. C. Lauterbur, Image formation by induced local interactions, examples employing nuclear magnetic resonance, *Nature*, vol. 242, pp.190-191, 1973.
- [Lawley 1963] D. N. Lawley and A. E. Maxwell, *Factor Analysis as a Statistical Method*, Butterworth, London, 1963.
- [Lee 1990] D. Lee, Coping with Discontinuities in Computer Vision: Their Detection, Classification, and Measurement, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.12, no.4, pp. 321-344, 1990.
- [Loève 1948] M. Loève, Fonctions Aléatoires de Second Order, in P. Lévy, *Processus Stochastiques et Mouvement Brownien*, Hermann, Paris, 1948.

- [MacSween 1992] R. MacSween and K. Whalet, *Muir's Textbook of Pathology* (Thirteenth edition), Edward Arnold, London, 1992.
- [Mansfield 1973] P. Mansfield and P. K. Grannell, NMR diffraction in solids, *J Phys. C: Solid State Phys.*, 6, pp. 422-426, 1973.
- [Marlill 1960] T. Marlill and D. M. Green, Statistical recognition functions and the design of pattern recognizer, *IEEE Trans. Electronic Computers*, EC-9, pp. 472-477, 1960.
- [Marr 1980] D. Marr and E. C. Hildreth, Theory of Edge Detection, *Proc. R. Soc. Lond.*, B 270, 187-217, 1980.
- [Moon 1973] R. B. Moon and J. H. Richards, Determination of intracellular pH by ^{31}P magnetic resonance, *J. Biol. Chem.*, vol.248, pp. 7276-7278, 1973.
- [Morris 1986] P. G. Morris, *Nuclear magnetic resonance imaging in medicine and biology*, Oxford, 1986, Clarendon Press.
- [Nagai 1984] H. Nagai and T. Aruna, Clinical Effects of Interferon in Malignant Brain Tumours, *Neurosurg. Review*, vol.7, pp.55-64, 1984.
- [Neave 1979] H. R. Neave, *Statistical Tables*, Allen and Unwin, London, 1979.
- [Neirinckx 1987] R. D. Neirinckx, L. R. Canning, I. M. Piper, D. P. Nowotnik, R. D. Pickett, R. A. Holmes, W. A. Volkert, A. M. Forster, P. S. Weisner, J. A.

Marriot and S. B. Chaplin, Tc-99m *d,l*-HMPAO: a new radiopharmaceutical for SPECT imaging of regional cerebral blood perfusion, *J. Nucl. Med.*, vol.28, pp.191-202, 1987.

[Nobel 1969] B. Nobel, *Applied Linear Algebra*, Prentice-Hall, Englewood Cliffs, N.J., 1969.

[NRPB 1983] NRPB (National Radiological Protection Board) 1983 Revised guidance on acceptable limits of exposure during nuclear magnetic resonance clinical imaging (NRPB *al hoc* advisory group on NMR clinical imaging).

[Persoon 1977] E. Persoon and K. S. Fu, Shape Discrimination Using Fourier Descriptors, *IEEE Trans. Systems, Man and Cybernetics*, vol.smc-7, no.3, pp.170-179, 1977.

[Pizer 1990] S. M. Pizer, T. J. Cullip and R. E. Fredericksen, Toward interactive object definition in 3D scalar images, in *3D Imaging in Medicine: Algorithm, Systems, Applications*, Hoehne KH *et al* , eds, Berlin, Springer-Verlag, 1990.

[Plante 1991] E. Plante and L. Turkstra, Sources of the Quantitative Analysis of MRI Scans, *Magnetic Resonance Imaging*, Vol.9, pp. 589-595, 1991.

- [Press 1988] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, 1988.
- [Purcell 1946] E. M. Purcell, H. C. Torrey and Nuclear induction, *Phys. Rev.*, 69, pp. 37-38, 1946.
- [Ralston 1978] A. Ralston and P. Rabinowitz, *A First Course in Numerical Analysis* (2nd ed), McGraw-Hill, New York, 1978.
- [Reich 1993] J. B. Reich, J. Sierra, W. Camp, P. Zanzonico, M. D. Deck and F. Plum, Magnetic Resonance Imaging Measurements and Clinical Changes Accompanying Transtentorial and Foramen Magnum Brain Herniation, *Ann. Neurology*, vol.33, no.2, pp.159-170,1993.
- [Riding 1986] M. Riding, L. Gerig, A. Girard and D. Stewart, Multivolumetric analysis of CT scans on patients with glioma, *Acta Radiologica Suppl.*, vol.396, pp.216-219, 1986.
- [Ropper 1986] A. H. Ropper, Lateral displacement of the brain and level of consciousness in patients with an acute hemispherical mass, *New England J. Med.*, vol.314, pp.953-958, 1986.

- [Ropper 1989] A. H. Ropper, A preliminary MRI study of the geometry of brain displacement and level of consciousness with acute intracranial masses , *Neurology* , vol.39, pp. 622-627, 1989.
- [Ross 1989] D. A. Ross, O. L. Walter, A. M. Ross, *et al*, Brain Shift, level of consciousness and restoration of consciousness in patients with acute intracranial hematoma, *J. Neurosurg.*, vol.71, pp.498-502, 1989.
- [Schoenberg 1946] I. J. Schoenberg, Contributions to the Problem of Approximation of Equidistant Data by Analysis Functions, *Quart. Appl. Math.*, Vol 4, pp. 45-99, 1946.
- [Scudder 1978] H. J. Scudder, Introduction to Computer Aided Tomography, *Proc. IEEE*, vol.66, no.6, pp. 628-637, July,, 1978.
- [Shani 1980] U. Shani, Filling Regions in binary raster images –a graph-theoretic approach, *SIGGRAPH'80* , pp.321-327, 1980.
- [Srivastava 1983] S. C. Srivastava and P. Pichards, Technetium labeled mono compounds, *Radiotracers for Medical Applications*, vol.I, ed Rayydn G V , Boca Raton: CRC Press, 1983.
- [Stark 1988] D. D. Stark and W. G. Bradley, *Magnetic Resonance Imaging*, The C.V. Mosby Company, 1988.

- [Stark 1992] D. D. Stark and G. William (ED), *Magnetic resonance imaging* (2nd ed), London, St Louis, 1992.
- [Tanna 1991] N. K. Tanna, M. I. Kohn, *et al* , Analysis of Brain and Cerebrospinal Fluid Volumes with MR Imaging: Impact on PET Data Correction for Atrophy –Part II: Aging and Alzheimer Dementia, *Radiology* , vol. 178, pp. 123-130, 1991.
- [Underwood 1987] S. R. Underwood, D. N. Firmin, R. H. Klipstein, R. S. Rees and D. B. Longmore, Magnetic resonance velocity mapping: clinical application of a new technique, *Br. Heart J.*, vol.57, pp. 404-412, 1987.
- [Valk 1985] J. Valk, C. Maclean and P. R. Algra, The localization of NMR signals in space, In *Basic principles of nuclear magnetic imaging*, New York, Elsevier Publishing Co., 1985.
- [Wallace 1980] T. P. Wallace and O. R. Mitchell, Analysis of Three-Dimensional Movement Using Fourier Descriptors, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, No.6, 1980.
- [Watanabe 1965] S. Watanabe, Karhunen-Loève expansion and factor analysis—theoretical remarks and application. *Proc. 4th Conf. Information Theory*, Prague, pp. 635-660, 1965.

- [Wehrli 1991] F. W. Wehrli, The Basis of MR Contrast, in *Magnetic Resonance Imaging of the Brain and Spine* (Ed. Atlas SW), Raven Press, New York, 1991.
- [Williams 1980] P. L. Williams and R. Warwick (Ed), *Gray's Anatomy* (36th ed), Churchill Livingstone, New York, 1980.
- [Wilson 1979] C. B. Wilson, D. Crafts and V. Levin, Brain tumours: criteria of response and definition of recurrence, In *Modern Concepts in Brain Tumour Therapy*, ed. Evans, AE, Castle House Publications, Tunbridge Wells, 1979.
- [Windham 1988] J. P. Windham, M. A. Abd-Allah, D. A. Reimann, J. W. Froelich, and A .M. Haggard, Eigenimage filtering in MR imaging, *J. Comput. Assisted Tomography*, vol.12, no.1, pp.1-9, 1988.
- [Young 1974] T. Y. Young and T. W. Calvert, *Classification Estimation and Pattern Recognition*, American Elsevier Publishing Company, New York, 1974.
- [Zahn 1972] C. T. Zahn and R. S. Roskies, Fourier Descriptors for Plane Closed Curves, *IEEE Trans. Computers*, Vol. C-21, pp 296-281, 1972.

Appendix A: Notation

The following notation is used in this thesis:

\propto	proportion to
∇	vector differential
\in	belong to
a, b	points or vectors
A, B	matrices or points
P_i	i th point
$P(\mu), P(\mu, \nu)$	B-spline curve, B-spline surface
$B_{i,k}(\mu)$	B-spline basis function
$B(\mu)$	spline blending function
u^*	the conjugate complex number of u
A^{-1}	inverse transformation of matrix A
A^T	transposition of matrix A
$ A $	determinant of matrix A
\mathcal{P}	perimeter of an object
\mathcal{A}	area of an object
\mathcal{S}	surface of an object

\mathcal{V}	volume of an object
γ_2	compactness ratio of a two-dimensional object
γ_3	compactness ratio of a three-dimensional object
ζ	elongation ratio of an object
ϱ	hemisphere variation ratio
γ_{3l}	compactness of the left hemisphere
γ_{3r}	compactness of the right hemisphere
ζ_l	elongation of the left hemisphere
ζ_r	elongation of the right hemisphere
r_1	area variation
r_2	absolute area variation
r_3	perimeter variation
r_4	absolute perimeter variation
r_5	central shift variation
r_6	absolute central variation
δ_3	three-dimensional central shift ratio
\mathcal{D}	shape distance or similarity parameter
\mathcal{Y}	symmetry parameter
K	Gaussian curvature
H	mean curvature

ω_1	normal class
ω_2	abnormal class
$\boldsymbol{P}(\mathbf{x})$	probability of event \mathbf{x}
$\boldsymbol{p}(\mathbf{x})$	probability density function
$\mathbf{P}(A B)$	probability of A conditioned on B
$\ell(\mathbf{x})$	likelihood ratio

Appendix B: Relevant Data

1. Measurement of the hemispheres

The mean of the measurement vectors:

5.16732500000000e-003
4.21540750000000e-002
3.47767500000000e-003
2.10797500000000e-002
5.21958000000000e-002
2.03128300000000e-002
4.96871675000000e-001
1.34170200000000e+000
1.02594000000000e+000
1.21553000000000e+000
5.62389999999999e-002
7.63161675000000e-001
4.23031325000000e-001
5.44506999999999e-001
7.26125000000000e-002
1.34373699999999e+000
1.03578000000000e+000
1.19913899999999e+000
5.73012560000000e-002
6.61426250000000e-001
4.83325000000000e-001

5.5726333333331e-001
 5.15489999999996e-002
 1.00000033899999e+000
 1.00000000011332e+000
 2.66278820000000e-001
 3.01325600000000e-001
 3.21325000000000e-003

The largest and the second largest eigenvalues:

$$\lambda_1 = 2.69518924342923e - 003,$$

$$\lambda_2 = 5.84319582659789e - 004.$$

The eigenvectors associated with these two eigenvalues (K-L transform matrix):

-5.29338454290179e-001	-1.58508687226221e-002
-7.07106228136634e-001	9.07284303723801e-007
2.22715185338047e-001	-1.49434333669380e-001
-5.26945057310401e-002	2.24015784947402e-002
1.10842970580517e-001	6.43547115011192e-001
-3.14021418803993e-001	-3.55870048286724e-001
2.37754500758411e-001	-6.60393760308820e-001
1.37168662823332e-002	2.93761091229954e-001
-1.30678055801462e-002	2.22361785310454e-001
-5.33749842724113e-002	3.84419999673403e-001
1.53646139398015e-001	6.44663540768545e-001
-1.64382536550713e-002	-2.34075527193076e-002
-8.41286004948773e-003	-8.34639739984296e-003
2.06340605579103e-002	1.27650886537039e-002
2.12840348971913e-002	-2.37164124206516e-002

-1.30086692434772e-002	3.57318346056132e-002
-1.34576841858421e-002	-1.73361753776225e-002
-1.09832406272382e-002	3.26143515307206e-002
2.84953680746699e-002	1.02549031088261e-002
-3.65387913423251e-002	3.20580216779796e-002
-2.52822814395638e-001	6.69340947693252e-003
1.00364492876765e-001	1.31673153315655e-002
-1.39152019971062e-001	-1.81490156594736e-002
1.45264341048078e-001	-5.86367509680665e-002
-2.63869959413414e-002	-1.71573102127200e-002
1.83263453788772e-001	5.51409347606966e-003
7.79549201295205e-003	-4.87527847070632e-003
-6.54822614331362e-003	-7.89422649078806e-003

Table B.1 The first and second feature vectors for the hemispheres

2.67332500000005e-003	-2.41398000000000e-002
3.18732500000007e-003	-2.51688000000000e-002
1.07732500000001e-003	-2.61438000000000e-002
1.89932500000006e-003	-2.21358000000000e-002
1.10932500000005e-003	-2.31668000000000e-002
-3.18674999999935e-004	-5.66180000000000e-003
-1.70567499999996e-003	-4.47580000000000e-003
3.71632500000007e-003	1.03620000000001e-003
3.36932500000008e-003	-1.38478000000000e-002
-9.16667499999995e-003	-8.13480000000000e-003
-4.17667499999996e-003	2.55920000000001e-003
1.32532500000004e-003	-8.54080000000000e-003
2.38732500000004e-003	-2.36248000000000e-002
4.97532500000008e-003	-3.55978000000000e-002
5.18325000000041e-004	-1.86508000000000e-002
1.23943250000000e-002	-2.08148000000000e-002

