Segmentation of Brain X-Ray CT Images using Seeded Region Growing

Alan Bub

Submitted to the Faculty of Engineering, University of Cape Town, in partial fulfilment of the requirements for the degree Master of Science in Engineering.
Cape Town,
March 1996

Declaration

I declare that this dissertation is my own work. It is being submitted for the degree of Master of Science in Engineering at the University of Cape Town. It has not been submitted before for any degree or examination at this or any other university.

A.M. Bub
(Signature of Candidate)
Signed by candidate
signature removed

Acknowledgements

I would like to thank the following people and institutions for their contribution towards this thesis.

Gerhard de Jager for his guidance and help.

Hannelie for expertly driving the scanner.

Magdel Shakelton for providing the images.

The Foundation for Research and Development for their financial assistance.

Everybody in the UCT image processing laboratory for the stimulating and interesting working environment.

Abstract

Three problems are addressed in this dissertation. They are intracranial volume extraction, noise suppression and automated segmentation of X-Ray Computerized Tomography (CT) images. The segmentation scheme is based on a Seeded Region Growing algorithm. The intracranial volume extraction is based on image symmetry and the noise suppression filter is based on the Gaussian nature of the tissue distribution. Both are essential in achieving good segmentation results.

Simulated phantoms and real medical images were used in testing and development of the algorithms. The testing was done over a wide range of noise values, object sizes and mean object grey levels. All the methods were first implemented in two- and then three-dimensions. The 3-D implementation also included an investigation into volume formation and the advantages of 3-D processing.

The results of the intracranial extraction showed that 9% *of the data in the relevant grey level range consisted of unwanted scalp (The scalp is spatially not part of the intracranial volume, but has the same grey level values). This justified the extraction the intracranial volume for further processing. For phantom objects greater than* $741.51mm^3$ *(voxel resolution* $0.48mm \times 0.48mm \times 2mm$*) and having a mean grey level distance of* 10 *from any other object, a maximum segmentation volume error of* 15% *was achieved.*

Contents

List of Figures

LIST OF FIGURES

LIST OF FIGURES

List of Tables

Section LIST OF TABLES

Chapter 1

Introduction

1.1 Background and Motivation for Research

X-Ray Computer Tomography (CT) has developed over the years as an essential medical tool that is used in most modern hospitals. The term *Tomography* refers to the cross-sectional imaging of an object from either transmission or reflection data collected by illuminating the object from different positions (Kak[1]). The impact of this technique in diagnostic medicine has been nothing short of revolutionary, since it has enabled doctors to view internal organs with high precision and safety to the patient. More recently, CT techniques are being used for operational planning and in some cases during surgery.

The use of Tomographic images for planning (finding the position of a tumour so that its position is known before surgery), diagnosis (determining if a lesion or cyst exists) or volume determination (finding the size of anatomical objects) all require image analysis. Image analysis is the process of deriving a description of an object from its image. This is of great theoretical and practical importance in medical image analysis. The problem with medical images is that the isolated regions of interest (ROI's) (e.g, tumours or cysts) cannot be directly obtained from X-Ray CT Images. The reason for this is that entire subject is imaged and thus the images formed contain all the ROI's. Therefore, delineating the the images to create images of ROI's, is always needed. Image segmentation is the partioning of an image into these ROI's (e.g, the tumours or cysts in medical images), based upon certain properties within those images.

Medical Image processing and specifically feature extraction (segmentation) from medical images, often requires the the outlining of predefined anatomic regions on the images. A common procedure is to manualy trace the features. This, however, has a number of drawbacks which will be described later. The alternative is to use computers to automatically or interactively assist with the segmentation. The focus of this dissertation is the development

of an automated method that uses X-Ray CT Images to find the volume and thus the outline of anatomical regions in the brain.

The Medical Physics Department of the University of Cape Town presently treats certain types of tumours, cysts or lesions using radioactive isotopes. The isotope is injected through tubes into the harmful tissue. The correct amount of isotope required is essential to both the planning and success of the treatment. It was found that the quantity of isotope used is directly dependent on the volume and type of abnormal tissue. The specific isotope used is not manufactured in the Republic of South Africa, therefore if too little is ordered there can be a serious delay in treatment. For these reasons, the planning of treatment of such abnormal growths depends greatly on the accuracy of the estimated volume. The volume is presently obtained via two distinct methods. The first involves extracting part of the abnormal tissue. Although accurate, it is generally undesired as an operation is required. The second is to obtain the volume from a set of X-Ray CT images. This is more desirable as:

- No surgery is required to find abnormal tissue volume.
- The exact position of of the abnormality can be found.
- Long term monitoring of the patient can be more easily and effectively done.
- It is much faster using a non-invasive method.

There are at present two methods implemented that use X-Ray CT in calculating the abnormal tissue volume. The first method is to scan the patient at various intervals and work on each scanned slice individually. The scanner used has a minimum slice depth of 2 mm, thus scanning a depth of 100 mm results in 50 scans. The operator then searches through the scans until a scan with the desired feature or abnormality is found. On the present system, the abnormality is outlined using a *track-ball*. The area of the enclosed curve is then automatically calculated. This area then needs to manually multiplied by the thickness of the slice to obtain a volume. The procedure is repeated for all the slices containing the relevant structure and a total volume obtained by summing the individual slice volumes.

There are a number of drawbacks with this method. The first is the speed of the operation. In order to do extract the object, the operator first needs to find it in the set of slices and then outline it. Both these tasks are time-consuming. The second is the accuracy of the tracking. From a physical standpoint, the track-ball is difficult to use and has a problem of occasionally *jumping* a number of pixels. The reason for this is unknown, but is not adjustable on the machine itself. From a software standpoint, the software is far more suitable for diagnostic purposes than planning (volume determination). An example of this, is that it only allows for tracking of objects with Jordan curves. A Jordan, or simple curve, is a closed curve that

does not cross itself (Parker[2]). In order to find the area[1]. of a non-Jordan curve, ie: a disc with a hole, the operator first required to outline both the inner and outer curves, obtain their areas and then subtract them to get the correct value. This is a time consuming process that is both inefficient and likely to lead to a loss of accuracy. It should also be noted that the software was implemented on the same machine used for the image reconstruction . This means that while scanning is being done, no images can be processed.

The second method to overcome the above problems, has been to transfer the set of images to a different machine that has more accurate extraction software. The software still requires the operator to manually outline the object, but alleviates the track-ball and closed curve problems described above. The problem with this solution is the method of transfer of the data. The images are first recorded onto 8mm tape, then moved to Kennedy Tape and finally to the machine with the better software. Both the tape media are slow and inefficient, thus increasing the overall time to complete the process. It also increases the amount of storage required as duplication of the data is required.

Although the alternate software does provide for a more accurate extraction, it still has the general problems associated with a human operator. The first is that the human approach of tracking features is both time consuming and tedious. This is especially true for volume estimation, which requires a three-dimensional scan of the subject to be processed. The biggest problem is, however, the effect of human subjectivity on results, which severely limits the ability to track objects over a long period of time. It also makes assigning a confidence value to the results extremely difficult. Implementing an automated or semi-automated method should solve the majority of the problems described above. According to Pfefferbaum [3], who measured the Cerebrospinal Fluid (CSF) with respect to aging, the use of an automated system enables researchers to have the same objective view of results over a period of time.

In summary, the research that led to this dissertation was motivated by: (a) The proven usefulness of Radiological Images (X-Ray Computed Tomography) for measuring neurological features. (b) The opportunity to work on and implement a solution that can be clinically evaluated, if it proves to be feasible, (c) a chance to improve the speed and accuracy over an existing system, (d) the cost advantage of implementing new software on an existing scanner, and (e) the requirement of an automatic or semi-automatic method to calculate volumes of intracranial structures.

[1]The volume would be obtained by multiplying the area by the thickness of the X-Ray CT slice

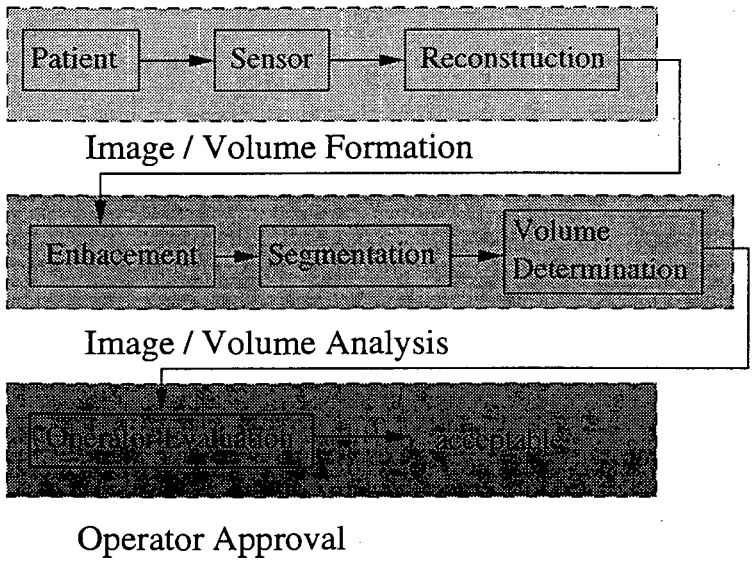


Figure 1.1: Block Diagram of a CT Volume determination System

1.2 Definition of the Problem

The problems addressed in this dissertation are defined in this section. System requirements, study objectives, study scope and study limitations are all discussed.

Figure 1.1 depicts a block diagram of the typical components in an Automated Volume Determination system using X-Ray Computed Tomography. The system has been divided into three categories, Image or Volume formation, Image or Volume analysis and Image or Volume Evaluation. Each component is further broken down into smaller modules. Ideally, each of these modules are self-contained and can be replaced without affecting the entire system.

The Image formation component is the first stage in the system, and is the area where the least amount of adjustment is possible. This involves the actual mechanical scanning and reconstruction algorithm. For this dissertation, the CT scanner and reconstruction software is assumed to be non-adjustable and improvements at this stage were not attempted. There are however many ways to form a volume from a set of slices.

The Image/Volume analysis component is the second stage in the system. This is where the majority of the work done in the dissertation is focused on. There are three modules, that are based on the basic image processing or machine vision paradigms. These are given by Jain[4] as Image enhancement, Image Analysis and Image Classification (In certain texts, the role of image enhancement is viewed as a preprocessing stage). Image classification, which in the context of this dissertation relates to identifying or labelling the segmented image, has

not been implemented. This has specifically been left to the operator. Thus, the two major problems addressed are image enhancement and image analysis. Volume formation techniques are also investigated, but form a smaller part of the overall problem.

The volume formation stage is required to fulfill the following:

- The slices must be correctly aligned with one another.
- The volume must maintain the original three dimensional structure.

The preprocessing or enhancement stage is required to fulfill the following:

- It must reduce the CT noise content.
- It must attempt to remove CT Artifacts.
- It must extract the intracranial Region of Interest.
- The above must be achieved without altering the diagnostic quality of the image.

The image analysis stage is required to fulfill the following:

- It must Segment the various tissue types (ie: Grey, White Tissue, CSF and various abnormalities).
- The volume of each segmented region must be determined.
- The automatically calculated volumes must agree closely with the input phantoms and manually obtained volumes.

1.3 Plan of Development

This section presents the plan of development for this dissertation.

Chapter 2: Chapter 2 gives a general introduction to the field of image segmentation. Further information related specifically to medical image segmentation is then given. An overview of the attempted methods used to measure the success of segmentation algorithm are also outlined. Lastly, a description of the brain structure and its visibility with respect to X-Ray Computed Tomography, is given.

Section 1.3: Plan of Development

Chapter 3: First, the relevant theory and imaging methods are presented. Chapter 3 deals with the theoretical background of Computed Tomographic Imaging. Topics include the theory of x-rays, formation of projection data, image reconstruction (the Radon transform) and an interpretation of the image values (Hounsfield Units). The theoretical analysis is very general due to the large number of techniques in use.

Chapter 4: Chapter 4 deals with the removal of noise in CT images. A theoretical background of the various noise types and artifact components are given. Methods of removing or reducing these, based on a developed noise model, are then described.

Chapter 5: Chapter 5 presents a new method for removing the intracranial region. This method is compared to previously applied methods and the differences explained. A description of the brain the axis of symmetry, and a method for finding it, are also given.

Chapter 6: Chapter 6 introduces the general Seeded Region Growing segmentation (SRG) algorithm. The front-end to the algorithm, which is the seed finding method, is then developed for two-dimensional case. The resulting volumes are then compared with traditional methods.

Chapter 7: Chapter 7 extends on the work in chapter 6, by rewriting the Seeding and Region Growing algorithm for three-dimensions. Reasons for working in 3D, for both general CT problems as well as specifically for the SRG algorithm, are given. Details concerning volume formation and phantom definitions are then covered. The results are finally compared against the two dimensional algorithm and traditional methods.

Chapter 8: Conclusions are drawn and recommendations made.

Chapter 2

Segmentation Overview and Issues.

This chapter first gives a brief overview on what image segmentation is, and it goals. Once this has been done, a number of techniques and their short-comings that have been previously tried, are given (Owing to the large number and varied types of segmentation techniques, details of their implementation will be omitted). This is followed by methods that can be used to measure the success of segmentation techniques. Finally an overview of the brain is given, as well as the visibility of the various structures with respect to X-Ray CT images.

2.1 Introduction to Segmentation and Current Techniques

Segmentation of an image entails the division or separation of the image into regions of similar attributes (Pratt [5]). A vast number of image segmentation techniques have been developed. Some of these methods are considered to be general while most are developed for a specific class of images. A question that is often posed, is what constitutes a *good* segmentation. Haralick and Shapiro[6] have tried to answer this by establishing the following qualitative guideline:

> Regions of an image segmentation should be uniform and homogeneous with respect to some characteristic such as gray tone or texture. Region interiors should be simple and without many small holes. Adjacent regions of a segmentation should have significantly different values with respect to the characteristic on which they are uniform. Boundaries of each segment should be simple, not ragged and be spatially accurate.

They go on to further say that achieving these goals is extremely difficult as most of uniform objects have holes and ragged boundaries. Also, the requirement that regions on either side

of a boundary have large differences can often cause otherwise separate regions to merge, causing the boundaries to be lost.

For the purpose of segmenting medical images, much of what has been said above applies. There is once again no fixed segmentation method or scheme. Most existing methods hope to emphasise or extract just one specific part or feature. Recently there has been an increase into the use of knowledge based segmentation methods (Lin[7], Kobashi[8]), but these have not been attempted or only used sparingly in this dissertation. Eggleston[9] found in laboratory tests that only a few of a range of segmentation methods were successful in segmenting medical images. For CT medical images, which falls under the Specular data class), he claims that either singly or a combination of region growing (Brice[10],Adams[11],Sandor[12]), Region Grouping/Splitting (Horowitz[13], Chen[14]), Edge Linking or Edge Grouping are the best to use.

Other previously used approaches include using geometric models to estimate the area or volume of a specific structure. These would typically use the objects diameter or width and relate them to geometric objects such as circles, rectangles or ellipses in 2-D and spheres, cylinders or ellipsoids in 3-D (Mahaley[15],Franke[16]. Planimetric methods methods that require an operator to trace the outline the edges of objects in each slice have been used (Mahaley[15], Breiman[17]). Standard image processing techniques such as each detection algorithms, thresholding of image grey levels and split-merge methods have also been tried (Jain[4]). Semi-automated methods that use histograms (Otsu[19],Kapur[20]) or threshold the image based on *priori* knowledge about objects in the scene (Sandor[21][12]) have also been implemented. Other segmentation schemes, such as that outlined in Lei[22], are based on statistical methods.

Although geometric methods are fast at calculating volumes, high error rates have been found. The main reason for this is the inaccuracy of the models with respect to anatomical shapes. Planimetric methods have smaller errors ranging from 5% to 15% (Breiman[17], Albright[23]), but require the operator to hand trace each slice and thus are time consuming. Hand segmentation is also subjective, which limits its use for monitoring subtle long term trends (Pfefferbaum[3]). Edge detection algorithms (Canny[18]) speed up the extraction of the object border, but are most often confused by image artifacts such as noise and complex image structures. Thresholding is subject to the image contrast to noise ratios (CNR's) and the subjectivity of the operators choice of threshold value.

Although many of the other techniques were investigated, most of the work done in this dissertation is based on the segmentation methods outlined by Eggleston[9]. As most of these methods are established and been used in the past, it was decided to investigate extending them from the present 2-D to a 3-D implementation. The ability to implement these algorithms in 3-D, adds new impetus for the lower level segmentation methods as well as to see

if any advantages over the 2-D cases can be found.

2.2 Measuring the success of a segmentation method

In general, a new or experimental process requires an ideal solution towards which to strive. The progress of the process can then be measured against this ideal solution or yardstick and a measure of the success obtained. Hussain [24] comments on this fundamental problem: 'One of the biggest hindrances to testing algorithms and how they perform is the lack of *ground-truth.* For much of the data we need to process, it is often difficult to establish the ground-truth. However, without this, we cannot verify the performance of our algorithms.'

In neurological medical imaging, where the regions being segmented are highly complex, the task of measuring the success or pe7rformance of a segmentation method is difficult. Often in this field, the evaluation is left up to the subjective view of the program author or operator. Horn [25] comments on this situation by stating 'when the output relies on human interpretation, it is more difficult to determine whether the system was successful in solving the vision problem presented.' This, coupled with the lack of standard medical test images (even more of a problem with volume data), has often resulted in an inaccurate methods of quantifying algorithm performance. Another problem that makes this more difficult is the large differences in images from various scanners.

2.2.1 Hand-Segmented comparisons

In an attempt to improve the above-mentioned situation a number of methods can be applied. The first and probably the most accurate way of achieving this, is to take a test image and have a number of specialists agree on a hand-segmented image. This could then be used as a reference or *ground-truth* image and the results compared to it. This would go a long way to solve the subjectivity problem as a number of inputs are required. It would also allow the analysis of real anatomical structures. There are, however, a few problems with this approach. The first is the logistics of getting the same specialists together for the duration of the development. The second, but much less of a problem than with a single operator, is the subjectivity of the human observers. For these reasons (the first contributing more), it was decided to test the algorithm by not only using hand segmented images, but also by adding phantom objects.

2.2.2 Phantom Comparison

The main advantage of using phantoms or artificially added objects is that the size, shape and noise levels can be predefined. This enables an accurate assessment of the algorithms performance to be made. It also allows for the testing of the algorithms sensitivity to noise, change in object size or shape. Thus the advantages of artificially adding objects are:

- Results can be compared objectively as there is a *ground-truth.*
- The robustness of the algorithm to noise can be assessed.
- The sensitivity to object size can be assessed.
- The sensitivity to object shape can be assessed.

The disadvantage of the method is that the model being used, never quite matches that of the real image. For the purpose of this study, simple structures were used to simulate phantom lesions, tumours, or other abnormal tissue. These structures consisted of ellipses (2-D) and ellipsoids (3-D). The software has been designed that multiple overlaps of these phantoms can be used in the same image. Thus, if a more complex structure is required, two ellipsoids can be merged and only the overall shell is added to the image or volume (The intersection points are treated as one object). An example of such a phantom is shown in figure 2.1 (The lesion is in the lower right hand section of the brain slice).

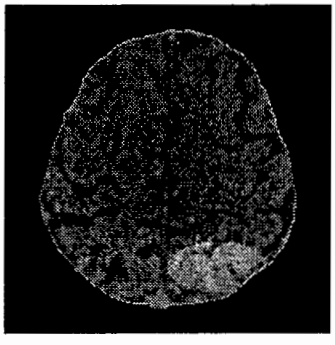


Figure 2.1: CT Image with artificially added lesion

2.3 Brief Overview of Brain Structure

In this section a description of the brain structure is given. The regions described are mainly those visible using X-Ray Tomography. They have also been restricted to the upper cranial portions or the Cerebral Hemispheres.

2.3.1 The Cerebral Hemispheres

There are four Cerebral hemispheres that make up the largest portion of the brain. These are the Frontal, Parietal, Temporal and Occipital hemispheres. An illustration of these are given in 2.2. Each has a rough hemispherical shape, and has on its surface a large number of grooves or **Sulci**. The ridges between these grooves are known as **Gyri**. If one takes a cut across the cerebral hemisphere, it can be roughly seen to have three layers. On the surface is a layer of grey matter, known as the cerebral cortex. Within this is a mass of white matter and deep inside this lies a number of smaller masses of grey matter.

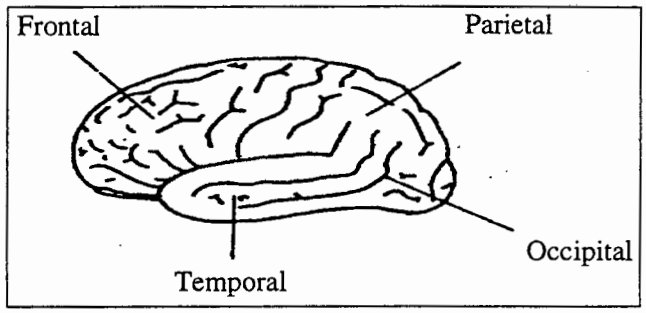


Figure 2.2: The lobes of the Cerebral Hemisphere

The white matter within the hemispheres, consists of three types of fibres. These are Commissural Fibres, Association Fibres and Projection Fibres. The Commissural Fibres run between the cerebral hemispheres The majority of these are collected together into the corpus callosum, a huge bundle containing millions of fibres. The Association fibres connect different area of the cerebral cortex within one hemisphere. Projection fibres connect the cerebral hemisphere with other parts of the nervous system. The first two are more likely to be visible using X-Ray CT (See section 2.4 for details on tissue visibility).

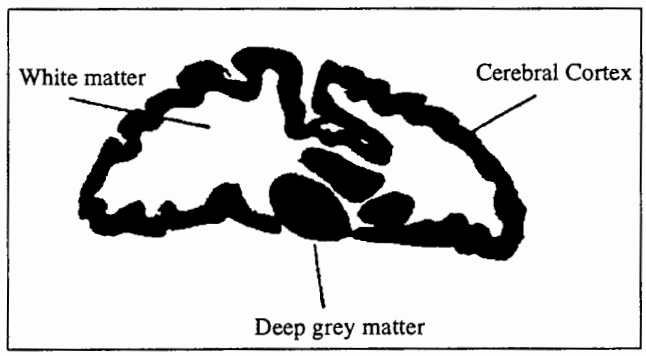


Figure 2.3: A horizontal section through a Cerebral Hemisphere

The deep grey areas in the hemisphere have a large number of components. These are

shown in figure 2.4. The **Corpus Striatum** is a large mass of grey matter with a large number of subcomponents. Deep inside the temporal lobe are the **Amygdaloid body** and **Hippocampus**. The **Diencephalon** lies toward the medial surface of the hemisphere. Its largest component is the **Thalamus**. There are three other smaller components making up the Diencephalon, these are the **hypothalamus** , **subthalamus** and **epithalamus**. The deep grey matter, as with most tissue (see section 2.4) has a low belief interval.

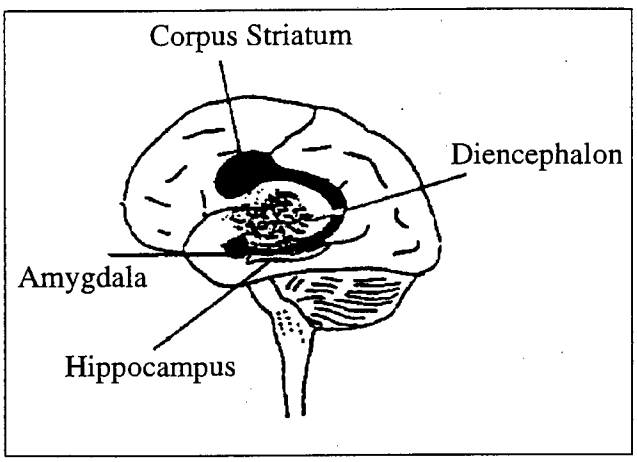


Figure 2.4: The deep grey matter of the hemisphere

2.3.2 The Ventricular System and Cerebrospinal Fluid

The Cerebrospinal Fluid (CSF)

The cerebrospinal fluid or CSF is a clear, watery liquid (Thus it is highly visibile in X-Ray CT Images). It is found in the Ventricular system of the brain, the spinal cord and sub-arachnoid space. The brain and spinal cord float in it and are thus supported with little mechanical stress. The fluid is not stagnant but flows constantly, being produced in the ventricals and absorbed into the superior saggital sinus.

The Ventricular System

The ventricular system consists of a number of cavities within the brain. They are formed from a hollow tube of ectoderm. The basic pattern of the ventricles is given in Figure 2.6. A C-shaped lateral ventricle lies within each cerebral hemisphere. These two cavities communicate via the intervacular foramina with the third ventricle. This is a narrow slit-like cavity with the diencephalon in the midline. At the posterior end of the third ventricle, the narrow cerebral aqueduct runs from the third ventricle to through the midbrain. At its lower end it expands

to form the fourth ventricle lying between the cerebellum and the brain stem. The fourth ventricle has three openings inferiorly through which the cerebrospinal fluid escapes. The ventricular system has a high degree of visibility, and therefore a more detailed description is given.

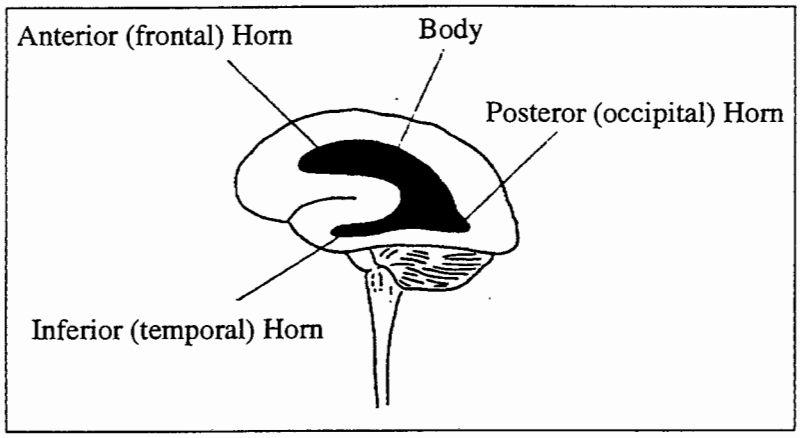


Figure 2.5: The Lateral Ventricle

The Lateral Ventricle: Figure 2.5 gives a description of the major parts of the lateral ventricle. Each Cerebral hemisphere contains a lateral ventricle which is roughly C-shaped. Within the temporal lobe, lies the **inferior** or **temporal horn** of the ventricle from where the ventricle passes backwards and upwards. Projecting into the occipital lobe is the **posterior or occipital horn**. The cavity now passes upwards and forwards beneath the corpus callosum to form the **body** of the ventricle. At its anterior end the **anterior** or **frontal horn** projects into the frontal lobe. At the junction of the body and anterior horn lies the **interventricular foramen**. It communicates medially with the third ventricle.

The third ventricle: This is a arrow, slit like cavity between the diencaphelon on either side. The pineal gland lies at its ipper, posterior extremity. Its lateral wall is formed by the thalamus and hypothalamus. The later being separated by the shallow hypothalmic sulcus. In this dissertation, the third ventricle lies at the lower extremities of the scanned slices. The scans that include this ventricle also include the eye-sockets and other facial structures. These are not part of the original segmentation requirement and are thus often not included.

Cerebral Aqueduct: This is a narrow, tube which leads from the postero-inferior end of the third ventricle to the top of the fourth ventricle. It runs through the mid brain and is surrounded by the per-aqueductal grey matter.

Section 2.3: Brief Overview of Brain Structure

The fourth Ventricle: The fourth ventricle is a tent-shaped cavity lying between the cerebellum and the brain stem. Its anterior surface is diamond shaped and forms the hollow **rhomboid fossa** which lies on the posterior surface of the pons and medulla. The peak of the tent projects backwards into the cerebellum. At its upper end the ventricle receives the **cerebral aqueduct**. It is continuous at its lower end, with a very narrow **central canal** which runs down the spinal cord. On either side of the canal are three bundles which connect the brain stem with the cerebellum. These are the superior, middle and inferior cerebellar peduncles. Between the superior cerebellar penducles is a sheet of white matter, the **superior medullary velum**. Between the inferior cerebellar penducles is the **inferior medullary velum**. Three openings are present in this layer. They are the **median aperture** on the mid-line and the **lateral apetures** on either side. These openings permit the escape of cerebrospinal fluid.

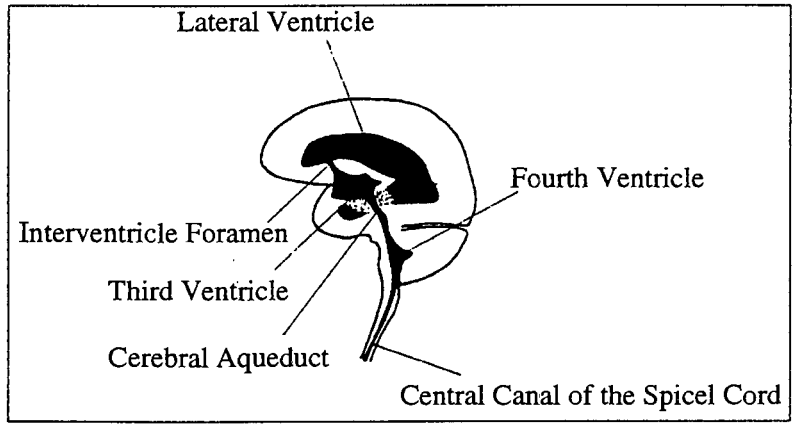


Figure 2.6: The Ventricular System and Cerebrospinal Fluid

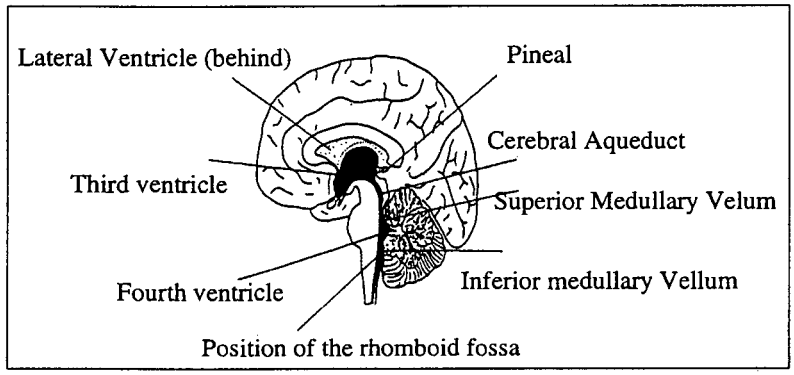


Figure 2.7: A median saggital section of the brain to show midline features of the ventricular system.

2.4 X-Ray CT tissue discrimination.

The anatomical structures of the brain results in five categories that need to be segmented: *cerebrospinal fluid, grey matter, white matter, bone* and *fat* (In practice there was found to be a sixth region, but this was seen to be the skull tissue interface). Due to the differences in anatomy visible in different modalities, each sensor has its own unique characteristics that justify its usage (Gonzalez [26]). Lin [7]) assigns an assessment to each of the five categories to be segmented and uses it to hypothesise whether a region is a specific anatomical part or not. In the case of this dissertation, the assessment is used to show the accuracy or belief in CT imaging as opposed to other modalities. It also gives an insight into the region most likely to be correctly segmented. Lin [7] has given Linguistic terms to the degree with which an anatomical structure is visible and can be discerned from the background. These are given in table 2.1. Table 2.2 illustrates the visibility in terms of the linguistic representation of the five major anatomy categories.

Linguistic Terms	*Belief Intervals*
true	[0.99 1.00]
very high	[0.85 0.99]
high	[0.65 0.85]
medium	[0.50 0.65]
low	[0.30 0.50]
very low	[0.15 0.30]
none	[0.00 0.15]

Table 2.1: Linguistic terms and their corresponding numerical intervals

Anatomy Categories	*Linguistic Terms*		
	X-ray CT	PD MRI	T_2 - weight MRI
Cerebrospinal fluid	high	medium	very high
Gray Matter	very low	very high	low
White Matter	very low	very high	medium
Fat	medium	high	low
Bone	true	none	none

Table 2.2: Linguistic scales of anatomical visibility

From the table 2.2 it should be clear that X-ray computed tomography is limited with respect to white and grey matter discrimination. This can be further seen when looking at figure 2.8, which represents a distribution of the pixel intensities in a typical CT brain image (Later real data will be used, but for this purpose an idealised representation is given). The CSF, gray

Section 2.4: X-Ray CT tissue discrimination.

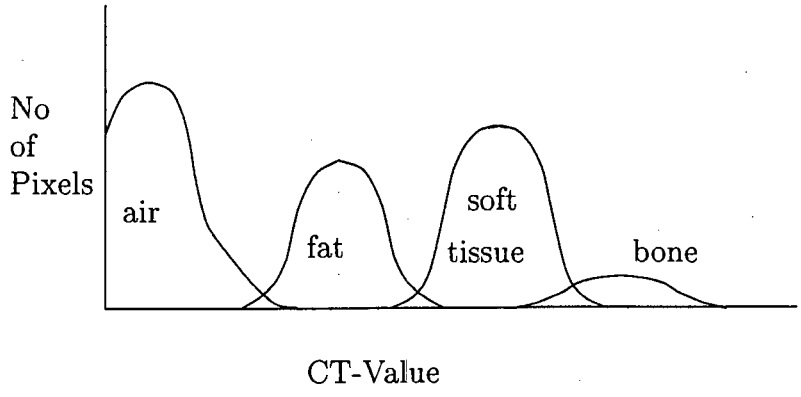


Figure 2.8: Intensity distribution of different anatomical structures in x-ray CT images

and white tissue distributions are represented as an unimodal Gaussian and only the bone and fat are clearly discernible.

The gray level differences between grey tissue, white tissue, CSF and any abnormalities like tumours, lesion and cysts are very small. The gray levels also differ depending on the patient and the scanner used. For this reason it is essential that the segmentation algorithm be as adaptive as possible.

Chapter 3

Overview of CT Imaging

In this chapter, the basic algorithm for recovering the image of the cross section of an object from projection data will be described. X-Ray CT will be used as the basis for examples and descriptions.

Although, from a mathematical standpoint, the solution of how to construct a function from its projections dates back to the paper by Radon[27], it was only when Hounsfield [28] invented of the X-ray Computed Tomographic Scanner that this become practical. His invention showed that it is possible to compute high-quality cross-sectional images with an accuracy now reaching one part in a thousand. This is in spite of the fact that the projection data does not strictly satisfy the theoretical models underlying the efficiently implementable reconstruction algorithms.

3.1 Line Integrals and Projections

A line integral represents the integral of some parameter of and object along a line. In x-ray CT the line integral is the attenuation of the x-rays as they propagate through the biological tissue. In this case, the object is modelled as a two-dimensional distribution of x-ray attenuation constants and a line integral represents the total attenuation suffered by a beam of x-rays as it travels through the object (Herman [29]).

The coordinate system defined in Fig. 3.1 is used to describe line integrals and projections. The object is represented by a two dimensional function $f(x, y)$ and each line integral by the (θ, t) parameters. The equation of line $\overline{AB}$ in Fig. 3.1 is

$$x \cos\theta + y \sin\theta = t \tag{3.1}$$

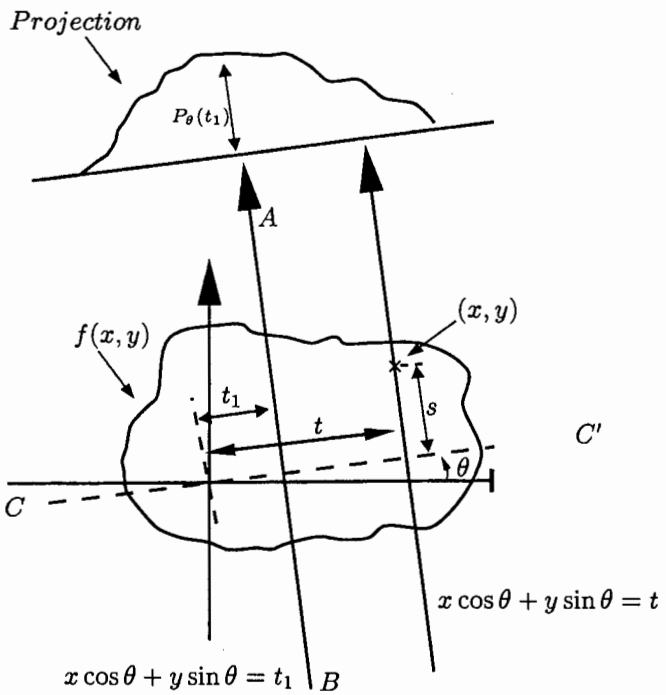


Figure 3.1: Description of Projection Formation

and using this relationship the line integral $P_\theta(t)$ is defined as

$$P_\theta(t) = \int_{(\theta,t)} f(x,y)ds \tag{3.2}$$

Using a delta function, this can be rewritten as

$$P_\theta(t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - t)dxdy \tag{3.3}$$

The function $P_\theta(t)$ is known as the Radon transform of the function $f(x,y)$. A projection is formed by combining a set of line integrals. The simplest form of projection that can be constructed is a collection of parallel ray integrals as is given by $P_\theta(t)$ for a constant θ. This is known as a parallel projection and an example is shown in Fig. 3.2. This is usually measured by moving a x-ray source and detector along parallel lines on opposite sides of an object.

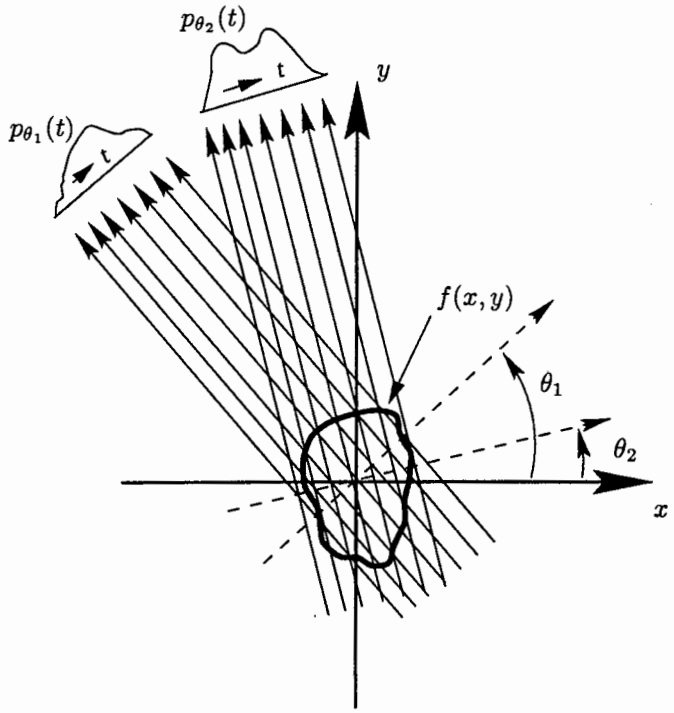


Figure 3.2: Formation of multiple Parallel Projections

3.2 Fourier Slice Theorem

The Fourier Slice Theorem is derived by taking the one-dimensional Fourier Transform of a parallel projections and noting that it is equal to a slice of the two dimensional Fourier Transform of the original object. It thus follows, that given projection data, it should be possible to estimate the object by simply performing a two-dimensional inverse Fourier transform.

The two-dimensional Fourier transform of the object function $f(x, y)$ is defined as

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\exp^{-j2\pi(ux+vy)}\,dxdy \tag{3.4}$$

and a projection at an angle, θ, $P_\theta(t)$, and its Fourier transform by

$$S_\theta(\omega) = \int_{-\infty}^{\infty} P_\theta(t)\exp^{-j2\pi\omega t}\,dt \tag{3.5}$$

The simplest way to illustrate the Fourier Slice Theorem is give an example for a projection

at angle $\theta = 0$. First, consider the Fourier Transform of the object along the line in the frequency domain given by $v = 0$. The Fourier transform integral given in 3.4 now simplifies to

$$F(u,0) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\exp^{-j2\pi(ux)}\,dxdy \tag{3.6}$$

and because the phase factor is no longer dependent on y, we can split the integral into two parts,

$$F(u,0) = \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f(x,y)dy\right]\exp^{-j2\pi(ux)}\,dx \tag{3.7}$$

From the definition of a parallel projection, it is clear that the term in brackets is the equation for a projection along lines of constant x or,

$$P_{\theta=0}(x) = \int_{-\infty}^{\infty} f(x,y)dy \tag{3.8}$$

Substituting this in Equation 3.6 the following is obtained,

$$F(u,0) = \int_{-\infty}^{\infty} P_{\theta=0}(x)\exp^{-jw\pi ux}\,dx \tag{3.9}$$

The right-hand side of Equation 3.9 represents the one-dimensional Fourier transform of the projection $P_{\theta=0}$; thus there is the following relationship between the vertical projection and the 2-D transform of the object function:

$$F(u,0) = S_{\theta=0}(u) \tag{3.10}$$

Equation 3.10 is the simplest form of the Fourier Slice Theorem. It is clear that this result is independent of the orientation between the object and the coordinate system. An example of this is shown in Fig 3.3. If the (t,s) coordinate system is rotated by the angle θ, the Fourier transform of the projection defined in Equation 3.8 is equal to the two-dimensional Fourier transform of the object along the line rotated by θ. This leads to the Fourier Slice Theorem

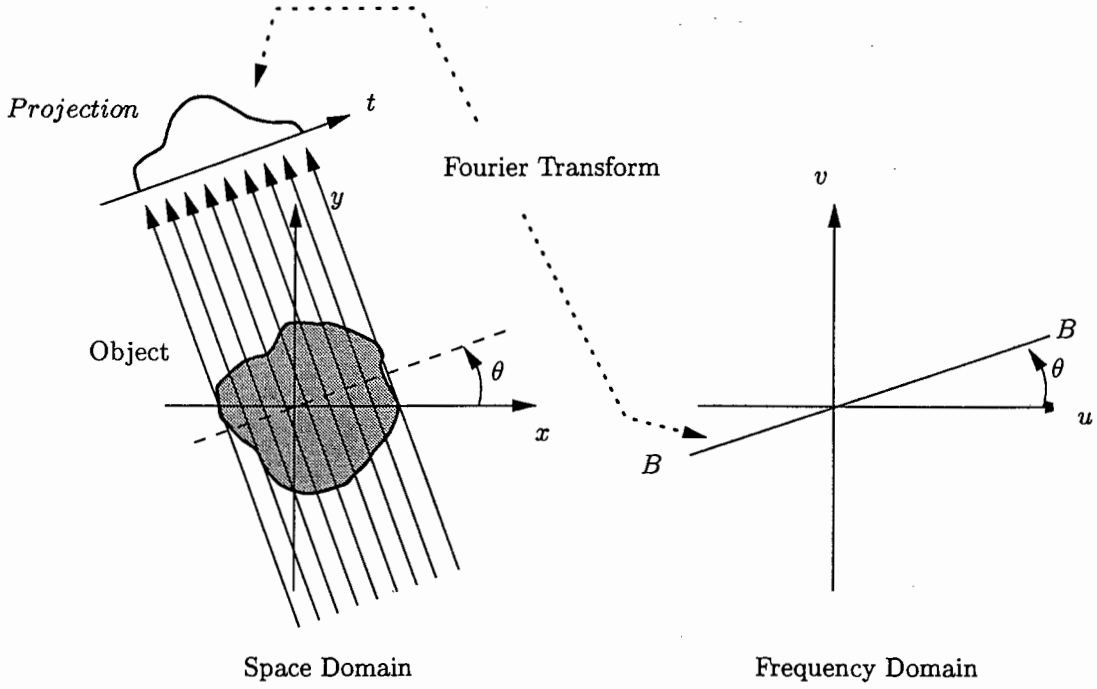


Figure 3.3: Fourier Slice theorem

which is states as Kak[30].