-4.73346749999999e-002	9.56302000000001e-002
1.59483250000001e-002	-2.69688000000000e-002
-2.47386749999999e-002	7.16662000000001e-002
-6.9976749999997e-003	1.72742000000000e-002
1.65133250000000e-002	1.22572000000000e-002
3.42432500000000e-003	-1.92948000000000e-002
-9.0006749999994e-003	4.06802000000000e-002
-2.25846749999999e-002	1.05602200000000e-001
2.48503250000000e-002	1.94952000000000e-002
-2.0346749999993e-003	-1.38608000000000e-002
1.56953250000000e-002	-1.29438000000000e-002
-8.0826749999997e-003	-1.34879999999999e-003
-1.70032500000000e-002	5.59980000000000e-002
-2.48325000000001e-002	3.51284800000000e-002
-1.34906749999999e-002	4.67302000000000e-002
-1.50016749999999e-002	2.95714800000000e-002
-1.82813250000000e-002	1.41132500000000e-002
-1.6836749999997e-002	6.15988000000000e-003
-1.1646749999993e-002	3.47513250000001e-002
-1.36372000000000e-002	1.80208000000000e-002
-1.00432500000000e-002	9.88313325000000e-003
3.47813325000001e-003	-2.28726000000000e-002
1.51183250000001e-002	-1.64127200000000e-002
1.58153250000000e-002	1.24840200000000e-002
2.21034679999999e-002	1.79472000000000e-002
7.00360000000000e-003	2.19979999999999e-002
1.00032500000007e-002	1.63820000000000e-002

2. Measurement of the lateral ventricles

The mean of the measurement vectors:

4.556705000000000e+000
3.168734333333333e+000
3.044637766666667e+000
2.429040000000000e+000
2.335640000000000e+000
2.620496666666667e+000

The largest and the second largest eigenvalues:

$\lambda_1 = 6.15528286260169e + 000,$

$\lambda_2 = 1.61555241719883e + 000.$

The eigenvectors associated with these two eigenvalues (K-L transform matrix):

1.75334677880302e-001	3.44084457168007e-001
4.84526201856396e-001	-6.352408441242754e-001
1.19535247769535e-001	1.95363034667512e-002
-2.25217303792164e-001	4.640404827698294e-001
7.60388778342517e-001	2.92839874550090e-001
3.02141534435874e-001	4.20227132442509e-001

Table B.2 The first and second feature vectors of the lateral ventricles

-1.519496666666667e+000	-9.730799999999999e-001
-1.581296666666667e+000	-7.981800000000000e-001
-1.825466666666667e+000	-2.381400000000000e+000
-1.857696666666667e+000	-9.026383999999999e-001
-5.739666666666667e-001	-1.718540000000000e+000

-5.68696666666668e-001	-4.12679999999999e-001
-7.02696666666668e-001	3.49026000000001e-001
5.49803333333332e-001	-1.71846400000000e+000
-7.51606666666667e-001	-9.12479999999999e-001
8.87303333333332e-001	-8.23980000000000e-001
-1.42063333333333e+000	-1.39140000000000e+000
-1.08166666666667e-001	-1.38130000000000e-001
-1.30729666666667e+000	-1.92036000000000e+000
-3.00730333333333e+000	-6.92253600000000e-001
-1.19985400000000e+000	-4.00010000000000e-001
1.83690333333333e+000	2.38962000000000e+000
-3.19800333333333e+000	-7.89384000000000e+000
-1.19640666666668e-001	1.35960000000001e-001
9.02006666666668e-001	2.79933333333333e-001
1.63696666666661e-001	3.19933333333333e+000
3.33230333333333e+000	1.37686000000000e-001
2.25846666666668e-001	4.02066666666661e-002
-5.68610333333332e-001	3.68652600000000e-001
-3.30300333333333e+000	7.42034000000000e+000
8.62300000000000e-001	1.81422400000000e-001
7.60609666666667e-001	6.19134000000000e-001
1.27054666666667e+000	5.63833333333331e-001
-3.00153333333332e-002	-2.52708400000000e-001
2.80706666666668e-002	3.79999999999999e-001
-7.13366666666668e-001	1.44666666666661e-002
3.27040000000000e-001	4.11266666666668e-001
3.37176666666668e+000	1.82940000000000e+000
2.81040000000000e+000	4.97999999999999e-001
2.19933333333333e+000	1.30010400000000e+000
2.63333333333333e+000	2.39910200000000e+000
2.20069666666661e+000	3.27099999999999e+000

1.42046800000000e+000	1.76310400000000e+000
1.29766666666668e+000	3.41069999999999e+000
4.71820800000000e-001	3.33310600000000e+000
-2.16466666666661e-001	2.84114035000000e+000
-9.00033333333331e-001	2.00606666666661e+000
4.00010040000000e-001	1.53097999999993e+000
9.53189999999991e-001	1.30069999999999e+000

3. Measurement of the falx cerebri

The mean of the measurement vectors:

-1.93787500000000e-003
2.59227500000000e-003
-8.65000000000000e-006
2.83200000000000e-004
-6.23901500000000e-002
5.98963000000000e-002
-5.04950000000000e-004
1.22255250000000e-002
1.00475000000000e-004
2.59227500000000e-003
3.78200000000000e-004
3.79750000000000e-004
1.24871800000000e-001
-3.90219250000000e-002
3.69388500000000e-002
-2.90525000000000e-004
1.15775750000000e-002
2.61456325000000e-001
-5.98205000000000e-003

5.985200000000000e-003
 1.055000000000000e-005
 2.919975000000000e-003
 4.368224750000000e-001
 -1.937875000000000e-003
 -1.002750000000000e-004
 -2.813500000000000e-004
 2.393000000000000e-004
 1.768493500000000e-001

The largest and the second largest eigenvalues:

$$\lambda_1 = 6.28931810482974e - 002,$$

$$\lambda_2 = 7.08482693973263e - 003.$$

The eigenvectors associated with these two eigenvalues (K-L transform matrix):

8.64121286757205e-008	2.04927267818663e-011
1.48984943016840e-007	-3.76721843392500e-010
4.67213174829609e-001	-1.04873974800299e-003
8.46110876359156e-002	3.09887859836863e-003
1.56655450192657e-001	-1.98572989796142e-003
1.59535249454048e-003	1.57420753690544e-001
5.24220587358435e-004	1.50476341762418e-001
2.43181975409899e-003	3.59791549371909e-001
1.37268662823332e-003	-2.97391091229954e-001
1.20768088501462e-003	3.12371653810454e-001
-5.44947842724112e-004	3.48149999763304e-001
3.53487139289051e-004	6.66443540678545e-001
-1.46832536570531e-003	-2.43045529123076e-001

-1.21266004948783e-003	-8.83739737784294e-002
2.02340506679105e-003	-1.87680886525039e-003
3.12784077891913e-002	-2.69064124106716e-002
-1.80063392434662e-002	3.75918326088122e-002
-1.83457641848521e-002	1.43081773766225e-002
-1.19742401292882e-002	3.54113702037334e-002
1.48935864076997e-002	3.02549031088261e-002
5.90539991142310e-002	-1.82051702167977e-002
2.25160281477568e-001	-1.36970494769252e-003
-2.50546492718765e-001	-1.61623315315565e-003
-4.13995209711206e-001	-2.01814957465936e-003
-2.82425601344808e-001	-9.76836506896065e-004
-1.36689159943441e-001	5.71175731021200e-004
3.34826352837877e-001	8.41540376660699e-004
4.99079542125205e-001	5.54287278707063e-004

Table B.3 The first and second feature vectors of the falx cerebri

-1.54723500000000e-002	-8.83000000000009e-005
-1.32171350000000e-001	-4.73000000000343e-005
-1.52679350000000e-001	-7.8300000000497e-005
-1.37054350000000e-001	-1.05300000000035e-004
-1.45843350000000e-001	-9.1300000000411e-005
4.06796500000000e-002	-6.32999999999886e-005
-1.57318350000000e-001	-1.71300000000042e-004
-3.99734999999996e-003	-1.35299999999998e-004
-3.06093500000000e-002	2.48699999999984e-004
-1.00677350000000e-001	-1.18300000000009e-004
-1.38031350000000e-001	-1.66300000000034e-004
-1.14105350000000e-001	-1.02300000000043e-004
-1.09710350000000e-001	-1.81300000000022e-004
-1.62201350000000e-001	-2.17300000000064e-004

-1.24847350000000e-001	-1.46300000000053e-004
-7.77283500000000e-002	-1.41300000000017e-004
2.45269650000000e-001	2.34700000000065e-004
-3.50934999999997e-003	-8.32999999999926e-005
-4.24234999999995e-003	-8.52999999999952e-005
2.87994650000000e-001	4.44700000000075e-004
7.09536500000000e-002	-2.329999999999726e-005
1.82769650000000e-001	4.30700000000052e-004
2.44781650000000e-001	5.66700000000062e-004
1.05621650000000e-001	8.77000000000325e-005
1.27838650000000e-001	3.77000000000351e-005
-5.30703500000000e-002	-9.23000000000072e-005
1.41021650000000e-001	-6.729999999999593e-005
2.39104620000000e-001	4.23000005678243e-004
2.01243330000000e-001	3.49999990000000e-004
1.77723400000000e-001	2.79800301204700e-004
1.89990000000000e-001	1.89426000000053e-004
1.90000230000000e-001	7.56300210000073e-005
1.47356500000000e-001	1.58999999999976e-004
1.2530733333332e-001	2.84720770000003e-004
1.43659940000000e-001	-3.11046000000013e-005
9.78314730000000e-002	1.999999999999312e-004
9.99237777130000e-002	-3.51063700000031e-005
7.32064733320000e-002	4.027001999999617e-005
5.27634180000000e-002	-2.97343310783000e-005
2.41037650000000e-002	-2.38270000000052e-005
3.20713065000000e-002	-4.38740000000063e-005
-3.23450000007072e-006	-1.57700362110003e-007
7.35273318250000e-003	1.00000007333325e-005

Appendix C: The Program for Image Segmentation

```
/*
 * canny.c---The basic routines performing a Canny's edge detection.
 *
 */

#include <math.h>
#include <stdio.h>

#define EDGE0 0
#define EDGE2 254
#define EWINDOWSIZE 3 /* IF THE FILTER IS EVENSIZED */
#define XDIR 1
#define YDIR 2

void canny(magmax,hthresh,lthresh,image,xsize,ysize,shortim>windowsize,sigma
          hfrac,lfrac,gx,gy,mag,hist,histsize,nms,edgemap,gm,gmp,temp)
int *image; /* The image array as an array of unsigned char. */
int *xsize; /* The x-dimension of the image */
int *ysize; /* The y-dimension of the image */
short *shortim,*temp; /* scratch space */
int *windowsize; /* The size of the window */
float *gm,*gmp; /* arrays to hold masks */
double *sigma; /* The sigma of the gaussian in pixels. */
/* RANGE -- (0.0,somebignumber) */
double *hfrac; /* Chooses the upper threshold in hysteric */
/* thresholding to cause the seed pixels to be */
/* chosen from the upper (hfrac*100) percent of */
/* the pixels in the magnitude (edginess) image */
/* RANGE -- [0.0,1.0] */
double *lfrac; /* Multiplies the upper threshold to get the lower
/* hysteric threshold */
/* RANGE -- [0.0, 1.0] */

unsigned char *edgemap; /* An array of char with 255 values at the */
/* edgepoints and 0 elsewhere */

short *gx;
short *gy;
short *mag;
int *magmax;
int *hist;
int *histsize;
```

```

int *hthresh, *lthresh;
unsigned char *nms;

{
short *mgx,*mgy,*mmag,*shortimage;
int *mhist,mhthresh,mlthresh, mhistsize,dummyoutx,dummyouty,mxsize,
    mysize,mmagmax;
unsigned char *iptr, *optr;
short *temp_image;
float *gmask,*gpmask;

mxsize = *xsize;
mysize = *ysize;

mhistsize = *histsize;

shortimage = shortim;
mgx = gx;
mgy = gy;
mmag = mag;
mhist = hist;
temp_image = temp;
gmask = gm;
gpmask = gmp;

dubimg(image,mxsize,mysize,shortimage);

shortimage = shortim;

gfilter(shortimage,mxsize,mysize,XDIR,*windowsize,*sigma,mgx,
    &dummyoutx,&dummyouty,gmask,gpmask,temp_image);

shortimage = shortim;
temp_image = temp;
gmask = gm;
gpmask = gmp;

gfilter(shortimage,mxsize,mysize,YDIR,*windowsize,*sigma,mgy,
    &dummyoutx,&dummyouty,gmask,gpmask,temp_image);

```



```

shortimage = shortim;
temp_image = temp;
gmask = gm;
gpmask = gmp;

magnitude(mgx,mgy,mxsize,mysize,mmag,&mmagmax);

non_max_supp(mmag,mgx,mgy,mxsize,mysize,edgemap);

find_edges(edgemap,mmag,mxsize,mysize,mmagmax,*hfrac,*lfrac,
           mhist,mhistsize,&mhthresh,&mlthresh);

if (magmax != (int *) 0L)
{
    *magmax = mmagmax;
}

if (hthresh != (int *) 0L)
{
    *hthresh = mhthresh;
}

if (lthresh != (int *) 0L)
{
    *lthresh = mlthresh;
}
}

magnitude(gx,gy,xsize,ysize,mag, max)
short *gx,*gy,*mag;
int xsize,ysize,*max;
{
    short *xpixelptr,*ypixelptr,*magpixelptr,nearestshort();
    int pixelcount,themax = 0;
    float gradx,grady;

    for(pixelcount = 0, xpixelptr = gx, ypixelptr = gy, magpixelptr = mag;
        pixelcount < xsize*ysize;

```

```

        pixelcount++, xpixelptr++, ypixelptr++, magpixelptr++)
    {
        gradx = (float) *xpixelptr;
        grady = (float) *ypixelptr;

        if ((*magpixelptr = nearestshort(sqrt((float) (gradx*gradx +
        grady*grady)))) > themax)
            themax = *magpixelptr;
    }
    *max = themax;
}

non_max_supp(mag, gradx, grady, ncols, nrows, result)
    short *mag, *gradx, *grady;
    int    ncols, nrows;
    unsigned char *result;
{
    int rowcount, colcount, count;
    short *magrowptr, *magptr;
    short *gxrowptr, *gxptr;
    short *gyrowptr, *gyptr, z1, z2;
    short m00, gx, gy;
    float mag1, mag2, xperp, yperp;
    unsigned char *resultrowptr, *resultptr;

    for(count = 0, resultrowptr = result, resultptr = result +
    ncols*(nrows - 1);
        count < ncols;
        resultptr++, resultrowptr++, count++)
    {
        *resultrowptr = *resultptr = (unsigned char) 0;
    }

    for(count = 0, resultptr = result, resultrowptr = result + ncols - 1;
        count < nrows;
        count++, resultptr += ncols, resultrowptr += ncols)
    {
        *resultptr = *resultrowptr = (unsigned char) 0;
    }

    for(rowcount = 1, magrowptr = mag + ncols + 1, gxrowptr = gradx + ncols +

```

```

        gyrowptr = grady + ncols + 1, resultrowptr = result + ncols + 1;
        rowcount < nrows - 2;
        rowcount++, magrowptr += ncols, gyrowptr += ncols, gxrowptr += ncols
resultrowptr += ncols)
    {
        for(colcount = 1, magptr = magrowptr, gxptra = gxrowptr, gyptra =
gyrowptr, resultptr = resultrowptr;
            colcount < ncols - 2;
            colcount++, magptr++, gxptra++, gyptra++, resultptr++)
        {
            m00 = *magptr;
            if(m00 == 0)
            {
                *resultptr = (unsigned char) EDGE0;
            }
            else
            {
                xperp = -(gx = *gxptra)/((float)m00);
                yperp = (gy = *gyptra)/((float)m00);
            }

            if (gx >= 0)
            {
                if (gy >= 0)
                {
                    if (gx >= gy)
                    {
                        z1 = *(magptr - 1);
                        z2 = *(magptr - ncols - 1);

                        mag1 = (m00 - z1)*xperp + (z2 - z1)*yperp;

                        z1 = *(magptr + 1);
                        z2 = *(magptr + ncols + 1);

                        mag2 = (m00 - z1)*xperp + (z2 - z1)*yperp;
                    }
                    else
                    {
                        z1 = *(magptr - ncols);
                        z2 = *(magptr - ncols - 1);

                        mag1 = (z1 - z2)*xperp + (z1 - m00)*yperp;

                        z1 = *(magptr + ncols);

```

```

        z2 = *(magptr + ncols + 1);

        mag2 = (z1 - z2)*xperp + (z1 - m00)*yperp;
    }
}
else
{
    if (gx >= -gy)
    {
        z1 = *(magptr - 1);
        z2 = *(magptr + ncols - 1);

        mag1 = (m00 - z1)*xperp + (z1 - z2)*yperp;

        z1 = *(magptr + 1);
        z2 = *(magptr - ncols + 1);

        mag2 = (m00 - z1)*xperp + (z1 - z2)*yperp;
    }
    else
    {
        z1 = *(magptr + ncols);
        z2 = *(magptr + ncols - 1);

        mag1 = (z1 - z2)*xperp + (m00 - z1)*yperp;

        z1 = *(magptr - ncols);
        z2 = *(magptr - ncols + 1);

        mag2 = (z1 - z2)*xperp + (m00 - z1)*yperp;
    }
}
}
else
{
    if ((gy = *gyptr) >= 0)
    {
        if (-gx >= gy)
        {
            z1 = *(magptr + 1);
            z2 = *(magptr - ncols + 1);

            mag1 = (z1 - m00)*xperp + (z2 - z1)*yperp;

            z1 = *(magptr - 1);

```

```

        z2 = *(magptr + ncols - 1);

        mag2 = (z1 - m00)*xperp + (z2 - z1)*yperp;
    }
    else
    {
        z1 = *(magptr - ncols);
        z2 = *(magptr - ncols + 1);

        mag1 = (z2 - z1)*xperp + (z1 - m00)*yperp;

        z1 = *(magptr + ncols);
        z2 = *(magptr + ncols - 1);

        mag2 = (z2 - z1)*xperp + (z1 - m00)*yperp;
    }
}
else
{
    if (-gx > -gy)
    {
        z1 = *(magptr + 1);
        z2 = *(magptr + ncols + 1);

        mag1 = (z1 - m00)*xperp + (z1 - z2)*yperp;

        z1 = *(magptr - 1);
        z2 = *(magptr - ncols - 1);

        mag2 = (z1 - m00)*xperp + (z1 - z2)*yperp;
    }
    else
    {
        z1 = *(magptr + ncols);
        z2 = *(magptr + ncols + 1);

        mag1 = (z2 - z1)*xperp + (m00 - z1)*yperp;

        z1 = *(magptr - ncols);
        z2 = *(magptr - ncols - 1);

        mag2 = (z2 - z1)*xperp + (m00 - z1)*yperp;
    }
}
}

```

```

        if ((mag1 >= 0.0) || (mag2 > 0.0))
        {
            *resultptr = (unsigned char) EDGE0;
        }
        else
        {
            *resultptr = (unsigned char) EDGE2;
        }
    }
}

```

```

dubimg(charimage,ncols,nrows,shortimage)
    int *charimage;
    short *shortimage;
    int ncols, nrows;
{
    int i;
    int *inptr;
    short *outptr;

    for(i=0,inptr = charimage,outptr = shortimage;i<ncols*nrows;i++,
inptr++,outptr++)
    {
        *outptr = (short) *inptr;
    }
}

```

/* gauss filter */

```

gfilter(inimage, inx, iny, direction, masksize,sigma, grad, outx, outy,
        gmask,gprimemask,tempimage)
    short *inimage,*grad;
    int inx,iny,direction,masksize,*outx,*outy;
    float sigma;
    float *gmask, *gprimemask;
    short *tempimage;
{
    int orthogdir,status,max,min;

    make_gaussian_mask(gmask, gprimemask, masksize, sigma, 4.0);

```



```

switch(direction)
{
    case XDIR:
        orthogdir = YDIR;
        break;
    case YDIR:
        orthogdir = XDIR;
        break;
    default:
        fprintf(stderr,"No such direction in gauss_filter (in gauss.c).\n");
        exit();
        break;
}

status=0;

correlate_img(inimage,inx,iny,gmask,masksize,orthogdir,
              tempimage,outx,outy,&max,&min,&status);

correlate_img(tempimage,*outx,*outy,gprimemask,masksize,direction,
              grad,outx,outy,&max,&min,&status);
}

correlate_img(image_ptr, incols, inrows, filter, windowsize, direction,
              result, outcols, outrows, maxval, minval, status)
short *image_ptr;
short *result;
int inrows, incols, windowsize, direction;
float filter[20];
int *outcols, *outrows, *maxval, *minval, *status;
{
    int rowcount,colcount,findex,halfwindow,currentpixel,theacol,zerooflag = 0;
    int therow,totalpixels, windowcolpixels,themaxval = 0, theminval = 0;
    short *inbegrowptr, *inendrowptr, nearestshort();
    short *inbegcolptr, *inendcolptr, *inposptr1, *inposptr2;
    short *outrowptr, *outcolptr, *outposptr;
    float currentresult;

    if (windowsize % 2 == 0)
    {

        *status = EWINDOWSIZE;

```

```

        return;
    }

    halfwindow = windowsize/2;
    totalpixels = inrows * incols;
    windowcolpixels = halfwindow * incols;

    switch(direction)
    {
        case XDIR:
            for(rowcount = 0, inbegrowptr = image_ptr, inendrowptr = image_ptr +
incols - 1, outrowptr = result;
                rowcount < inrows; rowcount++, inbegrowptr += incols,
outrowptr += incols)
            {
                for(colcount = 0, inposptr1 = inbegrowptr, outposptr =
outrowptr;
                    colcount < incols ;
                    colcount++, inposptr1++, outposptr++)
                {
                    currentresult = 0;
                    zeroflag = 0;
                    for(findex = 0, inposptr2 = inposptr1 - halfwindow,
thecol = colcount - halfwindow;
                        findex < windowsize;
                        findex++, inposptr2++, thecol++)
                    {
                        if(thecol < 0)
                        {
                            currentpixel = (int) (*(inposptr2 + incols));
                        }
                        else
                        {
                            if(thecol >= incols)
                            {
                                currentpixel = (int) (*(inposptr2 - incols));
                            }
                            else
                                currentpixel = (int) *inposptr2;
                        }
                        if (!zeroflag)

```

```

        currentresult += filter[findex]*currentpixel;
    }
    if (themaxval < currentresult) themaxval = currentresult
    if (theminval > currentresult) theminval = currentresult
    *outposptr = nearestshort(currentresult);
}
}

*outrows = inrows;
*outcols = incols;
*maxval = themaxval;
*minval = theminval;

case YDIR:
    for(colcount = 0, inbegcolptr = image_ptr, inendcolptr = image_ptr
totalpixels - incols, outcolptr = result;
        colcount < incols;
        colcount++, inbegcolptr++, inendcolptr++, outcolptr++)
    {
        for(rowcount = 0, inposptr1 = inbegcolptr, outposptr = outcolptr
rowcount < inrows;
            rowcount++, inposptr1 += incols, outposptr += incols)
        {
            zeroflag = 0;
            currentresult = 0;
            /* scan filter */
            for(findex = 0, inposptr2 = inposptr1 - windowcolpixels,
therow = rowcount - halfwindow;
                findex < windowsize;
                findex++, inposptr2 += incols, therow++)
            {
                if(therow < 0)
                {
                    currentpixel = (int) (*(inposptr2 + totalpixels));
                }
                else
                {
                    if(therow >= inrows)
                    {
                        currentpixel = (int) (*(inposptr2 - totalpixels));
                    }
                }
            }
        }
    }
else
    currentpixel = (int) *inposptr2;

```

```

        }
        if (!zeroflag)
            currentresult += filter[findex]*currentpixel;
    }
    if (themaxval < currentresult) themaxval = currentresult
    if (theminval > currentresult) theminval = currentresult
    *outposptr = nearestshort(currentresult);
}
}

*outrows = inrows;
*outcols = incols;
*maxval = themaxval;
*minval = theminval;

default:
    return;
}
}

```

```

make_gaussian_mask(gmask, gprimemask, masksize, sigma, maxresponse)
    float gmask[20],gprimemask[20],sigma, maxresponse;
    int masksize;
{
    int i, maskcenter, count, findex;
    float delta, currentx,gconst,gprimeconst;

    if (masksize % 2 == 0)
    {
        printf("Even masksize in make_gaussian_mask (in gauss.c).\n");
        exit();
    }

    maskcenter = masksize/2;
    delta = 1.0/(11 - 1);

    gconst = 1.0*maxresponse/((sqrt(TWO_PI)) * sigma);
    gprimeconst = maxresponse/(sigma*sigma);

    for(i = -maskcenter, findex = 0;findex < masksize;i++, findex++)
    {
        gmask[findex] = 0;
        gprimemask[findex] = 0;
    }
}

```

```

        for(count = 1, currentx = i - .5; count <= 11; count++,
currentx += delta)
    {
        gmask[findex] += gconst * exp(-currentx*currentx/(2*sigma*sigma));
        gprimemask[findex] += gprimeconst * currentx *
exp(-currentx*currentx/(2*sigma*sigma));
    }
    gmask[findex] /= 11;
    gprimemask[findex] /= 11;
}
}

```

```

/*
 *   Thin.c--- to thin all edges down to 8 connected edges. This routine is
 *               called from canny.c by default.
 *
 */

```

```

#include <stdio.h>

```

```

#define U 010
#define E 020
#define I 040
#define MM 0100
#define M 0200
#define NMM 200
#define absval(f) (((f)<0)?(-(f)):(f))

```

```

int flags = 0;
int flagv = 0;
int rinc[10] = {0,0,-1,-1,-1,0,1,1,1,0};
int cinc[10] = {0,1,1,0,-1,-1,-1,0,1,1};
int mrow[NMM],mcol[NMM],mlabel[NMM],maxmult,maxlabel;
int row,col;
char *pic;

```

```

thin(edges,height,width)
int height,width;

```

```

unsigned char *edges;

int frameno,r,c,change,trans,neigh,fp,first,gap,gapopen;
int i,j;
int mini,sumr,sumc,label,npt,Mfound;
float avgr,avgc,mindist,dist;
char *ppic;
char *temp;
unsigned char *temp2;
row = height;
col = width;
pic = (char * ) malloc(row*col,sizeof (char));

temp2 = edges;
temp=pic;
for (i=0; i<height * width; i++)
    { *temp = (char)*temp2;
      temp++;
      temp2++;
    };

    for (fp=0;fp<2;fp++)
        do {
maxlabel = maxmult = change = 0;

ppic = pic;

        for (r=0;r<row;r++) {
for (c=0;c<col;c++) {
if ((*ppic&0377)!=0) {
    switch(fp) {
        case 0:
        case 1:
neigh = trans = 0;
for (i=1;i<=8;i++) {
if (neighbour(r,c,i)) {
neigh++;
first = i;
}
if ((neighbour(r,c,i)==0) ^
    (neighbour(r,c,i+1)==0)!=0)
trans++;
}
if (neigh<2 || neigh>6 ||
    (neigh==2 && trans<4) ||