> The Fourier transform of a parallel projection if an image $f(x, y)$ taken at angle θ gives a slice of the two-dimensional transform, $F(u, v)$, sub-tending an angle θ with the u-axis. In other words, the Fourier transform of $P_\theta(t)$ gives the values of $F(u, v)$ along line $\overline{BB}$ in Fig. 3.3.

The Fourier slice theorem relates the Fourier transform of a projection to the Fourier transform of the object along a single radial line. Thus given the Fourier Transform of a projection at enough angles, the projections can be assembled into a complete estimate of the two-dimensional Fourier transform of the object. This can then be inverted to arrive at an estimate of the object. This provides a conceptual model of Computer Tomography. In practice, implementations usually require a different approach. However, methods are all based on the Fourier Slice Theorem and thus will not be described here. An in depth review of the Fourier Slice Theorem and some of the different reconstruction algorithms are given by Kak[1], Roberts[31] and Herman[32].

3.3 Measurement of Projection Data and the Hounsfield Unit

In this section, a brief outline of the nature of Projection data and the derivation of Hounsfield Unit for x-ray Computer Tomography, is given For a more detailed description on the subject the reader can refer to Brooks[33], Bates[34] or Kak[1].

The Fourier Slice theorem described in Section 3.1, and thus the tomographic reconstruction algorithms, are based on projection. In x-ray CT the measured data (ie: an approximation to the line integral shown in Figure 3.1 and formulated in Equation 3.3), represents the attenuation of x-rays through an object. The attenuation of an x-ray beam is dependent on the energy of each photon and since the x-rays used in imaging contain a range of energies, the total attenuation is a more complicated sum of the attenuation at each point along the line.

The hounsfield unit is used to measure the attenuation of an X-Ray through the body. In terms of the reconstructed image, they unit can be considered to be the value of the grey level pixels. To simplify the derivation of the Hounsfield unit, the case for the monochromatic (single energy) x-ray projection is first investigated. If N_d is the number of photons exiting and N_{in} is the number of photons entering an object in a given time, the relationship between the two is,

$$N_d = N_{in} \exp\left[-\int_{ray} \mu(x,y)ds\right] \tag{3.11}$$

or equivalently,

$$-\int_{ray} \mu(x,y)ds = \ln \frac{N_{in}}{N_d} \tag{3.12}$$

where ds is an element of length and where the integration is carried out along line $\overline{AB}$ in Figure 3.1 (This also assumed the beam is sufficiently small). The left hand side constitutes a ray integral for a projection. Thus measurements like $\ln(\frac{N_{in}}{N_d})$ taken for different rays at different angles may be used to generate projection data $\mu(x,y)$. This result is only valid for x-ray beams that consist of mono-energetic photons. In practice, the beam is not mono-energetic and Equation 3.11 does not hold, and must be replaced by

$$N_d = \int S_{in}(E) \exp\left[-\int \mu(x,y,E)ds\right] dE \tag{3.13}$$

where $S_{in}(E)$ is the energy spectral density of the incident photons. $S_{in}(E)dE$ is the total number if incident photons in the energy range E and $E + dE$. This equation also includes the fact the linear attenuation coefficient μ, at a point (x, y), is a function of energy. The measure energy spectrum of the exiting photons is given by

$$S_{exit}(E) = S_{in}(E) \exp\left[-\int \mu(x, y, E)ds\right] dE \tag{3.14}$$

Given the fact that the x-ray sources in CT scanning are polychromatic (have a number of energy levels) and that the attenuation coefficient is energy dependent, what parameter does the x-ray CT scanner reconstruct? To answer this question, McCullough [35] has introduced the notion of *effective energy of a CT scanner*. It is described as that monochromatic energy at which a given material will exhibit the same attenuation coefficient as measure by the scanner. It follows from further work by McCullough [36] that it is a good assumption that the measured coefficient $\mu_{measured}(x, y)$ is related to the actual coefficient $\mu(x, y, E)$ by

$$\mu_{measured} = \frac{\int \mu(E)S_{exit}(E)dE}{\int S_{exit}(E)dE} \tag{3.15}$$

This expression applies only when the output of the detectors is proportional to the total number of photons incident on them. Similar expressions can be derived for detectors that measure photon energy or total energy deposition per unit mass. From equation 3.15 one can say that the CT scanner calculates the linear attenuation coefficient of tissue at some effective energy. The numbers that are actually put out by the computer attached to the scanner are integers. These integers have been give the name Hounsfield units and are denoted by NU. The relationship between the linear attenuation coefficient and the corresponding Hounsfield unit is

$$HU = \frac{\mu - \mu_{water}}{\mu_{water}} 1000 \tag{3.16}$$

where μ_{water} is the attenuation coefficient of water and the values of both μ and μ_{water} are taken at the effective energy of the scanner. The value $HU = 0$ corresponds to water and the value $H = -1000$ to air ($\mu = 0$).

Section 3.3: Measurement of Projection Data and the Hounsfield Unit

Chapter 4

Intracranial Volume Extraction

The tomographic images that are being dealt with in this thesis are axial brain slices. These, like all tomographic images, consist of anatomical structures in a background. In general, the background is considered to be any area of the image that is not part of the subject being analysed. The background does not usually contain any relevant information, it complicates the image analysis task and increases the processing time required (Raff [37],Windham[38]). For these reasons, it was beneficial to remove the background before any analysis or processing begins. In the case of intracranial brain images, the skull, eyes, scalp and other structures that lie outside the intracranial cavity were not of interest.

The chapter first evaluates whether the removal of the intracranial volume was required. This was done by finding the percentage volume of skull and scalp in the X-Ray CT images. Two methods for removing the intracranial methods were then attempted. The first consisted of modifying existing thresholding methods with adaptive thresholding ones. and the second, implementing a symmetry based extraction method. These methods, which extracted the intracranial volume from the skull, scalp and background, were the first steps in analysing or segmenting the brain structure in the image.

4.1 Percentage Volume of Skull and Scalp

Although the removal of the background should be beneficial, it was necessary to estimate the quantitative effects it has on image analysis. The reason for doing this was to ensure that removal of the background was indeed beneficial, and not just a waste of processing time. To do this, an estimate of what percentage of the data consisted of skull and, more importantly, scalp was needed. The scalp was emphasised as they have similar CT grey scale values to the intracranial volume and would effect any algorithm that is based on the image grey levels or their distribution. Background structures such as equipment used for positioning the head

Section 4.1: Percentage Volume of Skull and Scalp

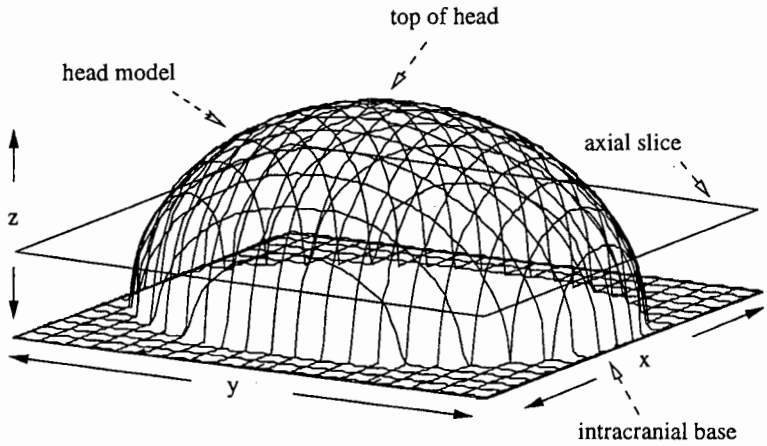


Figure 4.1: Elliptical hemisphere used in modelling head

are not a problem, as they have CT values far from the relevant data, and are easily removed. A reasonable starting point was to model the head images as a series of ellipses. These ellipses are then joined, forming an ellipsoid that was used as the final model. The ellipsoid in Figure 4.1 gives an example of such a model. In order to achieve a sufficiently useful model, the ellipsoid's dimensions need to approximate that of the human head and skull. These dimensions were taken from the Standard Measurements Texts[1] such as Newton[39] and then related to the ellipsoid model. A slice through these hemisphere is given in Figure 4.2.

The following dimensions based on Newton[39] were used: (1) The height to width ratio for the ellipsoid, in the x-y plane, was chosen at 12:10 ($x_1 : y_1$ or $x_2 : y_2$). In reality, the axial height to width ratio of the head varies from the base of the intracranial volume (just above the eye socket) to the top of the scalp. This variation is between 14:10 and 11:10. Thus, the head becomes slightly more circular higher in the axial plane. For this model, the averaged height to width ratio was used. (2) The z-depth was chosen to be 80% of the width of the skull in the x-y plane. Despite the rough measurements made in defining the model, it provided a starting framework for analysing the effect that the skull and scalp have on image statistics. The model could have been made more complex, but for the purposes of this dissertation, it was deemed unnecessary.

To determine the percentage skull and scalp the difference between **two** hemispheres needed to be found. The first hemisphere represented the skull/intracranial boundary and the second the scalp/air boundary. By finding the difference between the two hemispheres , an estimation of the amount of scalp and skull included in the analysis can be made.[2] To find out how much larger the second hemisphere should be made, the distance from the inside of the skull to the

[1]Details concerning skull and head measurements can be found in most head or skull atlasas

[2]Like the head-rests, it was found that skull's CT values were outside of the data that is to be segmented, thus the focus was mainly on the percentage scalp present.

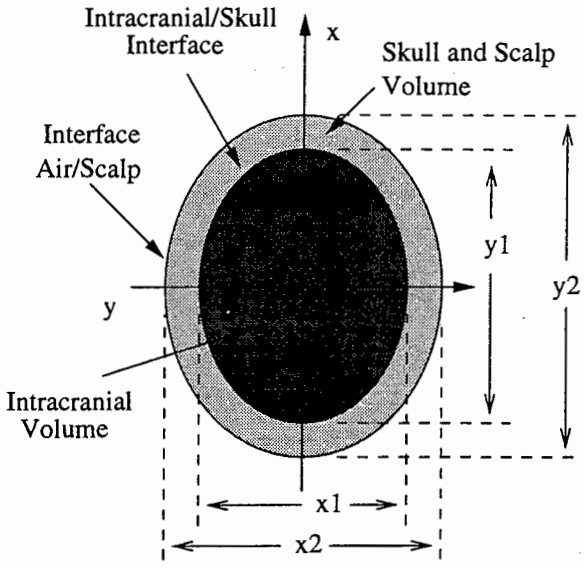


Figure 4.2: Elliptical hemisphere used in modelling head

scalp was measured ($x_2 - x_1$ and $y_2 - y_1$). This was done on the major and minor axes of the axial CT images and the result is given as a percentage increase in diameter ($\frac{y_1}{y_2}$ or$\frac{x_1}{x_2}$). The increase was found to be approximately 15%. This was first done experimentally using a number of images and then confirmed by the Standard Human Measurements text.

With the knowledge that there is a 15% increase in the diameter from the first to the second hemisphere, the size of the second ellipse can be determined. If one normalises the first hemisphere (representing the intracranial volume) x-axis to 1, the second hemisphere (representing the head including skull and scalp) will be 15% larger at 1.15. The ellipsoid's y-axis then becomes 1.38 and the z-axis goes from 0.8 to 0.92. Applying this to the modelled ellipsoid results in the skull and scalp being approximately 23.3% of the total cranial volume (This result was verified by doing a pixel count of the intracranial volume compared to the skull and scalp).

Interface	x	y	z
skull/tissue	1	1.2	0.8
scalp/air	1.12	1.38	0.82

Table 4.1: The size ratio in the x,y and z axis for the ellipsoid model. The values are normalised with the respect to the x width.

The standard form for an ellipsoid, Swokowski [40], was used to model the head and this is given by:

Section 4.1: Percentage Volume of Skull and Scalp

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \tag{4.1}$$

This is then reformulating as a function of x, y

$$f(x,y) = \sqrt{c^2(1 - \frac{x^2}{a^2} + \frac{y^2}{b^2})} \tag{4.2}$$

$$\tag{4.3}$$

and integrated over the appropriate limits to obtain a volume V,

$$V = 4 \int_0^a \int_0^{\sqrt{b^2(1-\frac{x^2}{a^2})}} f(x,y)\, dydx \tag{4.4}$$

$$\tag{4.5}$$

The values of a,b and c determine the size, and thus volume, of the ellipsoid. They were are adjusted according to the ratios given in Table 4.1. The volume difference of the two ellipsoids is obtained and the percentage of skull and scalp is obtained according to Equation 4.6. In calculating the volume only a single quadrant is required, thus the need for a factor 4 increase as shown in Equation 4.4. Finally, the percentage volume difference V_{diff} between the head volume V_h and intracranial volume V_i is given by:

$$V_{diff} = \frac{V_h - V_i}{V_h} \tag{4.6}$$

The effects of the inclusion of the scalp on the image analysis was quantified by analysing the effect on the image pixel distribution. The distribution before and after the intracranial extraction were compared and are given in figure 4.11. The resulting difference between them was calculated at roughly 9%. This is different from the theoretical value 23.3% as the skull was not included. Thus, in total the scalp contributes roughly 9% of the total CT data between the grey levels 1000 to 1120.

4.2 Method 1: Simple Thresholding and Tracking

4.2.1 Thresholding

The first attempt at segmenting the brain from the background was to threshold the image to extract the skull and then track the internal skull boundary. This has been the dominant method used in extracting the intracranial volume and are based on the method outlined by Sandor[21]. The threshold scheme used was Simple Global Thresholding (Gonzalez[41]). Given an image f, where $f(x, y)$ is the grey level of point (x, y), a thresholded image $g(x, y)$ is defined as

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) \leq T \end{cases} \tag{4.7}$$

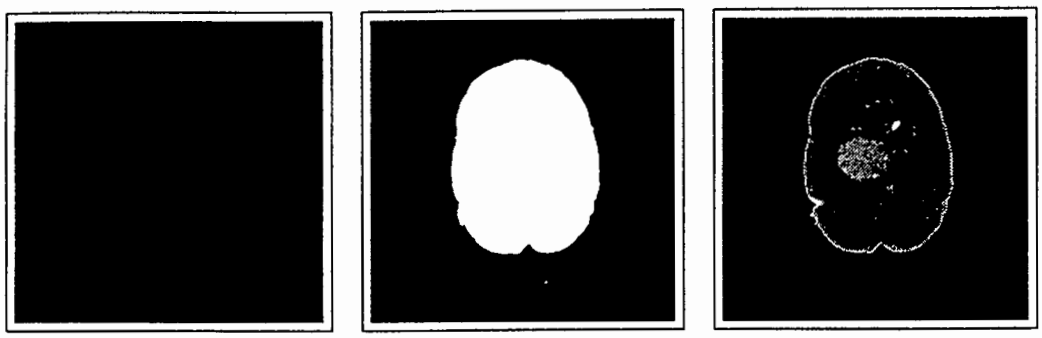
where T is a grey level value, and is referred to as the threshold value. The problem with these methods was that the threshold could not to be changed in varying circumstances. Considering this problem, a number of different **adaptive techniques** were investigated. A full description of each of the methods tried is given in Appendix B.

The best adaptive thresholding method was found to be the discriminant analysis method. Applying this method to the image in Figure 4.3(a) resulted in the segmentation given in Figure 4.3(b). As was expected, this did segment the background from the head, and it failed to segment the intracranial volume. The segmentation also included unwanted head rests and other equipment. One solution to this was to search the image in figure 4.3(b) and extract binary head image. This however had the following drawbacks:

- The skull and scalp are still included, which means that the intracranial volume is not extracted.
- Positioning objects, such as the head-rest in figure 4.3(b) are often connected to the head in the thresholded image, resulting in them being included in later processing.
- If there are many objects in the thresholded image, it may ambiguous to which one the head is. This is however less likely to occur.

The solution to this was to apply the same adaptive threshold, this time using as input the original image masked with the result of the previous thresholding (By masking, one means converting the image in 4.3(b) into a binary image and multiplying it with the original image). The masked image was adaptively thresholded at a value between the tissue and

deemed to have occurred an the process fails. The process also fails if no boundary is within acceptable limits. The acceptable limits and what is done when the process fails is given next.



(a) (b) (c)

Figure 4.4: Tracking the thresholded image and masking of the Original Image

An acceptable length was found to lie between 300 and 900 pixels depending on size and shape of the skull. If the number of points in the edge map lies below 300 pixels, the threshold used has not been low enough to extract the entire skull region. In this case the second threshold is reduced and the process repeated. The choice of reduction step is dependent on the amount of accuracy required. This has to be weighed against the processing time, as the smaller the step the more processing. A reduction of 3% was found to be acceptable, with a maximum reduction of 15% permitted before the process failed totally. If the number obtained is greater than 900, there was deemed to be a discontinuity in the bone structure. These discontinuities can be caused by various phenomena, such as dropout in the CT images or punctures in the skull used to insert tubes. They cause the outside of the skull region to be included by the tracking algorithm and therefore the larger CT values.

If the edge length falls within the acceptable limits, the interior region is then filled from the centre and used as a mask. The mask is then multiplied with original image to extract the region of interest. This is done for all the images in the scan sequence. If the edge length is outside the acceptable limits and the maximum threshold reduction is reached, the gaps in the image must be closed manually. This is done interactively by editing the original image and filling the gaps. In practice the images are processed automatically and the erroneous ones corrected afterwards.

4.2.4 Results

The results of the threshold and track method were at first very promising. For images of the type in Figure 4.3(a), an accurate extraction was obtained. An example of this is shown in

figure 4.4(c). However, there were a number of cases where the method failed. The reasons for these failures were mostly due to discontinuities in grey levels along the skull. Despite the fact that the thresholds used were adaptive, the change in intensity along the skull made the algorithm inflexible to different circumstances within an image. Many adjustments were made to the basic algorithm, such as localised adaptive thresholding and trying to identify break points and closing them. Despite these attempts the algorithm still was to unreliable for different subjects and varying conditions. It thus could not be used as an automated method. The algorithm is however very fast, and if complimented with limited user interactions to close any breaks, would work fine.

4.3 Method 2: Symmetry and Shape Analysis

4.3.1 Introduction

The major drawback of the previous method was the limited use of the inherent knowledge contained in CT brain images. Using predefined or adaptive threshold for discriminating skull from other tissue and then trying to estimate skull size, is sure to fail for a certain percentage of images. For these reasons a new approach, based on a more flexible set of rules, needed to be applied. The method must be able to cope with varying head sizes, CT scanning conditions (ie: different scanners, dropout, head rests) and any reasonable skull abnormalities. The approach taken was to use knowledge of the head shape and symmetry in order to find the skull/tissue interface, and thus extract the intracranial region.

4.3.2 Finding the approximate brain centre and Axis of Symmetry

The first task in this segmentation method, is to find the **Axis of symmetry (AOS)** of the brain CT image. The axis of symmetry is the line, running though the brain, that gives the highest match of symmetry between the right and left sides of the brain. In the intracranial CT images this line is usually defined as running from the **falx celebri** to the **superior sagittal sinus**, at the appropriate anatomical level (Raff [37]). Finding these two points and calculating the line of symmetry was not however the best approach. This is due the falx celebri or superior sagittal sinus often not being visible, and also that the same method cannot be applied to lower axial slices. Despite the method not being implemented, it should not be overlooked, as it would allow a rotated head to be more accurately and easily repositioned. The head symmetry plane could be calculated by finding the two points in two slices and then extrapolating that for the entire volume. It would also solve a later problem of finding the AOF in the lower axial slices, in the final solution.

Section 4.3: Method 2: Symmetry and Shape Analysis

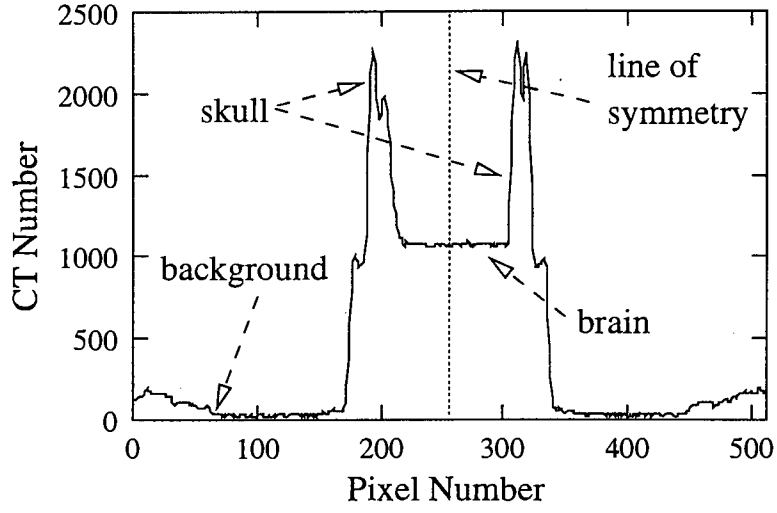


Figure 4.5: Row through typical CT scan

The method that was used was to scan the image rows from top to bottom at various intervals, and for each row calculate the point of symmetry. Figure 4.5 gives an example of a row through a typical CT slice. The two peaks represent the skull and the CT values between are the intracranial (brain) CT values. The symmetry is clearly visible in this example. In finding the point of best symmetry three assumptions regarding the patient were made:

- The patients head lay in the centre of the scanner and thus the image.
- The patients head was at a near vertical angle.
- The patient does not move while within the scanner (Generally assumed for all CT scans taken).

If these assumptions hold, it is likely that the symmetry point lies $\approx$ 50 pixels to either side of the image centre. This would enable the use of a limited search space to speed up processing. In most cases the first and third assumptions hold true, and the the 50 pixel approximation can be used. However, the second assumption, was not always valid. To rectify this problem, a method needed to be found that could find the axis of symmetry of a rotated head. The full method for a rotated head is outlined in a Appendix A (A brief overview is given later in the chapter), with the the simple case[4] given next:

The algorithm works by testing a number of possible or candidate centre points. For each point tested, the row is flipped about that point and the absolute difference between corresponding points are summed up. The point for which the summation is the minimum is taken as the

[4]The simple case has all the assumptions holding.

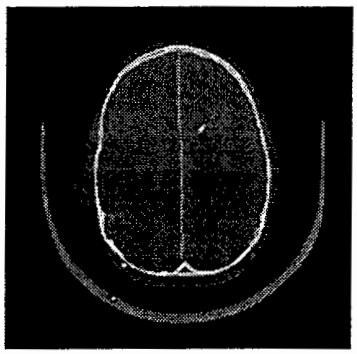


Figure 4.6: Calculated Line of Symmetry for Vertical Orientation

point of symmetry. Due to the finite size of the image, the range over which the summation is performed varies according to the position of candidate point being evaluated. This is better illustrated by equation 4.10, where $SymCentre(x_c, y)$ is the result of the summation. Using this definition, the algorithm can now be more clearly understood through the following: Letting x_c be the range of possible symmetry points,

$$\frac{1}{2}I_{width} - T \leq x_c \leq \frac{1}{2}I_{width} + T \tag{4.9}$$

where I_{width} is the image width and T is the extent from the centre in which the symmetry point can lie. The symmetry point X_{sym} can then be defined as the value of x_c for which

$$SymCentre(x_c, y) = \begin{cases} \sum_{x=0}^{x_c} |f(x,y) - f(2x_c - x, y)| & \text{if } x_c < \frac{1}{2}I_{width} \\ \sum_{x=x_c}^{I_{width}} |f(x,y) - f(2x_c - x, y)| & \text{otherwise} \end{cases} \tag{4.10}$$

is a minimum.

This procedure is repeated for a number of different rows and the symmetry value for each is stored. The resulting line of symmetry in figure 4.6 shows that the method has found a line between the falx celebri and superior sagittal sinus, which is the theoretical line of symmetry.

Another way of viewing this is to find the degree the head is rotated and then apply the inverse rotation. This has the effect of aligning the head vertically in the image. This was not done as rotating the image effects the positional accuracy of the pixels.

Once the line of symmetry has been found, the centre point of the brain image needed to be estimated. The centre point is not as important as the overall line of symmetry and thus did not need to be accurately found. This is due to the algorithm for segmenting the

intracranial region being designed such that large deviations in the position of the centre point could be dealt with. The search space for the centre position is restricted to values on the line of symmetry. The easiest method is to follow the line of symmetry, checking forward and backward the amount of tissue on the line. When the amount before and after are approximately equal, this can be considered the centre point.

4.3.3 Radial Ray casting

As stated in the introduction, the method used for segmenting the intracranial region is symmetry and shape analysis. To use the symmetry information, some technique was needed that would transform the data into a form that could be analysed. An useful technique was to radially scan the image from the calculated centre point and check how close points **mirrored** with respect to the axis of symmetry are (Details of the method are given later). The radial or polar scanning of the image is also desirable as it is aligned with the way the CT image was formed.

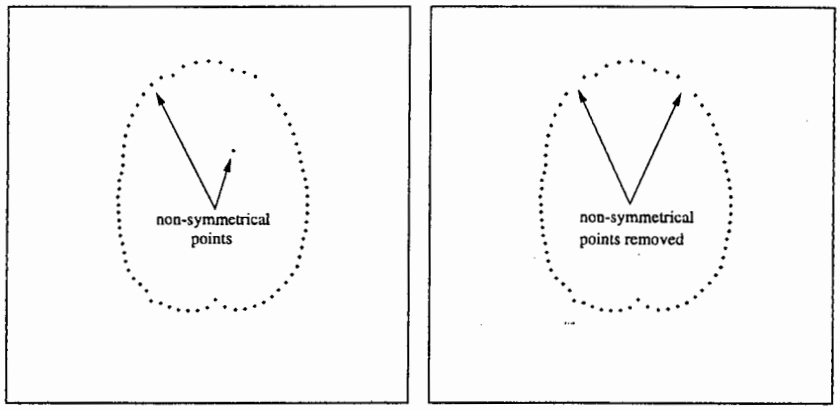
The radial method used, was to *cast* lines at different angles and mark the point on each line that had the highest CT number.[5] The angular step size must be chosen to ensure the same number of steps on either side of the line of symmetry. These steps also have to be at equal angles about the axis of symmetry (Any value that divides perfectly into 90 will work). An angle of 5 degrees was chosen to ensure that enough samples of the skull boundary were obtained. The method relies on the skull having a greatest CT value in the head region. The result of the radial scanning was a set of points which are almost symmetrical with respect to the axis of symmetry. An example of these points is given in figure 4.7(a), which was formed from the image in figure 4.6.

4.3.4 Checking Symmetry Relationship

Once the radial scan is completed, the resulting points can be measured for symmetry. For most axial cranial slices, the symmetry between the skull on either side of the calculated line of symmetry should be high. Based on this knowledge, each point should have a corresponding point in the opposite hemisphere. It should be noted that when calculating the symmetry coordinate system for a rotated head the line of symmetry calculated was used.

For each angle, the x and y points on the ray are subtracted from the corresponding point in the opposite hemisphere. If the error is greater than a certain value, both points are removed from the list of possible skull points. The threshold value is based on previously calculated average symmetry errors (from a sample of skull images) as well as the average

[5]This is similar to what was done for finding a point on the skull boundary (section 4.2.3), but the angles used cannot be randomly chosen.



(a) Values extracted from radial scanning

(b) Removal of Unsymmetrical point

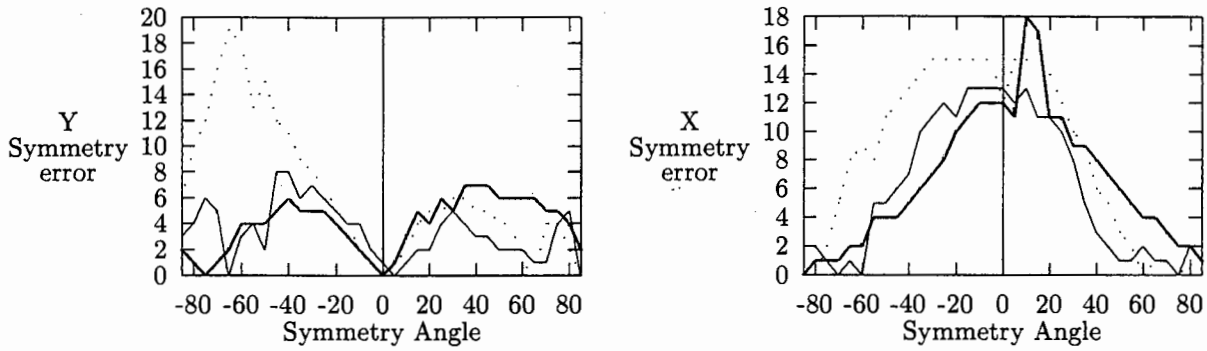
Figure 4.7: Results of radial scanning and symmetry checking

error of the image being processed. The mean or average of a number of skulls' symmetry errors was found to be 10, with a standard deviation of 2.5. To calculate the threshold value, a simple, yet effective, heuristic was applied. First the average symmetry error of the image being processed is found and then a new average is created with the previously calculated ensemble average. Assuming that most of the *correct* skull points lie within double the average value, the threshold was obtained by doubling the average. This value can be made smaller depending on the degree of accuracy required. If the threshold is made to small (ie: below the average error) and too many points are discarded, the later stage of tracking the boundary would be adversely affected. If the threshold is to large, then symmetry errors would be ignored and an incorrect boundary tracked. As the symmetry errors caused by abnormalities (skull breaks, fibroid cysts or plastic pipes) tends to be much greater than symmetry errors of the skull, the choice of threshold is less critical.

Despite this, it was found to be useful to adjust the threshold according to previously calculated skull symmetry changes. This is required, as the actual degree of symmetry of the skull changes from top to the bottom[6]. The threshold is therefore not fixed for the entire image, and an offset is added for the x and y symmetry changes, according to which angle is being checked. Figure 4.8 gives an idea of the change of symmetry in normal cranial axial images. The Symmetry angle is from $-85°$ to $+85°$, with $0°$ being the horizontal axis if the head is in the vertical position. Looking at the Y symmetry error, it is clear that the error is a minimum about the horizontal and increases as the vertical is approached. The opposite is true for the X error which has the error increasing toward the horizontal symmetry axis. These errors are consistent with the type of errors for this nature of elliptical structure. The

[6]Top to bottom refers to the skull in a single slice, not the top of the head to base of the intracranial region.

data calculated for the average x and y errors and the symmetry change are however based only on a small sample of clinical images and thus is not a definitive study. Further studies thus need to be done to verify these results.



(a) Y component of the symmetry error

(b) X component of the symmetry error

Figure 4.8: X,Y Symmetry Error

Figure 4.7(b) shows the results of checking the symmetry for the image in 4.6. In 4.7(a) there is a point in the upper right quadrant that clearly is not part of the skull. This change in symmetry is caused by the insertion of a pipe into brain to drain cystic fluid. The pipe has a higher CT value than the skull region and thus when finding the highest point on the ray, it was chosen. In figure 4.7(b) the point caused by the pipe as well as its corresponding symmetry point are removed.

4.3.5 Removing Abnormal Angular deviations

Despite checking the symmetry, not all irregularities were necessarily discarded. These irregularities are usually caused by objects with CT numbers similar to the skull, but were below the symmetry threshold. These objects lie approximately on the line of symmetry and extend equal distances into each hemisphere. Thus, they are situated in positions that did not violate the symmetry conditions.

As the symmetry check could not solve these problems, a post processing solution needed to be found. A common attribute regarding these points was that the angle they made with their neighbours[7] were more acute than points that lay on the skull. It was also found that there was a large difference between the cords connecting the points before and after. A cost function, based on these facts, was defined to establish whether the point should be kept. The formation of the cost function is as follows: Given that the angle made by the three points is

[7]Points at one angle step before and after.

$$\theta = \arccos(\frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|}) \tag{4.11}$$

where **a** and **b** are the vectors between the point being analysed and the previous and next points respectively, the final cost equation can be defined as

$$Point_{cost} = 180(1 - \frac{1}{\pi}\theta)(1 - \min(\frac{\|\mathbf{a}\|}{\|\mathbf{a}\| + \|\mathbf{c}\|}, \frac{\|\mathbf{b}\|}{\|\mathbf{b}\| + \|\mathbf{d}\|})) \tag{4.12}$$

The angle is converted to degrees and and then maximised by negating and adding 180. Equation 4.12 maximises the error based on certain requirements. In the case of an object lying on the line of symmetry, there will be a small chord and the a much longer chord. The cost function is biased towards a differential in the chord length as well as acuteness of the angle. This is the common case in most CT brain images and thus the reason for the cost function being optimised in this manner.

In analysing the cost function, the maximum and minimum values are 0 and 180. The main problem area was identified as adjacent points that form angles between 0 and 90 degrees. For this range, the cost function varies between 90 and 180. The average chord ratio is between 0.3 and 0.5 (5 degree steps), and below these are considered abnormal. The cost function for these ratios, in conjunction with angles between 0 and 90, varies between 63 and 180. The value used in practice was 90.

4.3.6 Post Processing of the remaining points

The points remaining after the above processes are not necessarily located on the inner edge of the skull. This is attributable to the maximum value being searched on the cast ray and not the skull tissue interface. The reason for not searching for the skull tissue interface was that it was deemed better to find the maximum point (assumed skull point) and from there search for the interface. This also proved to be faster in the final implementation. It was therefore required that the remaining points be moved to the skull tissue boundary. Once the points had been moved to the inner edge, some method of connecting them was required. This section addresses these two problems.

Section 4.3: Method 2: Symmetry and Shape Analysis

Finding skull tissue boundary

The method used in finding the optimal boundary position, was to calculate the greatest edge change in a small region along the ray. To do this, only a section of the ray, marginally longer than the width of the skull needed to be analysed. The problem with this is that the range of skull thickness varies between different areas of the skull as well as between patients. For the purposes of this study, the width, previously calculated as a percentage , was converted into a pixel length. The width varied between 5 and 10 mm, and for a scanner with resolution of 0.47 mm per pixel, this related to a maximum of 21.28 pixels. A search area of 30 pixels before and after the point being analysed was thus used. The increase assured that the interface is within the search area (Assuming of course the original point is on the skull).

The edge change was found as follows: The average value of the range to be searched was found. A mask of 5 pixels was then moved along the ray, and the average value found. If the average of the mask exceeded the average of the range, then the lowest point in the mask is used as the border point. The use of five pixels was to ensure that a spurious point did not cause a false position. In practice, the average value is not used but a value 20% higher than the average. This caused a small portion of the skull to be kept, but it was considered better to keep data than discard it (It should be noted that if the skull itself is included in the average, this results in an average somewhere between the skull and tissue). Due to the higher threshold, there were cases where no pixels above the ray average were found. When this occurred the threshold was reduced until one was found. The degree of reduction was once again dependent on the accuracy versus processing trade-off. The amount of times of an edge not being found was small, and thus a decrement of 5% justified.

There were occasions when the *averaging search method* did not achieve the desired results. This occurred when there was a depression in the CT values along the ray. This causes the average to be much lower than the skull or tissue, resulting in the tissue being marked as the boundary point. Another occurred for abnormally thick skull regions, were the averaging process returned a value above the tissue skull boundary. This resulted in the boundary point being in the middle of the skull region. Neither of the two errors resulted in large deviation from the true skull/tissue interface. More importantly, they do not include any of exterior scalp region which is more likely to corrupt an analysis.

Connecting final points

The final step in the intracranial extraction process, was connecting the boundary points calculated in the previous stages. The points need to be joined, as a closed boundary is required to extract the intracranial volume. The boundary must not intersect or cross itself as this would lead to ambiguities during extraction. Due to the method of ray casting, the

order of connecting the points is trivial. The points are connected according to their angle, thus point at angle 0 is connected to point at angle 5 and so on. The method of connecting is given next.

The first attempt at connecting the points, was to simply join them straight lines. This method works well for upper head slices in the cranial region, the reason being the absence of any complicated structures such as eye sockets and jaw and chin bones. With the upper cranial images an elliptical set of points is obtained and the angular difference between points is small. This results in a smooth boundary, despite connections with straight lines. Higher fitting orders were also tried, but the results did not improve enough to warrant the increase complexity and time.

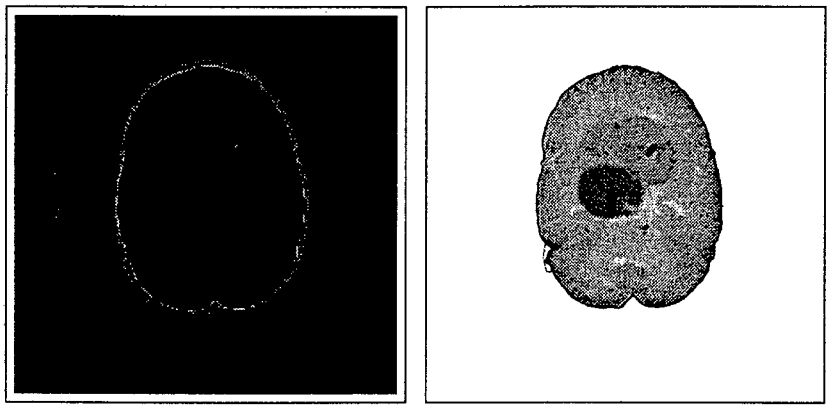
4.3.7 Results

A typical result of the extraction process is shown in Figure 4.9. The image was from the mid-cranial volume and thus does not include the eye sockets or nasal cavities. For slices of this type[8] the results were very good (The images analysed in this dissertation). The intercranial boundaries found using the symmetry and shape analysis method matched those of the adaptive thresholding method. The algorithm did not fail on images where discontinuities in the skull were present. This was a substantial improvement over the thresholding method. It also found more accurate boundaries, as thresholding does not always produce the optimum spatial positioning with respect to the skull/intracranial volume interface.

The algorithm was also applied to lower axial slices, where traditional thresholding and tracking methods (section 4.2) and those by Sandor[21] have failed. The reason for these methods not working, was their reliance on thresholding to form a closed skull boundary. For the lower axial images such boundaries can rarely be formed and therefore the failure. The symmetry method does not require this and thus can perform some degree of extraction. An example of this is shown in figure 4.10, where the intracranial volume has been correctly extracted. The reason for the sharp edges is due to the way the boundary points were connected. In spite of this, the majority of the skull and scalp has been removed.