```



```

        (neigh==3 && trans>4)) {
ppic++;
continue;
}
gap = gapopen = 0;
for (i=first+1;i<first+8;i++) {
    j = ((i - 1) % 8) + 1;
    if (gapopen) {
if (neighbour(r,c,j))
gapopen = 0;
    }
    else {
if (j%2==1 && neighbour(r,c,j)==0) {
gapopen++;
gap++;
}
    }
}
if (gap == 1) {
*ppic++ = 0;
change++;
}
else
    ppic++;
continue;

    default:
fprintf(stderr, "unknown pass!");
    }
}
else
ppic++;
    }
    } while (change!=0 && flags==0 && (fp==0 || fp==1));

temp2 = edges;
for (i=0; i<height * width; i++)
    {*temp2 = (unsigned) *pic;
    pic++;
    temp2++;
    };
}
/*****
addmult(row,col,label)

```

```

int row,col,label;

{
int i,oldlabel;

oldlabel = 0;
for (i=0;i<maxmult;i++)
if (mrow[i]==row && mcol[i]==col)
oldlabel = mlabel[i];
if (oldlabel) {
for (i=0;i<maxmult;i++)
if (mlabel[i] == oldlabel)
mlabel[i] = label;
}
else {
if (maxmult >= NMM)
fprintf(stderr,"M point overflow");
mrow[maxmult] = row;
mcol[maxmult] = col;
mlabel[maxmult++] = label;
}
}

/*****

neighbour(r,c,d)

int r,c,d;

{
int i,j;

i = r + rinc[d];
j = c + cinc[d];
if (i<0 || i>=row || j<0 || j>=col)
return(0);
else
return(pic[i*col+j]&0377);
}

/*  bscurve.c-- Subroutine to generate  a B-spline curve. The input is a
*          list of control points with byte format. The program will

```

```

*           produce a B-spline curve according to this control points.
*           The output is the B-spline curve with  $m/2$  ( = 1024/2 ) float
*           points.
*
*
*/

#include<math.h>
#include"shape.h"

#define m 512

float bound[1024];
unsigned char aryx[100],aryy[100];

bspline(int flag)
{
    int i,j,k,t,n,leng,size,im,noi;
    double b,u,v;
    float x,y,xx,yy,x0,y0;

    if(flag==0){
        _clearscreen(_GCLEARSCREEN);
        _settextposition(6,10);
        _outtext("input file name: ");
        scanf("%s",f_in);
        if ( ( in = open(f_in,O_BINARY|O_RDONLY)) == -1){
            printf("open failed on input file ");
            exit(0);
        }
        _settextposition(8,10);
        _outtext("output file name: ");
        scanf("%s",f_out);
        out = open(f_out,O_CREAT|O_TRUNC|O_WRONLY|O_BINARY,S_IREAD|S_IWRITE);
    }
    else {
        in=open("atob.out",O_BINARY|O_RDONLY);
        out=open("bspline.out",O_CREAT|O_TRUNC|O_WRONLY|O_BINARY,
        S_IREAD|S_IWRITE);
    }

    read(in,header,2*sizeof(short));
    leng = header[0];
    noi = header[1];

```

```

size = 2*leng;
n=leng-1;
header[0]= m;

write(out,header,2*sizeof(short));

for (im=0; im<noi;im++){
    read(in,buffer,size*sizeof(char));
    j=0; i=1;
    while ( j < size && i <= leng) {
        aryx[i] = buffer[j]; j++;
        aryy[i] = buffer[j]; j++;
        i++;
    }
    aryx[0]=2*aryx[1]-aryx[2];
    aryy[0]=2*aryy[1]-aryy[2];
    aryx[leng+1] = 2*aryx[leng]-aryx[leng-1];
    aryy[leng+1] = 2*aryy[leng]-aryy[leng-1];

t=0;
    for (j=0; j<m;j++) {
        u = (double ) j/((double) m);
        x=y=0.0;
        for ( i=0; i<n+3;i++){
            v = n*u-i+1;
            if ( v>-2 && v<=-1 )
                b = (2+v)*(2+v)*(2+v)/6;
            else if ( v>-1 && v <=0 )
                b = ( 4-6*v*v -3*v*v*v)/6;
            else if ( v>0 && v <= 1 )
                b = ( 4-6*v*v +3*v*v*v)/6;
            else if ( v > 1 && v<2 )
                b = (2-v)*(2-v)*(2-v)/6;
            else
                b =0.0;
            xx =(float) aryx[i];
            yy = (float) aryy[i];
            x = (float) xx*b + x ;
            y = (float) yy*b + y;
        }
    if(t==0){
        x0=x; y0=y;
        bound[t]=0; t++;
        bound[t]=0; t++;
    }
}

```

```

else{
    bound[t]= x-x0;    t++;
    bound[t]= y-y0;    t++;
}
    }
    write (out,bound,t*sizeof(float));
}
    close(in);
    close(out);
}

```

```

/* Using the fast B-spline basis function algorithm */
/* bsplfs.c---Subroutine to calculate a B-spline surface using open uniform
*          knot vectors ( control points ). The input is  a sequence of
*          control ( 2*n colums, i.e n control point in each slice; m
*          rows, i.e. m slices ) with binary ( unsigned char)
*          format. After inputing the input file, the programm will
*          ask the "thickness", it is the thickness between slices.
*          The output are three float format files "datx", "daty" and
*          "datz".
*
*
*/

```

```

#include<stdio.h>
#include<stdlib.h>
#include<fcntl.h>
#include<sys\types.h>
#include<sys\stat.h>

```

```

int k1=4,k2=4; /*  the degree of the basis function */
unsigned char buff[4096];
int buff1[16384],buff2[16384],buff3[16384];
unsigned short header[2];
void knot(int n, int c, int *tr);
void basis(int n, int c, float v, int *tc, float *bas);

```

```

main( int argc, char **argv)
{
    int in,out1,out2,out3,i,j,n,m,t1,t2,xt[64],yt[64],p1,p2;
    char f_in[32];

```

```

float u,v,x,y,z,tempx[64],tempy[64],delta1,delta2,x0,y0,z0,thick;

if ( argc!=3){
    printf("usage: bsplfs col row \n");
    exit(1);
}
p1 = atoi(argv[1]);
p2=  atoi(argv[2]);  /* number of fit points */
printf("input file name\n");
scanf("%s",f_in);
if ( ( in = open(f_in,O_BINARY|O_RDONLY)) == -1){
    printf("open failed on input file ");
    exit(0);
}
if ((out1 = open("datx",O_CREAT|O_TRUNC|O_WRONLY|O_BINARY,
S_IREAD|S_IWRITE))===-1){
    printf("cannot open file for writing\n");
    exit(errno);
}
if ((out2 = open("daty",O_CREAT|O_TRUNC|O_WRONLY|O_BINARY,
S_IREAD|S_IWRITE))===-1){
    printf("cannot open file for writing\n");
    exit(errno);
}
if ((out3 = open("datz",O_CREAT|O_TRUNC|O_WRONLY|O_BINARY,
S_IREAD|S_IWRITE))===-1){
    printf("cannot open file for writing\n");
    exit(errno);
}

printf("thickness = ");
scanf("%f",&thick);
read(in,header,2*sizeof(short));
n = header[0];
m = header[1];
read(in,buff,2*n*m*sizeof(char));
n--;
m--;

for(i=0;i<=n+k1;i++) xt[i]=0;
knot(n,k1,xt);
for(j=0;j<=m+k2;j++) yt[j]=0;
knot(m,k2,yt);
for(t2=0;t2<p2;t2++) printf("%c",2);
printf("\n");

```



```

    delta1=n-k1+2.0;
    delta2=m-k2+2.0;
    for(v=0.0,t2=0;v<delta2*(1+1.0/(9*p2-9.0)); v+=delta2/(p2-1.0),t2++){
for(j=0;j<=m+k2;j++) tempy[j]=0;
basis(m,k2,v,yt,tempy);
for(u=0.0,t1=0;u<delta1*(1.0+1/(9*p1-9.0)); u+=delta1/(p1-1.0),t1++){
for(i=0;i<=n+k1;i++) tempx[i]=0.0;
basis(n,k1,u,xt,tempx);
x=y=z=0.0;
for(j=0;j<=m;j++)
for(i=0;i<=n;i++){
x0=(float)buff[j*2*(n+1)+2*i];
y0=(float)buff[j*2*(n+1)+2*i+1];
z0=(float) j*thick;
x+=x0*tempx[i]*tempy[j];
y+=y0*tempx[i]*tempy[j];
z+=z0*tempx[i]*tempy[j];
}
buff1[t2*p1+t1]=(int)x;
                buff2[t2*p1+t1]=(int)y;
                buff3[t2*p1+t1]=(int)z;

}
printf("%c",1);
    }
printf("\n");
close(in);
        write(out1,buff1,t1*t2*sizeof(int));
        write(out2,buff2,t1*t2*sizeof(int));
        write(out3,buff3,t1*t2*sizeof(int));
close(out1);
close(out2);
close(out3);
exit (0);
}

```

```

void knot(int n, int c, int *tr )
{
int i;
for(i=c;i<=n+c;i++){
if(i<=n) tr[i]=i-c+1;
else tr[i]=n-c+2;
}
}

```

```

void basis(int n, int c, float u, int *tr, float *bas)
{
    int i,j,w;
    float d;

    for(i=0;i<n+c;i++){ /* the first level */
        bas[i]=0.0;
        if( u>= (float) tr[i] && u< (float) tr[i+1])
            bas[i]=1.0;
    }
    /* if( u==(float) tr[n+c]) bas[n+c]=1.0; */
    /* calculate N0.c(u)=(bas[0] ) and bas[1],bas[2] */
    for(j=2;j<=c;j++){
        for(w=0;w<=c-j;w++){
            if( bas[w]!=0)
                d=(u-tr[w])*bas[w]/((float)(tr[w+j-1]-tr[w]));
            else d=0.0;
            if(bas[w+1]!=0)
                d+=(tr[w+j]-u)*bas[w+1]/((float)(tr[w+j]-tr[w+1]));
            bas[w]=d;
        }
    }
    for(i=1; i<=n;i++){
        for(j=2;j<=c;j++){ /* calculate Ni.c(u) */
            w=i+c-j;
            if( bas[w]!=0)
                d=(u-tr[w])*bas[w]/((float)(tr[w+j-1]-tr[w]));
            else d=0.0;
            if(bas[w+1]!=0)
                d+=(tr[w+j]-u)*bas[w+1]/((float)(tr[w+j]-tr[w+1]));
            bas[w]=d;
        }
    }
}

```

Appendix D: The Program for Basic Geometric Measurements

```
/* measure.c----Subroutine to calculate basic geometric features which are
 *                described in Chapter 5.
 *
 */
#include<stdio.h>
#include<stdlib.h>
#include<fcntl.h>
#include<math.h>
#include<sys\types.h>
#include<sys\stat.h>

unsigned char buff[65536];
float bout[128];
double rout[64];
unsigned short header[64],cty,value,ovel;
unsigned char buff1[1];
int in,out,t,t1;
char f_in[32];
int m,im,noi,k=1;
unsigned long row,col,size;
void area(void);
void perimet(void);

void main(void)
{
    printf("input file name: ");
    scanf("%s",f_in);
    if ( ( in = open(f_in,O_BINARY|O_RDONLY)) == -1){
        printf("open failed on input file ");
        exit(1);
    }
    printf("the value: ");
    scanf("%u",&value);
    out=open("meas.dat",_O_BINARY|_O_CREAT|_O_WRONLY,_S_IREAD|_S_IWRITE);
    area();
    perimet();
    write(out,rout,40*sizeof(double));
    close(out);
}

void area(void)
```

```

{
    unsigned long i,j,sum,mx,my,smax,smin;
    double smean, sstd,ssum,sssum;

    read(in,header,64*sizeof(short));
    col=header[3];
    row=header[4];
    noi=header[6];
    /* write(out,header,64*sizeof(short)); */
    if ( col*row < 65536 ) size=col*row;
    else {
        k = col*row/65536;
        row=row/k;
        size = row*col-1;
    }
    ssum=0;smax=0;smin=6666;
    for (im=0,t=0;im<noi;im++,t+=3){
        sum=0; mx=0; my=0;
        for (m=0;m<k;m++){
            read(in,buff,size*sizeof(char));
            if ( row*col == 65536)
                read(in,buff1,1);
            for ( j=0;j<row;j++)
                for(i=0;i<col; i++)
                    if(buff[j*col+i]==(unsigned char)value){
                        sum++;
                        mx+=i;
                        my=my+j+m*row;
                    }
        }

        bout[t]=(float) sum;
        ssum+=(double)sum;
        sssum+=(double)sum*sum;
        if(smax<sum) smax=sum;
        if(smin>sum) smin=sum;
        bout[t+1]=(float) mx/sum;
        bout[t+2]=(float) my/sum;
        /* ctxu=(float)mx/sum;
        ctyu=(float)my/sum;
        cty=(unsigned short)ctyu;
        printf("area=%ld ctx=%f crty=%f\n",sum,ctxu,ctyu); */
    }
    smean=ssum/((double) noi);
    sstd=sssum-noi*smean*smean;

```

```

sstd=sqrt(sstd/noi);
rout[0]=1.0;
rout[1]=(double) smax;
rout[2]=2.0;
rout[3]=(double) smin;
rout[4]=3.0;
rout[5]=(double)smean;
rout[6]=4.0;
rout[7]=sstd;
close(in);
}

void perimet(void)
{
    unsigned short mask[9];
    int i1,j1,flag;
    unsigned short i,j,ctyl,lctxl,rctxl;
    float per, rdi,maxrd,minrd,sq,csum,ccsum,cmean,cstd,cmax,cmin,ctxu,ctyu;
    double psum,ppsum,pmean,pstd,pmax,pmin,comp;
    double xsum,xxsum,xmean,xstd,xmax,xmin;
    double nsum,nnsum,nmean,nstd,nmax,nmin;

    in = open(f_in,O_BINARY|O_RDONLY);
    read(in,header,64*sizeof(short));

    psum=ppsum=0; csum=ccsum=0; xsum=xxsum=0; nsum=nnsum=0;
    pmax=xmax=nmax=0; pmin=xmin=nmin=66666; cmax=0; cmin=6666;

    for (im=0,t=0;im<noi;im++,t+=3){
        maxrd=0.0; minrd=512.0;
        per=0;
        ctxu=bout[t+1]; ctyu=bout[t+2];
        for (m=0;m<k;m++){
            read(in,buff,size*sizeof(char));
            if ( row*col == 65536)
                read(in,buff1,1);
            /* counting the perimeter at the image edge */
            ctyl=(unsigned short) ctyu;
            for(j=0;j<row;j++){
                i=0;
                if(buff[j*col]==value){
                    per++;
                }
            }
            if ( j==ctyl) lctxl= i;
            sq = (double) (i-ctxu)*(i-ctxu)+(j-ctyu)*(j-ctyu);

```

```

        rdi=(float) sqrt(sq);
        if(rdi>maxrd) maxrd=rdi;
        if(rdi<minrd) minrd=rdi;
    }
    i=col-1;
    if(buff[j*col+i]==value && j!=row-1){
        per++;
    if(j==ctyl) rctxl= i;
        sq = (double) (i-ctxu)*(i-ctxu)+(j-ctyu)*(j-ctyu);
        rdi=(float) sqrt(sq);
        if(rdi>maxrd) maxrd=rdi;
        if(rdi<minrd) minrd=rdi;
    }

}
if ( buff1[0]==value){
    per++;
    i=255; j=255;
    sq = (double) (i-ctxu)*(i-ctxu)+(j-ctyu)*(j-ctyu);
    rdi=(float) sqrt(sq);
    if(rdi>maxrd) maxrd=rdi;
    if(rdi<minrd) minrd=rdi;
}
if(m==0)
    for(i=0;i<col-1;i++) if ( buff[i]==value){
        per++;
        j=0;
        sq = (double) (i-ctxu)*(i-ctxu)+(j-ctyu)*(j-ctyu);
        rdi=(float) sqrt(sq);
        if(rdi>maxrd) maxrd=rdi;
        if(rdi<minrd) minrd=rdi;
    }
    if(m==k-1)
    for(i=0;i<col-1;i++) if (buff[i]==value){
        per++;
        j=col-1;
        sq = (double) (i-ctxu)*(i-ctxu)+(j-ctyu)*(j-ctyu);
        rdi=(float) sqrt(sq);
        if(rdi>maxrd) maxrd=rdi;
        if(rdi<minrd) minrd=rdi;
    }
}
flag=0;
    for ( j=1;j<row-1;j++){
if(j==ctyl) flag=1;
        for(i=1;i<col-1; i++){

```



```

        for ( j1=-1;j1<2;j1++)
            for (i1=-1;i1<2;i1++)
                mask[(j1+1)*3+i1+1]=(unsigned short) buff[(j1+j)*col+i1+i]
                if(mask[4]==value && mask[1]+mask[3]+mask[5]+mask[7]<
4*value){
    if(flag==1) {lctxl= i; flag++;}
    if(flag==2) rctxl=i;
        if(mask[0]+mask[1]+mask[2]==0) per++;
        else if (mask[6]+mask[7]+mask[8]==0) per++;
        else if (mask[0]+mask[3]+mask[6]==0) per++;
        else if (mask[2]+mask[5]+mask[8]==0) per++;
        else per+=1.414214;
            sq = (double) (i-ctxu)*(i-ctxu)+(j-ctyu)*(j-ctyu);
            rdi=(float) sqrt(sq);
            if(rdi>maxrd) maxrd=rdi;
            if(rdi<minrd) minrd=rdi;
        }
    }
}
    psum+=(double)per;
    ppsum+=(double)per+per;
    if(pmax<per) pmax=per;
    if(pmin>per) pmin=per;
    comp=per*per/(4*3.1415927*bout[t]);
    csum+=comp;
    ccsum+=comp*comp;
    if(cmax<comp) cmax=comp;
    if(cmin>comp) cmin=comp;
    xsum+=maxrd;
    xxsum+=maxrd*maxrd;
    if(xmax<maxrd) xmax=maxrd;
    if(xmin>maxrd) xmin=maxrd;
    nsum+=minrd;
    nnsum+=minrd*minrd;
    if(nmax<minrd) nmax=minrd;
    if(nmin>minrd) nmin=minrd;
/*    bout[t1+4]=(float)lctxl;
    bout[t1+5]=(float)rctxl;    */
    printf("per= %f",per);
    printf(" comp=%f ", comp);
    printf("maxrd=%f minrd=%f\n",maxrd,minrd);
}
/* write(out,bout,6*noi*sizeof(float)); */
pmean=psum/((double)noi);

```

```

pstd=sqrt((ppsum-noi*pmean*pmean)/noi);
cmean=csum/noi;
cstd=sqrt((ccsum-noi*cmean*cmean)/noi);
xmean=xsum/noi;
xstd=sqrt((xxsum-noi*xmean*xmean)/noi);
nmean=nsum/noi;
nstd=sqrt((nnsum-noi*nmean*nmean)/noi);
rout[8]=5.0;
rout[9]=(double) pmax; /* Max-per */
rout[10]=6.0;
rout[11]=(double)pmin; /* Min-per */
rout[12]=7.0;
rout[13]=(double)pmean; /* Mean per */
rout[14]=8.0;
rout[15]=(double)pstd; /* Std-per */
rout[16]=9.0;
rout[17]=(double)cmax; /* Max-comp */
rout[18]=10.0;
rout[19]=(double)cmin; /* Min-comp */
rout[20]=11.0;
rout[21]=(double) cmean; /* Mean-comp */
rout[22]=12.0;
rout[23]=(double) cstd;
rout[24]=13.0;
rout[25]=(double) xmax; /*Max-maxrd */
rout[26]=14.0;
rout[27]=(double) xmin; /* Min-maxrd */
rout[28]=15.0;
rout[29]=(double) xmean; /* Mean-maxrd */
rout[30]=16.0;
rout[31]=(double) xstd; /* Std-maxrd */
rout[32]=17.0;
rout[33]=(double) nmax; /* Max-minrd */
rout[34]=18.0;
rout[35]=(double) nmin; /* Min-minrd */
rout[36]=19.0;
rout[37]=(double) nmean; /* Mean-minrd */
rout[38]=20.0;
rout[39]=(double) nstd; /* Std-minrd */

close(in);
}

```

```

/*
 * hishift.c----Subroutine to calculate shift of the hemispheres.
 *
 */

```

```

#include<stdio.h>
#include<stdlib.h>
#include<fcntl.h>
#include<math.h>
#include<sys\types.h>
#include<sys\stat.h>

```

```

unsigned char buffer1[65536],buffer2[65536];
float bout[256];
double rout[64];
unsigned short header[64],value1,value2;
unsigned char buff1[1];
int in1,in2,t;
int m,im,noi,k=1;
unsigned long row,col,size;
void cent(void);
void width(void);

```

```

void main(void)
{

```

```

    printf("hishift \n");
    if ( ( in1 = open("left.img",O_BINARY|O_RDONLY)) == -1){
        printf("open failed on input file ");
        exit(1);
    }
    value1=50;
    if ( ( in2 = open("right.img",O_BINARY|O_RDONLY)) == -1){
        printf("open failed on input file ");
        exit(1);
    }
    value2=100;

```

```

    cent();
    width();
}

```

```

void cent(void)
{

```

```

unsigned long i,j,sum1,sum2,mx1,mx2,my1,my2,smax,smin;

read(in1,header,64*sizeof(short));
read(in2,header,64*sizeof(short));
col=header[3];
row=header[4];
noi=header[6];
/*  write(out,header,64*sizeof(short));  */
if ( col*row < 65536 ) size=col*row;
else {
    k = col*row/65536;
    row=row/k;
    size = row*col-1;
}
for (im=0,t=0;im<noi;im++,t+=4){
    sum1=sum2=0; mx1=mx2=0; my1=my2=0;
    for (m=0;m<k;m++){
        read(in1,buffer1,size*sizeof(char));
        read(in2,buffer2,size*sizeof(char));
        if ( row*col == 65536){
            read(in1,buff1,1);
            read(in2,buff1,1);
        }
        for ( j=0;j<row;j++){
            for(i=0;i<col; i++){
                if(buffer1[j*col+i]==(unsigned char)value1){
                    sum1++;
                    mx1+=i;
                    my1=my1+j+m*row;
                }
                if(buffer2[j*col+i]==(unsigned char)value2){
                    sum2++;
                    mx2+=i;
                    my2=my2+j+m*row;
                }
            }
        }
    }
}

}

}

}
bout[t]=(float)mx1/sum1;
bout[t+1]=(float)my1/sum1;
bout[t+2]=(float)mx2/sum2;
bout[t+3]=(float)my2/sum2;
}

```

```

    close(in1);
    close(in2);
}

void width(void)
{
    unsigned short i,j,ctyl1,ctyl2,dmax,dmin,dsum,ddsum,dr,dl,w,sdlr;
    float dmean,dstd,r5;
    double dstdb;
    int out;

    in1 = open("left.img",O_BINARY|O_RDONLY);
    read(in1,header,64*sizeof(short));
    in2 = open("right.img",O_BINARY|O_RDONLY);
    read(in2,header,64*sizeof(short));
    dmax=0; dmin=512; dsum=0; ddsum=0;sdlr=0;
    out=open("meas.dat",O_BINARY|O_WRONLY|O_APPEND,S_IREAD|S_IWRITE);
    for (im=0,t=0;im<noi;im++,t+=4){
        for (m=0;m<k;m++){
            read(in1,buffer1,size*sizeof(char));
            read(in2,buffer2,size*sizeof(char));
            if ( row*col == 65536){
                read(in1,buff1,1);
                read(in2,buff1,1);
            }

            ctyl1=(unsigned short)bout[t+1];
            ctyl2=(unsigned short)bout[t+3];
            i=col-1;
            while(i>=0 && buffer1[ctyl1*row+i]!=value1) i--;
            dl=i-(unsigned short)bout[t];
            i=0;
            while(i<col && buffer2[ctyl2*row+i]!=value2) i++;
            dr=(unsigned short)bout[t+2]-i;
            if(dl>dr) w=dl-dr;
            else w=dr-dl;
        }
        sdlr=sdlr+dl+dr;
        if(dmax<w) dmax=w;
        if(dmin>w) dmin=w;
        dsum+=w;
        ddsum+=w*w;
        printf("%d ", im);
    }
    printf("\n");
    dmean=(float)dsum/((float)noi);

```

```

dstd=(float)ddsum-dmean*dmean;
dstdb=sqrt((double)dstd/noi);
r5=(float)dsum/((float)sdlr);
rout[0]=53.0;
rout[1]=(double) r5;
rout[2]=54.0;
rout[3]=(double)dmax;
rout[4]=55.0;
rout[5]=(double)dmin;
rout[6]=56.0;
rout[7]=(double)dmean;
rout[8]=57.0;
rout[9]=(double) dstdb;
write(out,rout,10*sizeof(double));

close(in1);
close(in2);
close(out);
}

```

```

/* File name: volume.c---calculate the volume of a 3-d object.
*
* The 3-d object consist of a sequence of 2-d images with
* uniform grey level. Usually, each 2-d image may contain
* different objects ( with different grey level ). To calculate
* the volume of a particular object, the grey level for this
* object is needed when the program is running.
* The thickness of each slice is supposed to be one voxel.
* In practice, a slice with thickness h voxels can be treated
* as a stack ( h ) of slices each of them has thickness 1.The
* volume between any two of these image slices (S1 and S2) is:
*
* 
$$v = h(\text{area}(S1) + \text{area}(S2) + \sqrt{\text{area}(S1)*\text{area}(S2)})/3$$

*
*/

```



```

#include<stdio.h>
#include<stdlib.h>
#include<fcntl.h>
#include<math.h>
#include<sys\types.h>
#include<sys\stat.h>

unsigned char buff[65536],buff1[1];
unsigned short header[64];
int in,out,im,noi,greyl,greyl2,t,m,k;
char f_in[32];
float vol, valued[16];
double rout[32];
long col,row,size,thick,area0,arean;
void perimet(void);

main()
{
    unsigned short i,j;
    long area,areap,centx,centy,centz;
    float gravtx,gravty,gravtz,areal;

    printf("input file name: ");
    scanf("%s",f_in);
    if ( ( in = open(f_in,O_BINARY|O_RDONLY)) == -1){
        printf("open failed on input file ");
        exit(0);
    }

    read(in,header,64*sizeof(short));
    out = open("meas.dat",O_CREAT|O_WRONLY|O_BINARY,S_IREAD|S_IWRITE);

    col=header[3];
    row=header[4];
    noi=header[6];
    printf("number of image(s) = %d\n",noi);

    printf("grey value = ");
    scanf("%d",&greyl);
    printf("overlay grey leve = ");
    scanf("%d",&greyl2);
    printf("thickness = ");
    scanf("%d",&thick);
    if ( col*row < 65536 ) size=col*row;