In order to analyse the effects of an accurate extraction, some measure of the image needed to be analysed. An appropriate measure, was the image grey level distribution. This was chosen due to its later use in the segmentation algorithm. Figure 4.11 shows the histogram of the image in figure 4.6. The dotted line is the histogram of the image including the scalp and skull. The first noticeable difference is that the original image does not go to zero in the usable CT range. This caused problems when analysing the tissue distribution for segmentation. Another major problem was that the distribution including the scalp distorted

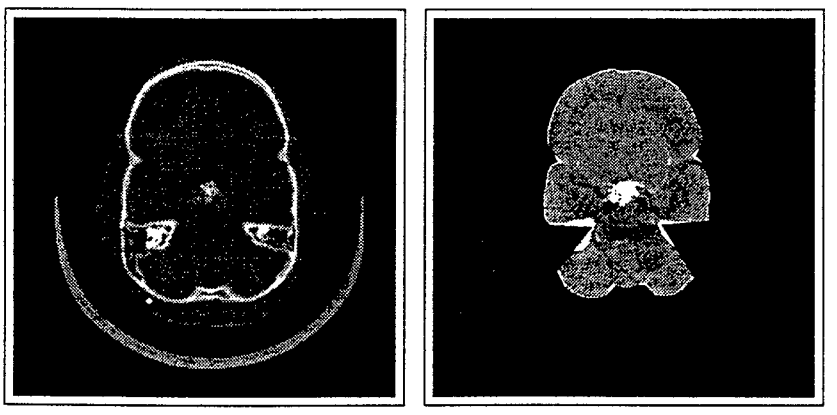
[8]Slices though the upper cranium



(a) Final boundary after connecting points

(b) Boundary after background removed

Figure 4.9: Middle Axial Image - Extraction



(a) Original Image

(b) Image with Background Removed

Figure 4.10: Lower Axial Image - Extraction

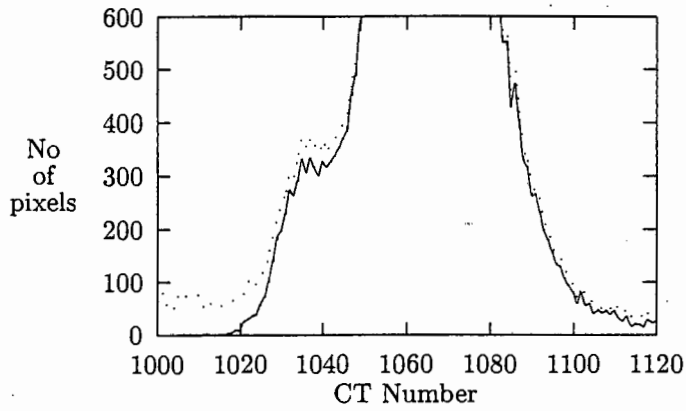


Figure 4.11: Comparison between Original and Intracranial volume.

the true distribution of the intracranial volume. Algorithms that analyse the cranial volume but calculate their statistics from the entire head would be affected by this difference.

The theoretical percentage volume difference was also checked with the real data. This gave an idea of the effectiveness of the model and its use in this context. In Section 4.1, it was predicted that the percentage of scalp and skull in a typical CT brain image is roughly 23%. Using Equation 4.6 on real CT data resulted in an average value of 20%, which agreed with the theoretical estimate.[9] This confirmed the need for the removal of the skull and scalp. It should be noted that of the split between skull and scalp is roughly a 60:40 to 70:30 split. This was calculated by thresholding the stripped background and doing a pixel count on the two regions. This showed that roughly 9% of the tissue being analysed in the histogram in figure 4.11 consists of scalp. This, if kept, would most certainly cause misclassifications in any future segmentation process.

[9]The difference of roughly 3% is due to the use of a simplified model.

Section 4.3: Method 2: Symmetry and Shape Analysis

Chapter 5

Noise Suppression in CT Images

5.1 Nature of Noise in CT Images

5.1.1 Introduction

The noise in CT images can be divided into two major contributions. The first being noise from the data acquisition process (forming of projection data) and the second from reconstruction errors. The second type of noise can in turn come from two different sources: The first is caused by an insufficiency of data, and the second is due to random noise in the measurements themselves. As can be appreciated, there are many reconstruction techniques, all with varying effects on the noise in the image. For this work, the noise analysis is based on the filtered back-projection algorithm described in Kak[1].

5.1.2 Polychromaticity Artifacts in X-Ray CT

In the energy ranges used for medical diagnostic purposes the linear attenuation coefficient for many tissues decreases with energy. For a polychromatic x-ray beam, this causes the low energy photons to be absorbed, so that the remaining beam becomes proportionally richer in high energy photons. Thus, the mean energy associated with the exit spectrum $S_{exit}(E)$, is higher than the incident spectrum $S_{in}(E)$. This phenomenon is known as *beam hardening* (Russ[43, pg.564-573]).

Beam hardening artifacts are represented by an elevation in CT numbers for tissues close to bone. The effect of this is to cause *whitening* of the tissue around the skull. It is also known as the *cupping effect*. This can cause errors when classifying tissue in these areas. Beam hardening can also cause dark and wide streaks to appear in cross section that include the petrous bones.

5.1.3 Aliasing Artifacts

Aliasing artifacts are usually caused by an insufficient amount of data. An insufficiency of data may occur either through under sampling of the projection data or because not enough projections were recorded. The distortion or noise that arise on account of data insufficiency are usually called the aliasing distortions. In general, the noise of this form manifests itself as streaks or star like patterns. The size and width of these artifacts are dependent on the number of projections and the number of samples per projections.

Kak[1] states that for a well-balanced $N \times N$ reconstructed image, the number of rays in each projection should be roughly N, and the total number of projections should also be roughly N. The requirement is met in almost all significant CT scanners and thus this source of noise was not of major concern. It was also assumed that for the images used, this type of noise can be ignored.

5.1.4 Noise in Reconstructed Images

The second type of error, due to random noise in the CT measurements (i.e. noise in the projection data) has been analysed by Herman[29]. A brief outline of the causes and nature of this noise are given and next it will then be show how these results can be practically used for the given CT Images.

The results of Herman[29] and Kak[1] show the second type of noise to consist of two major parts. The first, a continuously varying error due to electrical noise or roundoff errors, can be simply modelled as additive noise. The second type of noise is best shown as shot noise in X-Ray tomography. In CT images, the magnitude of this possible error is a function of the number of X-Ray photons that exit the object. In the case of the first error, the reconstructed image can be considered the sum of two images, the true image and the image resulting from the noise. In the second case the error analysis becomes more involved.

In Kak[1] the randomness of the measured projection is derived. The randomness is statistically described by the Poisson probability function. This analysis is then completed for the reconstruction algorithm and the variance and mean for each point in the image is derived. Although the mean and variance are based on the Poisson distribution, Lei[44] showed that for a large number of projections and rays per projection, the distribution approaches that of a Gaussian. Therefore, for this analysis, the noise generated by the reconstruction algorithm will be taken to have a Gaussian Distribution.

5.2 Noise Suppression Filter

Following the description and cause of the noise, a filter can be designed that attempts to restore the Image. This filter is directed at the noise described in section 5.1.4 and not the effect of beam hardening. As noted in the aims of the thesis, the requirement was to find the volume and size of the brain and any abnormalities (Tumour,cyst etc..). Due to strict clinical needs, the following requirements must be met by any filter that tries to improve the signal to noise ratio of CT images:

The filter must be edge-preserving and at the same time maintain partial volume information in the averaging sense. This simply means that when summing over a region containing partial volume effects, the results should be the same for both the original and restored image. As well as this, it should achieve as much noise suppression as possible. (Partial voluming occurs when a single slice or pixel contains a transition from one tissue type to another. As only one CT value (grey level) can be assigned to each point, there is an inaccuracy in the reading. This becomes more of a problem as the slice width increases causing the transition in the volume to increase. This problem is known as the *partial volume effect.*)

To achieve the above mentioned noise reduction and edge preservation goals, the proposed filter used both spatial and feature domain information. Spatial information refers to the relationship between adjacent pixels from particular tissue types. This is obtained from the structures visualised in the image. Feature domain information refers to the relationship between grey levels from a particular tissue and the physical properties of the tissue. The design of the filter is similar to that given by Windham [38], with greater emphasis placed on automated selection of the filter threshold (see later).

One can model the pixel grey level (P_{jk}) in a region corresponding to a specific tissue (tissue i) with Hounsfield Unit (HU_i) as:

$$P_{jk} = HU_i + w_{jk} \tag{5.1}$$

where the subscript jk represents the pixel location, i.e. jth row and kth column of the image, and w_{jk} represents a zero-mean white Gaussian noise with standard deviation σ. The model in Equation (5.1) provides a means for obtaining a least-square (LS) or maximum likelihood (ML) estimate if the pixel intensities.

Using the LS estimation, first the optimum grey level value $\hat{H}_x$ can be found. The least square error of the centre pixel and its neighbouring pixels are given by:

Section 5.2: Noise Suppression Filter

$$\begin{aligned}
{\varepsilon_x}^2 &= \sum_{i=-k}^{k}\sum_{j=-k}^{k}(\hat{H}_x - P_{ij})^2 \\
&= \sum_{i=-k}^{k}\sum_{j=-k}^{k}{\hat{H}_x}^2 - \sum_{i=-k}^{k}\sum_{j=-k}^{k}\hat{H}_x P_{ij} + \sum_{i=-k}^{k}\sum_{j=-k}^{k}{P_{ij}}^2
\end{aligned}$$

By taking the 1st-derivate,

$$\frac{\partial {\varepsilon_x}^2}{\partial \hat{H}_x} = 2\sum_{i=-k}^{k}\sum_{j=-k}^{k}\hat{H}_x - 2\sum_{i=-k}^{k}\sum_{j=-k}^{k}P_{ij} \tag{5.2}$$

and setting $\frac{\partial {\varepsilon_x}^2}{\partial \hat{H}_x} = 0$, the optimal grey level can be found.

$$\sum_{i=-k}^{k}\sum_{j=-k}^{k}\hat{H}_x = \sum_{i=-k}^{k}\sum_{j=-k}^{k}P_{ij} \tag{5.3}$$

Noting that $\hat{H}_x$ is a constant,

$$\sum_{i=-k}^{k}\sum_{j=-k}^{k}\hat{H}_x = N\hat{H}_x \tag{5.4}$$

and substituting 5.3 into 5.4, the optimum grey level $\hat{H}_x$ is given by:

$$\hat{H}_x = \frac{\sum_{i=-k}^{k}\sum_{j=-k}^{k}P_{ij}}{N} \tag{5.5}$$

Thus, the optimum grey level value for the pixel P_{jk} is the average of its neighbouring pixels. The model in Equation 5.1 assumes the additive noise is Gaussian, however in the calculation of the LS estimate this is not explicitly included. This was done to simplify the calculation. It should be noted that for this Gaussian model, the LS estimate coincides with the ML. For both these cases the result is simply the sample average Stark[45].

To justify the simplification in the LS calculation, the ML estimation has been included, and it is shown that it is indeed equivalent to the LS for the model in 5.1. The pixels surrounding

P_{jk} can be viewed as forming a random sample $(x_0...x_N)$ from a normal distribution for which the mean is unknown and the variance known. In this case the mean will be the most likely value of the pixel (P_{jk}). For the values $(x_0...x_N)$ the Maximum Likelihood function $L(\mu,\sigma)$ will be:

$$L(\mu,\sigma) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp^{-\frac{1}{2\sigma^2}\sum_{i=0}^{N}(x_i-\mu)^2} \tag{5.6}$$

The requirement was to find the value of σ and μ that maximised Equation 5.6. As σ was known, it was clear that this could be done by minimising the value of the exponent. i.e: Minimise $Q(\mu)$ where:

$$\begin{aligned} Q(\mu) &= \sum_{i=0}^{N}(x_i-\mu)^2 \\ &= \sum_{i=0}^{N} {x_i}^2 - 2\mu\sum_{i=0}^{N} x_i + N\mu^2 \end{aligned} \tag{5.7}$$

By taking the derivative of $Q(\mu)$,

$$\frac{\partial Q(\mu)}{\partial u} = -2\sum_{i=0}^{N} x_i + 2N\mu \tag{5.8}$$

and setting $\frac{\partial Q(\mu)}{\partial u} = 0$, the optimum mean value can be found,

$$\mu = \frac{\sum_{i=0}^{N} x_i}{N} = \overline{x_N} \tag{5.9}$$

Equation 5.9 shows that the value of pixel (P_{jk}), using the ML estimation, is the mean of the neighbouring pixels. This is the same result as equation 5.5, thus showing the LS and ML estimates are the same for the model used in equation 5.1. It also justifies the simplification of the Least Squares calculation shown above.

The approach to defining the filter was as follows. A neighbourhood around each pixel is considered. The Euclidean distance of each pixel grey level from that of the pixel in the centre was found. If this distance was smaller than a specific threshold value, η then that pixel is considered in the LS or ML estimation, otherwise it was not. The LS or ML estimate,

i.e., the average of the contribution grey levels is determined (as defined above) and saved for all the contributing pixels. The neighbourhood is then moved an the procedure repeated. Finally, the average of several estimates obtained for a particular grey level is calculated to obtain the filter output for that pixel.

5.3 Selection of the Threshold

The criteria of edge preservation is determined by the choice of threshold. As only pixels of grey level difference within the threshold are chosen, the edge information is better maintained than would be if the entire neigbourhood was averaged. Ideally, the threshold η selection was based on the noise standard deviation σ in the image, the contrast between adjacent tissues and partial volume averaging affects which are reflected in the sharpness of the edges in each image. In practice, the selection of the threshold was based on the probabilities of detection P_D and false alarm P_F. In the ideal case, it would be desirable to have probabilities of 0.0 and 1.0 for P_F and P_D respectively.

The probability of detection P_D is the chance of correctly identifying a pixel P_{jk}^d in the neighbourhood which represents the same tissue type as the pixel P_{lm}^d in the centre. These pixels are assumed to be uncorrelated and Gaussian distributed, with mean d and standard deviation σ. The difference ${D_{jk}}^d = {P_{lm}}^d - {P_{jk}}^d$ is therefore Gaussian distributed, with a zero mean and standard deviation $\sqrt{2}\sigma$ Stark[45]. Using the threshold value η for the absolute distance $AD_{jk} = |D_{jk}|$ yields the following probability of detection P_D:

$$P_d = \int_{-\eta}^{\eta} \frac{1}{2\sqrt{\pi}\sigma} \exp^{-\frac{x^2}{4\sigma^2}} dx \tag{5.10}$$

The probability of a false alarm P_F is the chance of wrongly classifying a pixel ${P_{jk}}^u$, which represents a different tissue, as one that represents the same tissue as that of the pixel ${P_{lm}}^d$ in the centre of the neighbourhood. As before, these pixels are assumed to be uncorrelated and Gaussian distributed with standard deviation σ, but different mean values u and d, respectively. The difference ${D_{jk}}^u = {P_{lm}}^d - {P_{jk}}^u$ is then Gaussian distributed with mean value $m = d - u$ and standard deviation $\sqrt{2}\sigma$. The probability of false alarm is thus

$$P_d = \int_{-\eta}^{\eta} \frac{1}{2\sqrt{\pi}\sigma} \exp^{-\frac{(x-m)^2}{4\sigma^2}} dx \tag{5.11}$$

In figure 5.1 a graph of the Probability of Detection subtracted from the Probability of False

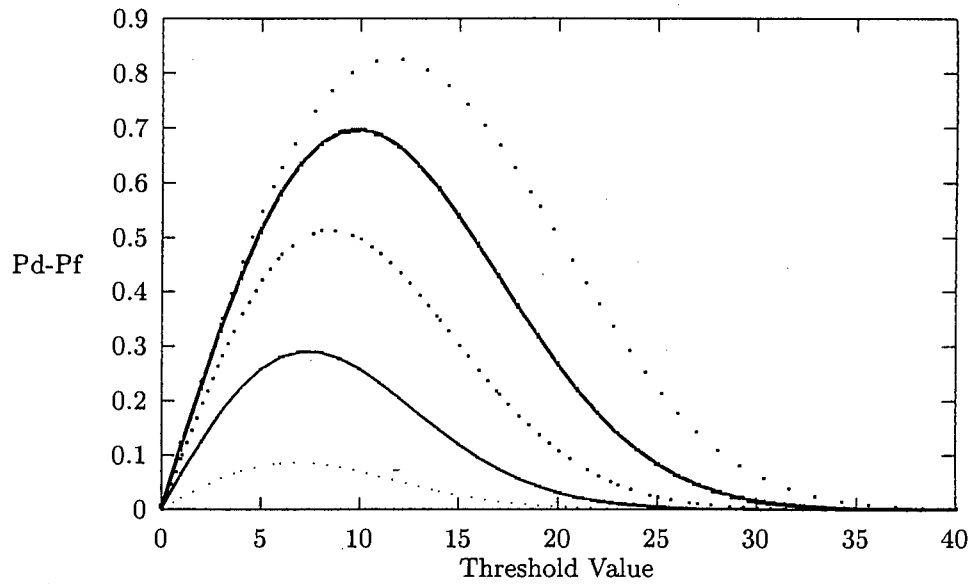


Figure 5.1: $P_d - P_f$ versus η for various values of m

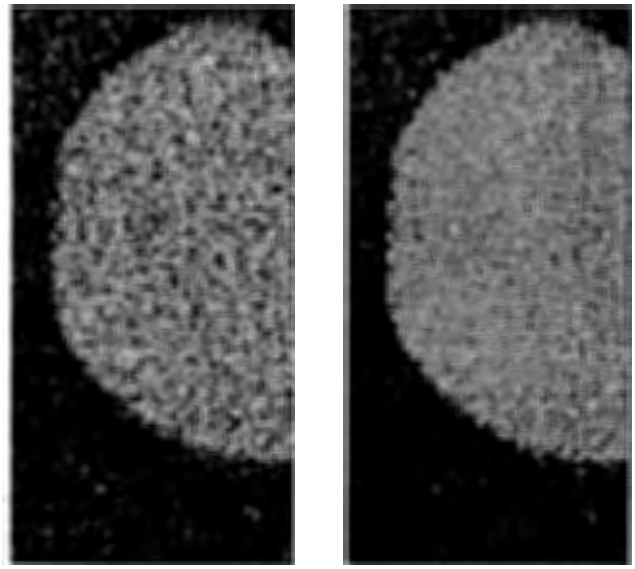
alarm versus η (threshold) is given. Each curve in the graph represents this relationship for a given d, the distance between the tissue mean CT values. The values of d, for the curves from bottom to top, are $4, 8, 12, 16, 20$ respectively. The significance of these curves is evident when choosing a threshold value η. The minimum distance between tissue means was found, for the test data used, to vary between 8 and 16 Hounsfield units. Using this as an estimate for m, a threshold based on figure 5.1 was chosen. The theoretical optimum threshold was found to be 11, which for the range specified earlier has a $P_D - P_F$ ranging from 0.6 to 0.7. These values also hold for changes in the noise variance σ.

5.4 Results of noise reduction

The above filter was applied to a number of clinical brain images, using various size neighbourhoods and thresholds. It was found that by increasing the neighbourhood size, the results improved. This is however rapidly offset by the required processing time. It was also found that the gain obtained using the average of multiple neighbourhoods for a single pixel was not worth the large increase in processing time. Thus, only a single neighbourhood was used for each pixel.

Increasing the threshold value also resulted in further noise suppression (a larger P_D), but it may have the effect of destroying edge and partial volume information (a larger P_F). It was found that the best threshold to use was approximately 10% larger than the theoretical value.

This was due to a higher trade off between partial voluming and noise reduction (The volume difference changed less dramatically than the noise suppression and thus was deemed to be justifiable). Repeated application of the filter was also tried. This further suppressed the noise and had less adverse effects on edge and partial volume regions. However, the increased processing time prevented this from being used practically.



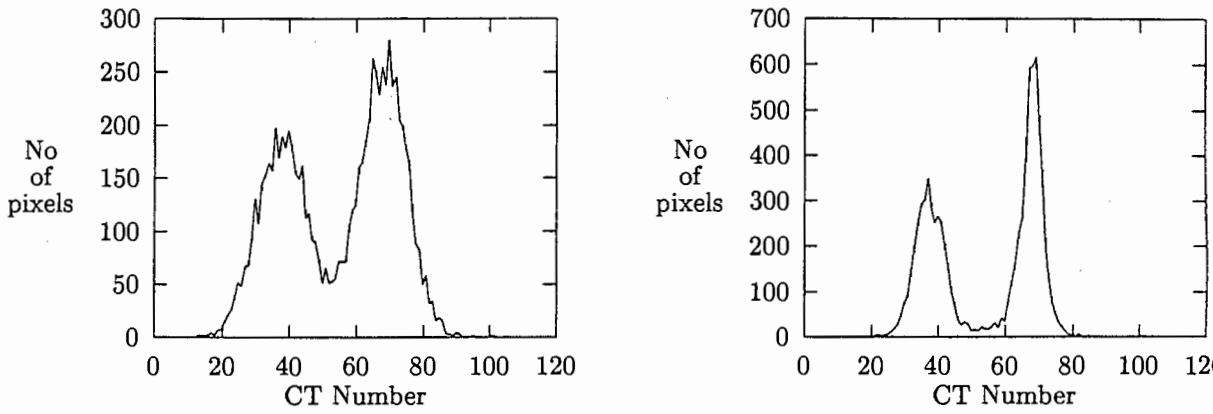
(a) Noisy Image (b) Filtered Image

Figure 5.2: Noise Reduction - Images

The 5.2(a) shows a portion of a unfiltered CT Image. The sub-image consists of two different tissue regions. The lighter region represents a cyst and the darker region the brain grey matter. This portion was used to highlight the two criteria that need to be fulfilled by the filter, i.e. edge preservence and maintaining of partial volume information.

The results of the filter are shown in Figure 5.2(b). The threshold used was 11 and the filter size was 6×6. The threshold was found using the aforementioned method and the filter size based on the minimum expected region size and available processing time. It is clear that in Figure 5.2(b), the filtered image, the cyst has a more uniform surface than that of Figure 5.2(a). It is also clear that edge of the cyst has not only been maintained, but also sharpened. The spatial relationship between the cyst and tissue has also been preserved.

To further show the effectiveness of the filter, a histogram of the Images in Figure 5.3 was included. The histogram in Figure 5.3(a) shows the effect the noise has on the cyst tissue. Due to the noise, there are a number of peaks centred about a grey level of about 70. After the filtering, the peak has become unimodal and much sharper. The grey level variance within



(a) Noisy Histogram

(b) Filtered Histogram

Figure 5.3: Noise Reduction - Histograms

the cyst has also been reduced (due to the averaging effect of the filter for tissue within the threshold), which will help greatly in the task of segmenting the various tissue types.

Although further tests regarding the optimum threshold and size of the filter need to be done, the results from the filter are encouraging. The uniformity of various tissue regions has been significantly increased, without adversely affecting the edges or partial volume information. More significantly, the performance of the later segmentation modules are vastly improved when applied to the filtered image as opposed the unfiltered one. For this reason, the noise reduction filter has become an integral component in the overall segmentation process.

Section 5.4: Results of noise reduction

Chapter 6

Seeded Region Growing

The following chapter is divided into two parts. The first describes the **Seeded Region Growing (SRG)** algorithm and its implementation, and the second addresses the problem of **Seed Selection**. When designing the segmentation process, an effort was made to make the system as modular as possible. In view of this, the seed selection process incorporates a number of distinct stages that need not only apply to the SRG algorithm, but to any algorithm[1] that requires some rough indication of the mean grey levels and position of objects in the image. Thus, the SRG (and final seed placing) can be viewed as one of many possible final processing methods that can be mixed with the preprocessing stages, to form a viable segmentation method. The use and advantages of such a system will be illustrated when describing how the preprocessing stages could be incorporated into other algorithms.

The method used in this dissertation was the SRG algorithm. The motivation for using Region Growing is:

- It is generally robust under varying noise conditions.
- It always arrived at some final segmentation.
- It can also be implemented to work on different image features.
- It is one of the few effective methods for the segmentation of medical images.

The last reason given was confirmed by Eggleston[9], who found, in laboratory tests of a range of segmentation methods, only a few were successful in segmenting medical images. For CT medical images he claims that either singly or a combination of region growing (all types), Region Grouping/Splitting, Edge Linking or Edge Grouping are the best to use. Therefore Region Growing as a segmentation technique for X-Ray CT Brain images was tried.

[1] Assuming of course that they are using X-Ray CT images

6.1 The Two Dimensional Seeded Region Growing Algorithm

Seeded region Growing (SRG) segments an image with respect to a number of predefined points. These points are often described as seeds and will be subsequently referred to as such. The algorithm starts with a number of these seeds which have been grouped into n distinct sets, say, $S_1, S_2, \cdots, S_n$. An individual set can consist of a single seed or a group of seeds. As these set of seeds are the starting point for the SRG algorithm, it is in the choice of the seeds that the decision of what is a feature of interest, what is irrelevant or noise, is embedded. It is possible to have only one starting seed point (single pixel) and then require a strict error cut-off (ie: If the error is to large then do not add the point to the region being grown), but this defeats the strength of the SRG and thus would not be an useful exercise.

The process of region growing evolves from the initial state of the sets $S_1, S_2, \cdots, S_n$. Each step of the algorithm involves the addition of one pixel to one of the n sets. To illustrate one step in the algorithm, it is useful to consider the state of the sets $S_1, S_2, \cdots, S_n$, after m iterations and then proceeding. Letting T be the single set of all as-yet unlabelled, pixels which border at least on of the grown regions (sets $S_1, S_2, \cdots, S_n$), and $N(x)$ as the set of immediate neighbours of pixel x, the following can be derived:

$$T = \left\{ x \notin \bigcup_{i=1}^{n} S_i \mid N(x) \cap \bigcup_{i=1}^{n} S_i \neq 0 \right\} \tag{6.1}$$

For the purposes of this study, a rectangular grid with immediate neighbours defined as those which are 8-connected to the pixel x, was used. If, for $x \in T$ we have that $N(x)$ borders just one of the sets A_i, then one can define $i(x) \in 1, 2, \cdots, n$ to be that index such that $N(x) \cap S_{i(x)} \neq 0$ and define $\delta(x)$ to be a measure of how different x is from the region it adjoins. The simplest definition for $\delta(x)$ is

$$\delta(x) = |g(x) - mean_{y \in A_{i(x)}}[g(y)]| \tag{6.2}$$

where $g(x)$ is the grey value of the image point x. If $N(x)$ meets two or more of the sets (S_i), $i(x)$ is taken to be of i such that $N(x)$ meets S_i and $\delta(x)$ is minimised (Alternatively, in this case, the pixel can be labelled as a boundary pixel and be appended to a set β of already flagged boundary pixels). The final pixel x that is added to $S_i(x)$ is defined as $x \in T$ such that

$$\delta(x) = min_{x \in T} \{\delta(x)\} \tag{6.3}$$

This then completes the $m+1$ iteration of the SRG algorithm. The process is repeated until all pixels have been allocated. The process the starts of with each S_i being just one of the seed sets. The definitions in 6.2 and 6.3 ensure that the final segmentation into regions is as homogeneous as possible. These equations can however be changed and adjusted to any distance measure that may arise from different features of the image being analysed. The algorithm description above provides a framework for one particular example, ie: grey level contrast, but once again can be adjusted for others.

Calculate Starting Set Values from initial groupings.
Label Starting Seed points.
Place neighbours of seed points (the initial T) in the SSDS.
While the SSDS is not empty
  Remove first item y from SSDS.
    Test the Neighbours if this item:
      If all neighbours of y which are already labelled (other than with the boundary label) have the same label:
        Set y to this label.
        Update running mean of corresponding region.
        Add neighbours of y which are neither already set nor already in the SSDS to the SSDS according to their value of δ .
      Otherwise
        Flag y with the boundary Label.
Merge Regions with corresponding seed identifiers.

Figure 6.1: The Generalised Seeded Region Growing Algorithm

6.2 Two dimensional seed selection

In standard region growing (Gonzalez[41]), the correct choice of the homogeneity parameter is critical in achieving a good segmentation. If this was the case for selection of seeds in Seeded Region Growing, no progress would have been made. Fortunately this is not the case, and if the regions are relatively noiseless, all that is required is that each seed pixel have a grey level that is typical of the region it comes from.

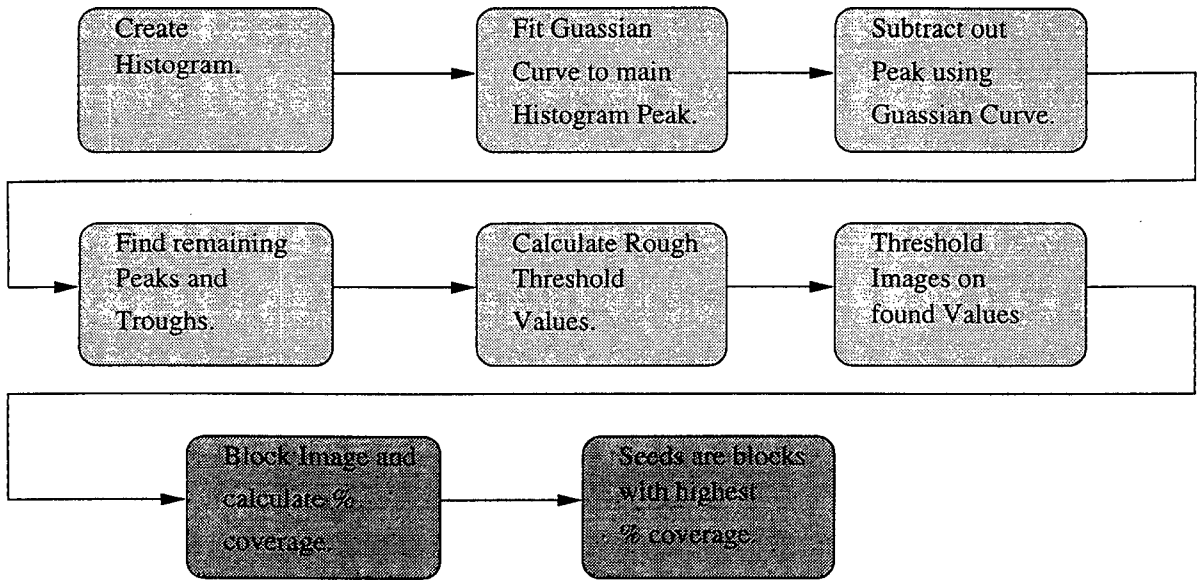


Figure 6.2: Flow Diagram of Seed Finding Procedure

In most Computer Tomography Images this is unlikely. The images are relatively noisy and it is possible that a single seed pixel may fall in a region that it does not belong to. This would result in a poor starting estimate of the regions properties and would result in the segmentation being incorrect. To prevent this, it is suggested by Adams[11] that small seed areas (or group of pixels), instead of single pixels be used. This is consistent with the original description of the SRG Algorithm given in the previous section. In the case of analysing Brain CT Images, it was soon clear that a group of seed points would be required.

Figure 6.2 gives a flow diagram of the seed finding process. The process modules are divided into two groups, the light-grey and dark-grey modules. The light-grey blocks represent the general segmentation modules and the dark-grey blocks the SRG specific modules. When the general modules are used in future, references to this chapter will be made.

6.2.1 Image Histogram

The first step in seed selection involves calculating the histogram (pixel intensity distribution) of the brain region. By analysing the structure of the histogram, the program estimates certain anatomical features of the brain. It is these features (or the lack thereof), that guide the selection of appropriate seed positions. In a normal healthy brain, the pixel intensity distribution has predictable features (shape) that can be effectively analysed. The two major features are the tissue regions and the Cerebrospinal Fluid region (CSF). Example of typical histograms from the mid-cranial region is given by the solid lines in figures 6.3(a) and 6.3(c).

The histogram of brain slices at different z-values have similar distributions. The reason for this is that the majority of matter in the intracranial volume consist of grey and white tissue, thus the distributions are similar for slices at different levels. The image histogram is related to the different structures and matter in the brain as follows: The main lobe on in the histogram represents the grey tissue mean, the fluid mean (with lower CT numbers) lies on the left hand shoulder of the main lobe and the white matter on the right hand shoulder of the main lobe.[2] In most cases the fluid mean, white matter and abnormal tissue is completely overlapped by the main lobe and thus not easily visible . This unimodal nature is a major problem when attempting to find a way of distinguishing the dominant tissue from the fluid or any abnormalities. Thus, a method needed to be found that would effectively unmask the CSF or abnormalities from the tissue.

The method used to separate these regions, is based on the assumption that the majority tissue has a roughly Gaussian distribution. The main lobe is modelled by a Gaussian distribution and this model is used to unmask the CSF or abnormalities. The use of a Gaussian distribution to model the tissue in CT Brain images is justified by Lei[44] as well as in the description of the noise filter in chapter 5.

The first identifiable region in the histogram, was the grey/white tissue mean. This was taken as being simply the peak of the histogram (NB: After the extraction of the intracranial region, the highest value in the histogram was assumed to be the tissue mean. Also, the background value of zero was ignored when calculating the histogram). The fluid or abnormal tissue mean was then found by subtracting the Gaussian tissue model from original image histogram. This subtraction effectively removes the main lobe tissue distribution, leaving the underlying fluid or abnormal tissue distribution. The remaining peaks to the low and high intensity side, of the now removed tissue, can be taken to be the mean of the fluid or/and abnormal tissue. As stated before, the CSF always lies on the low intensity side of the main lobe. Each of these peaks can also be assumed to the mean of a Gaussian distribution and thus their grey level distributions could be estimated. The next section details the method used to find the best fit Gaussian.

6.2.2 Fitting Gaussian Curve to Tissue Distribution

The method used to fit a Gaussian model to the histogram, was a straight-forward Least Mean Squared (LMS) error search of a range of possible Guassians. The search was however optimised in the sense that a highly reduced search space could be used. The search is a three variable problem and by finding good starting values for the three unknowns, one can reduce the search space. This is achieved by using the priori knowledge of the grey level distribution.

[2]Depending its type, abnormal tissue can have a mean value at various different grey levels

Thus, the fitting of the Gaussian was broken into three parts: (1) Finding suitable starting values, (2) From those values estimate the maximum possible extents that need to be search, and (3) Search through the reduced search spaced for an optimum.

Finding the starting estimates

The first step in fitting a Gaussian curve to the CT Tissue distribution, was to find the peak value in the histogram. As described above this main peak position corresponds to the tissue mean. This holds true for all the CT brain images, as the grey and white tissue constitutes the largest proportion of brain matter. Thus, the assumption that the main peak position corresponds to the tissue mean is justified and that it can be used as the initial starting mean for fitting the Gaussian distribution. It was found that the average of a number of adjacent points around the peak value was better. This was due to the peak value being, on occasion, being very high. Using such a high value resulted in an incorrect Gaussian fit, and thus the need to get an average height. Averaging between five and ten points either side of the peak was found to be suitable for estimating the starting mean and height positions.

The next step is to estimate the starting variance. This was done, by finding the half peak value width and substituting this into the Equation 6.4, where A is the height of the distribution at its mean, μ is the mean and σ^2 is the variance. The half peak was found by moving down the curve from the peak position until the points where found. This ensured that only the two half peaks closest to the histogram peak were obtained.

$$f(x : \mu, \sigma^2, A) = A \exp^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{6.4}$$

This is done by solving Equation 6.4 for σ^2 and substituting both half point estimates for x (Which in this case is the corresponding CT Values of the histogram x axis). The result of solving Equation 6.4 for σ^2 is:

$$\sigma^2 = \frac{-\frac{1}{2}(x-\mu)^2}{\log \frac{f(x)}{A}} \tag{6.5}$$

For the half height points $\frac{f(x)}{A}$ can be simplified to $\frac{1}{2}$ and Equation 6.5 can be rewritten as:

$$\sigma^2 = \frac{\frac{1}{2}(x-\mu)^2}{\log 2} \tag{6.6}$$

The x values corresponding to the two half peak values will in most cases give different values for σ^2. As the σ^2 being estimated is just a starting value, it was found that taking the average of the two was sufficient. The x values also enabled an estimation of the extents, on the x-axis CT Values, the Gaussian fit should cover. It would be pointless to include the entire range of CT Values to the fit, when only the 20-30% of the values to the left and right of the half peak x are relevant.

Step size estimates

When fitting the Gaussian the mean and height were incremented by a particular value, the *step* in the process. The value of these steps needed to be determined when fitting the Gaussian. The mean staring point is taken to be the lower of the two half peak x values and the ending point is the higher x value. The mean step is defined as the upper x value less the lower x value, divided by the number of steps wanted (Depending on computing power the number of steps can be adjustable).

$$MeanStep = \frac{x_{upper} - x_{lower}}{NoSteps} \qquad NoSteps \geq 2 \tag{6.7}$$

The same idea is used in finding the Height Steps Size. It would however be impractical to start the Height value from zero, and thus the starting point is taken as the half peak height. Once again a user defined number of steps can be defined, again depending on the computing power available. The height step can now be defined as:

$$HeightStep = \frac{\frac{1}{2}PeakHeight}{NoSteps} \qquad NoSteps \geq 2 \tag{6.8}$$

The last starting value that needs to be estimated is the variance of the Gaussian curve. Unlike the height or mean estimation, the variance estimation is not determined by any fixed values. It was also found that the range of variances that could be taken on by the best fit Gaussian varied a large amount depending on how the error was defined (see below). For this reason the range that the variance could take range from one eighth to eight times the value calculated in 6.6. The number of steps was once again user definable according to processing power. The variance step is defined as:

$$VarianceStep = \frac{8{\sigma^2}_{est} - \frac{1}{8}{sigma^2}_{est}}{NoSteps} \qquad NoSteps \geq 2 \tag{6.9}$$

Section 6.2: Two dimensional seed selection

Once the limits and step size values given by Equations 6.7 ,6.8 and 6.9 respectively have been calculated, the Gaussian distribution can be found. Determining the best fit involved iterating through the height, variance and mean values. For each combination of these variables, the Histogram data is subtracted from the fitted Gaussian and the error found. The set of parameters for which the error measure is a minimum determines the best fit. The next section outlines the error measures used and how they affected the best fit.

Error measures

In forming an error measure, certain criteria which the fit should satisfy needed to be identified. In this case, the over-riding factor was the need for the Gaussian to fit the main tissue distribution as accurately as possible. This needed to be achieved without destroying other histogram data. In order to achieve this, a number of error measures were tried.

Two standard error measures were applied. The first method was to have a minimum fit in the least squares sense. The total error is calculated using Equation 6.10. The second error measure was the City block or absolute difference error and the total error is calculated using equation 6.11.

For the Brain CT distributions, the expected differences between the two error measures should be small. The reason for this is firstly the close similarity between the histogram distribution and fitted Gaussian distribution. This coupled with the fact that only a small (but sufficient) sample of possible Gaussians are used[3], ensured that either could be used with similar results.

$$\varepsilon(LSE) = \sqrt{\sum_{i=x-x_{rng}}^{x+x_{rng}} (Gf(i) - H(i))^2} \tag{6.10}$$

$$\varepsilon(BLK) = \sum_{i=x-x_{rng}}^{x+x_{rng}} |Gf(i) - H(i)| \tag{6.11}$$

When implemented, a number of problems occurred. If equations 6.10 and 6.11 were used without restriction, the fitted curve could lie both above and below the original distribution [4]. Although this gave the least error (with respect to either error measure), when the final subtraction took place, negative values occured in the histogram. The options in dealing with negative values were either to set all the negative points to zero, or simply take the absolute value of resulting histogram. Doing this however, modified the physical structure

[3]The discrete step size also vastly restricts the possible solutions obtainable for the Gaussian fit.

[4]The reason for this is that the distribution of the original image distribution is not a perfect gaussian

of the original distribution and created data that was never originally present. This violated the criteria laid out in the introduction of this section and therefore a restriction on how the curve fits the original data needed to be added.

To solve this problem, a requirement was added that ensured the fitted Gaussian lay beneath the original histogram. This ensured that no negative values were introduced after subtraction, and thus no *new* data was introduced (Another way of viewing the a Gaussian fit lying above the original data, is that data is being subtracted that did not originally exist). The restriction was implemented by disregarding any fit that lay above the original histogram when searching for a solution. In mathematical terms this is expressed as:

$$Gf(i) - H(i) \geq 0 \qquad \text{for all values if } i \tag{6.12}$$

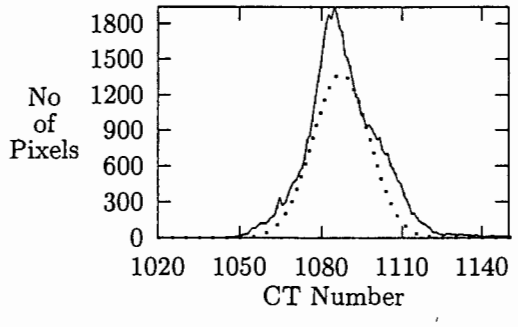
Using the restriction in equation 6.12, the results of fitting a Gaussian curve are given for two histogram distributions. Each distribution has had the different error measure applied and the results are given in figure 6.3. As was predicted, both the least squares and city block error measures gave similar results.

However, an immediate problem with the above method was that it did not fit the peak of the histogram very well. As the peak of the histogram represents the dominant tissue, it is essential that the Gaussian fit this. To achieve this, the error was weighted with increasing importance placed on the peak value. The weight was formed by multiplying the error by the ratio of the histogram value and the peak point. This has the effect of multiplying the error with a scalar between 0 and 1. In practice an average of 10 pixels around the maximum point was used as the peak point value. This ensured that spurious peaks did not cause extremely sharp distributions to be fitted. For most images, the peak value and average peak value were very close and therefore did not affect the Gaussian fit. Using the restriction in Equation 6.12 and the weighted error given by:

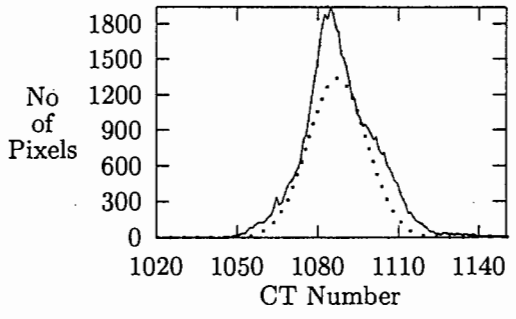
$$\xi(i) = \begin{cases} \frac{H(i)}{PeakAvg} & \text{if } \frac{H(i)}{PeakAvg} < 1 \\ 1 & \text{otherwise} \end{cases} \tag{6.13}$$

Equations 6.10 and 6.11 can be replaced with the following:

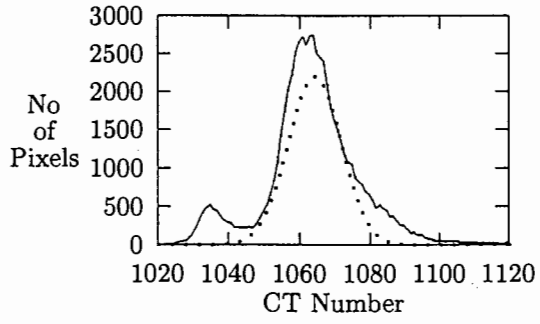
$$\varepsilon(WLSE) = \sqrt{\sum_{i=x-x_{rng}}^{x+x_{rng}} (Gf(i) - H(i))^2 \xi(i)} \quad \text{subject to} \quad Gf(i) - H(i) \geq 0 \tag{6.14}$$



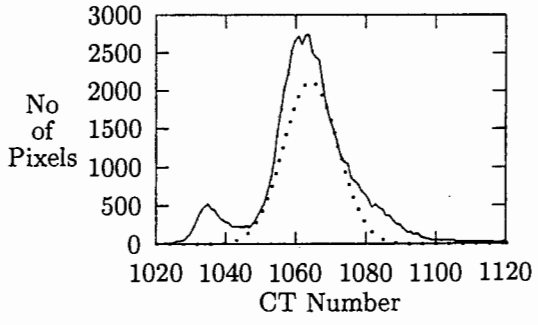
(a) Least Square Error



(b) City Block Error



(c) Least Square Error

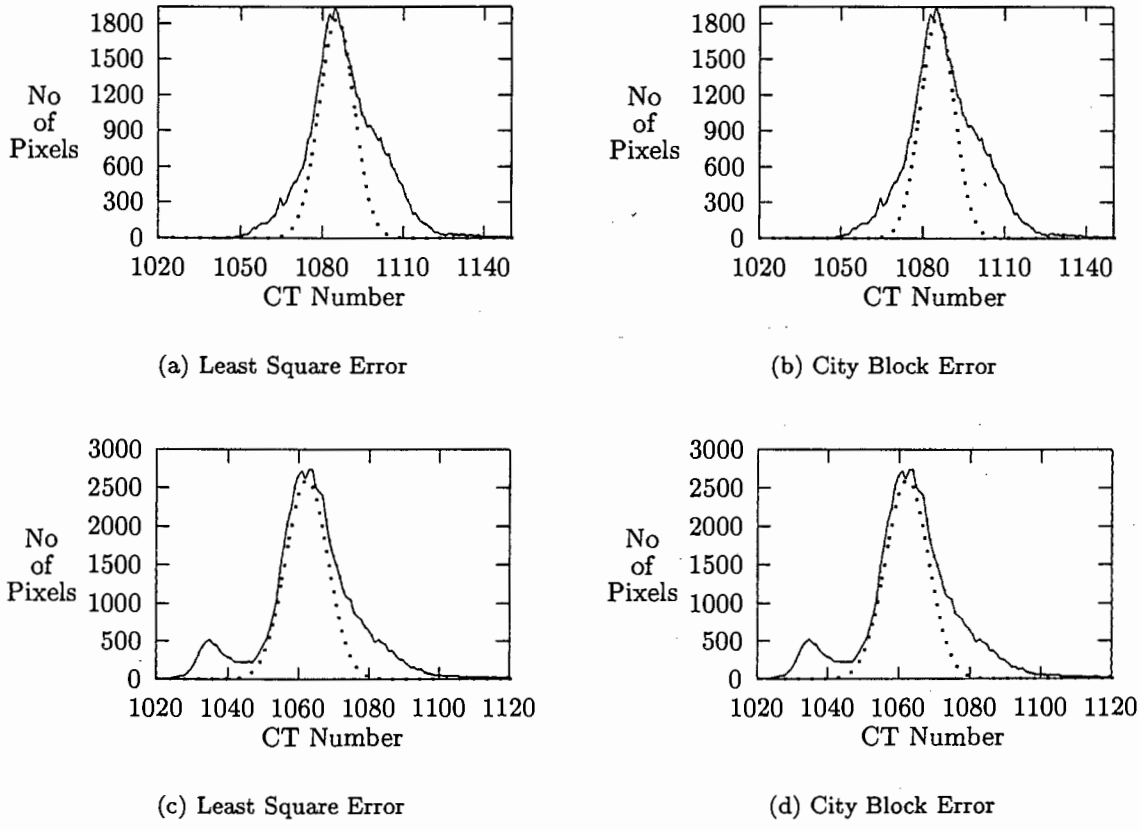


(d) City Block Error

Figure 6.3: Comparison between Gaussian fits with different error measures and no weighting.

$$\varepsilon(WBLK) = \sum_{i=x-x_{rng}}^{x+x_{rng}} |Gf(i) - H(i)|\xi(i) \qquad Gf(i) - H(i) \geq 0 \qquad (6.15)$$

Using Equations 6.14 and 6.15, resulted in the Gaussian fits given in figure 6.4. This time, the curve fitted the peak of the histogram far closer than in Figure 6.3. The difference between the two error measures was, once again, small (dominated by the weighting) and thus either could be used. The LSE however resulted in a marginally smaller variance for most of the images tested, and was used.



(a) Least Square Error

(b) City Block Error

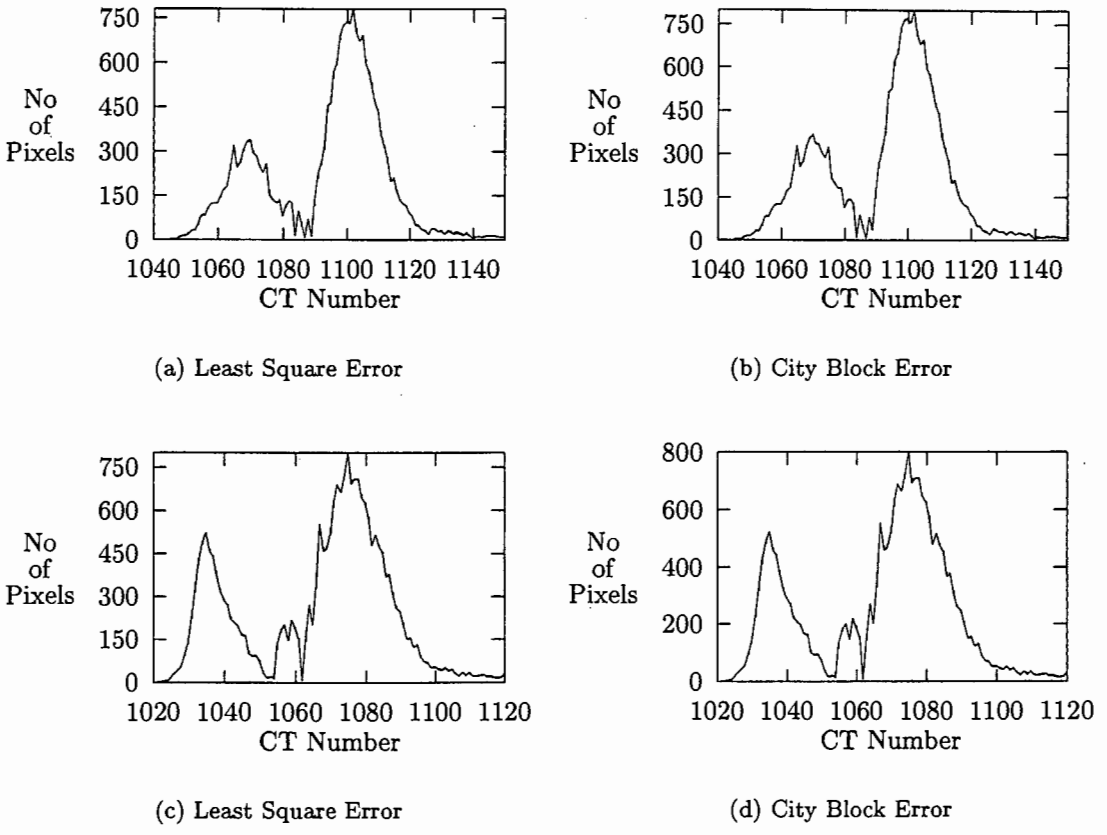
(c) Least Square Error

(d) City Block Error

Figure 6.4: Comparison between Gaussian fits with different weighted error measures

Once the Gaussian had been fitted, it was subtracted from the original distribution. The desired result was a new distribution with the dominant tissue removed. This, as stated previously, should allow masked brain matter, fluid and abnormal tissue to be more easily discerned in the histogram. Evaluation of the subtraction was however, not easily quantifiable. This was due to the large number of peaks present, noise and general variation in the distributions between images. This is evident when viewing the results of the subtraction in Figure 6.5. An effective way of testing the success of the Gaussian fit was to filter the sub-

tracted distribution, calculate the relevant peaks, and threshold the image on those extracted values. This was done in the next section.



(a) Least Square Error

(b) City Block Error

(c) Least Square Error

(d) City Block Error

Figure 6.5: Comparison between subtracted Gaussian fits with different error measures

6.2.3 Calculating Peaks,Troughs and Thresholds

The result of subtracting the Gaussian from the original image histogram is given in Figure 6.5. Each peak in the subtracted histogram represents a possible mean value of some tissue band in the CT image. The band grey level of this mean can be defined as the CT values from the peak to the corresponding left and right troughs. Thus for each peak in the histogram, it is assumed that the grey levels corresponding to the left and right adjacent troughs make up the constituting grey level for that particular tissue.

This assumption relies strongly on the definition of what is a peak or a trough. In general, a peak is defined as a point for which the points to either side have a lesser value. Conversely, a trough is defined as a point for which points on either side have a larger value. Thus P_i is a peak if $P_i > P_{i+1}$ and $P_i > P_{i-1}$ and P_i is a trough if $P_i < P_{i+1}$ and $P_i < P_{i-1}$,

where $1 \geq i \geq MaxPoints - 1$ ($MaxPoints - 1$ is the number of grey levels making up the histogram). In the rest of this section the task of finding troughs is just the inverse operation to finding peaks. In fact, troughs could be found by inverting the histogram and searching for peaks.

Using this definition for peaks and troughs does have problems. The first and foremost is its tendency to treat small variations on a large peak as separate peaks (The same naturally applies with troughs). To solve this problem certain criteria have been used in the search for the peaks and troughs. The criteria were designed to narrow down the search to only certain types of peaks and troughs. The two variables that can be adjusted to define this criteria are the Height to Width Ratio (HWR) and the Window Size. The HWR defines a minimum ratio for the the physical height of the peak to the a predefined width. The Window Size defines the how many points before and after the point being checked should be taken into consideration (eg: If the window size is 2 then point $P_{i-2}, P_{i-1}, P_{i+1}, P_{i+2}$ are used to check if P_i is a peak). Once again same criteria can be applied in reverse for the troughs.

Slope and Value Arrays The first step in utilising these criteria is to set up two arrays that provide information on how the point P_i relates to its surrounding points. A slope array is used to model each point P_i as either being a positive or negative slope, according to the following equation.

$$Slope_i = \begin{cases} 1 & \text{if } P_i > P_{i-1} \\ 0 & \text{if } P_i = P_{i-1} \text{ , } (i \geq 1) \\ -1 & \text{if } P_i < P_{i-1} \end{cases} \tag{6.16}$$

The second array that is set up is the value array. This array models each point P_i in terms of its actual value. Thus if peak $P_i = 100$ and $P_{i-1} = 30$, $Value_i = P_i - P_{i-1} = 70$ for $i \geq 1$.

Slope Array Utilisation Once these arrays have been defined, the peaks can be identified. How the slope array is utilised is described first. If the window size is set to 1, then in order for a peak to be found the points before and after the point must be 1 and -1 respectively. If the window size is 2 then 2 and -2 are required. One can clearly see that only points that rise and fall for more that 2 points will be considered a peak. This is generalised as follows:

Let Window size $= w$. If

$$\sum_{x=1}^{w} Slope_{i-x} \leq w \text{ and } \sum_{x=1}^{w} Slope_{i+x} \geq w \tag{6.17}$$

then P_i is a peak.

The window size does not necessarily have to be the threshold as is in equation 6.17. One can use a threshold that says if a certain percentage of the points are rising or falling within that window then consider the point a peak. This avoids the problem of noise on the data points. A problem with this method is the fact that it does not distinguish a single small rise from a single large rise. For CT images this is not much of a problem as the distance between peaks (and thus tissue means as described above) is the main criteria.

Value array utilisation In utilising the Value array, both the window size and the HWR are used. The basic principle is to add up the array values in the window and then average them by dividing by the window size. The value obtained is then compared against the HWR value entered by the user. This is done for both the window ahead and behind the point being checked. Mathematically this can be seen in the following.

Let Window size $= w$, if

$$\left|\frac{\sum_{x=1}^{w} Value_{i-x}}{w}\right| \geq HWR \text{ and } \left|\frac{\sum_{x=1}^{w} Value_{i+x}}{w}\right| \geq HWR \tag{6.18}$$

then P_i is a peak.

The HWR variable has a significant impact on the way the peaks are found, and it is recommended that it be set to zero unless specific circumstances require its use. The higher the HWR value, the greater the peak must be in order to be detected. For CT brain images the HWR should only be used if a high degree of confidence in the type of histogram generated is known.

Dealing with plateaus A problem with using the definition above, was that it did not take into account the occurrence of plateaus in the histogram. A plateau has no distinct peak and can be viewed simply as one or more consecutive levels in the histogram that have the same value. With respect to peaks and trough, this was not enough to fully describe the plateau, and information about the grey levels before and after the plateau are needed. The first possibility was the plateau being part of a greater peak and the second when the plateau grey level itself is the peak value. The following paragraph described the situations when these possibilities occur.

If the one side of a plateau satisfies the right side of Equation 6.17 and the other side satisfies neither, then that point is marked as a starting point of a possible plateau. This continues until the window satisfies one of the two possibilities in 6.17. If the point satisfies the left hand side, then that point is marked as then end of the plateau (Thus two consecutive peaks are formed, representing the start and end of a plateau. How one deals with a peak plateau was

not part of the peak finding process). The second possibility was that the window satisfies the right hand side of 6.17. This meant that there was a rise followed by a plateau followed by another rise. When this happens, the originally marked point is unmarked and the search for a peak continues normally.

Then final implementation of the peak detector involves the following steps:

1. Initialise the Slope and Value Arrays
2. Check 1st Point to see if it is a peak
3. Find Peaks before the window size, prior to the P_i, can be used.
4. Find Remaining peaks until window size, ahead of P_i, can be used.
5. Determine peaks when not enough points left to use window size ahead of P_i.
6. Check if Last point is a Peak.
7. Check for plateaus.

Figure 6.6: The Peak Detection Algorithm

Filtering the resulting peaks and Troughs

Once the peaks and trough have been found according to the method outlined in figure 6.6, they need to be filtered to provide a smaller range of possible objects (In the histogram, each peak represents some anatomical objects grey level distribution. The troughs represent the base of these peaks and the grey levels between them are sometimes referred to as a threshold band). The reason for reducing the number of threshold bands was that, in CT brain images, there are only a few discernible object types[5]. If there are, for example, fifty peaks, it is obvious that not all of them can be distinct tissue types. Thus, methods need to be found that reduce the number of peaks to a reasonable number. In doing this, some of the bands will be disregarded and other peaks will combine, forming larger bands. The way in which bands are disregarded or joined is critical, as it will determine the rough grey level distribution of the anatomical features in the intracranial volume.

[5]The number actual number varies with scanner quality, but is generally between 5 and 9.

Section 6.2: Two dimensional seed selection

Distance between peaks

The most important criteria in filtering the peaks and troughs was the distances between consecutive peaks. If two or more peaks are within a certain proximity to one another, it is unlikely that they belong to different tissue regions. This was based on the fact that the mean grey level values of different CT objects, are a minimum distance from each other (See Chapter 2). It was this minimum distance that was used to merge peaks together.

The first stage in combining the peaks, was removing obvious errors caused by noise. This involved searching for peaks that are within two CT values of each other. If the adjoining peaks are within this limit and and the points between them are of similar height (say 50 %, although for this distance the value is not critical) then the midpoint was taken as the new peak position. The extrema troughs on either side were used as the new troughs. If the secondary peak was small compared to the first, then the first is taken as the peak and troughs defined as before. In all the cases tested, there were no peaks within two CT values of each other that belonged to different tissue types. This justified the use of this method to combine peaks which were from the same anatomical object.

The next stage was to merge peaks with larger distances between CT values. Five grey levels was considered the uppermost limit that could be used for discrimination between different regions. This is based on the analysis done in calculating the threshold for the the noise filter described previously. The distance is reduced from the noise filter to lessen the chances of two regions being mistakenly combined. In the previous case, the in-between histogram values were not important, however due to the increased distance, merging was only considered if certain criteria are met.

Peaks of this distance were merged only if the point between them were within a percentage of the peak points. This required that all points between the peaks needed to be of similar value. In practise this meant only the minimum point (the trough value) must be within a percentage of the peak value. The percentage used depended on the degree of peak decimation required. A high value will retain both peaks, a low value will join them. It was found a value of 75% provided good final results.

Constituting region area

Another method used for removing peaks was calculating the total number of pixels between the troughs. This places importance on the band (trough-peak-trough) according to its contribution in the image. (Not unlike the height to width ratio described previously) It was found that if the number of pixels was less than 1000 (at best a block of 32 by 32), the following stage of the algorithm did not find relevant seed points. For this reason, all regions

less than 1000 were either excluded or joined to an adjacent peak. It should be noted that this criteria indirectly sets the resolution limits of the algorithm. It does this in the sense that any uniform connected region in the output region that is less than the threshold set (say 1000 pixels) will not be segmented.

Final trough adjustments

The final step in grey level band selection, was to adjust the troughs so that a higher degree of confidence exists that the bands contain only pixels from the same anatomical feature. As was described earlier, each region in CT images has an approximate Gaussian distribution. Using this model a confidence interval was developed. If the peak represents true value then values away become more ambiguous with respect to their true region. This posed a problem of what extent the band should be to ensure that incorrect pixels were not included.

The reason for the peak (or tissue mean) not being taken as the sole threshold value, is due to the problems of later seed placement. It was found that by using a single value[6], the region thresholded image became very sparse and meant that there were very few connected pixels. This made the placement of seeds very difficult, as it was not know if the seed was on an edge or not (A seed falling on a boundary, and thus overlapping two regions, caused incorrect starting statistics). By including values to the left and right of the peak more connected regions are formed, enabling accurate seed placement.

However, if one takes the entire range from trough to trough, one is including pixels that could fall into a different region. Thus, the peak half height is taken for the new trough positions (There are exception cases such as a plateau between peaks, in which case the original troughs remain). Although a trough does represent the best point to split two regions, at this stage, **the aim was not to segment the image**, but rather find just enough of the region so that seeds could be placed.

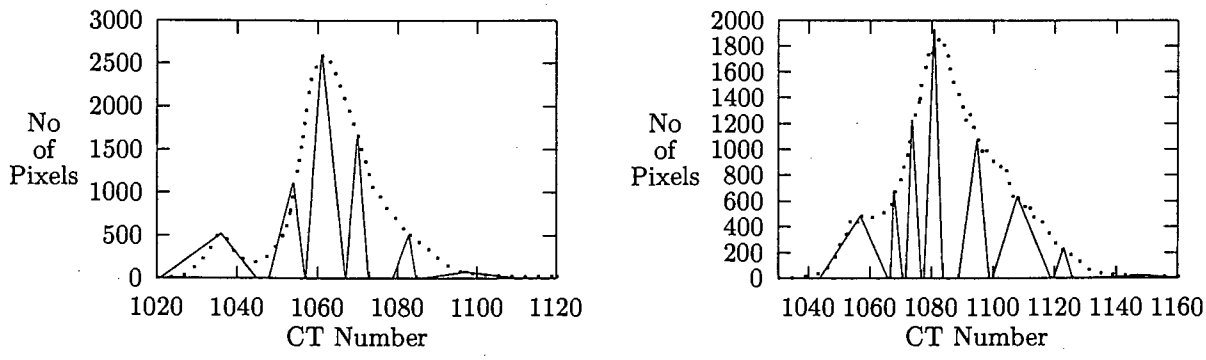
Results

The results of peak and trough finding module, for the CT images in Figures 6.8(a) and 6.9(a), are given in Figures 6.7(a) and 6.7(b) respectively. In these graphs, the peak of each of the triangles represents the peaks found using the above method. The peaks themselves represent the mean value of the different regions in the CT images. The troughs or zero points represent possible separation points between the different regions. Thus, the base of the triangle represents the range of grey level values that make up a particular region in the image (NB: There are areas between the triangle that have consecutive zeros. This implies

[6]The band is used to threshold the image

that there was not enough confidence to fix a hard boundary for these grey levels, and those pixels were not used when thresholding the image).

The results of thresholding the images according to the different bands in Figures 6.7(a) and 6.7(b), is given in Figures 6.8(b) to 6.8(f) and 6.9(b) to 6.9(e), respectively. As mentioned previously, quantifying the results of the CT segmentation is extremely difficult. This is both due to the physical structure of the images (except the ventricle) and the inherent uncertainty in the grey level values (See Chapter 2). For both sets of images, the best method to check the success of both the peak/trough finding and Gaussian subtraction was to first compare outlines of the ventricles to the thresholded ventricles. The rest of the tissue regions are then best judged by using an experienced operator. This was done by having the operator select a series of grey level bands that best segment the image. These were then compared to the bands found automatically. The results once again were favourable and there was a notable agreement on the chosen mean values between the human and machine methods.



(a) Bands for Figure 6.8(a)

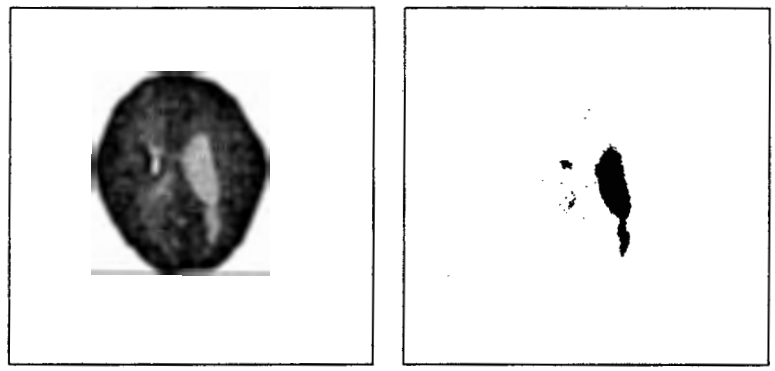
(b) Bands for Figure 6.9(a)

Figure 6.7: Calculated grey level bands for possible anatomical features.

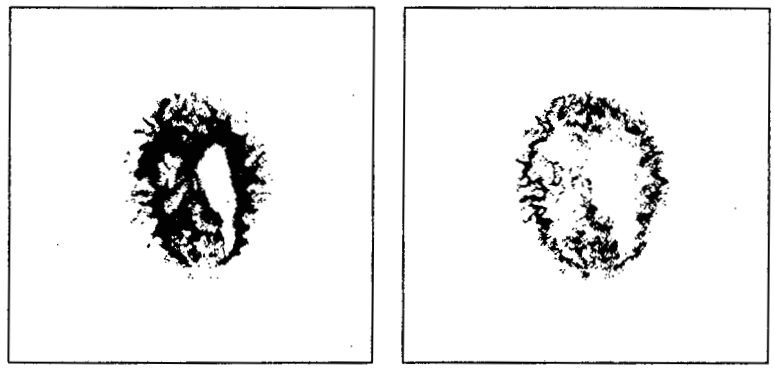
The bands and their corresponding thresholded image clearly show that modelling the dominant peak in the tissue distribution was justified. It has allowed for the extraction of tissue that is obscured by the main tissue distribution. This is especially clear when looking at Figure 6.7(b), where the distribution is unimodal, but after the fitted Gaussian was subtracted, the resulting peaks accurately represented the different regions. Thus, the method described up to now can be used as a simple and elegant preprocessor for various segmentation methods.

6.2.4 Image Blocking to find Seed Positions

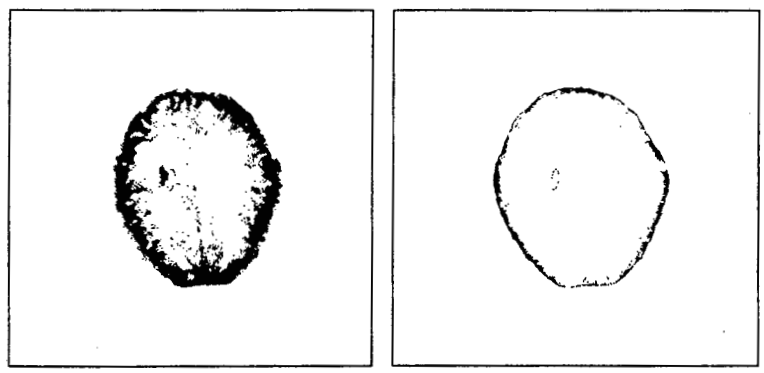
This section describes how the starting seed positions were found. Previously it was stated that the subtraction of the Gaussian tissue model and the formation of grey level bands



(a) Original Image (b) Threshold 1

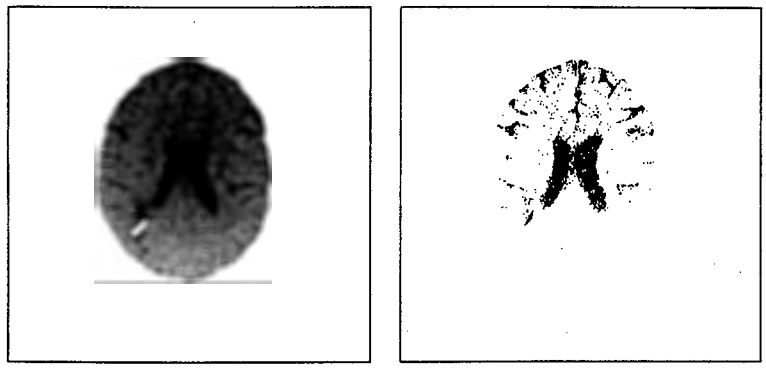


(c) Threshold 2 (d) Threshold 3



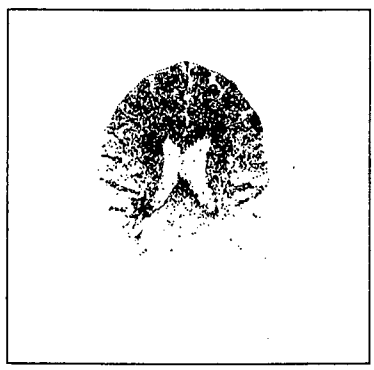
(e) Threshold 4 (f) Threshold 5

Figure 6.8: Images thresholded with calculated bands



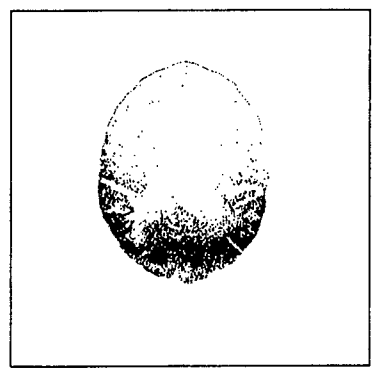
(a) Original Image

(b) Threshold 1



(c) Threshold 2

(d) Threshold 3



(e) Threshold 4

Figure 6.9: Images thresholded with calculated bands

(trough-peak-trough), could be used as a preprocessing module for the SRG or other segmentation algorithms. In the case of the Seeded Region Growing, the calculated bands were used to threshold the original image. The next stage is to use the thresholded image to calculate the starting seed positions.

The actual placement of the seed within each of the thresholded bands was not overly significant, provided it was placed in a *fairly* homogeneous position. The description of fairly homogeneous was calculated by the pixel to no pixel ratio within a fixed size block. More importantly it meant that all efforts should be made to ensure that the seed does not fall on the border of two regions. The thresholded bands in Figures 6.8 and 6.9 give the two basic types of structures that occur in the thresholded X-Ray CT images. The first consist of sparse and thin *finger-like* structures that are, to a certain degree, self similar.[7] The second consists of large, highly connected regions. In a normal brain slice, the first type generally applies to the grey and white tissue and the second to the cerebrospinal fluid and abnormalities such as cysts or lesions. In planning a method to place seeds, these different structure classes needed to be considered.

Despite the expectation that each band would result in only sparse and connected regions, it was found that each thresholded image band contained, to varying degrees, both types. It was decided that a method that could cope with both these, with only a change in parameters being required, would be ideal. This required viewing the regions as having similar structure, but just at different sizes. The method used to achieve this can be described as image blocking and it was implemented as follows:

The image gets thresholded according to each of the bands found in the previous stage. Each of the resulting thresholded images are then blocked. Blocking an image is done by breaking the image into blocks of size $n \times m$. The goal of blocking is two-fold. The first is to try find area of homogeneity in each band. The second is to break up large structures so that more seeds can be placed (This was the essentially idea behind blocking. Small blocks for thin structures, and large blocks for larger structures. This resulted in solutions that can be analysed using similar techniques). To measure the degree of homogeneity, the number of thresholded pixels in each block are summed and divided by the size of the block. The result of this calculation is defined as the **Block Percentage Coverage (BPC)**.

Using this description, given an image $f(x, y)$ with width I and height J, the number of blocks are the integer parts of $\frac{I}{n} * \frac{J}{m}$, where n, m are the same as before. In almost all cases CT scanners produce images of the size $0 \leq I, J \leq 2^r$ where $r \in \mathcal{Z}$. To have a perfect block fit would require n, m to be the set of image of type 2^r where r is the same as defined previously. In the images used, the head data is not on the edge of the image and thus any

[7] They are self similar in the sense that if one reduces the resolution and resizes the image, the structure remain similar.

block size could be used as long as the relevant data was blocked. The part of the image not divisible into a block can either be blocked with a reduced size or ignored. In the case of this dissertation it was ignored. Using this definition, the percentage coverage for each block was calculated by

$$Blk(i,j) = \frac{1}{nm} \sum_{x=im}^{(i+1)m} \sum_{y=jn}^{(j+1)n} f(x,y) \tag{6.19}$$

At this stage there was no point in working with the entire image, as the relevant data (BPC) is stored as single value. For this reason, a scaled block image was formed. The scaled image is reduced by a factor of $\frac{I*J}{m*n}$ where the variables I, J, m, n are the same as defined previously. The new image formed is of the size $\frac{I}{n} \times \frac{J}{m}$ and has $Blk(i,j)$ as the pixel values. The scaled image now represents the coverage of the original image with respect to the chosen block size.

An initial problem that needed to be considered was the discontinuities due to the blocking of the image. This happened when the block landed on a boundary in the image, and resulted in a low percentage coverage value. This in turn caused no seed to be placed for both regions. An alternative solution was to perform the blocking with a sliding window, thus forming a percentage coverage value for each point in the image. The reasons it was not used are the following:

- Although this would have solved the discontinuity problem, it dramatically increased the amount of processing time required. This was especially true for larger block sizes.
- The idea behind the SRG is that only a limited amount of starting points are required. It was found that using the blocking method resulted in enough seeds being found (despite the discontinuities) for the overall algorithm to be successful.
- While accurate seed coverage is ideal, it is not essential. This is in agreement with the definition of the SRG algorithm which requires that the choice of seed position, while being indicative of the region being segmented, need not be 100% accurate.

Thus, the method allowed for blocking the image at various scales, which was deemed more important than getting perfect percentage coverage. It also allows structures of different sizes to be analysed by the same algorithm.

The next stage in the seed placement algorithm was thresholding the scaled percentage coverage image. The threshold represented the minimum allowable BPC value that could be accepted. The BPC varies between 0 for no coverage and 1 for full coverage. Using a high

percentage coverage threshold[8] in conjunction with the blocking, resulted in a group of regions that related to area of high homogenity in the orignal images. Depending on the block size and threshold chosen, these regions ranged from being small (about the size of the block) to being large and highly connected.

The order, in terms of the block sizes, in which the images was blocked and thresholded was very important. It was preferable to first use the large block sizes and then the smaller ones. The reason for this was that by first using large blocks and thresholding, the large connected regions could be removed. This ensured that smaller blocks did not *re-block* these larger regions, causing overlaps.

Finding the best seed position within the thresholded regions needed to be considered next. Ideally the result of the thresholding procedure would be to produce a single pixel for each region. Despite the high thresholds used, this was not the case and the regions often had more than one possible seed site. To solve this, the image was labelled and the highest percentage coverage value in each region searched for. Labelling[9] of a binary image can be defined as giving every pixel that is connected, the same number (In this case 8-connectivity was used). Thus, if there are 4 groups in a binary image, after labelling an image would be formed with each group having a different grey level (The grey levels 1 to 4 in this case). Once the labelling is done, the process of finding the best seed placement within that region can be found. The original scaled image was searched, using the labelled image as a mask. The highest percentage coverage value was then taken as the optimum position.

Previously it was stated that there are two types of structures that could be identified. The highly connected regions and the sparse regions. For these structures, the selection of block size, percentage threshold and minimum region size all effect the manner in which seeds were selected. Different values were tried and compared to predicted estimates, the results of which are given next.

Seeding Sparse Regions

In order to place seeds on sparse regions, it was found that small block sizes, in conjunction with a lower BPC threshold, should be used. The decrease of block size has the effect of increasing the overall percentage coverage and the decrease in the BPC threshold has the effect of allowing more regions to be accepted. The block size should be approximately $\frac{1}{4}$ the size of the region that is being targeted. This ensured that the discontinuities involved with blocking do not influence the seed placement.

[8] 0.8 and above

[9] The code used was written by Patrick M. Kelly, Computer Research and Applications Group, Los Alamos National Laboratory.

Section 6.3: Implementation of Algorithm

For a block size of 3×3 (which was the minimum block size used), the resulting percentage coverage values varied between 0.70 and 1. For block sizes smaller than this, the percentage coverage was much higher. It was also found that the reduction of the block size had more of an effect on the number of seeds than the reduction in BPC threshold value. It could also be viewed in the sense that the decrease in block size has the automatic effect of increasing the region percentage coverage, making it less sensitive to the threshold.

The actual BPC value chosen was 0.75. This was considered the lowest acceptable value as, if it was further reduced, it was more likely that the seed point would lie on the border of two regions. This would cause the mean starting value of that seed to no longer reflect the region it was intended to grow. As mentioned previously the thresholded regions were searched and the pixel with the highest value used as the seed position. This ensured in most cases that the percentage coverage point chosen did not lie on a border.

Seeding large Connected Regions

When targeting large connected regions, it was found that large block sizes, in conjunction with a higher BPC threshold, should be used. The increase in block size has the effect of decreasing the the overall percentage coverage and the increase in the BPC threshold has the effect of breaking up the large regions. The block size should be approximately $\frac{1}{2}$ size of the region that is being targeted. This resulted in the larger regions being broken down into smaller objects. This enables more than one seed to be placed for these regions, which was found give better segmentation results. To futher break up these regions, the BPC value chosen was chosen to be above 0.85. In practice, a block sizes of 6×6 or 9×9 were used. There were however times when these block sizes failed to break apart large regions and thus only one seed was placed. In chapter 7 a method is given which deals with this situation.

6.3 Implementation of Algorithm

As was previously mentioned, the unique aspect of the seeded region growing algorithm is that pixels are added according to their position in a sequentially sorted data structure (SSDS). Thus, recapping from previously, when considering a new pixel at the beginning of each step in the SRG, the pixel at the beginning of the SSDS is taken. When adding a pixel to the list, it is placed according to its value of the ordering attribute. In this case, the SRG, stores the data according to δ given in equation 6.3. Thus, a major problem with using such a method is finding the appropriate data structure.

A standard structure for implementing a SSDS is a priority queue. Of all the priority queues, the simplest to implement and test is the one based on a linked list. Thus, a linked list was

first used. The linked list is efficient when deleting the smallest member, but is very inefficient when inserting in the correct order. According to Tenenbaum[46], the time required for such a search for n insertions is of the order $O(n^2)$ time. For large images or volumes (as will be the case in three dimensions), this is highly inefficient and cannot be used.

Other methods tested included the binary search tree implemented as a priority queue. This method has a best amortised time of $O(n \log n)$, but which can increase to a worst case of $O(n^2)$ for an unbalanced tree. The final method used, is based on the standard heap implementation of the priority queue are known as *Fibonacci Heaps*. This data structure has arbitrary deletion from an n-item heap in $O(\log n)$ amortised time and all other standard heap operations in $O(1)$ amortised time. This is an improvement over both the standard binary tree and the list implementation. It also has a slight improvement over the classic skew-heap priority queue described in Budd[47], which has an amortised running time of $O(\log n)$ for all operations.

As the use of the Fibonacci-heap is to be used as a priority queue, only two of the standard heap operations will be described. These are deleting the minimum and inserting an item into the heap. A detailed description and analysis of the data structure is given by Fredman [48] and this implementation is based on that.

6.4 Results

The results of the Seeded Region growing algorithm were measured according to the methods laid out in Section 2.2. The performance of the algorithm also needs to be seen in the light of section 2.4, which gives an idea of which anatomical structures can be effectively imaged using X-Ray Computer Tomography. The first set of results presented are comparisons between the SRG algorithm and a qualified professional operator. The problem of this type of comparison, regarding the operator, has been given in section 1.1 and will not be repeated here. The second set of results are based on predefined phantom objects.

6.4.1 SRG Results vs Hand Segmented Results

The two anatomical structures chosen to evaluate the algorithm, were a cyst and part of the ventricular system filled with CSF (The ventricular systems being imaged consisted of the lateral ventrical, part of 3d ventrical and anterior horn). Both were chosen because they allowed for the most accurate **hand** segmentation to be done. Grey and white matter are too finely spread and to difficult to be objectively identified and hand segmented (They are generally not used to get quantitative results, but rather to track trends over time). The results were obtained from two image sets. The cyst consisted of 41 consecutive slices (20

contained the cyst itself) and the ventricular system 20 slices (8 which contain the ventricular system). Both has a slice depth of of $2mm$.

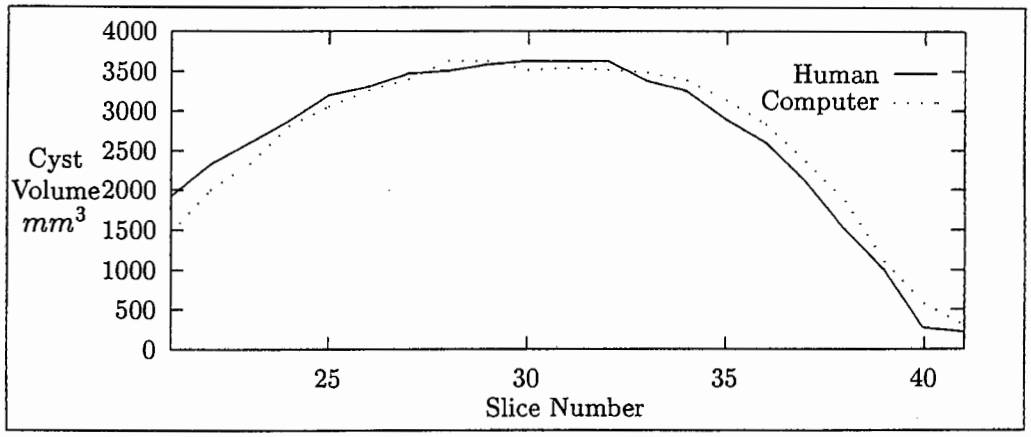
For each image in the set, the operator calculated the area of the relevant structure. From these, the total volume was be found (The method used for this has been outlined in section 1.1). The same method of determining the volume (from the segmented area) was used for the Seeded Region Growing results. This involved first extracting each segmented region, multiplying it by the thickness of the slice and then adding these to form a volume. A comparison between the hand and SRG results are given in figure 6.10. The x-axis represents the slice being segmented and the y-axis the volume of the structure in mm^3. Figures 6.10(a) and 6.10(b) represent the volume for the cyst and ventricular system respectively. From visual standpoint (can be subjective) the SRG and hand segmented methods had similar results.

For the cyst, the total volume by hand was calculated at $56.184cm^3$. The corresponding result for the SRG algorithm was $56.056cm^3$. This gave a percentage difference of 0.228%, which was well within required limits (The limits for calculating abnormal region volume were set by the amount of radioactive isotope that needed to be ordered. Generally a percentage difference of up to 10% is acceptable). The reasons for the success of the algorithm, for this type of structure, was the fact that the cyst:

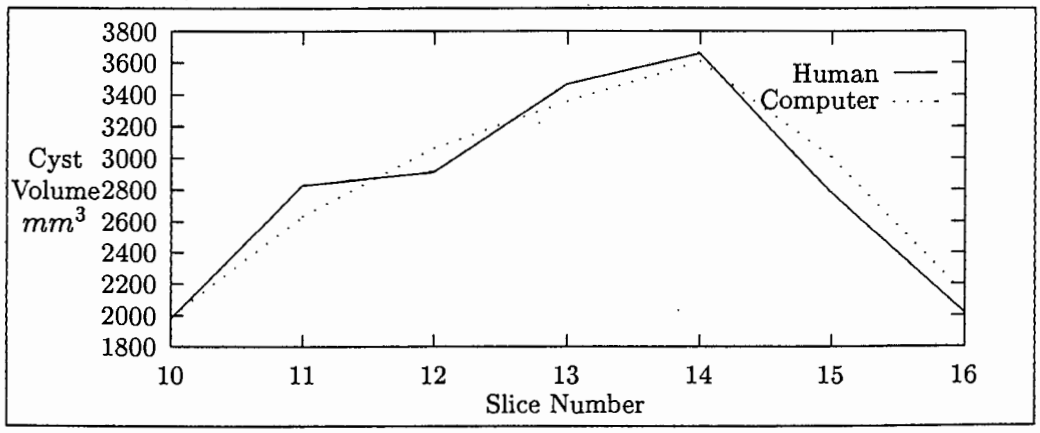
- Had a higher mean value than the Tissue and less importantly CSF, making its threshold less ambiguous.
- It is a tight, highly connected region, which resulted in more noise reduction than for sparse regions.
- Due to its large size, seeds could be placed with very high accuracy.