```

```

else {
    k = col*row/65536;
    row=row/k;
    size = row*col-1;
}

vol=0;
centx=centy=centz=0;
for (im=0;im<noi;im++){
    read(in,buff,size*sizeof(char));
    if ( col*row==65536 )
        read(in,buff1,1);
    area=0;
    for( j=0;j<row;j++){
        for (i=0;i<col;i++){
            if ( buff[j*col+i]==grey1|| buff[j*col+i]==grey2){
                area++;
                if(im<noi-1){
                    centx=centx+i*thick;
                    centy=centy+j*thick;
                    for(t=0;t<thick;t++) centz=centz+im*thick+t;
                }
                else centz=centz+im*thick;
            }
        }
    }
    if ( im !=0 ){
        area1 = (float)area*areap;
        area1 = (float) sqrt((double)area1);
        vol+=(float) thick*(area+areap+area1)/3.0;;
    }
    areap=area;
    if(im==0) area0=area;
    if(im==noi-1) arean=area;
}

gravtx=(float)centx/vol; gravty=(float)centy/vol;
gravtz=(float)centz/vol-0.5;
rout[0]=45.0;
rout[1]=(double) vol;
rout[2]=46.0;
rout[3]=(double) gravtx;
rout[4]=47.0;
rout[5]=(double) gravty;
rout[6]=48.0;

```

```

rout[7]=(double) gravtz;
valued[0]= vol;
valued[1]= gravtx;
valued[2]= gravty;
valued[3]= gravtz;
write(out,rout,8*sizeof(double));
close(in);
perimet();
close(out);
}

```

```

void perimet(void)
{
    unsigned char md1,md2,md3,md4,msum,mask[9];
    int i1,j1;
    unsigned long i,j,per,perp;
    float rdi,maxrd=.0,minrd=512.0,ss=.0;

    in = open(f_in,O_BINARY|O_RDONLY);
    read(in,header,64*sizeof(short));

    for (im=0;im<noi;im++){
        per=0;
        for(m=0; m<k; m++){
            read(in,buff,size*sizeof(char));
            if ( row*col == 65536)
                read(in,buff1,1);
            /*    counting the perimeter at the image edge    */

            for(j=0;j<row;j++){
                i=0;
                if(buff[j*col]==grey1){
                    per++;
                    rdi=sqrt((i-valued[1])*(i-valued[1])+(j-valued[2])*(j-valued[2])
(im*thick-valued[3])*(im*thick-valued[3]));
                    if(rdi>maxrd) maxrd=rdi;
                    if(rdi<minrd) minrd=rdi;
                }
                i=col-1;
                if(buff[j*col+col-1]==grey1 && j!=row-1){
                    per++;
                    rdi=sqrt((i-valued[1])*(i-valued[1])+(j-valued[2])*(j-valued[2])
(im*thick-valued[3])*(im*thick-valued[3]));

```

```

        if(rdi>maxrd) maxrd=rdi;
        if(rdi<minrd) minrd=rdi;
    }

}

if ( buff1[0]==grey1){
    per++;
    i=255; j=255;
    rdi=sqrt((i-valued[1])*(i-valued[1])+(j-valued[2])*(j-valued[2])+(im*thick-valued[3])*(im*thick-valued[3]));
    if(rdi>maxrd) maxrd=rdi;
    if(rdi<minrd) minrd=rdi;
}

if(m==0)
    for(i=0;i<col-1;i++) if ( buff[i]==grey1){
        per++;
        j=0;
        rdi=sqrt((i-valued[1])*(i-valued[1])+(j-valued[2])*(j-valued[2])+(im*thick-valued[3])*(im*thick-valued[3]));
        if(rdi>maxrd) maxrd=rdi;
        if(rdi<minrd) minrd=rdi;
    }

if(m==k-1)
    for(i=0;i<col-1;i++) if (buff[i]==grey1){
        per++;
        j=col-1;
        rdi=sqrt((i-valued[1])*(i-valued[1])+(j-valued[2])*(j-valued[2])+(im*thick-valued[3])*(im*thick-valued[3]));
        if(rdi>maxrd) maxrd=rdi;
        if(rdi<minrd) minrd=rdi;
    }

for ( j=1;j<row-1;j++){
    for(i=1;i<col-1; i++){
        if(im==0||im==noi-1){
            if(buff[j*col+i]==grey1){
                rdi=sqrt((i-valued[1])*(i-valued[1])+(j-valued[2])*(j-valued[2])+(im*thick-valued[3])*(im*thick-valued[3]));
                if(rdi>maxrd) maxrd=rdi;
                if(rdi<minrd) minrd=rdi;
            }
        }
    }

    msum=0; md1=0; md2=0; md3=0; md4=0;
    for ( j1=-1;j1<2;j1++)
        for (i1=-1;i1<2;i1++){
            mask[(j1+1)*3+i1+1]=buff[(j1+j)*col+i1+i];

```

```

        msum+=mask[(j1+1)*3+i1+1];
        if(i1==j1) md1+=mask[(j1+1)*3+i1+1];
        if(i1==-j1) md2+=mask[(j1+1)*3+i1+1];
        if(i1==0) md3+=mask[(j1+1)*3+i1+1];
        if(j1==0) md4+=mask[(j1+1)*3+i1+1];
    }
    if(mask[4]==grey1)
        if(mask[1]!=grey1||mask[3]!=grey1||mask[5]!=grey1 \
        ||mask[7]!=grey1){
            per++;
            rdi=sqrt((i-valued[1])*(i-valued[1])+(j-valued[2])*(j-
valued[2])+(im*thick-valued[3])*(im*thick-valued[3]));
            if(rdi>maxrd) maxrd=rdi;
            if(rdi<minrd) minrd=rdi;
        }
    }
}
}
if(im!=0)
    ss+= (float) (per+perp)*thick/2.0;
    perp=per;
}
rout[0]=49.0;
rout[1]=(double)area0+arean+ss;
rout[2]=50.0;
rout[3]=(double)maxrd;
rout[4]=51.0;
rout[5]=(double)minrd;
rout[6]=52.0;
rout[7]=(double)ss*sqrt((double)ss)/(10.63472311*valued[0]);
write(out,rout,8*sizeof(double));
close(in);
}

```

Appendix E: The Program for Shape Analysis

```
/*
 * shape.h--- header file for shape analysis subroutine.
 *
 */
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<conio.h>
#include<fcntl.h>
#include<sys\types.h>
#include<sys\stat.h>
#include<graph.h>

struct menus {
    char *option;
    char *optiontext;
} menu;

int in,out;
char f_in[32],f_out[32];
unsigned char buffer[512];
unsigned short header[2];
float data[1024];
dispf(int,int,int,int);
int getpt(struct menus *);
char *getptc(struct menus *);
atob(int);
bspline(int);
fft(int);
match(int);
sym(void);
seerl(int);

/*
 * shape.c --- shape analysis top level subroutine.
 *
 */

#include "shape.h"

static struct menus choice[4] =
{
    {"A","Auto measure"},
    {"M","Manual measure"},
```



```

        {"S","Symmetry measure"},
        {"E","Exit to DOS"}
    };
static struct menus mainmenu[6] =
{
    {"1","Ascii to binary"},
    {"2","B-Splines"},
    {"3","Fourier descriptors"},
    {"4","Match"},
    {"5","Show result"},
    {"6","Back to main menu"}
};
void autoch(),manual(),symmetry(),exitpro();

void main(void)
{
    char *text;

    for(;;){
        text=getptc(choice);
        *text=toupper(*text);
        switch(*text){
            case 'A': autoch();break;
            case 'M': manual();
            case 'S': symmetry();
            case 'E': exitpro();
        }
    }

    void exitpro(void)
    {
        _clearscreen(_GCLEARSCREEN);
        exit(1);
    }

    void autoch(void)
    {
        int flag=1;

        atob(flag);
        bspline(flag);
        fft(flag);
        match(flag);
    }
}

```

```

seerl(flag);
}

void symmetry(void)
{
int flag=1;

atob(flag);
bspline(flag);
sym();

}

void manual(void)
{
int option,flag=0;

for(;;) {
option=getpt(mainmenu);
switch(option){
case 1:atob(flag);
break;
case 2:bspline(flag);
break;
case 3:fft(flag);
break;
case 4:match(flag);
break;
case 5:seerl(flag);
break;
case 6:main();
}
}

}

#include"shape.h"

atob(int flag)
{
int length,slices,i, n=0;
FILE *str;

_clearscreen(_GCLEARSCREEN);
_settextposition(6,10);
_outtext("Input file: ");

```

```

scanf("%s",f_in);

if((str=fopen(f_in,"r"))==NULL) {
printf("cannot open file\n");
exit(1);
}
if (flag==0){
_settextposition(8,10);
_outtext("Out file: ");
scanf("%s",f_out);
    out = open(f_out,O_CREAT|O_TRUNC|O_WRONLY|O_BINARY,
S_IREAD|S_IWRITE);
}
else
out = open("atob.out",O_CREAT|O_TRUNC|O_WRONLY|O_BINARY,
S_IREAD|S_IWRITE);

    fscanf(str,"%d",&length);
    header[0]=(unsigned short) length;

while(!feof(str)){
fscanf(str,"%d",&i);
buffer[n]=i;
n++;
}
slices = n/(2*header[0]);
header[1]=slices;
write(out,header,2*sizeof(short));
write(out,buffer,2*length*slices*sizeof(char));

close(out);
fclose(str);
}

#include<math.h>
#include"shape.h"

#define m 512

float bound[1024];
unsigned char aryx[100],aryy[100];

bspline(int flag)

```

```

{
    int i,j,k,t,n,leng,size,im,noi;
    double b,u,v;
    float x,y,xx,yy,x0,y0;

    if(flag==0){
        _clearscreen(_GCLEARSCREEN);
    _settextposition(6,10);
        _outtext("input file name: ");
        scanf("%s",f_in);
        if ( ( in = open(f_in,O_BINARY|O_RDONLY)) == -1){
            printf("open failed on input file ");
            exit(0);
        }
    _settextposition(8,10);
    _outtext("output file name: ");
        scanf("%s",f_out);
        out = open(f_out,O_CREAT|O_TRUNC|O_WRONLY|O_BINARY,S_IREAD|S_IWRITE);
    }
    else {
in=open("atob.out",O_BINARY|O_RDONLY);
        out=open("bspline.out",O_CREAT|O_TRUNC|O_WRONLY|O_BINARY,
S_IREAD|S_IWRITE);
    }

    read(in,header,2*sizeof(short));
    leng = header[0];
    noi = header[1];
    size = 2*leng;
    n=leng-1;
    header[0]= m;

    write(out,header,2*sizeof(short));

    for (im=0; im<noi;im++){
        read(in,buffer,size*sizeof(char));
        j=0; i=1;
        while ( j < size && i <= leng) {
            aryx[i] = buffer[j]; j++;
            aryy[i] = buffer[j]; j++;
            i++;
        }
        aryx[0]=2*aryx[1]-aryx[2];
        aryy[0]=2*aryy[1]-aryy[2];
        aryx[leng+1] = 2*aryx[leng]-aryx[leng-1];
    }
}

```

```

    aryy[leng+1] = 2*aryy[leng]-aryy[leng-1];

t=0;
for (j=0; j<m;j++) {
    u = (double ) j/((double) m);
    x=y=0.0;
    for ( i=0; i<n+3;i++){
        v = n*u-i+1;
        if ( v>-2 && v<=-1 )
            b = (2+v)*(2+v)*(2+v)/6;
        else if ( v>-1 && v <=0 )
            b = ( 4-6*v*v -3*v*v*v)/6;
        else if ( v>0 && v <= 1 )
            b = ( 4-6*v*v +3*v*v*v)/6;
        else if ( v > 1 && v<2 )
            b = (2-v)*(2-v)*(2-v)/6;
        else
            b =0.0;
        xx =(float) aryx[i];
        yy = (float) aryy[i];
        x = (float) xx*b + x ;
        y = (float) yy*b + y;
    }
if(t==0){
    x0=x; y0=y;
    bound[t]=0; t++;
    bound[t]=0; t++;
}
else{
    bound[t]= x-x0; t++;
    bound[t]= y-y0; t++;
}
    }
    write (out,bound,t*sizeof(float));
}
    close(in);
    close(out);
}

#include "shape.h"

int getpt(struct menus hmenu[6])
{
    int i, option;
    char numb[80];

```

```

_settextwindow(1,1,25,80);
_clearscreen(_GCLEARSCREEN);
dispf(3,15,22,65);

for(i=0;i<=5;i++){
    _settextposition(5+(i*3),21);
    _outtext(hmenu[i].option);
    _settextposition(5+(i*3),29);
    _outtext(hmenu[i].optiontext);
    dispf(5+(i*3)-1,18,5+(i*3)+1,24);
}
_settextwindow(23,1,25,80);
do {
    _clearscreen(_GWINDOW);
    _settextposition(2,21);
    _outtext("Type option and press Enter:");
    gets(numb);
    option = atoi(numb);
}
while ((option<1 ) || (option >6));
_settextwindow(1,1,25,80);
return(option);
}

```

```

#include "shape.h"