By further testing, it was confirmed that anatomical structures that fulfilled the above set of items would result in a good segmentation. This was also the case for the next structure tested, which was the lateral ventrical (The criteria in this case were however, not as strong as for the cyst. The CSF has a mean closer to the tissue, its not as highly connected and was smaller than the cyst). The results for the hand segmentation is given in figure 6.10(b). The total volume for the hand segmentation, was $19.71cm^3$ and for the SRG was $19.47cm^3$. This gives a percentage difference of 0.24% (The success is can once again be contributed to criteria given above). These results are also not unexpected for the CSF structures, as according to the table 2.2, in section 2.4, they have high visibility in X-Ray CT images.

The output of the SRG algorithm are given in figure 6.11 and figure 6.12. Figure 6.11 represents the cyst images. Figures 6.11(a) to 6.11(e) are the results of the Region Growing, using the corresponding thresholds in Figures 6.8(b) to 6.8(f) for find the seed placing. It was useful to compare each threshold to the output of the SRG, as this gave an overall idea

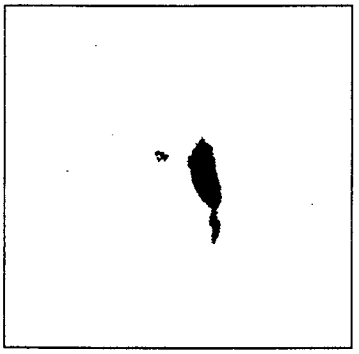


(a) Results for Cyst

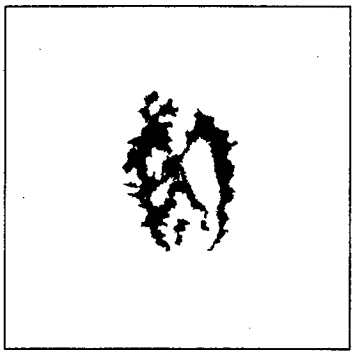


(b) Results for Ventricular System

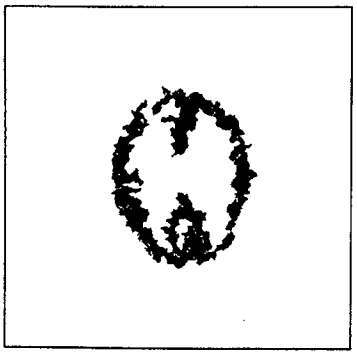
Figure 6.10: Results of Human vs SRG Segmentation



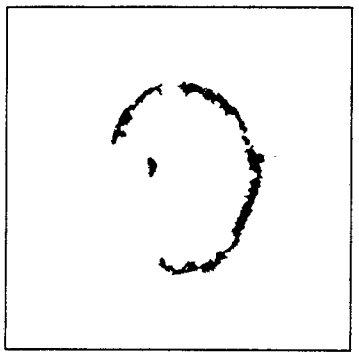
(a) SRG for Threshold 1: Lateral Ventrical



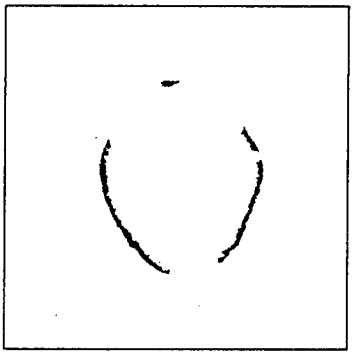
(b) SRG for Threshold 2



(c) SRG for Threshold 3



(d) SRG for Threshold 4



(e) SRG for Threshold 5

Figure 6.11: Results of SRG for Image in figure 6.8

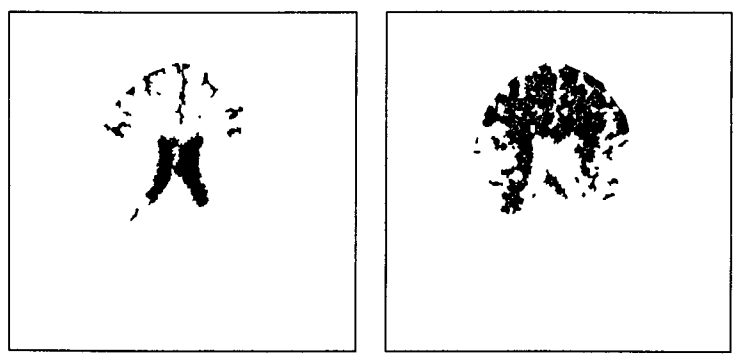
of the algorithm performance. A visual comparison seemed to indicate that the thresholded image produced a better segmentation. If this was the case, why would one need to use the SRG? It was however found that for the CSF, Cyst and other abnormalities, the thresholding had resulted in a smaller volume and incorrect spatial segmentation. The reason for the smaller volume could be attributed to the fact that not all the grey-levels were being used in the threshold band. Trying to modify the thresholding/peak finding process, so that segmentation could be done by that stage alone, did not work. Thus, the use of thresholding was only considered in the pre-processing stage.

On the whole, the SRG algorithm provided a good segmentation of the CSF regions, as can be seen by figure 6.11(a) and 6.10(b). The results of the segmentation of grey and white matter however still needed to be analysed. In comparison to the CSF, the expected segmentation of these regions would be less successful. This is attributable to a number of factors:

- The difficulty of placing seeds in fine and sparse regions.
- The inherent problem of image grey and white matter using X-Ray CT (See table 2.2 in section 2.4).
- The large effect of noise on small sparse structures causes difficulty when Region Growing.

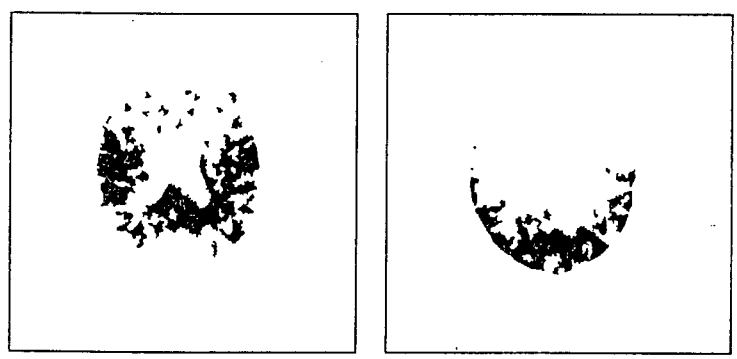
Despite these drawbacks, an attempt to do a partial segmentation of the general region was made. The results were however very difficult to analyse as to hand segment the tissue regions is not practical. From a visual standpoint (which once again should be noted can be subjective, but in this case unavoidable), the results seem to be acceptable, with the major loss of accuracy and mis-segmentation being for fine tissue regions. For these reason it may be better not to use a region grower like the SRG, with limited seeds as starting points for this type of tissue.

The results of the SRG for the 2nd image set closely reflected those obtained for the first set. The lateral ventrical and CSF were segmented with a high degree of accuracy. Some of the smaller CSF regions once again exhibited a degree of mis-segmentation. This was particularly evident in an area of the brain, typically near the skull area, where the CSF percolates upwards into superior sagittal sinus to the lacunae laterals and then drains through the arachnoid granulations. These regions of CSF are not segmented clearly, due to their small thin-like structure. This, like the for tissue regions, can be attributed to the difficulty of placing seeds in these areas. The rest of the results 6.12(b) to 6.12(d) are for the grey and white matter. In this image set, there appeared to have been some drainage to the base of the skull, probably due to excess contrast media. This caused an increase in grey level values in this area. This is evident in the threshold obtained in figure 6.9(e). The grey matter 6.12(b), as before, was less



(a) Lateral Ventrical and CSF

(b) Grey Tissue matter



(c) White Tissue matter

(d) Higher level tissue, probably due to contrast drainage.

Figure 6.12: Results of SRG for Image in figure 6.9

accurately segmented and is probably useful for following trends (Pfefferbaum[3]). The white matter 6.12 had results similar to that of the grey matter. It once again should be noted that the segmentation and even identification, of the white and grey matter in X-Ray CT images, is sub-optimal. This completes the hand segmented results (volume comparison), as well as the visual comparison of the original and segmented images. The next section measures the success of the algorithm with respect to pre-defined phantom objects.

6.4.2 Pre-defined Phantoms compared with SRG Segmented Results

Inititially, tests done using phantoms were done on individual slices. This was done by adding circular phantoms of various sizes, mean grey levels and noise variances to the image. The problem with this approach was that it did not sufficiently model the physical situation. In reality, the SRG algorithm is required to segment a three dimensional object that exists at various slice levels. To do this the phantom developed needed to be three-dimensional and incorporate, to some degree, the image formation process. The description of the phantom and how it is placed in the X-Ray CT volume, is given in chapter 7. The reason for this, is that chapter 7 introduces the volume formation process and elaborates on partial voluming, which influenced the formation of the phantom. All that needs to be known regarding the phantom in this chapter, is that it was inserted into a volume, and from this volume each slice was extracted and processed.

The characteristics of the phantoms were adjusted to simulate varying conditions that could exist in the images. The three main variables were the mean grey level, the size of the phantom and the noise variance. The mean grey level determined how close the phantom was to the main lobe in the histogram. The closer the two means, the worse the expected segmentation. The phantom chosen was a sphere, thus the size was adjusted by varying the radius. The larger the radius, the better the expected segmentation (A sphere was chosen as it represented a general anatomical obejct). The noise was zero mean Gaussian and was justified in chapter 5. By adjusting the variance, the shape of the phantom grey level distribution could be varied. For a large variance, the distribution is more spread, causing overlapping in the image histogram and therefore a worse segmentation.

In calculating the performance of the SRG algorithm, two errors were measured, the first was the difference in volume and the second the overlap difference. The overlap difference was found by doing a grey scale exclusive or (xor) on the original and segmented phantom. The reason for this was to measure whether the SRG is spatially accurate. For example, if the segmentation had the same volume, but the shape (spatially) was incorrect, the volume difference would be zero and the overlap difference high. It was however found that if the segmentation under segmented, the segmented volume lay within the phantom. Conversely, if there was an over segmentation, the entire phantom was always included. This voided the

need for a spatial error analyis and thus it has not been included.

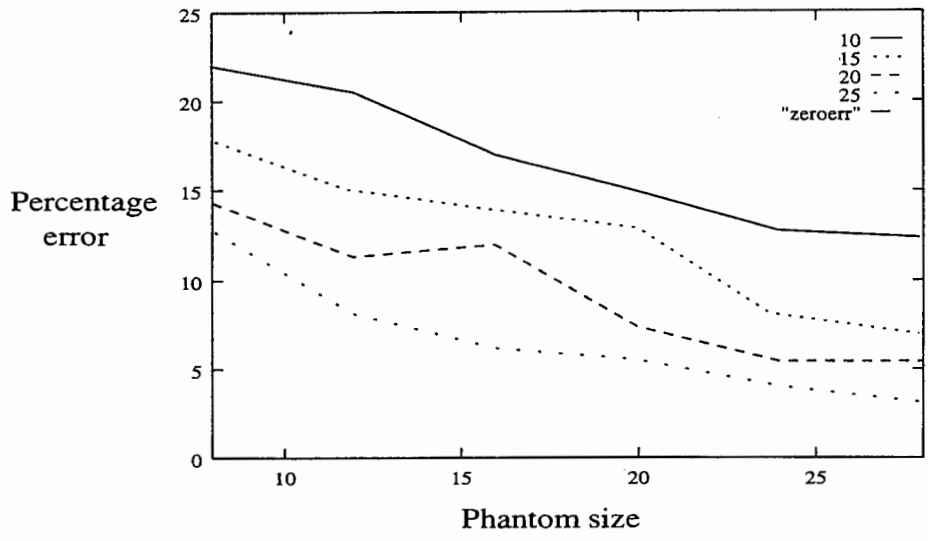


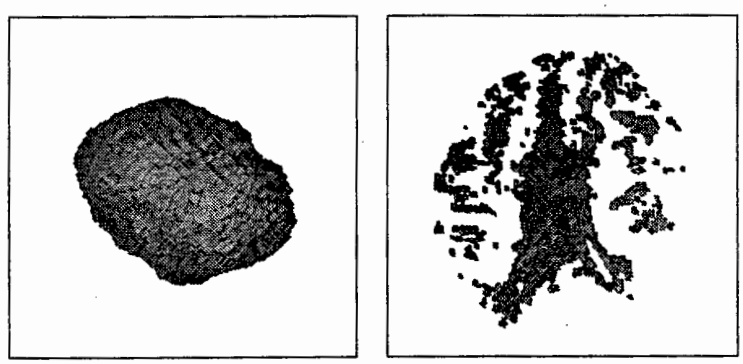
Figure 6.13: Curves of size versus Percentage error for varying mean grey levels.

The experiments were done for mean grey levels of 10 to 25 grey levels from the mean tissue value. The radii varied from 8 to 28 voxels in size and the noise variances from 30 to 130. The step sizes for the three were 5, 4 and 20 respectively. Figure 6.13 gives the percentage error versus phantom size for varying mean grey levels (The value for each point was averaged over a noise range of $\sigma = 30$ to $\sigma = 130$). A full set of results is given in appendix C. From the graph, it is clear that the error decreased as the phantom size was increased. From this graph it can be concluded that an anatomical object that encompass a circular area of radius greater than 12 can be segmented. The error for this was found to be less than 15%. This was however restricted to phantoms that have a mean grey level distance of 15 or above. Further results for the two-dimensional phantom results are given in chapter 7, where a comparison between the two and three dimensional was done.

6.4.3 Three-Dimensional Display of Results

To visually stress the results of the SRG algorithm and to get an idea of the anatomical structures being segmented, a three dimensional volume was formed. This was done by combining the relevant 2-D structures in each slice to form a volume (The z-axis was increased by a multiple of 4 to ensure that the depth perspective was correct). The volume was then rendered using the medical imaging package 3DVIEWNIX, which is specifically designed for viewing 3D anatomical structures. The volumes for the above test sets are given in Figures 6.14(a) and 6.14(b). On displaying the results to neurologists, it was concluded that a good

and reasonably accurate description of the structures were given. Figures 6.14(a) and 6.14(b) represents the cyst and CSF (ventricles) respectively. The angle of display has been chosen to emphasise the discontinuity caused by the slice by slice processing of the images. Although this is not immediately present in ventricular system, on close inspection the effects can be seen. These effects lead one to speculate whether the results of the slice by slice processing, by hand or automated, are accurate (The effect is obviously compounded by the formation of the data set being a two-dimensional and not three-dimensional. Processing in 3-D did not however cause as large discontinuities, which would tend to indicate that the 2-D processing rather than the image formation is the dominant factor)



(a) SRG 2-D - Stacked Cyst Volume

(b) SRG 2-D - Stacked CSF Volume

Figure 6.14: Three dimensional display of the segmented slices

Chapter 7

Seeded Region Growing - Extensions to 3-D

As was shown in the previous section, the SRG algorithm, with accurate seed placement is an effective method for segmenting homogeneous regions in CT x-ray brain images. This was especially true for the CSF, ventricles, various tumours, lesions and cysts. The next stage was to extend the SRG to higher dimensions, in this case 3-D. Before outlining the implementation of the 3-D algorithm the reasons for extending to the higher dimension with its increased computational and memory requirements needs to be justified.

7.1 Motivation for the 3-D SRG

The main reason for moving to a higher dimension can be realised when analysing the human operator's role in segmenting an object. The brain is a complex 3-D structure, and when scanned axially and converted into a 2-D image, much of the 3-D knowledge is lost. This loss of dimensionality can result in an incorrect segmentation by the operator. An example of this is shown for the object given in Figure 7.1. Figure 7.1(a) shows a 3-D *horse-shoe* shaped object that is tapered on the left hand side. A dotted quadrilateral gives the extents of an imaginary plane that cuts through this object. The plane is parallel to the xy-axis. The resulting slice when viewed looking into the xy-plane from the z-axis results in the 2-D image in figure 7.1(b). In this 2-D slice, object-II (the large object) will be segmented easily, however object-I has the possibility of being missed. Under normal conditions an experienced operator could use his or her priori-knowledge of the brain to assume which parts are connected. However, under abnormal conditions it would be virtually impossible to pick up such connected objects. As many of the circumstances under which CT is used are abnormal, the need for a method which can link seemingly unrelated objects together is crucial (It could be possible that later on the

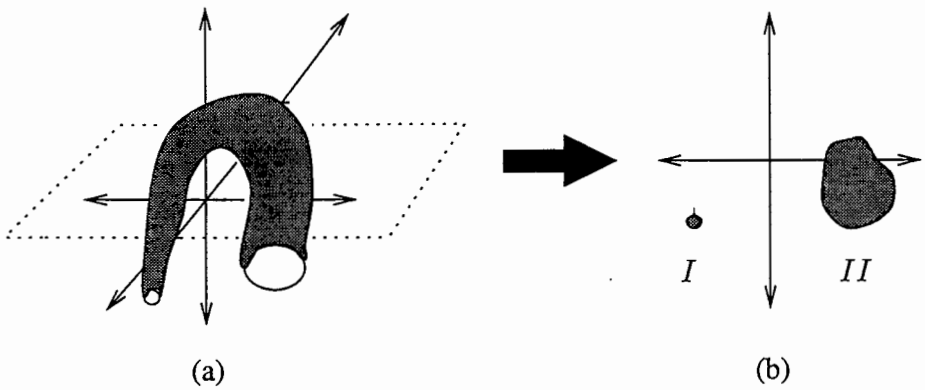


Figure 7.1: Example of Complex 3-D Object

operator would notice that the objects were the same, but this would require re-segmentation of the previous slices. This is both time consuming and tedious). A real example of this type of problem was shown in Figure 6.8(c) where, although correctly segmenting the large portion, both the SRG 2-D and human operator failed to segment a small portion of the right hand side of the lateral ventricle.

The three dimensional version of the SRG was able to cope with problems like this, as it integrates depth information (the z direction) into the growth of a region. Thus, the chances of an object that is small or obscure in a two dimensional image (but connected to a larger object in three dimensions) being connected to the initially seeded object was much greater. This essentially means that the 3-D SRG grows regions that have been disconnected due to the nature of 2-D slicing. It can also do this by growing around the discontinuities in the volume itself. An example of this type of discontinuity are pipes used for inserting contrast media or draining fluid from the intracranial volume.

Leading directly from this, another advantage when using the 3-D SRG is that no post-processing is required to join segmented regions. In the two dimensional case each slice is segmented separately, which causes a problem when the volume of an entire structure is required. The reason for this is that the segmented regions in each slice may have different labelled values. The different labels result from the number of threshold bands obtained during the preprocessing stage. As the number of thresholds in each slice varies, the assignment of labels also varies. One solution for the 2-D SRG was to post processes the labelled regions in the consecutive slices. This was done by comparing their mean values and spatial correspondence in the original slices. Another solution was to let the operator decide which regions in successive slices are related. Both of these methods added unnecessary time and complexity to the problem. By the nature of the 3-D region grower, the structure is already in a volume form and thus no post-processing program or operator is required. This considerably speeds

up and simplifies the overall running of the system.

Another advantage of processing in three dimensions, is the accuracy of seed placement. It is often difficult, in a 2-D image, to find a homogeneous place to put a seed. Using the entire volume, the probability of finding such a seed position is more likely. This was once again due to the higher dimensionality. Thus, to some extent the 3-D SRG placed less stringent requirements on seed placement.

7.2 Volume Formation

Before any processing can be done, a 3-D volume needs to be formed. For the majority of scanners presently available the standard methods used involves the combining of consecutive image slices Lin[49]. This section first gives the considerations and assumption that are used when combining or *stacking* the slices. It then gives methods used to combine the slices, thus forming the volume.

7.2.1 Considerations and assumptions when forming the Volume

For the scanner being used during this dissertation the image slices were taken axially at a resolution of $0.47mm \times 0.47mm \times 2mm$. The process for obtaining slices at different depths involved moving the bed (on which the patient lies) in or out of the scanner. The accuracy of the positioning of successive scans relied on two major factors. The first was the accuracy of the movement of scanning bed. The second and by far the more serious problem was that of patient movement. Both of these problems were essentially unpredictable and unavoidable (The machine error could be reduced through accurate calibration, however this was often not possible due to high usage).

The next immediate problem is that of the slice depth and the partial volume effect. A brief description of this effect was given in section 5.2. A more thorough description and its impact on volume formation will now be given. Partial voluming occurs when a single slice contains a transition from one tissue type to another. Figure 7.2 shows a cross-sectional slice through two imaginary regions. The block enclosed by the dashed lines represents a single voxel, with Δz being the slice width (voxel depth) and Δx the voxel width (The voxel height Δy usually equals Δx). It is clear that there are two different grey levels contained in the voxel and as only one CT value can be assigned to each point, there is an inaccuracy in the reading. The larger the volume of the voxel (ie: greater Δx, Δy or Δz) the greater the error is likely to be. The voxel depth for the scanner used was roughly 4 times greater than the height or width. This meant that the partial voluming was much greater in the z direction than in the x or y direction.

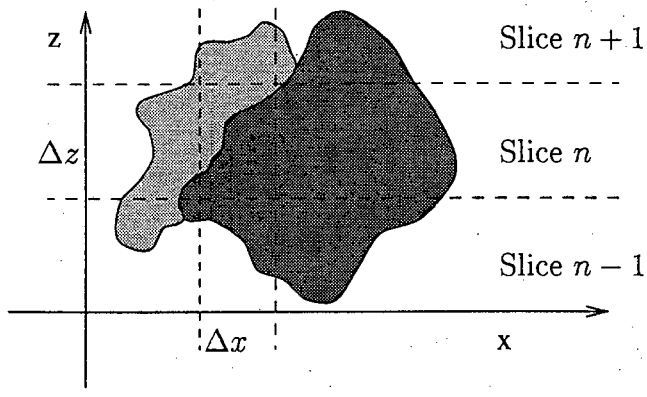


Figure 7.2: Description of a Partial Volume

Taking into account the above problems the following assumptions regarding them (and thus on forming a 3-D volume) were made:

- The scanner accuracy did not cause any mis-alignment between successive slices.
- Patient movement was negligible and did not cause any mis-alignment between successive slices.
- The value of any voxel was assumed to be the average of the physical regions contained.

These assumptions enabled the volume to be formed by combining consecutive slices. This could be done without worrying about the actual alignment of each slice with the ones above and below. There are however many ways to actually combine the slices. The objective of most these methods is to resize the voxel dimensions so that they are equal. This involves some sort of interpolation and will be given next.

7.2.2 Using interpolation to resize voxels

A major problem in X-Ray CT is the finite slice thickness available from a particular scanner. The thicker the scan, the greater the partial voluming and the less the accuracy. An immediate question that can be asked is if there is some way to improve accuracy of measurements/segmentation without altering the scanner. Lancester [50] used a geometric model to improve volume measurements of objects from tomographic images. These models are however based on an already segmented volume (in this case using the marching cubes algorithm) and thus are limited by the accuracy of the segmentation itself. To improve the accuracy of the segmentation (be that for the marching cubes algorithm or SRG) an investigation into the results of interpolating the data in the z direction was done.

Types of Interpolation The simplest method was to *expand* the depth of each voxel. This involved replacing each slice with dimension $0.47mm \times 0.47mm \times 2mm$ with 4 duplicate slices. This has no effect on the results of an segmentation that uses non-interpolated volumes (Although dealing with a cube voxels was easier). Interpolation schemes used which altered the data included Polynomial Interpolation, Rational Function Interpolation and Cubic Spline Interpolation. The details of these methods can be found in Press[pg.85-104][51]. The effect of the interpolation on the volume formed and the consequences to the segmentation are given next.

Reason for Interpolation - Advantages for X-Ray CT: The main reason for interpolation the was the following: Under the assumptions made above, the value of a voxel is formed by the average of the regions contained. Considers the boxed voxel in Figure 7.2: it will have a grey level value somewhere between the two regions. This would result in a voxel lying in the low confidence portion of the histogram shown in Figure 7.8. If one now considers the voxels in the slices above and below it (in the z-plane), they give some indication of the voxel tissue distribution between them. In figure 7.2 the above voxel was indicative of the tissue in the upper part of the centre voxel and the pixel below the lower part. If this information is used, the result of interpolation has the potential to partially restore the gray level distribution in the z-axis. This could result in a more accurate segmentation.

Another potential advantage is that a misclassification in the growing stage would result in a lessened volume error. This would be due to the smaller voxel size, with the reduction in error being proportional to the amount of interpolation. This may lead to the conclusion that infinite interpolation would result in infinite error reduction. This would not be the case, as after a certain degree of interpolation the grey levels (z-axis) do not change and would always be assigned to the same region.

Problems resulting from Interpolation: Although interpolating between planes of voxels produces a more visibly acceptable result, it sometimes can cause other problems. The most significant of which are the creation of apparent structure and the ignoring of real structure (Russ[43]). Another problem with interpolation occurs when the slice images are not accurately aligned. This will cause interpolation between different regions to be done. Further information related to interpolation of image slices is given by Cline[52, 53].

Results The result of interpolation for the cyst in Image Set 1 was a volume of $57.293cm^3$. This is marginally smaller than the SRG 3-D volume given later in Table 7.3. The SRG 3-D volume was $58.16cm^3$ which is $0.867cm^3$ larger. These results were echoed by the interpolated phantom studies, which showed that the interpolated SRG 3-D volumes lay between the SRG

2-D and non-interpolated SRG 3-D. This meant that the interpolated results were slightly more accurate than non-interpolated results. The increased accuracy however required a major increase in the memory (Section 7.6.1 gives an idea of the memory requirements for the standard SRG 3-D, this would be multiplied by 4 for the interpolated case) and significantly increased the running time of the algorithm. This minor increase in accuracy and the added requirements resulted in the interpolation not being feasible.

7.3 Development of a new phantom to account for Partial Voluming Effect

To evaluate the effect of any segmentation algorithm, phantom objects needed to be used. A number of different phantoms were tried and the major problem was the degree to which they modelled the real objects with respect to image formation. The more realistic the model, the more useful the results derived from its use. The simplest phantom was formed by adding an offset to the original image. Thus given the main tissue grey level mean T, a phantom lesion was created by offsetting an area of fairly constant T by a constant C. The first problem with this was identifying a good value for C. In practice C was selected such that $T - C$ lies between the CSF value and the main tissue mean. The second problem was that the Partial Volume phenomenon, which is occurs at the boundary between objects, is not adequately dealt with by using a simple offset.

Sandor[12] attempted to give a better model for the simplified case of segmenting CSF from Grey and White Tissue matter (Although the method can be generalised to more than two regions). Again the idea was to offset a constant C from each pixel of the phantom lesion. However, in his model the size of the constant was related to the level of the *estimated* Partial Voluming present, according to the following formula:

$$PV(\%) = 100 * \frac{C}{T - F} \tag{7.1}$$

where PV denotes the estimated Partial Voluming effect and T and F are the mean intensities of the Tissue and CSF pixels, respectively. Using this formulation an increase in the Partial Volume corresponds to an increase in the percentage of CSF in the voxel. Although this attempted to incorporate the Partial Volume effect, equation 7.1 only relates it to being a fraction of the mean region differences. This did not model the actual image formation process and thus does not effectively model the partial voluming effect. Another problem, which related to both methods described above, is the finding of an area of constant T. If any other region (say T_o less than T) is in that area that C is being offset from, the grey

level for those pixels becomes: $T_{new} = T - T_o - C$, which will not occur in a real lesion. This problem is exacerbated in the 3-D case where a large volume of constant T (to test for a large phantom) is unlikely to be found.

Due the lack of reality in the above mentioned models, a new phantom needed to be designed. As the Partial Volume is an interface problem between regions, it would seem logical to calculate the part of each region contained in the pixel or voxel. Doing this and using the assumption in Section 7.2.1 the correct grey-level for the boundary voxel can be found. Thus, if a voxel contains n regions, each with a volume V_n and grey-level G_n, the grey-level value for the Voxel is given by:

$$Vox = \sum_{i=0}^{n} V_n G_n / \sum_{i=0}^{n} V_n \tag{7.2}$$

Calculating V_n for the phantom was done using two methods. The first was an mathematical approach and the second a practical and realisable approach. The mathematical approach involved calculating the intersection between the equation for the phantom (in this case a sphere) and a 3-D voxel grid.[1] The percentage volume of phantom in each voxel was then found. The problem with the mathematical approach was the requirement that the phantom must be mathematically definable. This was often not possible and restricts the phantoms available. (The degree of complexity was also a limiting factor to the choice of function.) For these reason the mathematical approach was not used and the practical solution implemented.

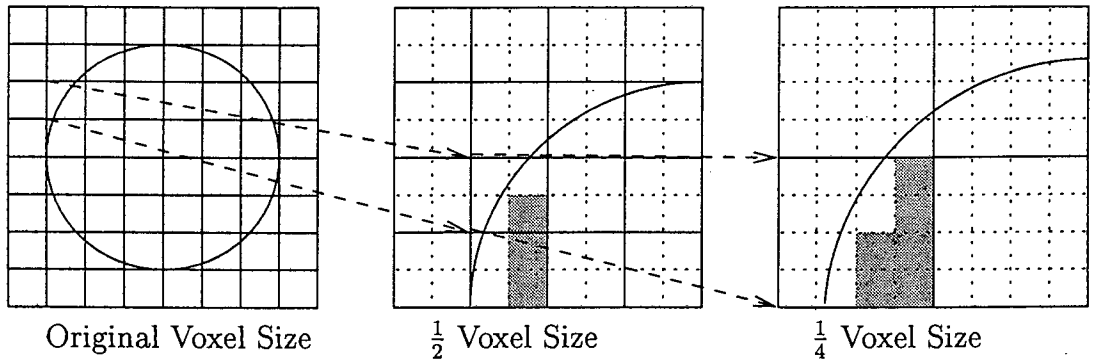


Figure 7.3: Calculation of a Partial Volume

The practical approach involved plotting the simulated lesion on a finer grid. The fine grid was then resized to the original input resolution. Figure 7.3 gives an example of how this fine grid was used to calculate the Partial Volume of the edge voxels or pixels. The first

[1]The grid dimensions correspond to the scanner slice dimensions.

grid represents a slice through a spherical region plotted at the resolution of the input scan. The second grid represents the same phantom plotted on a grid twice as fine (double the accuracy). The last circle is a plotted with grid four times as fine. In all of the figures, the solid boundary blocks represent a single voxel, thus as the grid gets finer, a single voxel in figure one is represented by 16 in figure 3. To calculate the partial , the sub-sampled voxels in each block are added up. It is clear for the first grid that the Partial Volume cannot be calculated (Using equation 7.2 it will always return 100%). The Partial volume for the second and third grids are 25% and 37.5% respectively. Doubling up the grid further return a value of 48.37 %. To ensure the accuracy of the Partial Volume, a grid eight times finer than the original was used. The procedure was repeated for every voxel in the volume and a new volume consisting of Partial Volume Percentages formed.

Adding the phantom to the CT image or volume was considered next. The placing of the phantom should not be dependent on the volume. This meant that the positions chosen should not be restricted to roughly constant grey level as was required by Sandor[12]. The phantom must however, still maintain the original noise characteristics of the volume. Using these criteria, the grey label of each voxel making up the phantom was calculated as follows:

$$G_{ph} = I_{ct}(x,y,z) * (1 - PV(x,y,z)) + P_{gr} * PV(x,y,z) + \eta \tag{7.3}$$

where I_{ct} is the input voxel value, PV is the Partial Volume phantom voxel value, P_{gr} is the chosen grey level for the phantom and η is the additive noise. The noise variance is obtained by calculating the variance in a roughly constant portion of the image. This is not generally part of the patient, but rather the background. This differed considerably from the methods given above, as the partial volume was not determined by the mean grey level of the phantom, but by the percentage of region contained in the sampled voxel CT grid.

The advantage of the method is that the phantom is based on **true** partial volumes and not some assumed grey level. This is because the CT slice thickness which, in essence, determines the degree of partial voluming has been included in the modelling.

The drawback of this model is mostly due to the memory requirements. For example, a sphere of radius $11.25mm$ computed on a grid of 16 times the resolutions, resulted in a volume of size $384 \times 384 \times 384$ voxels ($24 \times 24 \times 24$ voxels at the original CT resolution). The partial volume is stored as float values, with 0 for no part contained in the voxel, to 1 for the entire voxel being contained in the phantom. If one computed the partial volume model for this example, the total memory required would be 216 Mbytes. This was not available on the equipment being used and a more efficient method needed to be found. The solution was to deal with each slice individually. This meant, for the example above, a volume of size 0.84 Mbytes,

($24 \times 24 \times 384$) and a slice of 0.56 Mbytes. The partial volume can the be calculated by using 7.2 in the z direction. This substantially reduced the memory requirements and made the use of this phantom definition feasible.

7.4 Three-Dimensional Seeded Region Growing Algorithm

The Seeded Region Growing algorithm described in section 6.1 was initially implemented directly to three-dimensions. This was done without any change to the structure of the existing algorithm, except that growth into the z direction was allowed. The results of this conversion were not as good as expected. The main problem was not that voxels were incorrectly assigned to regions, but that too many voxels were being marked as boundary points. The reasons for this occurring and why these problems were not as prominent in the two-dimensional case, are given next.

7.4.1 Problems with direct conversion of SRG from 2-D to 3-D

The major factor was the higher inter-connectivity in three dimensions. While this has its advantages (see section 7.1), when looking at the following step in the SRG 2-D algorithm: **If all neighbours of the pixel being checked are the same value, then label the pixel with that value, else label it as a border pixel**, the following problems arose. In the 2-D case there were only 8 neighbours, this meant that the boundaries between regions could only be a single pixel deep. The boundaries that result from the rule are thus simple, single pixel linked curves. This allowed for easy post-processing when joining regions grown from seeds from the same threshold band (This is done by simply following the curve connecting those regions that come from the same threshold). It also results in an extremely fast implementation of the 2-D algorithm. Thus, in the 2-D case, allowing each seed placed to grow into an individual region and forming borders with regions grown from seeds of the same and opposite thresholded regions, was acceptable.

However, when using the direct SRG implementation in the 3-D case, if a region was one voxel away from another region, then all 9 surrounding voxels would be marked as a border points. This was exacerbated by the fact that one region can have many seeds, causing the same region to be split by a border. This and the high degree of interconnectivity in 3-D, resulted in a large percentage of the voxels being labelled as borders. This was especially apparent as the borders were no longer simple 2-D curves, but rather surfaces, often three voxels deep. Initially a similar type of post-processing which assigned the border pixels to regions was attempted. However, post processing the border surfaces proved to be far more complicated and before attempting any implementation, methods that adjusted the workings of the SRG

were first tried. These methods were based on not having voxels classed as boundaries, but rather merging regions. This required methods to deal with regions from different or the same threshold bands[2]. The solutions are given next.

7.4.2 Dealing with regions from the same Threshold Band

The first change that was made to the SRG was to allow regions grown from seeds from the same threshold band to merge. To do this required that each region stored which threshold band it was originally derived from. Thus each region stored the following information:

- The original threshold label it was derived from.
- Its starting label. Each seed has an unique starting label.
- The mean value of the region at any stage of the growing.
- The size of the region at any stage of the growing.

The procedure for adding to a region was then as follows: A voxel was removed from the SSDS (sequentially sorted data structure) and the 27 neighbouring voxels were checked. If the voxels were either from the same region or had not been assigned, then the voxel being checked was added to that region. If there were voxels from two or more different regions, then their originating threshold band was checked. After this check, if all the regions were from the same band, they were merged. The case when regions originating from different threshold bands are found is dealt with later.

The merging procedure worked as follows: The regions to be merged (there is a theoretical maximum of 8) were first sorted according to size. Merging required changing the label values of all the regions to a single value. As this involved iterating through all the voxels of a region, it was decided to have all the regions take the label of the largest region. This meant that the largest region was never iterated through. The method used to convert the label was simply to re-label the region (This method is different to the labelling algorithm given in chapter 6 and is given later in section 7.5.3). Once the two regions are merged in the volume, the new mean was calculated according to the following equation:

$$Mean_{12} = \frac{1}{NoVox_1 + NoVox_2}(mean_1 * NoVox_{reg1} + mean_1 * NoVox_2) \tag{7.4}$$

where $NoVox$ is the number of voxels in region. This was repeated until all the regions were merged. All new voxels bordering the merged region use this mean in calculating the error.

[2]The threshold bands are those selected by the peak detection stage.

The above merging, where the surrounding voxels are either of the same threshold band or unlabelled, did not contribute extensively to the problem described above. The problematic case is when there was a mix of regions from different threshold bands.

7.4.3 Dealing with regions from different Threshold Band

Staring once again from the point where the voxel was removed from the SSDS and the 27 neighbouring voxel are checked, if there were voxels from different region the following steps were taken: The error (Equation 6.3 in section 6.1) between each of the different regions and the voxel being considered was found. The voxel is added to the region with the minimum error. Once the voxel has been labelled, the surrounding regions are checked to see if any mergers need to take place (ie: This is required as two of the three regions maybe grown from the same threshold band).

An immediate problem with this method, was what to do when the error between two regions from different threshold bands was very small (The minimum error size can be set by the user). This generally occurred when voxels added lay in the low confidence intervals shown in Figure 7.8. To simply add a voxel into these regions based purely on the mean error would often lead to an incorrect voxels added. A solution was to take the neighbouring voxels into account. This was done by a majority rule system using a 3×3 kernel surrounding the voxel being considered.

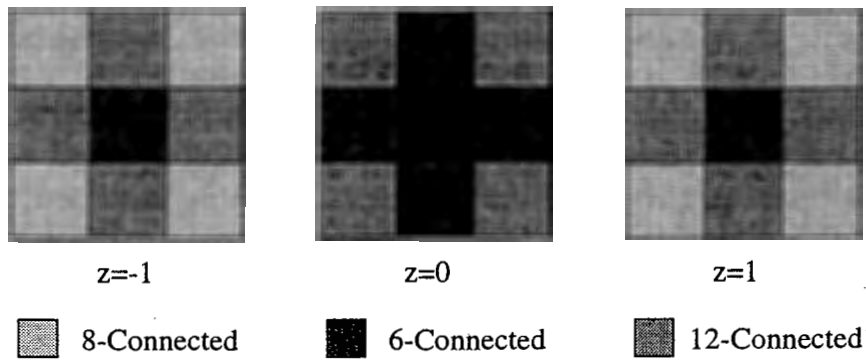


Figure 7.4: Connectivity definitions for three dimensional points (voxels)

The kernel has value $\frac{1}{2}$ for the 6-connected voxels, $\frac{1}{4}$ for the 12-connected voxels and $\frac{1}{8}$ for the 8-connected voxels. Figure 7.4 describes 6-, 8- and 12-connectivity for 3-D voxels. For each of the different regions present, their connectivity with the voxel being considered is calculated. The kernel values biases the voxel towards regions with a larger connected area. The surface connectivity for region i is defined as $\zeta(i)$ and calculated as follows:

$$L(x_0, y_0, z_0, i) = \begin{cases} 1 & \text{if } f(x + x_0, y + y_0, z + z_0) = i \\ 0 & \text{otherwise} \end{cases} \tag{7.5}$$

$$\begin{aligned} \Gamma_6(i) &= L(0,-1,0,i) + L(0,1,0,i) + L(1,0,0,i) + L(-1,0,0,i) \\ &+ L(0,0,1,i) + L(0,0,-1,i) \end{aligned} \tag{7.6}$$

$$\begin{aligned} \Gamma_8(i) &= L(1,1,1,i) + L(1,-1,1,i) + L(-1,1,1,i) + L(-1,-1,1,i) \\ &+ L(1,1,-1,i) + L(1,-1,-1,i) + L(-1,1,-1,i) \\ &+ L(-1,-1,-1,i) \end{aligned} \tag{7.7}$$

$$\begin{aligned} \Gamma_{12}(i) &= L(0,-1,-1,i) + L(0,1,-1,i) + L(1,0,-1,i) + L(-1,0,-1,i) \\ &+ L(0,-1,1,i) + L(0,1,1,i) + L(1,0,1,i) + L(-1,0,1,i) \\ &+ L(-1,-1,0,i) + L(1,1,0,i) + L(-1,1,0,i) + L(1,-1,0,i) \end{aligned} \tag{7.8}$$

$$\zeta(i) = \frac{1}{2}\Gamma_6(i) + \frac{1}{8}\Gamma_8(i) + \frac{1}{4}\Gamma_{12}(i) \tag{7.9}$$

The voxel is joined with the region that has the highest connectivity ζ. If the surface connectivity is equal for two are more regions, ζ is recalculated using Γ_6 and Γ_{12}. If the results are still equal then Γ_6 is used. In the case still no result, the voxel is unassigned and marked as such. Once the voxel has been joined to a region, a check is done to see if any mergers are required (e.g. If the added voxel has connected two or more regions). The merging of these regions follows that outlined in section 7.4.2.

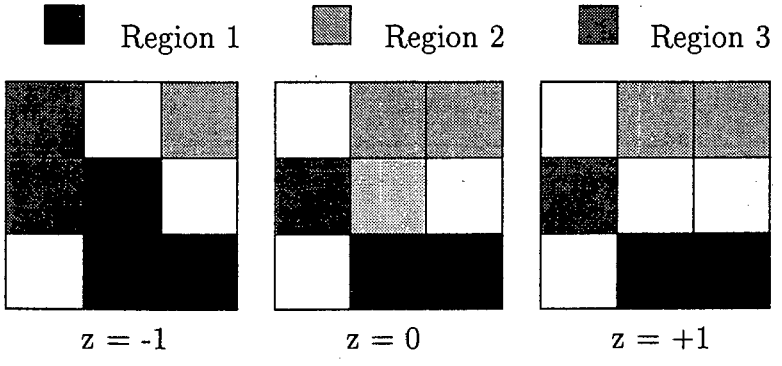


Figure 7.5: Calculating Surface Connection

Figure 7.5 depicts an example of finding the surface connectivity. The voxel being considered is in the centre of the $z = 0$ slice. The $z = -1$ and $z = +1$ slices lie below and above this slice respectively. Assuming that the error difference between the regions is too small to confidently assign the voxel being considered, the surface connectivity can be calculated. Using the kernel described above, the results obtained for the figure in 7.5 are given in Table 7.1. For this example, it is clear that the voxel will be merged with region 1.

Region No	6-Connected	8-connected	12-connected	Surface Connectivity
1	2	3	2	1.875
2	0	3	2	1.000
3	0	3	1	0.875

Table 7.1: Results for Figure 7.5

The final algorithm after incorporating the surface connectivity criterion is given in 7.6.

7.5 Selection of Seeds in three-dimensions

As was stated in section 6.2, the correct choice of the homogeneity parameter is critical in achieving a good selection. For region growing the homogeneity parameters are the seed mean values, and the requirements for the SRG 2-D hold true for the extension to three-dimensions. The selection of these seeds mirrored the method given in section 6.2, and a brief overview are method is given next:

1. Fit Gaussian to Grey Level Histogram distribution.
2. Subtract Gaussian from Grey Level Histogram distribution.
3. From subtracted Histogram, find *rough* grey Level bands corresponding to objects in brain volume.
4. Threshold the Volume according to the bands found in step 3.
5. Use blocking to find percentage coverage of each of the thresholded bands.
6. Seeds positions are those with the highest percentage coverage.

For the 3-D implementation, step 1 had to be analysed to see if a direct implementation could be done. This is given first and the results show that it was still applicable. Slight modifications were done to the Filtering of peaks (section 6.2.3) and will be repeated for

Calculate Starting Set Values from initial groupings.
Label Starting Seed points.
Place neighbours of seed points (the initial T) in the SSDS.
While the SSDS is not empty
  Remove first item y from SSDS.
    Test the Neighbours of this item:
      If all neighbours of y which are already labelled
      have the same label or are from the same starting band:
        Set y to this label.
        Merge groups that have same starting band.
        Update running mean of corresponding region.
        Add neighbours of y which are neither already set
        nor already in the SSDS to the SSDS according to
        their value of δ.
      Calculate the error δ of voxel y from each of the
      neighbouring regions.
      If the error is too large or the difference between
      the minimum and the rest to small, then calculate the label
      according to the maximum surface connectivity, then:
        Set y to this label.
        Merge groups that have same starting band.
        Update running mean of corresponding region.
        Add neighbours of y which are neither already set
        nor already in the SSDS to the SSDS according to
        their value of δ.
      otherwise, choose voxel label according to the error δ

Figure 7.6: The Seeded Region Growing Algorithm

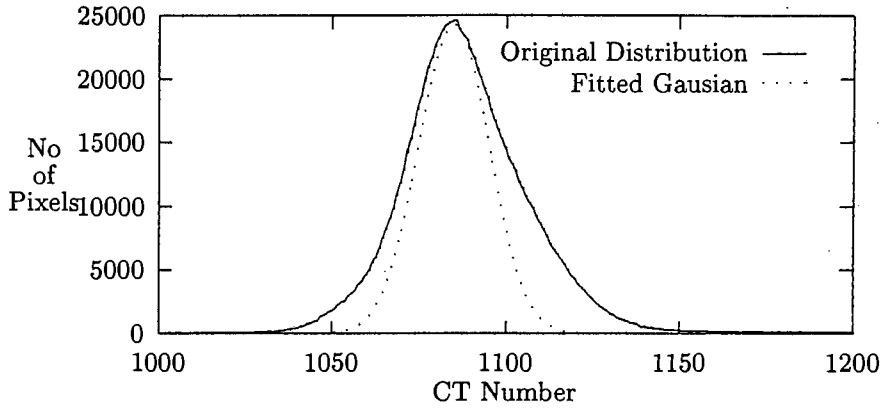


Figure 7.7: Fitted Gaussian for Volume Histogram

clarity. This ensured that steps 2 to 4 could be used without any change. The blocking stage needed to be changed to cope with the higher connectivity of the 3-D volume. The evaluation and changes made to the seed selection process are given next.

7.5.1 Evaluating the Volume CT Grey-Level distribution

The first consideration when implementing the 3-D SRG related to the volume histogram. As described in section 6.2.2, for each slice the histogram is found and Gaussian fitted to, and then subtracted from the main lobe. From the resulting distribution, a series of threshold bands was found. The questions regarding the 3-D implementation that needed to be answered were:

- How does the histogram distribution alter when done for a brain volume?
- Do the methods for calculating the grey level bands described in section 6.2.3 still hold?
- Will the thresholds found allow for accurate seed placement?

The change in the distribution from 2-D to 3-D can be described as follows: The volume histogram is essentially the sum of the individual distributions. Each of these distributions contains, to a varying degree, the grey level distributions of each tissue type. By adding these distributions, the effects of noise are effectively reduced. However with the volume, the dominance of the main tissue lobe over other CSF or abnormal tissue types becomes greater. This should not pose a problem if the mean grey levels between the different objects is sufficient. If the difference is small, the volume histogram would tend to obscure it more than individual slice histograms.

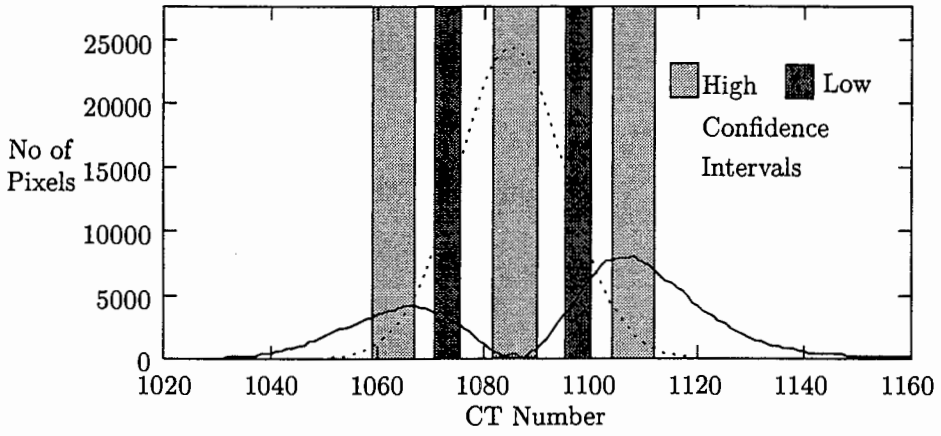


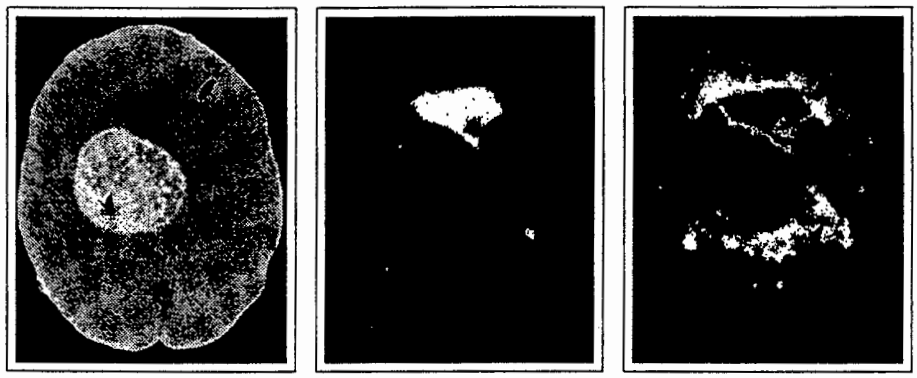
Figure 7.8: Subtracted Volume Histogram with fitted Gaussian

With this in mind, it needed to be established if it was still feasible to use the same method in three-dimensions. As was previously stated, the threshold bands were not used to segment the volume but as a means of finding rough seed locations. This, coupled with the fact that the bands are formed using the high confidence portion of the distribution, allowed for the method to be used. The high confidence intervals are the gray levels from the peak point, to the corresponding grey levels at the 70% of the peak points. These grey levels have a higher probability of belonging to a specific region and by using these interval, the low confidence regions between peaks were not included in the thresholding. Figure 7.8 gives an example of the subtracted distribution with the fitted Gaussian overlaid. In this figure the three high confidence intervals are shown. It is these grey levels within these intervals that are used as the threshold bands.

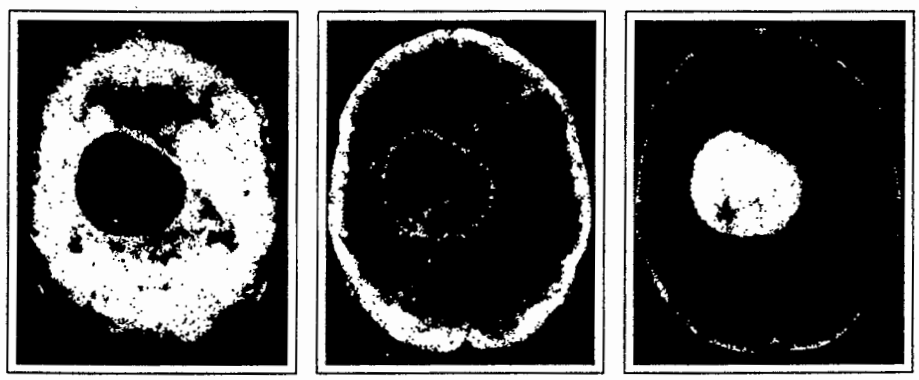
The volume, as for 2-D slices, was thresholded according to the different bands. The results for slices extracted from the thresholded volume are given in Figure 7.11 and 7.10. The slices extracted correspond with the 2-D thresholds shown in Figure 6.9 and 6.8. A Comparison of the two showed that they were very similar. This enabled the same technique to be used in 3-D case. In fact, according to the operator, the 3-D method gave a better rough threshold for the CSF regions. For the rest of the tissue, the difference was difficult to gauge.

7.5.2 Modification to Filtering Peaks and Trough of the subtracted Histogram

The result of subtracting the fitted Gaussian from the original distribution was a set of peaks that represented the previously masked grey level distributions of regions in the Volume. Not all the peaks represented a different region and they needed to be filtered out or merged. This section clarifies and extends the 2-D methods given in section 6.2.3 for the 3-D case.

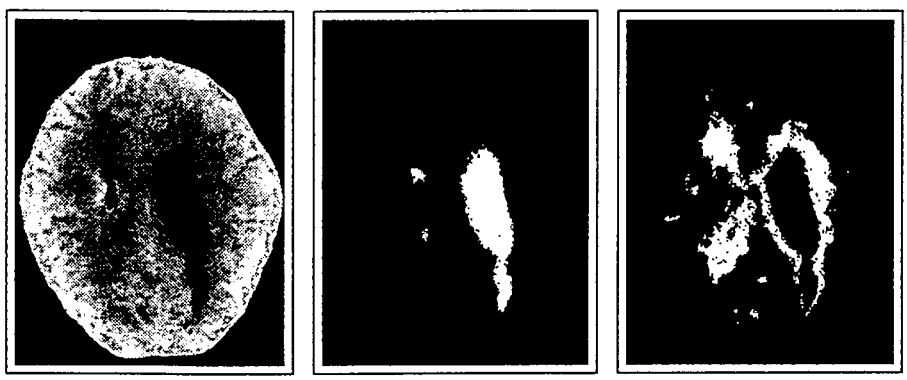


(a) Original Image (b) Threshold 1 (c) Threshold 2

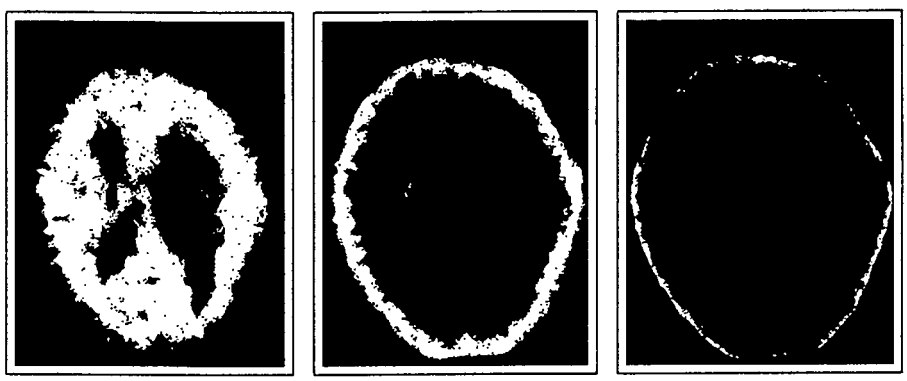


(d) Threshold 3 (e) Threshold 4 (f) Threshold 5

Figure 7.9: Threshold Results for axial slice from Image Set 1 - Low axial slice.

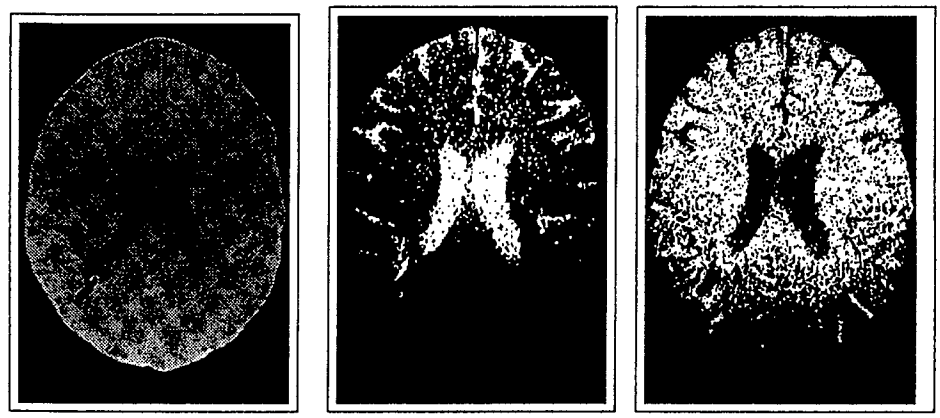


(a) Original Image (b) Threshold 1 (c) Threshold 2

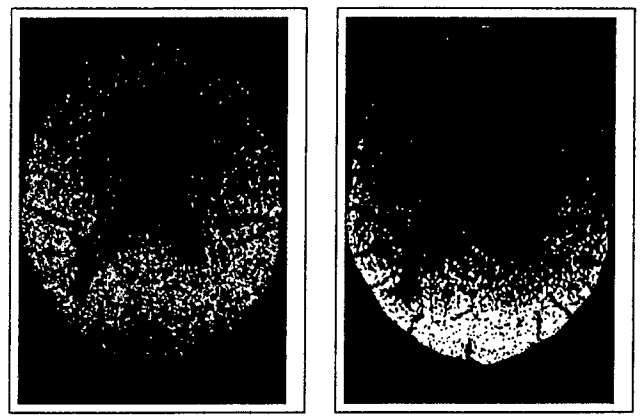


(d) Threshold 3 (e) Threshold 4 (f) Threshold 5

Figure 7.10: Threshold Results for axial slice from Image Set 1 - High axial slice. Corresponds to 2-D thresholds in Figure 6.8



(a) Original Image (b) Threshold 1 (c) Threshold 2



(d) Threshold 3 (e) Threshold 4

Figure 7.11: Threshold Results for axial slice from Image Set 2. Corresponds to 2-D thresholds in Figure 6.9

Section 7.5: Selection of Seeds in three-dimensions

Distance between peaks: If two or more peaks are within a certain proximity to one another, it is unlikely that they belong to different regions. This is based on the minimum distance between mean grey level values of different CT regions. The approach to merging is also based on the change in height between the peaks. Given the set of peaks $P_0, P_1, \cdots, P_n$ and corresponding troughs $T_0, T_1, \cdots, T_n, T_{n+1}$ and taking two peaks P_i and P_{i+m} $(i+m \leq n)$ the peak to trough height ratio PTH was defined as:

$$PTH = \min\left(\frac{P_i}{T_{min}}, \frac{P_{i+m}}{T_{min}}\right) \tag{7.10}$$

where T_{min} is defined as $\min(T_{i+1}, \cdots, T_{i+m-1})$. The PTH was used by comparing it to a predefined table and making a decision on whether the peaks are to be merged.

	Grey Levels Apart				
PTH Value	1-2	3-5	6-8	9-11	> 11
0.91-1.00	√	√	√	√	×
0.71-0.80	√	√	√	×	×
0.61-0.70	√	√	×	×	×
0.50-0.60	√	√	×	×	×
< 0.50	√	×	×	×	×

Table 7.2: Merging criteria based on Grey level distance and height changes between peaks.

Table 7.2 gives an example of the type of criteria used to merge peaks. The closer the grey level values of the peaks, the less the restriction on the PTH value. The merging starts with the highest peaks and mergers recursively until one of the conditions is not met. It should be noted that the peak that is merged is still used to check if other peaks can join the band. The mergers stop only when two adjacent peak are not within the restrictions of Table 7.2.

Constituting region area: To further reduce the number of bands or peaks, the total number of voxels per band was considered. This rates the importance of the band according to its contribution in the image. It was found that if the number of voxels was less than 1152 (block of $12 \times 12 \times 8$), the next stage of the algorithm sometimes did not find accurate seed points. For this reason bands less than 1152 were either excluded or joined to an adjacent peak. It should be noted that this criteria indirectly set the resolution limits of the algorithm. It does this in the sense that any uniform connected region in the output region that is less than the threshold set (say 1152 pixels) will not be segmented.

7.5.3 Modifications to the Volume blocking technique for finding Seed Positions in 3-D

The seed placement technique followed the same procedure as the 2-D SRG, up to calculating the threshold ranges. The main differences are from the volume thresholding onwards. The reasons for not using the same method are based on the difference in structure between an image slice and a volume of data. The changes in three dimensional volume blocking to find seed position were as follows: Once the regions have been thresholded they are labelled. The labelling procedure considered all 8-connected voxels as belonging to the same region and assigns a value for each one. Only regions of a large enough size are considered for further processing. This has the effect of reducing the search space as only the extents of the regions need to be processed and not the whole volume. The choice of minimum region size has been found to be not that critical, as the small non-connected (not part of a larger structure) regions are generally not of any medical use (Sandor [12] has a post processing filter that removes all regions found smaller than 20 pixels in 2-D, this was be extrapolated to about 200 voxels in 3-D).

The labelling technique used is based on a simple 2-D method that has been extended to 3-D. It is proportionally not as fast as the previous 2-D method described above, but is easier to implement and work with in 3-D. The method required two volumes of equal size to the input volume and a stack data structure capable of holding 3-D vectors. The algorithm raster scans the volume until a voxel of value greater than zero is found. This is taken as the starting point of the region to be labelled. The same position in the bit volume and output volume is then marked. The output volume value is set to a region starting number (usually 1). All 26 neighbours in the original volume and corresponding bit volume are then checked. If the neighbour in the original volume is greater than zero, then it is placed on the stack and marked output bit volume. After this, the point on the top of the stack is removed and used as the next starting point. Its position in the output volume is set to the region starting number and the procedure repeated. The only difference now, is that the neighbouring voxels greater than zero are checked against the bit volume. If the voxels in the bit volume are set, then the neighbours are not added to the stack. The labelling of a single region is complete when the stack is empty. The raster scan continues, ignoring voxels already marked in the output volumes. When a new voxel is found, the region starting number is incremented. At the completion of the labelling algorithm, a labelled output volume and a list of region information is given.

The next stage in the seed placement algorithm was to process each region so that the best seed position could be found. The method is based the 2-D case, but due to the higher connectivity a method had to be altered for dealing with very large regions. As before, blocking based on a minimum percentage coverage value criteria was used. However, the

percentage coverage values chosen were always above 90% due to the higher connectivity and the labelled region blocked in relation to the physical dimensions of the voxels. This worked as follows: If an image was scanned at a voxel resolution of $x = y = \frac{240}{512} = 0.47\ \frac{mm}{pixels}$ at a slice width of $z = 2mm$, there is a ratio of $4 : 4 : 1$. When choosing the block value size, it must be proportional to this ratio. Once again, two different block sizes were used. The first aimed to place seeds for small regions and a $2 \times 2 \times 2$ block size was used. The second block size was chosen to ensure that larger regions would be broken up. This ensured a number of seeds being placed within. For this a block size value of $6 \times 6 \times 2$ voxels worked well. As the blocking for each region and at different scales (block sizes) is done separately, the resulting seeds must be checked to ensure that no overlaps occurred. If there is an overlap, the seeds are spatially merged to form a single seed.

Once the region has been blocked and thresholded[3], the labelling process was repeated. The resulting regions were divided into two groups: Those small enough so that single seeds are sufficient, and the rest large enough so that more than one seed would be required. The criteria for deciding on the which region required more than one seed, was based on the blocking dimensions. As a broad estimate, it was decided that each seed should effectively grow ten times its starting volume. This means for a seed volume of 50, $(5 \times 5 \times 2)$, a threshold value of 500 was chosen. The validity of this decision is based purely on a heuristic interpretation of the problem, however it was found to be successful in implementation.

If the region required one seed, then the blocked region (before applying the percentage coverage threshold value) was searched and the point with the highest value chosen as the seed point. The algorithm allowed for the point to be refused if it does not have a high enough value, but due to the nature by which the region is formed, this was not necessary. The search for the highest percentage coverage value works fine for small regions, but as soon as the region is to large, the use of a single seed can result in *under-grown* region. This did not happen if sufficiently spaced seeds are found for the larger regions. The method used for finding these seeds is described next.

Dealing with large regions

The reason for having more than one seed was that more than one seed gave a higher chance of a large object being successfully grown. It also places a less stringent requirement on what otherwise would be a single seed position. The method for finding these positions works as follows: Given the thresholded percentage coverage region and its bounds, a minimum distance between possible seeds needs to be found. This was to achieve an even spread of seeds in the region. This distance is defined by the size of the bounding box of the region and

[3]The threshold value is the percentage coverage value, which can vary between 0 for full coverage and 0 for no coverage.

a density constant and is given by:

$$d_{min} = \sqrt{(\frac{X_{max} - X_{min}}{n_x})^2 + (\frac{Y_{max} - Y_{min}}{n_y})^2 + (\frac{Z_{max} - Z_{min}}{n_z})^2} \tag{7.11}$$

where the X, Y, Z are the are the dimensions of the bounding block and n_x, n_y, n_z are the **density constants**. The role of the density constants are two-fold. The first is to decide roughly how many seeds should be placed (relative to the size of the box) and secondly to allow biasing of seed positioning in any direction. If one has $n_x = n_y = n_z$ (which was eventually always used), the minimum distance is just some percentage of the box diagonal. This implies that the seed density remains constant no matter what the box size.

Once the minimum distance is found, the bounding box is scanned until one of the voxel values exceeds a threshold for its percentage coverage value. This is taken as the first seed value and placed into list. The scan continues until another point exceeds the threshold, this point is then compared to all the points in the list and if the distance is greater than d_{min}, it is added to the list. The distance between the point being scanned and the points in the list is given by

$$Dist_{pl} = \sqrt{(x_p - x_l)^2 + (y_p - y_l)^2 + (z_p - z_l)^2} \tag{7.12}$$

The process is repeated until the entire region has been scanned. Due to the much larger region size, the choice of percentage coverage threshold was set very high and values greater than 90% were used. The choice of density constants was chosen at 8. This gave a good even spread of seeds, without over seeding or under-seeding any regions. Although the method does not provide optimum solution, it does achieve a sufficient spread of seeds so that the region can be grown.

7.6 Implementation Considerations

The definition of the Seeded Region Grower, as given in Chapter 6, section 6.1, is a generic formula and applies to both 2-D and 3-D spaces. Although changes to the criteria for adding voxels have been made, the implementation of the three-dimensional seeded region grower is basically the same as in the two dimensional case. The differences required in actual implementation were the use of specific data structures due to the high memory requirements of volume data.

7.6.1 Memory Requirements

The first data structure that needs to be considered is the brain Volume. The images from the CT scanner are 512×512 *short* (2 byte) images. In order for effective 3-D processing and to try reduce partial voluming effects, a slice depth of $2mm$ was used. Thus, a scan depth of 10cm at a $2mm$ slice depth requires $2 * 512 * 512 * \frac{100}{2} = 26214400$ bytes, or 25 Mbytes of memory. If one now considers interpolation to obtain a square voxel size, the memory requirement increase four-fold to roughly 100 Mbytes.

The next data structure is the priority queue, used to keep track of the next voxel to be considered in the SRG. The Fibonacci heap, as described earlier, is used. It stores the key (the error (float) value) and its position in the volume (x - short ,y - short ,z - unsigned char). The memory for 1 element is thus 9 bytes. The number of elements in the heap at any one time varies, but a good estimate would be half of the number of voxels in the volume. This results in roughly 60 Mbytes for the example above. (225 Mbytes for the interpolated case.)

Finally an intermediate and output volume are also required. These have the same dimensions as the input volume and thus require similar amount of memory. Adding these together, a maximum memory requirement of roughly 125M byte for the non interpolated and 450 Mbyte would be required. The typical memory requirements are less than the ones calculated above, but it should be clear that moving from 2-D to 3-D requires a vastly more memory.

7.6.2 Reducing memory requirements

As the above memory requirement is large, the SRG algorithm would require an expensive computer to operate. As these machines were not available (and are not likely due to costs), the first priority was to reduce the memory requirements of the 3-D algorithm. Two methods were used to achieve this reduction. The first decreases the amount of data in memory that is not being processed and the second implements the volume data structure more efficiently.

The first method used to reduce the memory requirement, is to extract only that part of the volume that is used in the analysis. In chapter 4, the region of interest (ROI) or intracranial area (although it can be thought of as a volume due to the slice thickness) was extracted. The Volume of Interest (VOI) is formed by stacking the slices forming a square block of data. Each slice contains empty space (regions outside the ROI), that do not need to be included in the block volume. The method used is to find the maximum and minimum volume bounds and then extract the data into a new volume with the reduced dimensions.

Once the intracranial volume was extracted, the data was converted from a short data type to a byte date type. This was achieved by noting that the brain images (skull and scalp removed) contain less than 255 useful grey-levels. The gray levels chosen were between 1000

and 1255. These values worked for all the images from the scanner used, but might have to be adjusted for different scanners. This resulted in a reduction of half the input volume.

The final technique alters the data structure that is used to keep track whether a pixel has been accessed or not. This is a binary value, and thus requires only 1 bit. The smallest data type (depending on machine architecture) is 8 bytes. Using this for such a data structure would therefore waste 7 bits, or using the figures derived for the entire volume, 10.94 Megabytes. For this reason, a bit storage type was implemented that allows manipulations of individual bits. The bit vector wastes at maximum 7 bits of data and thus is a vast improvement over using standard data types.

If one assumes that the x-y dimensions of the image is reduced from 512×512 to 400×400 (a reduction of 40%), and applies the above modifications, the new memory requirement is found to be 58 Mbytes. This halves the memory needed and enables the program to run on most modern workstations. The size of the interpolated volume memory use is reduced to roughly 230 Mbytes. The above example used 50 slices. In most CT volume scans, the number of slices are less than this and thus the values here must be seed as the upper bound of memory usage.

7.7 Results

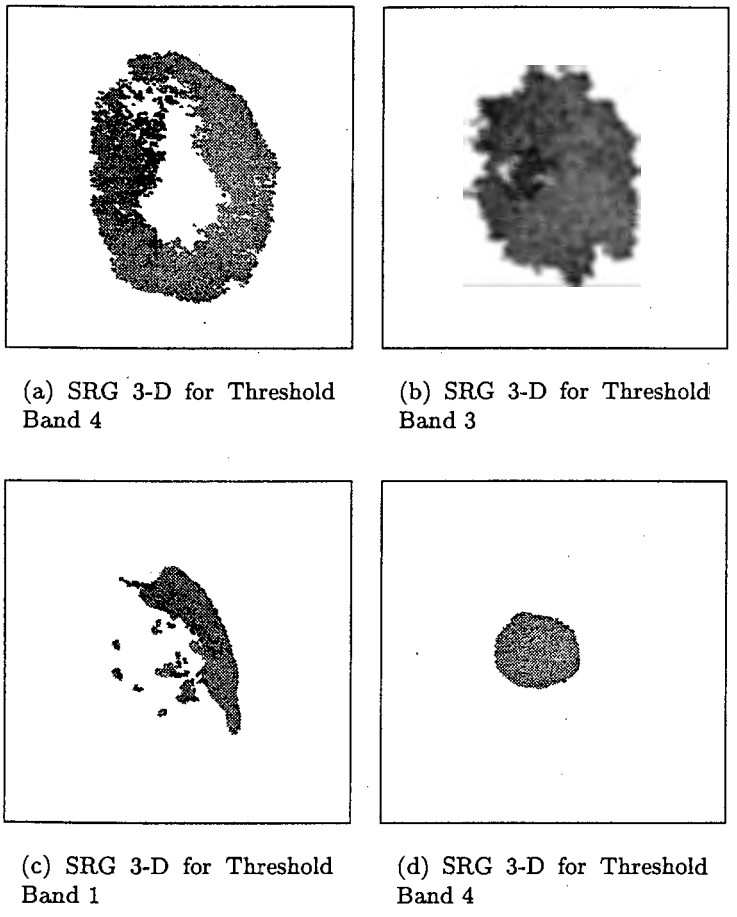
The results are divided into two sections. The first gives the results for real anatomical objects, such as cysts or the Cerebrospinal fluid. The former useful as a demonstration of a practical implementation, but does not provide very useful quantifiable results. This is due to the *ground truth* or original input volumes being unknown. It is also somewhat ambiguous to compare the three dimensional SRG results with the hand segmented results as was done previously. This is due to the fundamental differences in segmentation procedures. If a three dimensional hand segmentation was available, it could be used. For these reasons, phantom objects were used. The second section gives the results of the segmentation with respect to these phantoms, and compares them with the two dimensional results.

7.7.1 Results for 3-D anatomical objects

The results of the three dimensional extension of the Seeded Region Growing are given in Figures 7.12 (a)-(d) and Figures 7.13(a)-(d). Figure 7.12(d) corresponds to the cyst given in Figure 6.14(a) and 7.12(a) to the lateral ventricle segmented in Figure 6.14(b). The main difference between the two and three dimensional results, from a visual perspective, is the smoothness in the 3-D structure. There seems to be less of a noticeable quantisation step between slices in the z-axis. This can be directly attributed to the incorporation of depth or

z-direction information. It could also indicate a better segmentation of the volume. This is however subjective and would need to be confirmed by experienced radiographers.

More important than the visual differences is the volume obtained. For the three dimensional case, the volume of the cyst in Figure 7.12(d) was found to be $58.162cm^3$. This was 3.756% and 3.521% larger than that of the two dimensional and hand-segmented cases respectively. The volume of the CSF differed considerably more, and was calculated at $21.2cm^3$. This is 8.16% larger than the hand segmented and 7.07% larger than the two dimensional results. The method of calculating the volume was once again by finding volume shell and counting all the voxels contained inside.



(a) SRG 3-D for Threshold Band 4

(b) SRG 3-D for Threshold Band 3

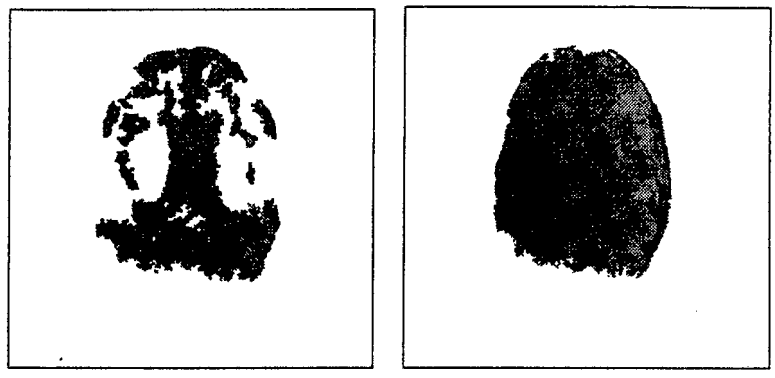
(c) SRG 3-D for Threshold Band 1

(d) SRG 3-D for Threshold Band 4

Figure 7.12: Results of SRG 3-D for Image Set 1 - Corresponding Threshold Bands are given in Figures 7.10 and 7.9

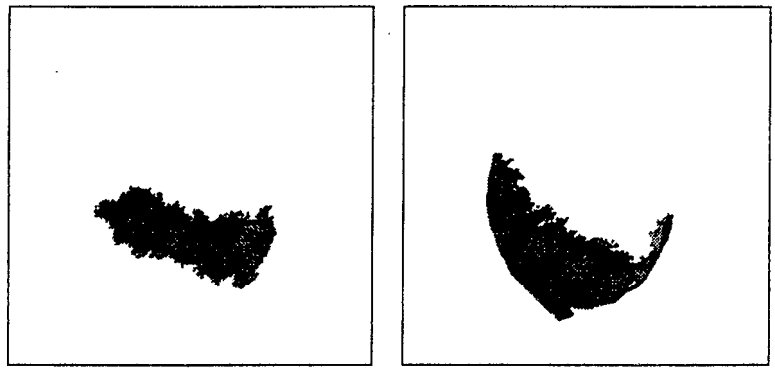
The reason for the increased volume can be attributed to the higher connectivity of the three dimensional algorithm. This was less pronounced for the cyst than the Cerebrospinal fluid (CSF), as the cyst formed a solid compact object. This is opposite to the CSF and ventricles,

which are formed by both compact and sparse regions. The sparse region are under segmented or mis-segmented in the two dimensional algorithm, but not in the three-dimensional case, thus the higher volume. The hand segmented case reflects the results of the two dimensional algorithm more closely, as these sparse objects are often ignored or deemed too difficult to segment.



(a) SRG 3-D for Threshold Band 1

(b) SRG 3-D for Threshold 2



(c) SRG 3-D for Threshold 3

(d) SRG 3-D for Threshold 4

Figure 7.13: Results of SRG 3-D for Image Set 2 - Corresponding Threshold Bands are given in Figure 7.11

7.7.2 Results for 3-D phantom lesions

As was done with the 2-D algorithm, the accuracy of the SRG 3-D implementation was tested using phantom lesions. The phantom model used was the new phantom model described in section 7.3. Comparisons between the 2-D and 3-D algorithms were done for both simple and more complicated phantoms. The simple phantom used was a sphere. The phantom

was inserted into the volume (as described in section 7.3) and when evaluating for the 2-D algorithm each slice was extracted and processed. The volume of the segmented phantoms were then calculated and compared to the input size. This was done for varying noise, phantom size and phantom mean tissue value. Due to the number of variables affecting the segmentation, it would be impractical to plot all of the results in this chapter. For this reason, a detailed set of results has been given in Appendix C. However, to illustrate important results, extracts of these and other results will be given here.

Upper and Lower band Percentage errors

Before describing how noise, mean or phantom size affect the results, it was deemed essential to find the upper and lower bound percentage volume error. The percentage error is the percentage volume difference between the input and segmented phantoms. In theory the upper bound error (maximum) should be obtained for the highest noise value, smallest phantom size and minimum separation between tissue means. The lower bound is obtained for opposite values. These results were important as they would give an overall idea whether further testing or implementation of a complete system would be feasible. The upper bound variables for this dissertation were $\sigma^2 = 130$, a $radius = 8$ and a mean separation of 5 grey levels. The lower bound variables are $\sigma^2 = 30$ (normal noise level for the scanner used), $radius = 28$ and a mean separation of 25 grey levels. Figure 7.14 shows the error for the above values. The negative percentage error (y-axis) means an over segmentation of the phantom and a positive value an under segmentation. The upper bound case resulted in a value of -48% and the lower bound case -2.5%.

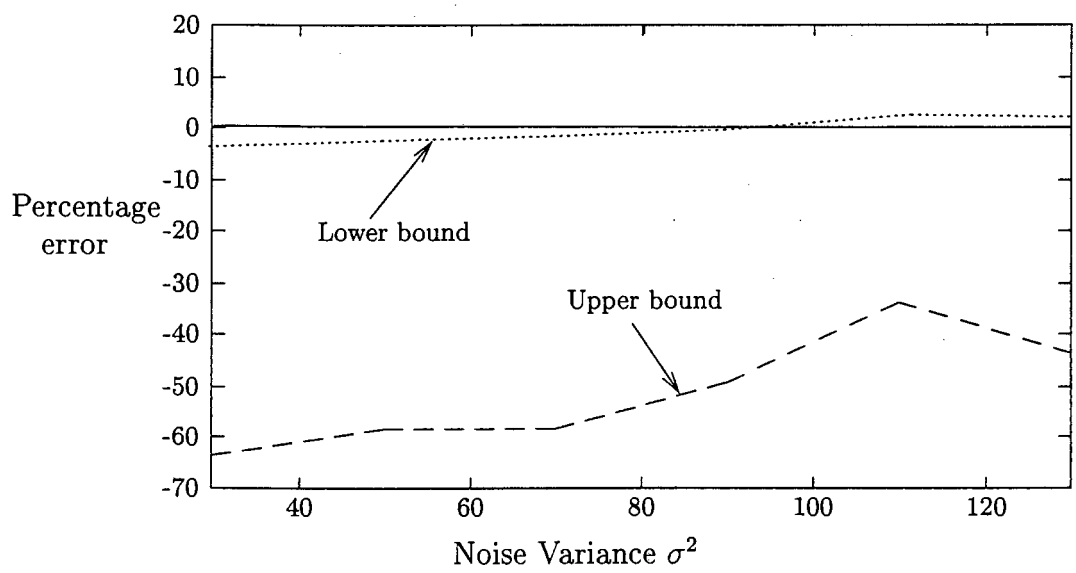


Figure 7.14: Upper and lower error bounds for SRG 3-D

In an ideal situation, the rest of the results (for the varying parameters) should fall between these values. As can be seen in Figure 7.14, for a variation in the noise parameter, this is not the case. The main reason for this, is the effect of the phantom position on the segmentation. In order to quantify the effect of position, phantoms would need to be placed randomly in various parts of the volume. This was not possible due to time and processing constraints and could only be done for a few isolated positions. From these experiments, a feasible upper and lower bound estimate of -45% to 10% was formed. This large swing in percentage error, initially seemed to large to be acceptable. However, if one looks at figure 7.15 it is apparent that the large errors are for small phantom sizes. The minimum sized phantom that was practically discernible in the SRG 2-D has a radius of 12 (Radii below this value, although segmented, required substantial user intervention.). Thus using 12 as a minimum size, the lower and upper bound average errors for the 3-D case are -1% and -15% respectively.

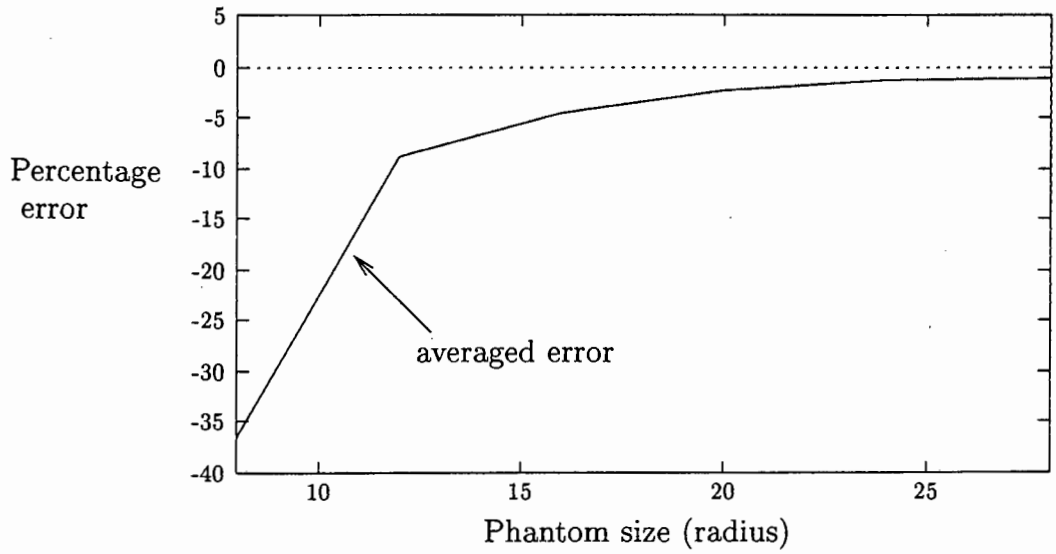


Figure 7.15: Averaged error vs phantom size

Effects of size, mean distance and noise on segmentation results.

The previous section focused on the upper and lower error bounds. This gave an idea whether or not the algorithm was feasible. The next task was to evaluate how the algorithm responded to the changing of the various variables. This was crucial as it would enable an operator to place a degree of confidence on the results (in practise the input is clearly **not** known) according to the rough output size, noise level and segmented object mean grey level.

The first variable that can be varied is the size of the phantom. In this case, the phantom was a sphere and adjusting the size meant changing the radius. The sphere is an useful phantom as almost any anatomical object can be made from a series of different size spheres. The

smallest sphere that can be accurately segmented generally gives an idea for the rest of the anatomical objects.