```

```

char *getptc(struct menus hmenu[4])
{
    int i;
    char *text;

    _settextwindow(1,1,25,80);
    _clearscreen(_GCLEARSCREEN);
    dispf(2,10,23,60);
    _settextposition(4,24);
    _outtext("SHAPE ANALYSYS PROGRAM");
    for(i=0;i<4;i++){
        _settextposition(8+(i*4),21);
        _outtext(hmenu[i].option);
        _settextposition(8+(i*4),29);
        _outtext(hmenu[i].optiontext);
        dispf(8+(i*4)-1,18,8+(i*4)+1,24);
    }
}

```



```

}

_settextwindow(24,1,25,80);
    _clearscreen(_GWINDOW);
_settextposition(1,21);
_outtext("Type option  and press Enter:");
gets(text);
_settextwindow(1,1,25,80);
return(text);
}


#include<math.h>
#include"shape.h"

#define swp(a,b) tempr=(a); (a)=(b); (b)=tempr;
int isign=-1;

fft(int flag)
{
    int nn, noi,i,im;

    if ( flag==0){
        _clearscreen(_GCLEARSCREEN);
        _settextposition(6,10);
        _outtext("input file name: ");
        scanf("%s",f_in);
        if ( ( in = open(f_in,O_BINARY|O_RDONLY)) == -1){
            printf("open failed on input file ");
            exit(0);
        }
        _settextposition(8,10);
        _outtext("output file  name: ");
        scanf("%s",f_out);
        out = open(f_out,O_CREAT|O_TRUNC|O_WRONLY|O_BINARY,
S_IREAD|S_IWRITE);
    }
    else {
        in = open("bspline.out",O_BINARY|O_RDONLY);
        out = open("fft.out",O_CREAT|O_TRUNC|O_WRONLY|O_BINARY,
S_IREAD|S_IWRITE);
    }

        read(in,header,2*sizeof(short));
        nn =  header[0];

```

```

        noi = header[1];
        write(out,header,2*sizeof(short));

for ( im=0; im<noi; im++){
read(in,data,2*nn*sizeof(float));
four1 (nn);
if(isign==-1)
    for (i=0;i<2*nn;i++)
        data[i] =data[i]/((float) nn);
    write(out,data,2*nn*sizeof (float));
}
close(in);
close(out);

}

```

```

four1(nn)
int nn;
{
int n,mmax,m,j,istep,i;
float wtemp,wr,wpr,wpi,wi,theta;
double tempr,tempi;

n = nn <<1;
j=0;
for(i=0;i<n; i+=2) {
if ( j>i ) {
swp(data[j],data[i]);
swp(data[j+1],data[i+1]);
}
m = n >> 1;
while ( m>=2 && j>=m ) {
j-=m;
m>>=1;
}
j +=m;
}
mmax=2;
while( n > mmax ) {
istep=2*mmax;
theta=6.28318530718/(isign*mmax);
wtemp=sin(0.5*theta);
wpr=-2.0*wtemp*wtemp;
wpi = sin(theta);

```

```

wr=1.0;
wi=0.0;
for( m=0; m<mmax; m+=2) {
for( i=m; i<n; i+=istep) {
j=i+mmax;
tempr=wr*data[j]-wi*data[j+1];
tempi=wr*data[j+1]+wi*data[j];
data[j]=data[i]-tempr;
data[j+1]=data[i+1]-tempi;
data[i]+=tempr;
data[i+1]+=tempi;
}
wr=(wtemp=wr)*wpr-wi*wpi+wr;
wi=wi*wpr+wtemp*wpi+wi;
}
mmax=istep;
}
}

```

```

#include<math.h>
#include"shape.h"

```

```

#define pi 3.1415926535897932384626

```

```

int in2;
char f_in2[32];
float data2[1024];
double rho[1024],psi[1024];
FILE *fp;

```

```

match(int flag)
{

```

```

    int i,n,n0,stp,im;
    double phi,d,dd,cc1,cc2,cc,bb,aa,a,b,alpha,theta;
    double s,rotat,x,y,x0,y0;

```

```

    _clearscreen(_GCLEARSCREEN);
    _settextwindow(4,10,20,70);
    if (flag==0){
_settextposition(4,2);
_outtext("input file1 name: ");
scanf("%s",f_in);
if ( ( in = open(f_in,O_BINARY|O_RDONLY)) == -1){

```

```

        printf("open failed on input file ");
        exit(0);
    }
    _settextposition(6,2);
    _outtext("input file2 name: ");
    scanf("%s",f_in2);
    if ( ( in2 = open(f_in2,O_BINARY|O_RDONLY)) == -1){
        printf("open failed on input file ");
        exit(0);
    }
}
else{
    _settextposition(4,2);
    _outtext(" input the file to compare with: ");
    scanf("%s",f_in2);
    if ( ( in2 = open(f_in2,O_BINARY|O_RDONLY)) == -1){
        printf("open failed on input file ");
        exit(0);
    }
}
in=open("fft.out",O_BINARY|O_RDONLY);
}

fp=fopen("match.out","a");

    read(in,header,2*sizeof(short));
    read(in2,header,2*sizeof(short));
    n= header[0];
    rho[0]=0.0;
    psi[0]=0.0;
    for(im=0; im<header[1]; im++){
read(in,data,2*n*sizeof(float));
read(in2,data2,2*n*sizeof(float));
aa =0.0;
bb =0.0;
rho[0]=0.0;
psi[0]=0.0;
for ( i=2; i<2*n; i+=2) {
a= (double) data[i]*data2[i]+data[i+1]*data2[i+1];
b= (double) data[i]*data2[i+1]-data[i+1]*data2[i];
rho[i/2]= sqrt(a*a+b*b);
if ( a==0.0 && b!= 0.0) psi[i/2]=pi/2.0;
else if ( a== 0.0 && b==0.0 ) psi[i/2]=0.0;
else psi[i/2]=atan(b/a);
bb= bb+ (double) (data2[i]*data2[i]+data2[i+1]*data2[i+1]);
aa= aa+ (double) (data[i]*data[i]+data[i+1]*data[i+1]);

```

```

}
d=9999999.9;
/* for(n0=0; n0<n; n0++){    all boundary start at original (0,0)
alpha=2.0*pi*((double)n0)/((double) n);    */
n0=0;

        alpha=0;
cc1=0.0; cc2=0.0;
for(i=1;i<n;i++){
cc1= cc1 + rho[i]*sin(psi[i]+i*alpha);
cc2= cc2 + rho[i]*cos(psi[i]+i*alpha);
}
if ( cc2 == 0.0 && cc1 !=0.0) phi = -pi/2.0;
else if ( cc2==0.0 && cc1==0.0 ) phi=0.0;
else phi = atan(-cc1/cc2);
cc=0.0;
for(i=1; i<n; i++){
cc= cc+ rho[i]*cos(psi[i]+i*alpha+phi);
s = cc/bb;
if( s < 0 ) {
s = -s;
cc = -cc;
}
dd=0.0;
for(i=2; i<2*n; i+=2){
theta=((double)i/2.0)*alpha+phi;
x0=data2[i]*cos(theta)-data2[i+1]*sin(theta);
y0=data2[i+1]*cos(theta)+data2[i]*sin(theta);
x= s*x0; y= s*y0;
dd = dd + sqrt((data[i]-x)*(data[i]-x)+
(data[i+1]-y)*(data[i+1]-y));
}
if( dd < d){
/* stp=n0;    */
rotat = phi*180.0/pi;
d = dd;
}
/* }    */

        fprintf(fp,"\n");
fprintf(fp,"rotate = %lf scale = %lf\n",rotat,s);
fprintf(fp,"minimum = %lf\n",d);
}

        close(in);
        close(in2);
fclose(fp);
}

```

```

#include<math.h>
#include"shape.h"

#define swp(a,b) tempr=(a); (a)=(b); (b)=tempr;
float data1[1024],data2[1024],data3[1024];
FILE *fp;

sym()
{
int nn, noi,i,im,isign;
float d, sum;

in = open("bspline.out",O_BINARY|O_RDONLY);
fp=fopen("sym.out","a");

    read(in,header,2*sizeof(short));
    nn = header[0];
    noi = header[1];

for ( im=0; im<noi; im++){
read(in,data,2*nn*sizeof(float));
four2(nn,-1);
    for (i=0;i<2*nn;i++)
        data2[i]=data1[i];
    four2(nn,1);
    for (i=0;i<2*nn;i++)
        data3[i] =data1[i];
sum=0;
for(i=1; i<nn; i++){
d= data2[2*i]+data3[2*i];
if( d> 0) sum+=d;
else sum=sum-d;
}
    fprintf(stderr,"smy = %.4f ",sum);
fprintf(fp, "%d %.4f \n",im,sum);
}
close(in);
fclose(fp);

}

```



```

four2(int nn, int isign)
{
    int n,mmax,m,j,istep,i;
    float wtemp,wr,wpr,wpi,wi,theta;
    double tempr,tempi;

    for(i=0; i<2*nn; i++) data1[i]=data[i];
    n = nn <<1;
    j=0;
    for(i=0;i<n; i+=2) {
        if ( j>i ) {
            swp(data1[j],data1[i]);
            swp(data1[j+1],data1[i+1]);
        }
        m = n >> 1;
        while ( m>=2 && j>=m ) {
            j-=m;
            m>>=1;
        }
        j +=m;
    }
    mmax=2;
    while( n > mmax ) {
        istep=2*mmax;
        theta=6.28318530718/(isign*mmax);
        wtemp=sin(0.5*theta);
        wpr=-2.0*wtemp*wtemp;
        wpi = sin(theta);
        wr=1.0;
        wi=0.0;
        for( m=0; m<mmax; m+=2) {
            for( i=m; i<n; i+=istep) {
                j=i+mmax;
                tempr=wr*data1[j]-wi*data1[j+1];
                tempi=wr*data1[j+1]+wi*data1[j];
                data1[j]=data1[i]-tempr;
                data1[j+1]=data1[i+1]-tempi;
                data1[i]+=tempr;
                data1[i+1]+=tempi;
            }
            wr=(wtemp*wr)*wpr-wi*wpi+wr;
            wi=wi*wpr+wtemp*wpi+wi;
        }
        mmax=istep;
    }
}

```

```
for(i=0; i<2*nn; i++) data1[i]=data1[i]/((float) nn);
}
```

```
#include "shape.h"
```

```
#define BRIGHT_WHITE 15
```

```
#define RED 4
```

```
#define WINDOW_SIZE 14
```

```
seerl(int flag)
```

```
{
```

```
FILE *input;
```

```
char text[80];
```

```
int line_number=0;
```

```
int save_color;
```

```
struct rccoord save_position;
```

```
_clearscreen(_GCLEARSCREEN);
```

```
save_color=_getcolor();
```

```
if((input=fopen("match.out","r"))==NULL){
```

```
_settextposition(3,10);
```

```
_outtext("Error opening the file");
```

```
}
```

```
else {
```

```
_clearscreen(_GCLEARSCREEN);
```

```
_settextwindow(5,1,20,80);
```

```
_settextcolor(BRIGHT_WHITE);
```

```
while(fgets(text,sizeof(text),input)){
```

```
_outtext(text);
```

```
if(++line_number%WINDOW_SIZE==0){
```

```
_settextcolor(RED);
```

```
save_position=_gettextposition();
```

```
_settextwindow(5,1,22,80);
```

```
_settextposition(22,1);
```

```
_outtext("Press any key to continue");
```

```
_settextcolor(BRIGHT_WHITE);
```

```
do
```

```
getch();
```

```
while(kbhit());
```

```
_settextposition(22,1);
```

```
_outtext(" ");
```

```
_settextwindow(5,1,20,80);
```

```
_settextposition(save_position.row,  
save_position.col);  
}  
}  
fclose(input);  
}  
_settextwindow(1,1,25,80);  
_settextcolor(save_color);  
_settextposition(25,10);  
_outtext("Press Enter to continue");  
getchar();  
}
```

Appendix F: The Program for Surface Measurement

```
/* Name:   measurf.c---compute surface Gaussian curvature and mean curvature
 * Description: This program computes the Gaussian curvature and mean
 *              mean curvature of a surface.
 *              The surface is a B-splines surface it is generated through
 *              a set of surface trace (control) points. The control points
 *              are obtained from a sequence of 2-D image slices. So during
 *              the computing procedure, you are asked to give the thickness
 *              between the slices. The input of the program is the control
 *              points ( float format). According to the control points, the
 *              program will produce a B-splines surface. Then it will compute
 *              the Gaussian curvature (K) and mean curvature (H) at each
 *              point of the surface. There are totally m1*m2 sample points
 *              on the surface. According the sign of K and H, the surface :
 *              divided into four regions: ellipic ( with gray level 250 ),
 *              parabolic (with gray level 200 ), planar ( with gray level
 *              150 ), and hyperbolic (with gray level 90 ).
 * Usage:   measurf col row
 *
 *
 *
 */
#include<stdio.h>
#include<stdlib.h>
#include<fcntl.h>
#include<sys\types.h>
#include<sys\stat.h>
#include<math.h>

unsigned char buff[8192];
float curvat[512];
double rout[64];
unsigned char surface[1024];
float tempx[64],tempy[64],d1tempx[64],d1tempy[64],d2tempx[64],d2tempy[64];
int xt[64],yt[64];
char f_in[32], f_out[32];
unsigned short header[2];

void knot(int n, int c, int *tr);
void dbasis(int n, int c, float v, int *tc, float *bas,float *d1bas,
float *d2bas);

main(int argc, char **argv)
```

```

{
    int k1=4,k2=4;
    int p1,p2;
    int in,out,out2,out3;
    int i,j,k,m,n,t1,t2;
    int ell,par,pla,hyp;
    float a,b,u,v,x0,y0,z0,mb,nb,d1mb,d1nb,d2mb,d2nb,delta1,delta2;
    float xu1,yu1,zu1,xv1,yv1,zv1,xuv,yuv,zuv,xu2,yu2,zu2,xv2,yv2,zv2;
    double E,F,G,L,M,N;
    double fb,ft,fth,max,min,mean,std,kt,sum,h,maxh,minh,meanh,svd,svdh,sum2;
    double sumh,sumh2,maxell,minell,sumell,sumell2,meanell,svdell,sumpar2;
    double maxhyp,minhyp,sumhyp,sumhyp2,meanhyp,svdhyp,maxpar,minpar,sumpar;
    double meanpar,svdpar,maxpla,minpla,sumpla,sumpla2,meanpla,svdpla;
    float ellf,parf,plaf,hypf,thick;

    if ( argc!=3){
printf(" usage: measurf col row \n");
exit(1);
    }
    p1 = atoi(argv[1]);
    p2 = atoi(argv[2]);
    printf("input file name\n");
    scanf("%s",f_in);
    if ( ( in = open(f_in,O_BINARY|O_RDONLY)) == -1){
        printf("open failed on input file ");
        exit(0);
    }

    printf("output file name\n");
    scanf("%s",f_out);
    out = open(f_out,O_CREAT|O_TRUNC|O_WRONLY|O_BINARY,S_IREAD|S_IWRITE);

    printf("thickness  = ");
    scanf("%f",&thick);

    read(in,header,2*sizeof(short));
    n = header[0];
    m = header[1];
    read(in,buff,2*n*m*sizeof(char));
    n--;
    m--;
    out2 = open("measf.dat",O_CREAT|O_TRUNC|O_WRONLY|O_BINARY,
S_IREAD|S_IWRITE);
    out3 = open("meas.dat",O_CREAT|O_WRONLY|O_BINARY,S_IREAD|S_IWRITE);
    ell=par=hyp=pla=0;