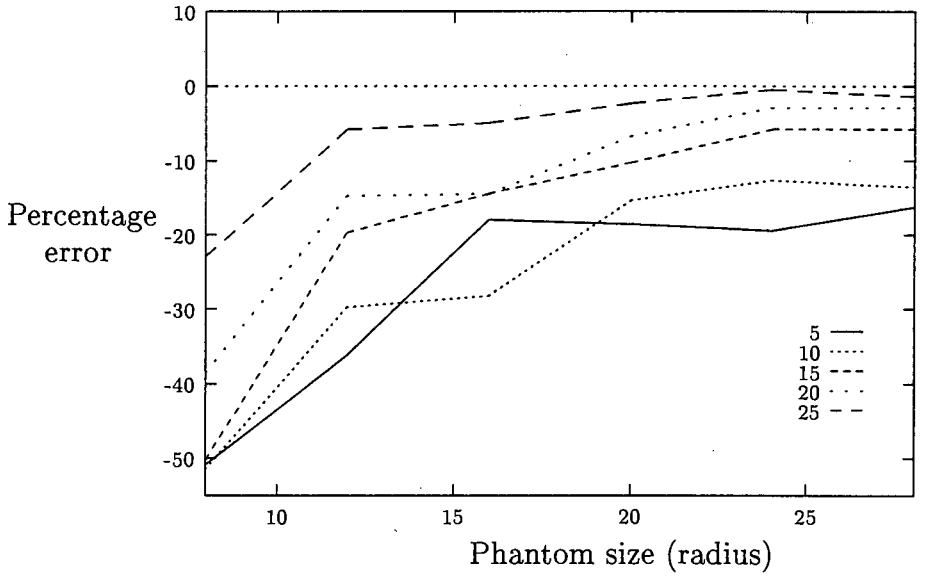


Figure 7.16: Percentage Error vs phantom size for different phantom mean values.

Figure 7.16 shows the percentage error versus phantom size for various phantoms means. Each value is averaged over a noise range of $\sigma = 30$ to $\sigma = 130$. The results show that the percentage error decreased as the size of the phantom increased. The decrease was the greatest between 8 and 12 for spherical phantoms. The slope is quasi-logarithmic in the sense that the change in error gets smaller as the size increases. It was found that the change in error is negligible above radii of 16. These results showed that anatomical objects, encompassing an approximate sphere of radius greater than 16, can be segmented with an percentage volume error below 15%. These values apply to the complete algorithm including seed finding. It should be noted that results were better when manual seed selection was done. This was expected as placing of seeds was more accurate.

The size of the object was shown to be the dominant factor in the segmentation. This was due to the following reasons:

- The noise filter has a greater effect on a larger object (It averages, thus the more samples the better the average).
- The larger the object, the greater the contribution in the volume histogram. It is also less likely to be masked by other tissue in the histogram.
- This makes it more likely to be found by the peak detection process.
- It will be found by a larger seed, making the starting statistics more accurate.

- The larger the object, the less likely it is to be affected by noise. This ensures that the region growing criteria remain more constant during growing.

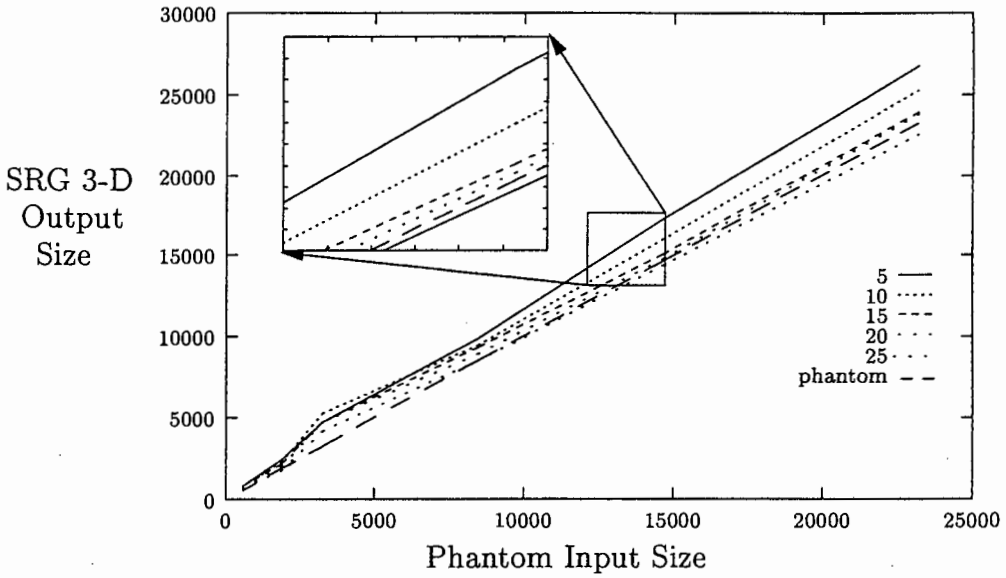


Figure 7.17: SRG 3-D Volume vs phantom Volume for different mean values.

After the size of the object, the next influential variable is the effect of the object mean grey level. As was described earlier, the problem in segmenting X-Ray CT images lies is the domination of the grey and white matter (the majority) over other intercranial materials. To evaluate the effect of the closeness of the phantom mean to the tissue mean value, phantoms of various distances were segmented. These ranged from 5 to 25 CT gray levels from the tissue mean. The effect of this is effectively seen when looking at figure 7.17. It consists of plots of the segmented volume versus the input volume (The graph is for a noise value of $\sigma^2 = 70$, which is above the average noise of the scanner. A full set of graphs is given in C.3). The line depicting the phantom is thus at 45°. The extracted portion of the graph shows that as the mean distance decreases, the segmented volume deviates further from the input volume.

Another way of viewing the quantifiable effects of the phantom mean distance is given in section C.2. These are graphs of the segmented volume versus noise for the different phantom means. Each is for a fixed size. As in figure 7.17, the deviation from the input phantom increases with decreased mean. What is however clearer in these graphs, is that there is an increase as the mean distance decreases. (ie: There is a non-linear relationship between the mean and volume error.) From these graphs and especially for larger phantoms, a distinct step occurred at a distance of about 15 grey levels from the dominant tissue.

The last significant variable considered to influence the segmentation, was that of noise. The type of noise found was detailed in chapter 5. The typical noise value of the X-Ray CT scanner

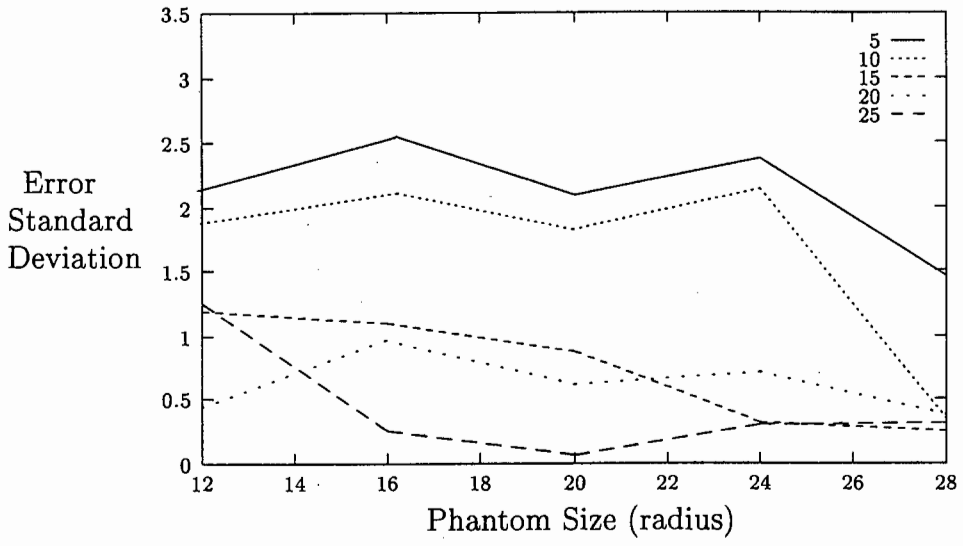


Figure 7.18: Standard deviation in percentage volume error due to noise - 3-D SRG case

used was $\sigma^2 = 30$. They noise values used in the experiments was varied between this value and $\sigma^2 = 130$.[4] The effects of the noise on the segmentation were more difficult to estimate, as only a limited number of trials could be done given time considerations. The graphs in appendix C.5, however do give an indication of the effect of noise. The graphs show the variation in percentage error due to noise. Each plot is for an individual mean and phantom size. It was found that under higher noise conditions, the segmentation deviated away from the true phantom volume. Another way to view the effect of noise is to see how stable the segmentation algorithm is, over the noise band. Figure 7.18 is the standard deviation of the percentage volume error over the noise band. From figure 7.18 it is clear that the maximum deviation is roughly 2.5%. This shows that the SRG 3-D algorithm is robust under varying noise conditions.

7.7.3 Comparison of 2-D and 3-D algorithms

The results of both the two and three dimensional algorithms were within the required limits. This section compares the two methods, to see if the assumed advantage (for the SRGD 3-D) given at the beginning of this chapter, are realised. The comparison was first made between the real images and then between the images with phantoms. A speed comparison of the overall process and memory requirements will also be done.

[4]The Gaussian noise distribution values were generated by the *Normal* class, using the *ACG* random number generator class (with varying seeds and a size parameter of 98), of the libg++ class library supplied with the g++ C++ compiler.

2-D and 3-D Anatomical Comparison An effective comparison was made in section 7.7.1 and has not been repeated here. The results of the comparison have however, been given in table 7.3 for completeness.

Image Set	Hand	SRG 2-D	SRG 3-D
1 (cyst)	56.11	55.97	58.16
2 (ventrical)	19.71	19.47	21.20

Table 7.3: Comparison of Hand, SRG 2-D and SRG 3-D segmented anatomical volumes. All volumes are in cm^3

2-D and 3-D Phantom comparison Figure 7.21 gives the results of the segmentation for a mean distance of 15. The dotted curve represents the SRG 3-D result, the lower dashed curve the SRG 2-D results and the straight solid line is the actual phantom size. The most striking difference between the two is the under-segmentation of the phantom by the two-dimensional SRG. This result is not unexpected and can be explained as follows: The 2-D SRG is segmenting a number of disks (slices of the sphere). These disks become smaller towards the upper and lower portions of the sphere. At a certain point the disk becomes undetectable by the SRG 2-D and thus is not segmented. This can be caused either by the disk being too small or being swamped by noise. Either way, the result is a smaller segmentation (Section 7.1 in the motivation for a 3-D algorithm explained this problem). The reason for the over segmentation of the three dimensional algorithm was given in section 7.7.1. This explanation was with reference to the anatomical objects, but and can also be applied to the phantoms.

Figure 7.19 shows the percentage error versus phantom size for different mean values. The solid plots represent the two dimensional results. Both algorithms perform at similar levels for mean values greater than 10 grey levels from dominant tissue. The plot with the high error is for a distance of 5. The two-dimensional SRG failed to segment the phantom for this mean distance. The variation in results with size is most noticeable for smaller phantoms. In these cases the 2-D algorithm performs better. The average percentage error between the two between phantom sizes of 8 and 12 was approximately 5%. For larger phantoms the 3-D algorithm performs better, with an average improvement of 2%.

The effect of noise on the two algorithms was most found to be the major difference between the two algorithms. With the results on average being quite similar, the percentage volume error variance for the phantoms was calculated. The results for the three dimensional case were given in figure 7.18 and for the 2-D case in Figure 7.20. From these graphs, it should be clear that the standard deviation for the two dimensional case is higher (Maximum at 2.5 for

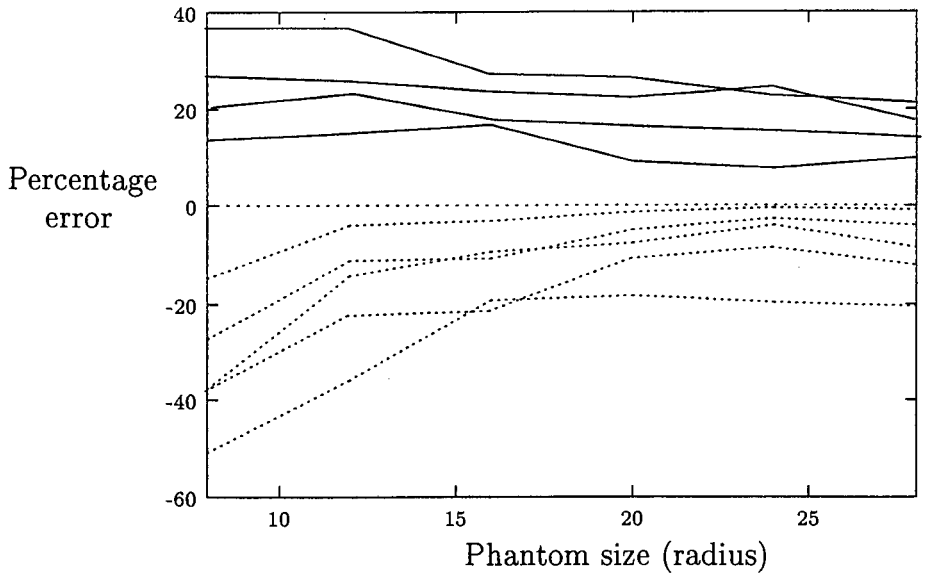


Figure 7.19: A comparison between the SRG 2-D and 3-D algorithms of size versus percentage error. This was done for various mean grey level distances from the dominant tissue mean. The dotted lines represents the SRG 3-D algorithm and the sold lines the SRG 2-D algorithm.

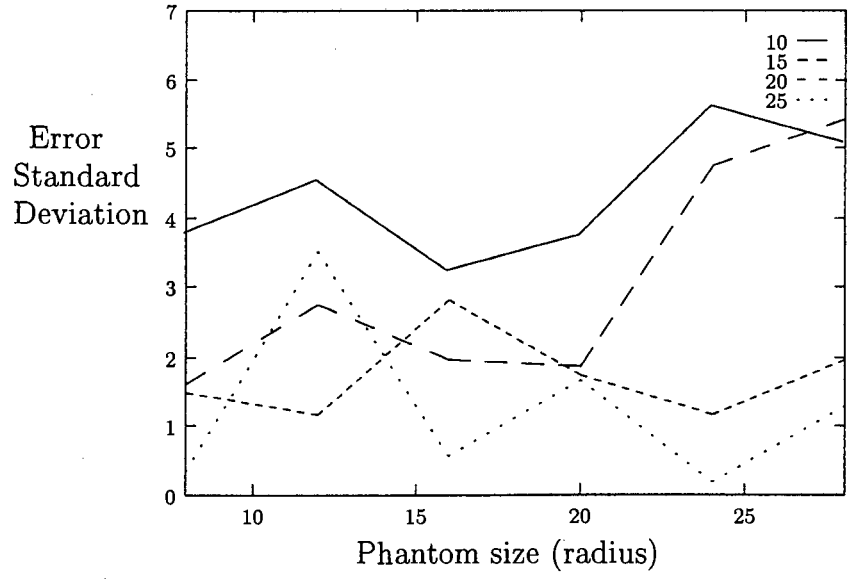
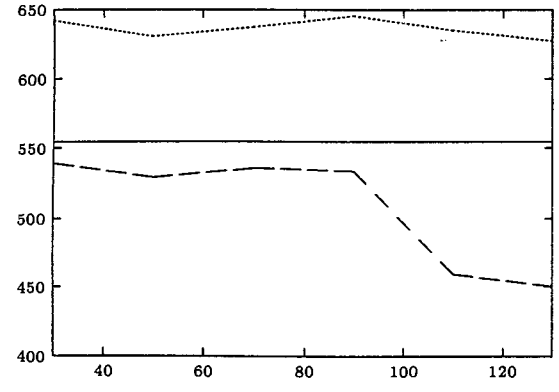


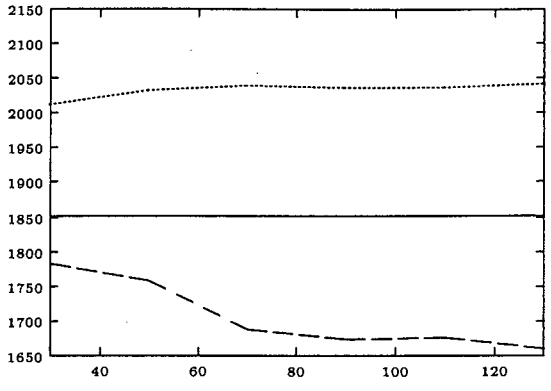
Figure 7.20: Standard deviation in the percentage volume error due to noise - 2-D SRG case.

3-D case and 5.5 for the 2-D case). It also becomes greater with increasing noise. This shows that the 3-D processing of the volumes results in a more consistent volume for varying noise conditions. For both the 2-D and 3-D SRG the volume found decreased with the amount of noise present.

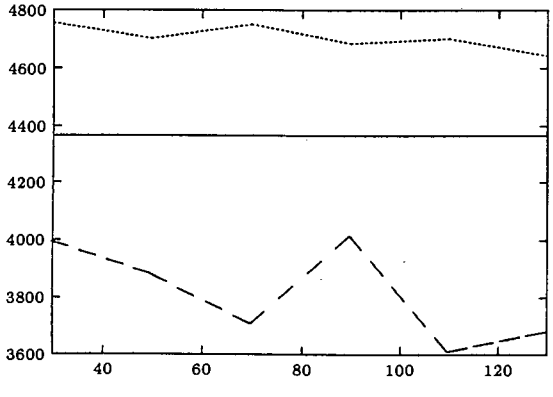
An important factor that is not highlighted by the results, is the failure of two-dimensional algorithms for certain slices. When this occurs, the slice needs to be manually segmented. This occurred for phantoms where either the mean distance was small or the noise was high. It was also found that the larger the size of the phantom, the less likely it was to be missed in the segmentation. It was however, not as significant as the mean distance or noise.



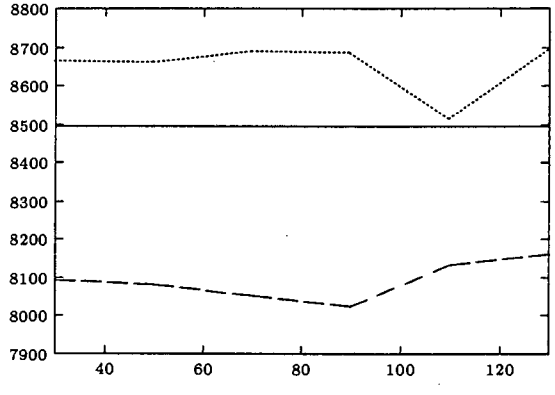
(a) Phantom Radius 8



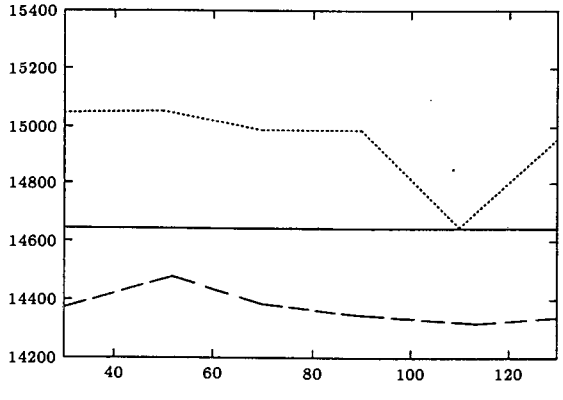
(b) Phantom Radius 12



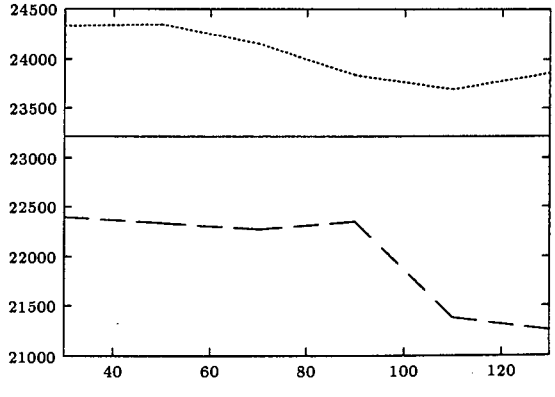
(c) Phantom Radius 16



(d) Phantom Radius 20



(e) Phantom Radius 24



(f) Phantom Radius 28

Figure 7.21: SRG 2-D vs SRG 3-D for phantom lesions. The y-axis is the output phantom volume and the x-axis is the noise level σ^2. The input volume is the volume of a sphere with the given input radius.

Chapter 8

Conclusions and Recommendations

8.1 Summary of Work

The first problem addressed in this dissertation was the extraction of the intracranial volume. It was first established whether it was worthwhile, from a segmentation point of view and not a processing one, to do this. A simple model of the head was developed and from this, the estimated scalp volume was determined. The scalp is spatially not part of the brain, but does have the same grey level values. This meant that including the scalp would give an incorrect grey level distribution for the brain. It was shown that approximately 9% of the histogram contains CT values from the scalp, and thus it was concluded that extraction was required. The next stage involved implementing an extraction algorithm. The initial step was to use existing methods, specifically by Sandor[21], to see if they could be used. The adaptive thresholding approach failed (did not produce the required closed boundary) on many images, especially the lower cranial slices. A new method, based on the shape and symmetry of the skull, was then developed. This required the head to be vertically aligned in the scanner, which resulted in a method used to re-align head being implemented (Appendix A). The new algorithm proved to be reliable and always produced a closed boundary. The method was used alone in the extraction process but could be used in conjunction with the adaptive thresholding algorithms outlined in Appendix B.

Once the intracranial region had been extracted, a filter to reduce the noise was implemented. The filter had to maintain the Partial Volume of the anatomical regions in the averaging sense. It also had to preserve the edge details of the regions. The filter was based on CT images (or volumes) having additive zero-mean Gaussian white noise. This model provided a means of finding the Least Squares or Maximum Likelihood estimate for each pixel based on surrounding pixels. The results of this showed that the best estimate was simply to average the neighbouring pixel values that lie above a certain threshold. The threshold value is chosen

from a set of curves based on the mean separation of tissue regions and the noise variance. Figure 5.1 shows these curves and the threshold chosen should maximise the Probability of detection minus the Probability of false classification. A reasonable range of values which provided good noise reduction while still maintaining edge and Partial Volume information, was found to be between 9 and 11 grey levels. The neighbourhood size was less critical than the threshold value, and the choice was based on a trade off between processing time and having enough pixels from a particular region to enable averaging. The results given in section 5.4, showed that the filter was successful. This was particularly clear when viewing the image histogram.

After the above pre-processing stages were completed, the segmentation algorithms were designed. The technique chosen for the actual segmentation was a Seeded Region Growing algorithm. Much emphasis was placed on designing a number of segmentation modules that could be interchanged or extended to higher dimensions. This resulted in the development of the technique of fitting and subtracting a Gaussian to the main histogram (tissue) peak, in order to help establish the rough grey levels region bands. The bands were found using a peak detection algorithm to the subtracted histogram and then grouping adjacent peaks according to size and distance. From this stage, a number of different techniques could have been implemented, thus emphasising the modular approach. Sandors [12] classic algorithm, or methods described by Windham[38], or even simple thresholding could have been used. The approach taken was to find seeds in each of the regions found in the peak detection stage and grow them with a minimum error ordered region grower. The result of the SRG 2-D closely matched that of the hand-segmented results. This showed that computer segmentation could be used as an effective aid in assisting the human operator. The consistency of the results also fulfilled the criteria for a good anatomical segmentation method.

Despite the success of the 2-D algorithm, it became clear that the 2-D nature of the processing itself was a restriction on what were essentially 3-D structures. The logical extension was thus to process the entire set of data as a volume. This resulted in various aspects concerning 3-D processing (with respect to CT imaging) being investigated. These included the advantages of 3-D over 2-D in terms of connectivity, ways of creating a volume from a set of 2-D slices, effects of interpolation on the segmentation and whether the increased processing time associated with 3-D processing was justified. The extension of the entire segmentation process to three dimension was then implemented. The modules were all extended with changes being made to cope with the higher connectivity of the volume. A method using surface connectivity between voxels was used to classify voxels with similar error measures. The testing of the 2-D and 3-D algorithms was done for real anatomical objects and phantoms. This resulted in a new technique for modelling phantoms, that incorporate the Partial Voluming effects caused by image formation process. Finally the results of the 2-D and 3-D methods were compared and evaluations made.

Thus, for the entire segmentation the objectives were met. The limitations of each of the methods were outlined and evaluated with respect to the project requirements. A general frame-work regarding CT segmentation and the problems involved were outlined. Finally, the use of region growing for segmenting CT brain images was shown to be an effective tool in aiding the human operator.

8.2 Conclusions

Based on the results and findings of this dissertation, the following conclusions may be draw:

1. Intracranial Volume Extraction

   (a) The need for intracranial volume extraction was justified as 9% of the image grey level distribution consisted of scalp.

   (b) All Thresholding and Boundary Tracking Techniques failed due to grey level dropout along the skull boundary.

   (c) A algorithm based on symmetry and shape enabled accurate extraction under varying conditions.

   (d) The failure rate of symmetry and shape based extraction technique was lower than that of the thresholding and tracking.

   (e) The skull could be vertically aligned provided the initial rotation was within ± 20 degrees from vertical.

   (f) Axial slices containing eye sockets and lower ventrical system (3rd and 4th ventricals) could be extracted.

2. Noise Reduction Filter

   (a) Additive Zero-Mean white Gaussian noise was a reasonable model for X-Ray CT images.

   (b) The filter used maintained edge and Partial Volume information, while maximising noise reduction.

   (c) Threshold values between 9-11 grey levels for the noise filter were found to be sufficient to maintain the above condition.

   (d) A Neighbourhood size of between 7 and 11 pixels was large enough to provide the necessary averaging.

(e) The filter improved the results of the segmentation procedure.

3. Seed Finding Procedure

(a) Fitting a Gaussian to the image histogram enabled obscured tissue to be identified.

(b) Peak detection and a combination of size and distance measures reduced the number of Region threshold bands in the histogram.

(c) A block percentage coverage of above 80% proved to be adequate in finding homogeneous regions to place seeds.

(d) For slice processing, a block size of 2×2 for sparse regions and 5×5 for large connected regions allowed for both characteristic region types to be seeded.

(e) For volume processing the best block size was square if the volume was interpolated to resize voxels. Otherwise, it was best to choose the block size in relation to the voxel resolution.

(f) For the voxel ratio of width versus height versus depth of 4:4:1, a block size of $2 \times 2 \times 1$ (for small sparse regions) and $5 \times 5 \times 2$ (for large connected region) was chosen.

4. Seed Region Growing Algorithm.

(a) The SRG algorithm incorporated knowledge through the choice of seeds. This enabled automated (this dissertation) or manual selection (an operator) of seeds, the main criteria for a good segmentation.

(b) The use of a sequentially sorted data structure in deciding which pixels to add, ensured that the regions were grown according to minimum error and were not effected by outlying points.

(c) The SRG was shown to be robust under noise conditions varying between $\sigma^2 = 30$ to $\sigma^2 = 130$.

(d) The SRG 3-D was able to reliably segment objects consisting of a constant region of radius greater than 12 voxels. The voxel dimensions of the scanner used were $0.47mm \times 0.47mm \times 2mm$ and thus the small object in mm has about radius of $5.64mm$ and a volume $741mm^3$.

(e) The error for the SRG 2-D and SRG 3-D ranged between 5% and 15% depending on noise, shape and separation between region means. This was within the requirements of this project.

(f) The 3-D implementation out-performed the 2-D one, especially for objects that are connected in 3-D but difficult to segment in 2-D.

(g) The Seeded Region Grower fulfilled all the requirements layed out in the dissertation. It provides a fast, accurate and most important reproducible method for find the volume of various anatomical regions in the brain.

8.3 Recommendations

As a result of the findings and conclusions of this dissertation the following recommendations regarding the implementation of the system:

- The Noise Reduction Filter threshold should be between 9 and 11 grey levels.
- The Noise Reduction Filter neighbourhood size should be between 7×7 and 11×11 pixels. For the 3-D filter it should be based on the same x and y values, but for the z value it should take the voxel resolution into account.
- For the Seed Selection process, bands containing less than 1000 (32×32) pixels and 2048 ($16 \times 16 \times 8$) voxels in the histogram should be merged or ignored.
- Block Sizes for calculating percentage coverage should be 2×2 and 5×5 for 2-D and $2 \times 2 \times 1$ and $5 \times 5 \times 2$ for 3-D.
- For the Seed Selection process, a minimum of 80% percentage coverage should be used.

Due to the finite development time, only a small number of X-Ray CT slices and volumes could be tested. It is thus important that a clinical implementation and trial of the technique be done. The recommended areas that should be investigated during such a trial are:

- A comparison of a wide range of block sizes (as used in seed placement) to the recommended ones for volumes and slices.
- A comparison of manually places seed positions to the automated ones over a large group of subjects. This would expose any problems related to automated seed placement.
- A thorough comparison between the SRG and manually obtained volumes needs to be done. This would enable the algorithm to be compensated for, if any repeated deviations occur.

It is also recommended that the following topics should be further investigated.

Section 8.3: Recommendations

- The extending of the intracranial volume extraction to 3-D, with emphasis on using symmetry to find the orientation of the head in the volume.
- The use of other features such as spatial position or region variance in calculating the growing error criteria δ.
- The incorporation of edge-information into the region growing process, through a similar process outlined by Pavlidis[54].

Bibliography

[1] Kak A. C. and Slaney M., *Principles of Computerized Tomographics Imaging.* IEEE Press, 1987.

[2] Parker P., *McGraw-Hill Dictionary of Scientific and Technical Terms.* McGraw-Hill, New York, 4th ed., 1984.

[3] Pfefferbaum A., Zatz L. M., and Jernigan T. L., "Computer interactive method for quantifying cerebrospinal fluid and tissue in brain CT scans: Effect of aging," *Journal of Computer Assisted Tomography*, vol. 10, no. 4, pp. 571–578, 1986.

[4] Jain A. K., *Fundamentals of Digital Image Processing.* Prentice-Hall International, Inc, 1989.

[5] Pratt W. K., *Digital Image Processing.* John Wiley and Sons, Inc., second ed., 1991.

[6] Haralick R. M. and Shapiro L. G., "Survey: Image segmentation techniques," *Computer Vision, Graphics and Image Processing*, no. 29, pp. 100–132, 1985.

[7] Chen S.-Y., Lin W.-C., and Chen C.-T., "Medical image recognition based on dempster-shafer reasoning," in *SPIE Proceedings. Medical Imaging VI: Image Processing* (Loew M. H., ed.), vol. 1652, pp. 555–566, 1992.

[8] Linda G. Shapiro M. K.abd, "Knowledge-based organ identification from ct images," *SPIE Medical Imaging VI: Image Processing*, vol. 1652, pp. 544–554, 1992.

[9] Eggleston P., "Object segmentation techniques for use in laboratory visual automation systems," in *Intellegent Robots and Computer Vision XI: Algorithms, Tech.*, vol. 1825-31, pp. 314–324, Nov. 1992.

[10] Brice C. and Fennema C., "Scene analysis using regions," *Artificial Intelligence*, vol. 1, pp. 205–226, 1970.

[11] Adams R. and Bischof L., "Seeded region growing," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 641–646, June 1994.

[12] Sandor T., Mtcalf D., and Kim Y., "Segmentaiton of brain CT images using the concept of region growing," *Int J Biomed Comput*, vol. 29, pp. 133–147, July 1991.

[13] Horowitz S. L. and Pavlidis T., "Picture segmentation by tree traversal algorithm," *J. Assoc. Comput. Mach*, vol. 23, no. 2, pp. 368–388, 1976.

BIBLIOGRAPHY

[14] Chen S.-Y., Lin W.-C., and Chen C.-T., "Split-and-merge image segmentation based on localized feature analysis and statistical tests," *CVGIP:Garphical; Model and Image Processing*, vol. 53, pp. 457–475, Sept. 1991.

[15] Mahaley S. M., Gillespie Y. G., and Hammet R., "Computerised tomography brain scan tumor volume determination," *J. Neurosurgery*, vol. 72, pp. 872–878, 1990.

[16] Franke C. L., Versteege C. W. M., and van Gijn J., "Best fit method. a simple way for measuring the volume of an intercerebral hematoma," *Neuroradiology*, vol. 30, pp. 73–75, 1988.

[17] Breiman R. S., Beck J. W., Korobkin M., Glenny R., Akwari O. E., Heaston D. K., Moore A. V., and Ram P. C., "Volume determination using computed tomography," *AJR*, vol. 138, pp. 329–333, 1982.

[18] Canny J., "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 679–698, 1986.

[19] Otsu N., "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man Cybernetics*, vol. SMC-9, no. 1, pp. 62–66, 1979.

[20] Kapur J. N., Sahoo P. K., and Wong A., "A new method for grey-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics and Image Processing*, vol. 29, pp. 273–285, 1985.

[21] Sandor T., Kalender W. A., Hanlon W. B., and Rumbaugh C., "Automated calvaria analysis from computerized axial tomographic scans," *Comp. Biomed Res*, vol. 14, pp. 119–124, 1985.

[22] Lei T. and Sewchand W., "Statistical aproach to x-ray CT imageing and its applications in image analysis - part II: A new stochastic model-based image segmentation technique for x-ray CT image," *IEEE Transaction on Medical Imaging*, vol. 11, pp. 53–61, March 1992.

[23] Albright R. E. and Fram E. K., "Microcomputer-based techniques for 3-D reconstruction and volume measurements of computed tomographic images. part 1: Phantom studies," *Invest. Radiol.*, vol. 23, no. 882-885, 1988.

[24] Hussain Z., *Digital Image Proccessing - Practical applications of Parallel Processing Techniques.* Ellis Horwood, 1991.

[25] Horn B. K. P., *Robot Vision.* McGraw-Hill, 1986.

[26] Gonzalez C. F., Grossman C. B., and Masdeu J. C., *Head and Spine Imaging.* A Willey Medical, 1985.

[27] Radon J., "Über die bestimmung von funktioen durch ihre integralwerte tangs gewisser mannigfaltigkeiten (on the determination of functions from their integrals along certain manifolds).," *Bertichte Saechsiche Akad. Wissenschaften (Leipzig), Math. Phys.*, vol. 69, pp. 262–277, 1917.

[28] Hounsfield G. N., "A method of and apparatus for examination of a body by radiation such as x-ray or gamma radiation." Patent Specification 1283915, The Patent Office, 1979.

[29] Herman G. T., "On the noise in image produced by computed tomography," *Computer Graphics and Image Processing*, vol. 12, pp. 271–285, 1980.

[30] Kak A. C., *Digital Image Processing Techniques - Image Reconstruction from Projections.* New York: Academic Press, 1984.

[31] Roberts B. and Kak A., "Image reconstruction from projections," in *Handbook of Pattern Recognition and Image Processing* (Young T. and Fu K., eds.), Academic Press, 1986.

[32] Herman G. T., *Image Reconstruction from Projections.* New York: Academic, 1980.

[33] Brooks R. A., "A quantitative theory of the hounsfield unit and its application to dual enrgy scanning," *J. Comput. Assit. Tomog.*, vol. 1, pp. 487–493, 1977.

[34] Bates R. H. T., Garden K. L., and Peters T. M., "Overview of computerized tomography with emphasis on future developments," *Proc. IEEE*, vol. 71, pp. 356–372, Mar. 1983.

[35] McCullough E. C., Baker H. L., Houser L. W., and Reese D. F., "An evaluation of the quantitative and radiation features of a scanning x-ray transverse axial tomograph: The emi scanner," *Radiat, Phys*, vol. 3, pp. 709–715, June 1974.

[36] McCullough E. C., "Photon attenuation in computed tomography," *Med. Phys.*, vol. 2, pp. 307–320, 1975.

[37] Raff U. and Newman F. D., "Automated lesion detection and lesion quantification in mr images using autoassociative memory," *Medical Physics*, vol. 19, pp. 71–77, Jan/Feb 1992.

[38] Soltanian-Zaded H., Windham J. P., and Peck D. J., "Semi-automatic brain morphometry from CT images," *SPIE The International Society for Optical Engineering*, vol. 2167, 1994.

[39] Newton T. H. and Potts D. G., eds., *Radiology of the Skull and Brain.* The C. V. Mosby Company, Saint Louis, 1971.

[40] Swokowski E. W., *Calculus With Analytic Geometry.* PWS. Kent, 1992.

[41] Gonzalez R. C. and Woods R. E., *Digital Image Processing*, ch. 7, pp. 443–457. Addison-Wesley, 1992.

[42] Lisle D. A., *Imaging for Students.* Arnold, 1996.

[43] Russ J. C., *The Image Processing Handbook.* IEEE PRESS, second edition ed., 1995.

[44] Lei T. and Sewchand W., "Statistical aproach to x-ray CT imaging and its applications in image analysis - part I: Statistical analysis of x-ray CT imaging," *IEEE Transaction on Medical Imaging*, vol. 11, pp. 53–61, March 1992.

BIBLIOGRAPHY

[45] Stark H. and Woods J. W., *Probabilty, Random Processes and Estimation Theory for Engineers, Englewood Cliffs.* Prentice-Hall Inc.,Englewood Cliffs, N,J,, 1986.

[46] Tenenbaum A. M., Kangsman Y., and Augenstein M. J., *Data structures using C.* Prentice Hall International Editions, 1990.

[47] Budd T. A., *Classic Data Structures in C++.* Addison Wesley, 1994.

[48] Fredman M. L. and Tarjan R. E., "Fibonacci heaps and their uses in improved network optimization algorithms," *Journal of the ACM*, vol. 34, pp. 596–615, July 1987.

[49] Lin W.-C., Chen S.-Y., and Chen C.-T., "A new surface interpolation technique for reconstructing 3d objects from serial cross-sections," *Computer Vision, Graphics and Image Processing*, pp. 124–143, 1989.

[50] Lancester J., Eberly D., and Alyassin A., "A gemoetric model for measurement of surface distance, surface area and volume from tomographic images," *Medical Physics*, vol. 19, pp. 419–431, Mar/Apr 1992.

[51] Press W. H., Flannery B. P., and Teukolsky S. A., *Numerical Recipes in C, The Art of Scientific Computing.* Cambride University Press, 1990.

[52] Cline H. E., Lorenson W. E., Hart H. R., Lorenson W., and Ludke S., "3-D reconstruction of the brain from magnetic resonance images using a connectivity algorithm," *Magnetic Resonacne Imaging*, vol. 5, pp. 345–352, 1987.

[53] Cline H. E., Lorensen W. E., Ludke S., Crawford C. R., and teeter B. C., "Two algorithms for the three-dimensional reconstruction of tomograms," *Medical Physics*, vol. 15, pp. 320–327, 1988.

[54] Pavlidis T. and Liow Y.-T., "Integrating region grwoing and edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 1278, pp. 225–233, 1990.

[55] Zabrodsky H., Peleg S., and Avnir D., "Symmetry as a continuous feature," *IEEE Transaction on Pattern Analysis and Machine Intellegence*, vol. 17, pp. 1154–1166, Dec. 1995.

[56] Otterloo P. J. V., *A Contour-Oriented Approach to Shape Analysis.* Prentice Hall International, 1991.

[57] Castleman K. R., *Digital Image Processing.* Prentice-Hall, 1979.

[58] Pun T., "Entropic thresholding , a new approach," *Computer Graphics and Image Processing*, vol. 16, pp. 210–239, 1981.

[59] Tsai W., "Moment preserving thresholding: A new approach," *Computer Vision, Graphics and Image Processing*, vol. 29, pp. 377–393, 1985.

[60] Brink A. D., "The selection and evaluation of grey-level thresholds applies to digital images," Master's thesis, Rhodes University, Oct. 1987.

[61] Brink A. D., "Grey-level thresholding of images using a correlation criterian," *Pattern Recognition Letters*, no. 9, pp. 335–341, 1989.

[62] Kittler J. and Illingworth J., "Minimum error thresholding," *Patter Recognition*, vol. 19, no. 1, pp. 41–47, 1986.

[63] Cseke I. and Fazekas Z., "Comments on gray-level thresholding of images using correlation criterian," *Pattern Recognition Letters*, no. 11, pp. 709–710, 1990.

[64] Brink A. D., "Comments on grey-level thresholding of images using a correlation criterian," *Pattern Recognition Letters*, no. 12, pp. 91–92, 91.

Section BIBLIOGRAPHY

Appendix A

Implementation of Rotational Symmetry Search

This appendix outlines a method to find the degree of rotation of the head in the image slice. The method assumes that the centre of gravity (COG) of the image lies on the line of symmetry of the head. This assumption was justified experimenatly in chapter 4.2.2. The method used checks the symmetry of the image, for various angles, about the centre of gravity. The angle with the best symmetry gave the amount of rotation of the head. The method to be used is designed specifically for X-Ray CT image slices. A more general solution to finding the symmetry of an object is covered by Zabrodsky [55] and further information can be found in Otterloo[56]. The following outlines the optimized method used in this dissertation:

Given an Image $f(x, y)$ with the dimensions X,Y for width and height, the objective is to find the axis of symmetry of an object for which the centre of mass is approximately known. The centre of mass is given by the point (X_c, Y_c) and defines the centre of the new search co-ordinate system. From this centre point, lines at varying angles are calculated for symmetry. The line which corresponds to the highest symmetry measure will be considered the axis of symmetry. This Appendix reviews the mathematics involved with entire process.