```

```

sum = sum2= sumh = sumh2=0.0;
sumell=sumell2=sumhyp=sumhyp2=0.0;
sumpar=sumpar2=sumpla=sumpla2=0.0;
max =maxh= maxell =maxhyp = -9999.0;
maxpar=maxpla=-9999.0;
min = minh= minell = minhyp =9999.0;
minpar=minpla=9999.0;
for(i=0;i<=n+k1;i++) xt[i]=0;
knot(n,k1,xt);
for(j=0;j<=m+k2;j++) yt[j]=0;
knot(m,k2,yt);
for(t2=0;t2<p2;t2++) printf("%c",2);
printf("\n");
delta1=n-k1+2.0;
delta2=m-k2+2.0;
for(v=0.0,t2=0;v<delta2*(1.0+1.0/(9*p2-9.0)); v+=delta2/(p2-1.0),t2++){
for(j=0;j<=m+k2;j++){
tempy[j]=0.0;
d1tempy[j]=0.0;
d2tempy[j]=0.0;
}
    dbasis(m,k2,v,yt,tempy,d1tempy,d2tempy);
    for(u=0.0,t1=0;u<delta1*(1+1.0/(9*p1-9.0)); u+=delta1/(p1-1.0),t1++){
for(i=0;i<=n+k1;i++){
tempx[i]=0.0;
d1tempx[i]=0.0;
d2tempx[i]=0.0;
}
    dbasis(n,k1,u,xt,tempx,d1tempx,d2tempx);

        xu1=yu1=zu1=0.0;
        xv1=yv1=zv1=0.0;
        xuv=yuv=zuv=0.0;
        xu2=yu2=zu2=0.0;
        xv2=yv2=zv2=0.0;
for(j=0;j<=m;j++){
for(i=0;i<=n;i++){
        x0=(float)buff[2*j*(n+1)+2*i];
        y0=(float)buff[2*j*(n+1)+2*i+1];
        z0=thick*j;
mb=tempy[j]; nb=tempx[i];
d1mb=d1tempy[j]; d1nb=d1tempx[i];
d2mb=d2tempy[j]; d2nb=d2tempx[i];
        xu1 = x0*d1nb*mb + xu1;
        yu1 = y0*d1nb*mb + yu1;

```



```

        zu1 = z0*d1nb*mb + zu1;
        xv1 = x0*d1mb*nb + xv1;
        yv1 = y0*d1mb*nb + yv1;
        zv1 = z0*d1mb*nb + zv1;
        xuv = x0*d1nb*d1mb + xuv;
        yuv = y0*d1nb*d1mb + yuv;
        zuv = z0*d1nb*d1mb + zuv;
        xu2 = x0*d2nb*mb + xu2;
        yu2 = y0*d2nb*mb + yu2;
        zu2 = z0*d2nb*mb + zu2;
        xv2 = x0*d2mb*nb + xv2;
        yv2 = y0*d2mb*nb + yv2;
        zv2 = z0*d2mb*nb + zv2;
    }

}

E = (double)xu1*xu1+yu1*yu1+zu1*zu1;
F = (double) xu1*xv1+yu1*yv1+zu1*zv1;
G = (double) xv1*xv1+yv1*yv1+zv1*zv1;
L = (double) xu1*yv1*zu2+xv1*yu2*zu1+xu2*zv1*yu1-xu2*yv1*zu1-
yu2*zv1*xu1-zu2*yu1*xv1;
M = (double) xu1*yv1*zuv+xv1*yuv*zu1+xuv*zv1*yu1-xuv*yv1*zu1-
yuv*zv1*xu1-zuv*yu1*xv1;
N = (double) xu1*yv1*zv2+xv1*yv2*zu1+xv2*zv1*yu1-xv2*yv1*zu1-
yv2*zv1*xu1-zv2*yu1*xv1;
fb = E-G-F*F;
ft = L*N-M*M;
fth = E*N+G*L-2*F*M;
if ( fabs(fb)<=0.00000000000001 &&fabs(ft)<= 0.00000000000001)
kt =0.0;

else if ( fabs(fb)<=0.00000000000001) {
    kt=9999.99;
    printf(" fb=0 \n");
}
else kt=ft/(fb*fb);
curvat[t1]=(float) kt;
sum += kt;
sum2 +=kt*kt;
if ( kt < min) min = kt;
if ( kt > max) max = kt;
if ( fabs(fb)<=0.0000000001 && fabs(fth)<=0.0000000001) h = 0.0;
else if (fabs(fb)<=0.0000000001 ) {
    h=9999.99;
    printf(" fb=0 \n");
}

```

```

        else h=fth/(2*fb*sqrt(fb));
        if ( h > maxh) maxh=h;
        if ( h < minh ) minh=h;
    sumh+=h;
    sumh2+=h*h;
    if ( kt >0.0001){
surface[t1]=250;
sumell+=kt;
sumell2+=kt*kt;
if(kt>maxell) maxell=kt;
if(kt<minell) minell=kt;
        ell++;
    }
    if ( kt<-0.0001) {
surface[t1]=90;
sumhyp+=kt;
sumhyp2+=kt*kt;
if(kt>maxhyp) maxhyp=kt;
if(kt<minhyp) minhyp=kt;
        hyp++;
    }
    if ( fabs(kt) <= 0.0001 && fabs(h)>0.006 ){
        surface[t1]=200;
sumpar+=h;
sumpar2+=h*h;
if(h>maxpar) maxpar=h;
if(h<minpar) minpar=h;
        par++;
    }
    if ( fabs(kt) <= 0.0001 && fabs(h) <= 0.006){
        surface[t1]=150;
sumpla+=h;
sumpla2+=h*h;
if(h>maxpla) maxpla=h;
if(h<minpla) minpla=h;
        pla++;
    }
}

write(out,curvat,t1*sizeof(float));
write(out2,surface,t1*sizeof(char));
printf("%c",1);
}

mean = sum/(double) (t1*t2);

```

```

    meanh = sumh/((double) (t1*t2));
    sdv = (sum2-((float)t1*t2)*mean*mean)/((float) t1*t2);
    sdv = sqrt(sdv);
    sdvh = (sumh2-((float)t1*t2)*meanh*meanh)/((float) t1*t2);
    sdvh = sqrt(sdvh);
    ellf =(float) ell/(t1*t2);
    meanell=0; sdvell=0;
    if(ell!=0){
meanell = sumell/((float) ell);
        sdvell = (sumell2-((float)ell)*meanell*meanell)/((float) ell);
        sdvell= sqrt(sdvell);
    }
    parf = (float) par/(t1*t2);
    meanpar=0; sdvpar=0;
    if(par!=0){
        meanpar = sumpar/((float) par);
        sdvpar = (sumpar2-((float)par)*meanpar*meanpar)/((float) par);
        sdvpar= sqrt(sdvpar);
    }
    plaf=(float) pla/(t1*t2);
    meanpla=0; sdvpla=0;
    if(pla!=0){
        meanpla = sumpla/((float) pla);
        sdvpla = (sumpla2-((float)pla)*meanpla*meanpla)/((float) pla);
        sdvpla= sqrt(sdvpla);
    }
    hypf = (float) hyp/(t1*t2);
    meanhyp=0; sdvhyp=0;
    if(hyp!=0){
        meanhyp = sumhyp/((float) hyp);
        sdvhyp = (sumhyp2-((float)hyp)*meanhyp*meanhyp)/((float) hyp);
        sdvhyp= sqrt(sdvhyp);
    }
}

/*
fprintf(fp,"          min          max          mean
stdv      perct      \n");
fprintf(fp," =====
===== \n");
fprintf(fp,"Globe K  %.10f  %.10f  %.10f  %.10f\n",min,max,mean,sdv);
fprintf(fp,"Globe H  %.10f  %.10f  %.10f  %.10f\n",minh,maxh,meanh,sdvh);
fprintf(fp,"Ellip K  %.10f  %.10f  %.10f  %.10f
%f\n",minell,maxell,meanell,sdvell,ellf);
fprintf(fp,"Parab H  %.10f  %.10f  %.10f  %.10f
%f\n",minpar,maxpar,meanpar,sdvpar,parf);
fprintf(fp,"Planar H %.10f  %.10f  %.10f  %.10f

```

```

%f\n",minpla,maxpla,meanpla,svdpla,plaf);
fprintf(fp,"Hyper K %.10f %.10f %.10f %.10f\n",minhyp,maxhyp,meanhyp,svdhyp,hypf);
*/
    for(i=0;i<28;i++) rout[2*i]=53.0+(double)i;
    rout[1]=(double) min; /* Min Globe K */
    rout[3]=(double) max;
    rout[5]=(double) mean;
    rout[7]=(double) sdv;
    rout[9]=(double) minh; /* Min Glob H */
    rout[11]=(double) maxh;
    rout[13]=(double) meanh;
    rout[15]=(double) sdvh;
    rout[17]=(double) minell; /* Ellip K */
    rout[19]=(double)maxell;
    rout[21]=(double) meanell;
    rout[23]=(double) sdvell;
    rout[25]=(double) ellf;
    rout[27]=(double) minpar; /* Parab H */
    rout[29]=(double) maxpar;
    rout[31]=(double) meanpar;
    rout[33]=(double) sdvpar;;
    rout[35]=(double)parf;
    rout[37]=(double) minpla ; /* Planar H */
    rout[39]=(double) maxpla;
    rout[41]=(double) meanpla;
    rout[43]=(double) sdvpla;
    rout[45]=(double) plaf;
    rout[47]=(double) minhyp; /* Hyper K */
    rout[49]=(double) maxhyp;
    rout[51]=(double) meanhyp;
    rout[53]=(double) sdvhyp;
    rout[55]=(double) hypf;
    write(out3,rout,56*sizeof(double));
    close(in);
    close(out);
    close(out2);
    close(out3);
    exit(0);
}

```

```

void knot(int n, int c, int *tr )
{
    int i;

```

```

for(i=c;i<=n+c;i++){
if(i<=n) tr[i]=i-c+1;
else tr[i]=n-c+2;
}
}

```

```

void dbasis(int n,int c,float u,int *tr, float *bas,float *d1bas,float *d2bas;
{

```

```

int i,j,w;
float d;

```

```

for(i=0;i<n+c;i++){ /* the first level of zero derivative */
bas[i]=0.0;
if( u>=(float) tr[i] && u< (float) tr[i+1])
bas[i]=1.0;
}

```

```

if( u==(float) tr[n+c]) bas[n+c]=1.0;
for(i=0;i<n+c-1;i++){ /* the first level of the first derivative */
if((tr[i+2]-tr[i])!=0)
d=bas[i]/((float)(tr[i+2]-tr[i]));
else d=0.0;
if((tr[i+2]-tr[i+1])!=0)
d=-bas[i+1]/((float)(tr[i+2]-tr[i+1]));
d1bas[i]=d;
}

```

```

for(i=0;i<n+c-2;i++){ /* the first leve of the second derivative */
if((tr[i+3]-tr[i])!=0)
d=2.0*d1bas[i]/((float)(tr[i+3]-tr[i]));
else d=0.0;
if((tr[i+3]-tr[i+1])!=0)
d=-2.0*d1bas[i+1]/((float)(tr[i+3]-tr[i+1]));
d2bas[i]=d;
}

```

```

for(i=0;i<n+c-1;i++){ /* the second level of the zero derivative */
if(bas[i]!=0)
d=(u-tr[i])*bas[i]/((float)(tr[i+2]-tr[i]));
else d=0.0;
if(bas[i+1]!=0)
d+=(tr[i+2]-u)*bas[i+1]/((float)(tr[i+2]-tr[i+1]));
bas[i]=d;
}

```

```

for(i=0;i<n+c-2;i++){ /* the second level of the the derivative */
if((tr[i+3]-tr[i])!=0)
d=(bas[i]+(u-tr[i])*d1bas[i])/((float)(tr[i+3]-tr[i]));

```