The first requirement, is to find the equation of the lines in terms of the image co-ordinate system. It should be noted, that all equations are based in terms of the image co-ordinate system. For the image in Figure A.1, the line $\overline{P_1 P_2}$ is derived. Points $P1, P2$ are given by:

$$\begin{aligned} P_1 &= (X_c + Y_c \tan(\theta), 0) \\ p_2 &= (X_c - (Y - Y_c) \tan(\theta), Y) \end{aligned} \tag{A.1}$$

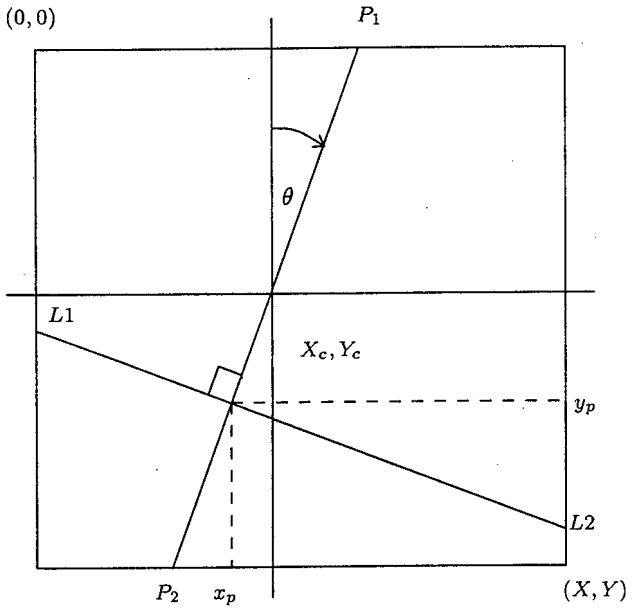


Figure A.1: Symmetry Co-ordinate system

From these points, the gradient of line $\overline{P_1P_2}$ is

$$\begin{aligned} m1 &= \frac{0 - Y}{(X_c + Y_c \tan(\theta)) - (X_c - (Y - Y_c)\tan(\theta))} \\ &= \frac{-1}{\tan(\theta)} \end{aligned} \tag{A.2}$$

and for the line equation of form $y = mx + c$, with $y = Y_c$ and $x = X_c$, c is

$$\begin{aligned} Y_c &= \frac{-1}{\tan(\theta)} X_c + c \\ c &= Y_c + \frac{1}{\tan(\theta)} X_c \end{aligned} \tag{A.3}$$

and the final equation for line $\overline{P_1P_2}$ is

$$y = \frac{-1}{\tan(\theta)} x + Y_c + \frac{1}{\tan(\theta)} X_c \tag{A.4}$$

All the lines perpendicular to the line in Equation A.4 are checked for symmetry. Before

this can be done, the equation of the perpendicular lines need to be found. The point of intersection of the perpendicular line with the symmetry line is given by x_p, y_p, with y_p being the running variable. Thus, given y_p the x_p is

$$\begin{aligned} y_p &= \frac{-1}{\tan(\theta)}x_p + Y_c + \frac{1}{\tan(\theta)}X_c \\ x_p &= \frac{y_p - (Y_c + \frac{1}{\tan(\theta)}X_c)}{\frac{-1}{\tan(\theta)}} \end{aligned} \tag{A.5}$$

The gradient of the perpendicular line, m_2 is given by $m_2 = -\frac{1}{m_1}$ which is

$$m_2 = \tan(\theta) \tag{A.6}$$

As before with the symmetry line, the equation for the perpendicular line is given by $y_\perp = \tan(\theta)x_\perp + c$

$$\begin{aligned} y_p &= \tan(\theta)x_p + c \\ &= \tan(\theta)\left[\frac{y_p - (Y_c + \frac{1}{\tan(\theta)}X_c)}{\frac{-1}{\tan(\theta)}}\right] + c \\ c &= y_p - \tan(\theta)\left[\frac{y_p - (Y_c + \frac{1}{\tan(\theta)}X_c)}{\frac{-1}{\tan(\theta)}}\right] \\ &= y_p + (\tan(\theta))^2\left[y_p - (Y_c + \frac{1}{\tan(\theta)}X_c)\right] \end{aligned} \tag{A.7}$$

Using this, the perpendicular line $\overline{L1L2}$ in Figure A.1 is defined as

$$y_\perp = \tan(\theta)x_\perp + y_p + (\tan(\theta))^2\left[y_p - (Y_c + \frac{1}{\tan(\theta)}X_c)\right] \tag{A.8}$$

Using A.8 the maximum extents of the line with respect to the image $f(x, y)$ can be found. The easiest way to achieve this, is to solve A.8 for the lines making up the border of the image.

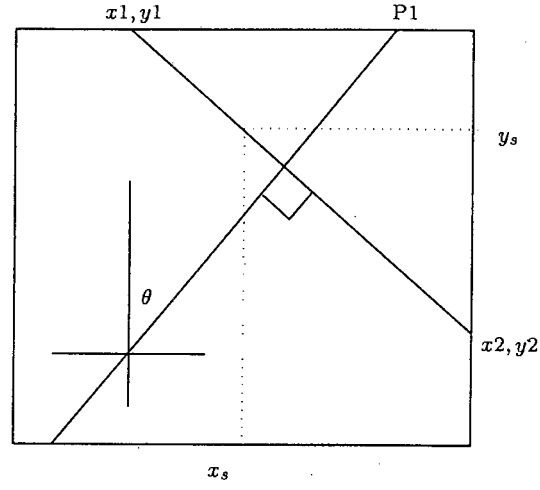


Figure A.2: Symmetry Search Area

$$
\begin{aligned}
y_\perp &= Y \\
x_\perp &= \frac{Y - (y_p + (\tan(\theta))^2 \left[y_p - (Y_c + \frac{1}{\tan(\theta)} X_c)\right])}{\tan(\theta)} && \text{(A.9)} \\
y_\perp &= 0 \\
x_\perp &= \frac{-(y_p + (\tan(\theta))^2 \left[y_p - (Y_c + \frac{1}{\tan(\theta)} X_c)\right])}{\tan(\theta)} && \text{(A.10)} \\
x_\perp &= X \\
y_\perp &= X \tan(\theta) + y_p + (\tan(\theta))^2 \left[y_p - (Y_c + \frac{1}{\tan(\theta)} X_c)\right] && \text{(A.11)} \\
x_\perp &= 0 \\
y_\perp &= y_p + (\tan(\theta))^2 \left[y_p - (Y_c + \frac{1}{\tan(\theta)} X_c)\right] && \text{(A.12)}
\end{aligned}
$$

The points for which both x and y co-ordinates are within the image, ie: $0 \leq x \leq X$ and $0 \leq y \leq Y$, are taken as the maximum extents of the line. There is an exception for the lines which pass through the origin, the point (X, Y) or both. Thus if there are duplicate points, only one is taken.

The equations for the symmetry line, perpendicular line, and the perpendicular line extents, have now been formulated. Using these, the centre extent and search extent for each perpendicular line can be defined. The centre extent is the points on the line that will be tested for as the point of symmetry. The search extent is the distance along each line, from the chosen centre point, that will be used in the calculation of the symmetry value. This is illustrated in Figure A.2.

The line at angle θ is assumed to be close to the line of symmetry, thus the point of intersection of the two x_p, y_p is assumed to be close to the point of symmetry for the line $y_\perp$. Thus, it is from this point, and not the line midpoint, that the centre search distance must be calculated. The distance between x_p, y_p and the maximum extent points $(x_1, y_1), (x_2, y_2)$ is found and the smaller one used to calculate the search area.

$$D_{min} = \min(\sqrt{(x_1 - x_p)^2 + (y_1 - y_p)^2}, \sqrt{(x_1 - x_p)^2 + (y_1 - y_p)^2}) \tag{A.13}$$

The final distance along the perpendicular line is obtain by taken a percentage of minimum length. The higher the percentage, the large the search area. As it is assumed the line at angle θ, is close to the symmetry line, the percentage lower bound determines how confident this assumption is. In theory it would be optimal to search the entire line, but due to processing limitations this is not possible. In practise, a percentage of 30% was found to be well above the lower bound and thus used.

$$D_{search} = D_{min} PCT \qquad (0 \leq PCT \leq 1) \tag{A.14}$$

For the checking of symmetry on line $y_\perp$, x is used as the running variable. The centre search x extents $x_{csearch}$ is now defined in terms of the search distance D_{search}.

$$x_p - \cos(\theta) D_{search} \leq x_{csearch} \leq x_p + \cos(\theta) D_{search} \tag{A.15}$$

The reason for using the smaller chord A.13 is now apparent. If the larger would be used and the percentage value chosen high enough, the search could exceed the image limits. Thus, by using the smaller chord this is prevented. The next value that needs to be found, is the extents over which the symmetry measure will be calculated. A measure of symmetry is found, by flipping the line about the centre point being evaluated and subtracting the corresponding points from either side. In order to do this, the extents over which the subtraction is done must be equal. The centre point does not necessarily bisect the perpendicular line and thus the shorter of the cord from centre point to the maximum line extents must be used. The equation is the same as in A.13 except x_p, y_p is replaced by a point x_s, y_s in the range $x_{csearch}$.

$$SymMeasure(x_s, y_\perp(x_s)) = \sum_{x=x_s-X_{length}}^{x_s} (f(x, y_\perp(x)) - f(2x_s - x, y_\perp(2x_s - x))) \quad \text{(A.16)}$$

where X_{length} is

$$X_{length} = \min(\sqrt{(x_1 - x_s)^2 + (y_1 - y_s)^2}, \sqrt{(x_1 - x_s)^2 + (y_1 - y_s)^2}) \cos(\theta) \quad \text{(A.17)}$$

The symmetry measure for all the x_s values in the range defined in A.15 and the smallest one taken as the symmetry point. This is repeated for all the lines perpendicular to the assumed symmetry line and the symmetry measures summed. This is then repeated for the angles $-45 \leq \theta \leq 45$ and the smallest, is considered as the symmetry line. It should be notes, that when choosing a perpendicular line $y_\perp$, y_p is the running variable, but when scanning the perpendicular, x_s is the running variable. The reason for this, is that the running variable is always chosen in correspondence with the acute angle. This increases the accuracy of the line being calculated when dealing with digital images.

Appendix B

Automatic Thresholding Algorithms - Review

In the chapter concerning intracranial region extraction, the first method tried used automatic thresholding algorithms. It was assumed that using these methods the skull/tissue interface could be segmented and later tracked, thus extracting the region (The value from the algorithms is a starting point, as explained in the original chapter). The standard algorithms assume that the input is an rectangular image, however for the purposes of this method, pixels with a CT values midway between air and water were not included. Values below these were not set in the histogram. The number of pixels is thus not the size of the image, but rather the contributing number of pixels to the histogram.

The threshold chosen by the automatic algorithms is critical as it will determine how much of the intracranial tissue is either removed or retained when tracking the skull. In the case of inclusion of skull there is no problem, however the exclusion of any tissue must be avoided. Trying to hand select a threshold from the histogram proves to be a difficult and costly (in a time sense), as it can only be done by experimentation by a human operator.

With regard to the above facts, a method for unsupervised segmentation of the skull and tissue are needed, as described previously. Many such methods do exists and before developing an entirely new method, these were tried. The majority, when applied to a thresholding problem, assume that the histogram is somewhat bimodal and base their result on some discriminatory fact between a foreground and background object. This is evident in the methods by Castleman [57] and Horn [25]. As an initial try, these methods were implemented and tried and the overall results obtained. A review of the algorithms tried are given next.

B.1 Discriminant Analysis

Otsu [19] proposed a method which automatically selects an optimum threshold by evaluating the threshold according to the discriminant analysis theory. As with most thresholds methods, the original image is divided into two classes C_0 (object) and C_1 (background) by some threshold level T. C_0 thus represents pixels with grey levels $[0, 1, ..., T]$ and C_1 represents pixels with grey levels $[T+1, T+2, ..., n]$. The image histogram is calculated and then normalised. Each bin has a value between $0-1$ and the total value of 1 and can be regarded as a probability distribution:

$$P_i = f_i/N \tag{B.1}$$

where f_i is the frequency of occurrence of grey level i, and N is the number of contributing pixels in the histogram. The probabilities of class occurrence ω and the class mean levels μ for each of the two classes are given by:

$$\omega_0 = Pr(C_0) = \sum_{i=0}^{T} p_i \tag{B.2}$$

$$\omega_1 = Pr(C_1) = \sum_{i=T+1}^{n} p_i \tag{B.3}$$

and

$$\mu_0 = \sum_{i=0}^{T} Pr(i|C_0) = \sum_{i=0}^{T} \frac{p_i}{\omega_0} = \frac{\mu(T)}{\omega(T)} \tag{B.4}$$

$$\mu_1 = \sum_{i=T+1}^{n} iPr(i|C_1) = \sum_{i=T+1}^{n} i\frac{p_i}{\omega_1} = \frac{\mu_{Tot} - \mu(T)}{1-\omega(T)} \tag{B.5}$$

where

$$\omega(T) = \sum_{i=0}^{T} P_i \tag{B.6}$$

and

$$\mu(T) = \sum_{i=0}^{T} iP_i \tag{B.7}$$

are the zeroth 1st-order cumulative moments of the histogram up to grey level T respectively, and

$$\mu_{tot} = u(n) = \sum_{i=0}^{n} iP_i \tag{B.8}$$

is the total mean grey level of the original image. It should be evident and is verified by Otsu that the following relationships are true for any choice of T:

$$\omega_0\mu_0 + \omega_1\mu_1 = \mu_{Tot}, \qquad \omega_0 + \omega_1 = 1 \tag{B.9}$$

The class variance are given by:

$$\sigma_0^2 = \sum_{i=0}^{T}(i-\mu_0)^2 Pr(i|C_0) = \sum_{i=0}^{T}(i-\mu_0)^2 \frac{p_i}{\omega_0} \tag{B.10}$$

$$\sigma_1^2 = \sum_{i=T+1}^{n}(i-\mu_1)^2 Pr(i|C_1) = \sum_{i=T+1}^{n}(i-\mu_1)^2 \frac{p_i}{\omega_1} \tag{B.11}$$

Discriminant criterion measures (or measure of class separability) are introduced to evaluate the choice of threshold T:

$$\lambda = \frac{\sigma_B^2}{\sigma_W^2}, \qquad \kappa = \frac{\sigma_{Tot}^2}{\sigma_W^2}, \qquad \eta = \frac{\sigma_B^2}{\sigma_T^2} \tag{B.12}$$

where

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_{Tot})^2 + \omega_1(\mu_1 - \mu_{Tot})^2 \tag{B.13}$$

$$= \omega_0\omega_1(\mu_1 - \mu_0)^2 \text{using eqn B.9} \tag{B.14}$$

$$\sigma_W^2 = \omega_0\sigma_0^2 + \omega_1\sigma_1^2 \tag{B.15}$$

and

$$\sigma_{Tot}^2 = \sum_{i=0}^{n}(i-\mu_{Tot})^2 p_i \tag{B.16}$$

where σ_B^2,σ_W^2 and σ_{Tot}^2 are the between-class variance, within-class variance and the total variance respectively. Otsu assumes that adequately thresholded classes would be segmented

in grey levels and, conversely, a threshold giving the the best separation in classes in grey levels would be the most appropriate threshold. The problem is thus reduced to one of finding a threshold level T that maximises one of the criterion measures in equation B.12. Using the fact that the following relation always holds:

$$\sigma_W^2 + \sigma_B^2 = \sigma_{Tot}^2 \tag{B.17}$$

the discriminant criteria maximising λ, κ or η are equivalent. As σ_{Tot}^2 is independent of the chosen threshold T and σ_B^2 is based on 1st-order statistics (class) means rather than 2nd-order statistics (as with σ_W^2), η is the simplest measure with respect to T. It is therefore adopted as the criterion measure to evaluate the chosen threshold T.

The optimum threshold is selected by:

$$\eta(T) = \sigma_B^2(T)/\sigma_{Tot}^2 \tag{B.18}$$

Using equation B.13, and substituting equations B.4 to B.8:

$$\sigma_B^2(T) = \frac{[\mu_{Tot}\omega(T) - \mu(T)]^2}{\omega(T)[1 - \omega(T)]} \tag{B.19}$$

As σ_{Tot}^2 is independent of T, only $\sigma_B^2(T)$ is considered. The optimal threshold T_{opt} occurs when $\sigma_B^2(T)$ is a maximum, that is:

$$\sigma_B^2(T_{opt}) = \max_{0 \le T < n} \sigma_B^2(T) \tag{B.20}$$

B.2 Entropy-Based Method

Pun [58] represented two algorithms for threshold selection based on the entropy of the histogram. Kapur, Sahoo and Wong [20] analysed and corrected Puns algorithm and proposed a new thresholding method also based on the entropy of the histogram. If $f_1, f_2, ..., f_n$ are the image $f's$ grey level frequencies, then let $p_1, p_2, ..., p_n$ be the grey level probability distribution with

$$p_i = f_i/N, \qquad \sum_{i=0}^{n} f_i = N \tag{B.21}$$

where N is the total number of pixels contributing to the histogram and $n+1$ is the number of grey levels in the image f. Pun defines the entropy of the histogram as

$$H_n = -\sum_{i=0}^{n} p_i \ln p_i \tag{B.22}$$

Two probability distributions are derived for the above and below threshold classes. A: For discrete values 0 to T and B: For discrete values $T+1$ to n:

$$A : \frac{p_1}{P_0(T)}, \frac{p_2}{P_0(T)}, ..., \frac{p_T}{P_0(T)} \qquad B : \frac{p_{T+1}}{P_1(T)}, \frac{p_{T+2}}{P_1(T)}, ..., \frac{p_n}{P_1(T)} \tag{B.23}$$

where $P_0(T)$ and $P_1(T)$ are the class probabilities given by:

$$P_0(T) = \sum_{i=0}^{T} p_i, \qquad P_1(T) = \sum_{i=T+1}^{n} p_i = 1 - P_0(T) \tag{B.24}$$

From equation B.22 the class entropies are defined as:

$$H_0(T) = -\sum_{i=0}^{T} p_i \ln p_i, \qquad H_1(T) = -\sum_{i=T+1}^{n} p_i \ln p_i = H_n - H_0(T) \tag{B.25}$$

The entropies associated with the class distributions are then:

$$\begin{aligned} H_A(T) &= -\sum_{i=0}^{T} \frac{p_i}{P_0(T)} \ln(\frac{p_i}{P_0(T)}) \\ &= \ln P_0(T) + \frac{H_0(T)}{P_0(T)} \end{aligned} \tag{B.26}$$

and

$$H_B(T) = -\sum_{i=T+1}^{n} \frac{p_i}{P_1(T)} \ln(\frac{p_i}{P_1(T)})$$

$$= \ln(1 - P_0(T)) + \frac{H_1(T)}{1 - P_0(T)} \tag{B.27}$$

the sum of the entropies $H_A(T)$ and $H_B(T)$ is defined as $\psi(T)$. From equations B.26 and B.27:

$$\psi(T) = \ln P_0(T)P_1(T) + \frac{H_0(T)}{P_0(T)} + \frac{H_1(T)}{P_1(T)} \tag{B.28}$$

$$\psi(T) = \ln P_0(T)(1 - P_0(T)) + \frac{H_0(T)}{P_0(T)} + \frac{H_n - H_0(T)}{1 - P_0(T)} \tag{B.29}$$

$$\tag{B.30}$$

The value of T for which $\psi(T)$ is a maximum is the chosen threshold value. This is when the maximum information between the object and background distributions is obtained.

B.3 Moment Preserving Method

Tsai [59] proposes a moment-preserving approach to bilevel thresholding:

The j^{th} moment m_j of image f is defined as

$$m_j = \frac{1}{N} \sum_x \sum_y f^j(x, y) \tag{B.31}$$

where

N = The total number of pixel in the image, and

$f(x, y)$ = The pixel's grey level value at (x, y).

Let f_i be the frequency of occurrence of gray level i and $p_i = f_i/N$. Moments can be computed from the histogram of image as follows: (In this case N is the number of contributing pixels, not the number of pixels in the image)

$$m_j = \frac{1}{N} \sum_{i=0}^{n} f_i(i)^j$$

$$= \sum_{i=0}^{n} p_i(i)^j \tag{B.32}$$

Tsai defines m_0 to be 1. If the image f is thought of as a blurred version of an ideal binary image consisting of pixels with grey levels z_0 and z_1 where $z_0 < z_1$, then the moment preserving algorithm attempts to select a threshold value T such that if all the below threshold level pixels in f are replaced by z_0 and all the above by z_1, the first three moments of f are preserved in the resulting binary image. The binary image can be thus considered as an ideal un-blurred version of the image f.

Let the binary image binary image be denoted by g and let p_0 and p_1 be the fractions of the below threshold and above threshold in f, respectively. the first the moments of g are then:

$$m'_j = \sum_{i=0}^{1} p_i (z_i)^j \qquad \text{for } j = 1, 2, 3. \tag{B.33}$$

Note that $p_0 + p_1 = 1$. The first three moments are preserved in g by letting

$$m'_j = m_j \qquad \text{for } j = 1, 2, 3. \tag{B.34}$$

Using equations B.33 and B.34:

$$\begin{aligned} m_0 &= p_0 z_0^0 + p_1 z_1^0 && \text{(i) ...zeroth} \\ m_1 &= p_0 z_0^1 + p_1 z_1^1 && \text{(ii) ...mean} \\ m_2 &= p_0 z_0^2 + p_1 z_1^2 && \text{(iii) ...variance} \\ m_3 &= p_0 z_0^3 + p_1 z_1^3 && \text{(iv) ...skewness} \end{aligned} \tag{B.35}$$

Solving the Moment-Preserving Equations Noting that $m_0 = 1$ equations B.35 (i) and (ii) can be rewritten as

$$\begin{bmatrix} 1 \\ m_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ z_0 & z_1 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \end{bmatrix} \tag{B.36}$$

The first three moments (m_1, m_2 and m_3) are obtained from the histogram using equationB.33. Equation B.36 is then solved using Cramer's rule:

Section B.3: Moment Preserving Method

$$\begin{aligned} p_0 &= \frac{1}{z_1 - z_0} \begin{vmatrix} 1 & 1 \\ m_1 & z_1 \end{vmatrix} \\ p_1 &= 1 - p_0 \end{aligned} \tag{B.37}$$

Equation B.37 shows that the evaluation of p_0 and p_1 us dependent on z_0 and z_1 which can be solved as follows:

1. The following linear equations are solved to obtain a set of auxiliary values c_0 and c_1:

$$\begin{aligned} c_0 m_0 + c_1 m_1 &= -m_2 \\ c_0 m_1 + c_1 m_2 &= -m_3 \end{aligned} \tag{B.38}$$

Using Cramers Rule:

$$\begin{aligned} c_0 &= \frac{1}{m_2 - m_1^2} \begin{vmatrix} -m_2 & m_1 \\ -m_3 & m_2 \end{vmatrix} \\ c_1 &= \frac{1}{m_2 - m_1^2} \begin{vmatrix} m_0 & -m_2 \\ m_1 & -m_3 \end{vmatrix} \end{aligned} \tag{B.39}$$

2. The following polynomial is then solved to obtain z_0 and z_1:

$$z^2 + c_1 z + c_0 = 0 \tag{B.40}$$

giving:

$$z_0 = \frac{1}{2}\left[-c_1 - \sqrt{(c_1^2 - 4c_0)}\right] \tag{B.41}$$

$$z_1 = \frac{1}{2}\left[-c_1 + \sqrt{(c_1^2 - 4c_0)}\right] \tag{B.42}$$

the four equations described by B.35 are then solved for p_0 and p_1 using the method described above. The threshold T is chosen by determining the p_0-tile of the histogram of image f such that

$$p_0 = \frac{1}{N} \sum_{z_j \leq T} p_j \tag{B.43}$$

If no discrete grey level exists which corresponds exactly with the p_0-tile of the histogram, the closest grey level to it is chosen as the calculated threshold.

B.4 Maximum Correlation Method

Brink [60, 61] presents a threshold selection technique using a correlation criterion. Consider two sets of data $X = x_1, x_2, .., x_s$ and $Y = y_1, y_2, .., y_s$ each containing s elements, where x_i and y_i are their data values, $p(x_i)$ and $p(y_i)$ their associated probabilities and $p(x_i y_i)$ the probability associated with their product.

The coefficient of correlation ρ for the two sets of data X and Y is defined as:

$$\rho_{XY} = \frac{E_{XY} - E_X E_Y}{\sqrt{V_X V_Y}} \tag{B.44}$$

where E_X and E_Y are the mathematical expectations of the respective sets of data, E_{XY} is the expected value of their product and V_X and V_Y are the variance of X and Y, given by:

$$E_X = \sum_{i=1}^{s} x_i p(x_i) \tag{B.45}$$

$$E_Y = \sum_{i=1}^{s} y_i p(y_i) \tag{B.46}$$

$$E_{XY} = \sum_{i=1}^{s} x_i y_i p(x_i y_i) \tag{B.47}$$

$$V_X = E_{XX} - [E_X]^2 \tag{B.48}$$

$$V_Y = E_{YY} - [E_Y]^2 \tag{B.49}$$

where

$$E_{XX} = \sum_{i=1}^{s} x_i^2 p(x_i) \tag{B.50}$$

$$E_{YY} = \sum_{i=1}^{s} y_i^2 p(x_i) \tag{B.51}$$

The correlation coefficient ρ_{XY} can take on values from -1 to 1 with -1 corresponding to an inverse correlations, 1 an absolute correlation an 0 no correlations between X and Y. Brink

applies the correlation criteria to an image as follows:

Let X represent the set of grey level values g of the original image and Y represent the grey levels (black or white) of the thresholded bilevel image. Brink uses the below and above threshold means $\mu_0(T)$ and $\mu_1(T)$ of the original image X to represent the 'black' and 'white' levels of the bilevel image Y. These means are defined and used by Kittler and Illingworth [62] (see later) and are given as:

$$\mu_0(T) = (\sum_{i=0}^{T} ip_i)/(\sum_{i=0}^{T} p_i) \tag{B.52}$$

$$\mu_1(T) = (\sum_{i=T+1}^{n} ip_i)/(\sum_{i=T+1}^{n} p_i) \tag{B.53}$$

where T is the threshold level with $0 \leq n$, and p_i is the probability of grey level i obtained from the normalised histogram:

$$p_i = f_i/N \tag{B.54}$$

where N is the number of contributing pixels in the image, and f_i is the frequency of occurrence of grey level i.

Only E_X and V_X are independent of the threshold as they are obtained directly from the original image X. The coefficient of correlation between the two images X and Y for threshold T becomes:

$$\rho_{XY}(T) = \frac{E_{XY}(T) - E_X E_Y(T)}{\sqrt{V_X V_Y(T)}} \tag{B.55}$$

where

$$E_X = \sum_{i=0}^{n} i_i p_i \tag{B.56}$$

$$E_Y(T) = \sum_{i=0}^{T} \mu_0(T) p_i + \sum_{i=T+1}^{n} \mu_1(T) p_i \tag{B.57}$$

$$E_{XY}(T) = \sum_{i=0}^{T} i\mu_0(T)p_i + \sum_{i=T+1}^{n} i\mu_1(T)p_i \quad \text{(B.58)}$$

$$E_{XX} = \sum_{i=0}^{n} i^2 p_i \quad \text{(B.59)}$$

$$E_{YY} = \sum_{i=0}^{T} \mu_0^2(T)p_i + \sum_{i=T+1}^{n} \mu_1^2(T)p_i \quad \text{(B.60)}$$

with the variance given by:

$$V_X = E_{XX} - [E_X]^2 \quad \text{(B.61)}$$

$$V_Y(T) = E_{YY}(T) - [E_Y(T)]^2 \quad \text{(B.62)}$$

The threshold maximising the correlation function $\rho_{XY}(T)$ is found by an iterative process and chosen as the desired threshold. Cseke and Fazekas [63] comment that Brink's method leads to the same function to be maximised as Otsu's method, despite their different approaches. They reason this as follows:

From Otsu's method [19]:

$$\eta(T) = \frac{\sigma_B^2(T)}{\sigma_{Tot}^2} \qquad \text{from equation B.18, and}$$

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_{Tot})^2 + \omega_1(\mu_1 - \mu_{Tot})^2 \qquad \text{from equation B.13}$$

Using Brink's method, the pixels belonging to class C_0 take grey level μ_0 and the pixels belonging to class C_1 take grey level μ_1, therefor the following statements are true:

$$E_Y(T) = \mu_0(T)\omega_0(T) + \mu_1(T)\omega_1(T) = E_X \quad \text{(B.63)}$$

$$E_{YY}(T) = \mu_0^2(T)\omega_0(T) + \mu_1^2(T)\omega_1(T) = E_{XY}(T) \quad \text{(B.64)}$$

therefore

$$E_{XY}(T) = E_X E_Y = E_{YY} - E_Y^2 = V_Y(T) \quad \text{(B.65)}$$

Cseke and Fazeka thus show that the separability quotient $\eta(T)$ used by Otsu's method is the square of the correlation coefficient ρ_{XY} used in Brink's method:

$$\rho_{XY}^2 = \frac{V_Y^2(T)}{V_X V_Y(T)} = \eta(T) \tag{B.66}$$

Brink [64] agrees with Cseke and Fazeka but states that this is only true if C_0 and C_1 are represented by μ_0 and μ_1 respectively, which is not always true. Brink claims that processing time an be reduced by pre-selecting black and white values from the output image which constitutes an advantage over Otsu's method. He also notes that Brinks's method will select identical thresholds to Otsus method (using μ_0 and μ_1), even if different black and white levels are used for the output image.

B.5 Minimum Error Thresholding

Kittler and Illingworth[62] propose a method which assumes that the respective object and background populations are normally distributed with distinctive means and standard deviations. This was an interesting method to use, as it seems to use the model for CT brain images described by Lei[44] (Although the skull does not have a clearly defined Gaussian distribution, it was hoped the Gaussian distribution of the intracranial region would lead to an optimum threshold).

Let image $f's$ pixels assume grey levels, g, from $[0, n]$. The image histogram is defined as $h(g)$ and the probability density function of the mixture population comprising the grey levels of object and background pixels is then $h(g)/N$, which is defined as $p(g)$. It is assumed that each of the two components $p(g|0)$ and $p(g|1)$ (0 and 1 representing the background and foreground classed respectively) are normally distributed with mean μ_0 and μ_1, standard deviations σ_0 and σ_1 and *a priori* probabilities P_0 and P_1, that is:

$$p(g) = \sum i = 0^1 P_i p(g|i) \tag{B.67}$$

where

$$p(g|i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp - \frac{(g-\mu_i)^2}{2\sigma_i^2} \qquad \text{Normal Distribution} \tag{B.68}$$

There exists a grey level T for which grey levels g satisfy

$$P_0 p(g|0) > P_1 p(g|1) \quad \text{for } g \leq T \text{ and} \tag{B.69}$$

$$P_0 p(g|0) < P_1 p(g|1) \quad \text{for } g > T \tag{B.70}$$

This threshold T is the Bayes Minimum Error threshold at which the two modelled normal distributions best fit the population mixture. Substituting equation B.68 and taking the logarithm of both sides, the threshold value can be chosen by solving the quadratic equation:

$$\frac{(g-\mu_0)^2}{\sigma_0^2} + 2\log\sigma_0^2 - 2\log P_0 = \frac{(g-\mu_1)^2}{\sigma_1^2} + 2\log\sigma_1^2 - 2\log P_1 \tag{B.71}$$

but μ_0 and μ_1, σ_0 and σ_1 and P_0 and P_1 of the mixture density $p(g)$ are usually not known. These parameters can be estimated from the histogram $h(g)$ using fitting techniques, but this is computationally expensive (see chapter on fitting Gaussian). Kittler and Illingworth propose a simpler technique for finding the optimum threshold T:

Let the image f be thresholded with an arbitrary grey level T, with each of the two resulting pixel populations being modelled by normal densities $h(g|0,T)$ and $h(g|1,T)$. Parameters $\mu_0(T)$, $\mu_1(T)$, $\sigma_0(T)$ $\sigma_1(T)$, $P_0(T)$ and $P_1(T)$ are estimated from the histogram as follows:

$$P_0(T) = \sum_{g=0}^{T} h(g) \quad , \quad P_1(T) = \sum_{g=T+1}^{n} h(g) \tag{B.72}$$

$$\mu_0(T) = \frac{(\sum_{g=0}^{T} h(g)g)}{P_0(T)} \quad , \quad \mu_1(T) = \frac{(\sum_{g=T+1}^{n} h(g)g)}{P_1(T)} \tag{B.73}$$

$$\sigma_0^2 = \frac{(\sum_{g=0}^{T}(g-\mu_0(T))^2 h(g))}{P_0(T)} \quad , \quad \sigma_1^2 = \frac{(\sum_{g=T+1}^{n}(g-\mu_1(T))^2 h(g))}{P_1(T)} \tag{B.74}$$

The conditional probability, $e(g,T)$ of grey level g being replaced in the image by a correct binary value is:

$$e(g,T) = \frac{h(g|i,T)P_i(T)}{h(g)} \qquad \text{where } i = \begin{cases} 0 & g \leq T \\ 1 & g > T \end{cases} \tag{B.75}$$

where $h(g|i,T)$ is now defined as:

Section B.5: Minimum Error Thresholding

$$h(g|i,T) = \frac{1}{\sqrt{2\pi}\sigma_i(T)} \exp - \frac{(g-\mu_i(T))^2}{2\sigma_i^2(T)} \tag{B.76}$$

In the authors analysis the denominator in equation B.75 is ignored since the image histogram $h(g)$ is independent of both i and T. Taking the logarithm of the numerator of equation B.75, adding $\log\sqrt{2\pi}$ and multiplying by -2:

$$\varepsilon(g,T) = \frac{(g-\mu_i(T))^2}{\sigma_i^2} + 2\log\sigma_i(T) - 2\log P_i(T) \qquad \text{where } i = \begin{cases} 0 & g \le T \\ 1 & g > T \end{cases} \tag{B.77}$$

Kittler and Illingworth introduce a criterion function $J(T)$ which characterises the average classification error for the whole image:

$$J(T) = \sum_{g=0}^{n} h(g)\varepsilon(g,T) \tag{B.78}$$

Substituting equation B.76:

$$\begin{aligned} J(T) = &\sum_{g=0}^{T} h(g)\left[\frac{(g-\mu_0(T))^2}{\sigma_0^2} + 2\log\sigma_0(T) - 2\log P_0(T)\right] \\ &+ \sum_{g=T+1}^{n} h(g)\left[\frac{(g-\mu_1(T))^2}{\sigma_1^2} + 2\log\sigma_1(T) - 2\log P_1(T)\right] \end{aligned} \tag{B.79}$$

substituting in equations B.72, B.73 and B.74 into equation B.79:

$$\begin{aligned} J(T) = 1 &+ 2[P_0(T)\log\sigma_0(T) + P_1(T)\log\sigma_1(T)] \\ &-2[P_0(T)\log P_0(T) + P_1(T)\log P_1(T)] \end{aligned} \tag{B.80}$$

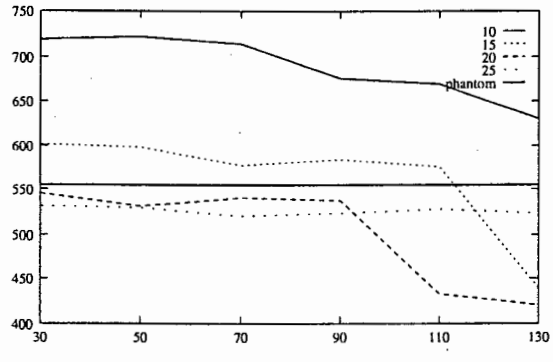
For a given threshold level T the criterion function $J(T)$ indirectly reflects the amount of overlap between the Gaussian models of the object and background populations. The value of threshold T minimising the criterion function $J(t)$ is the minimum error threshold, because with this threshold value the overlap between the density functions of the object and background is the smallest, effectively yielding the smallest classification error.

Appendix C

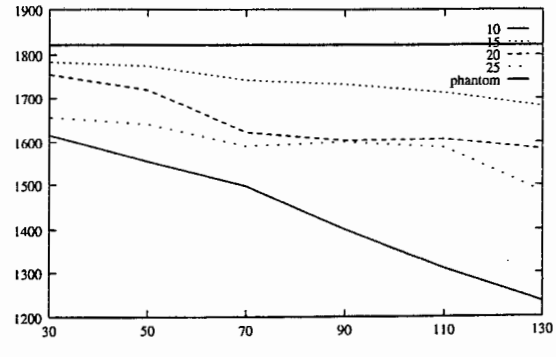
Results of SRG 2D and 3D Segmentation Algorithms

C.1 Effect of phantom grey level distance from tissue mean - SRG2D

Figure C.1: SRG2D: Phantom size versus Noise for varying phantom mean distances from main tissue mean - The x-axis and $y-axis$ represents the noise variance σ^2 and SRG volume (given in voxels - multiply by voxel volume to give size in mm^3), respectively.

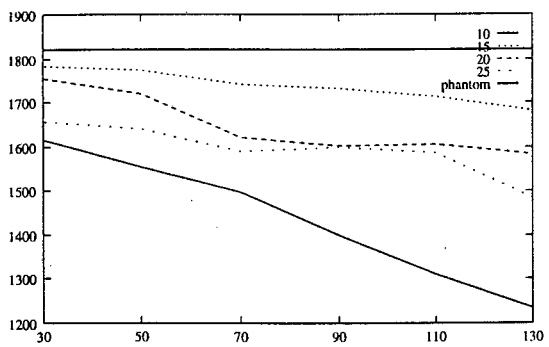


(a) SRG2D: Input phantom radius = 8.

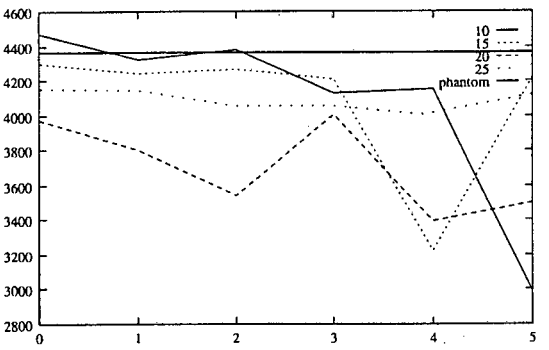


(b) SRG2D: Input phantom radius = 12.

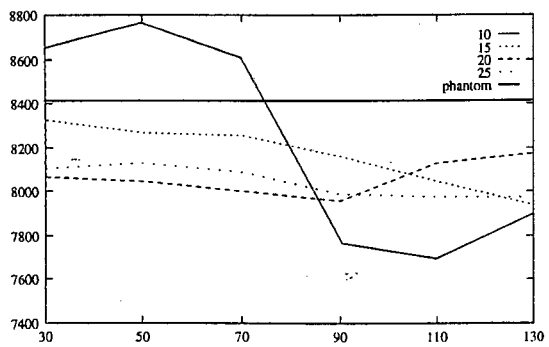
Section C.1: Effect of phantom grey level distance from tissue mean - SRG2D



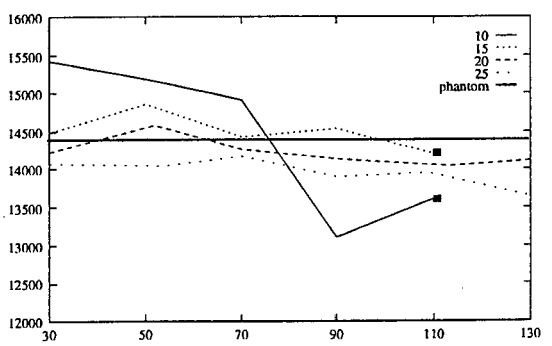
(c) SRG2D: Input phantom radius = 12.



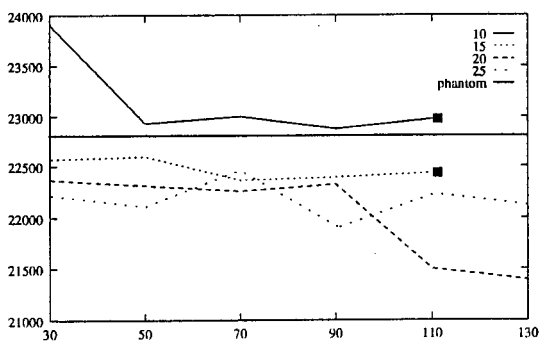
(d) SRG2D: Input phantom radius = 16.



(e) SRG2D: Input phantom radius = 20.



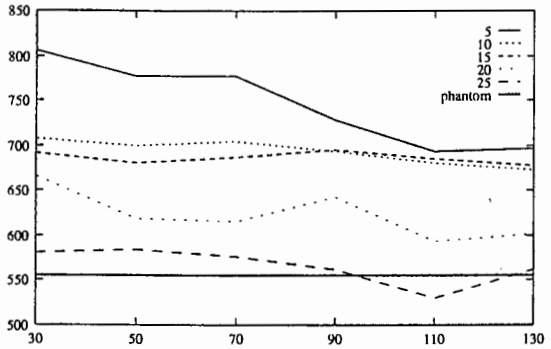
(f) SRG2D: Input phantom radius = 24.



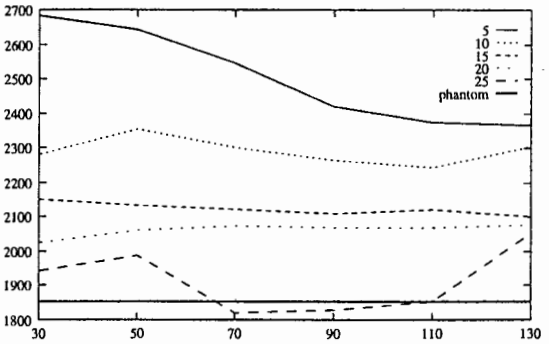
(g) SRG2D: Input phantom radius = 28.

C.2 Effect of phantom grey level distance from tissue mean - SRG 3-D

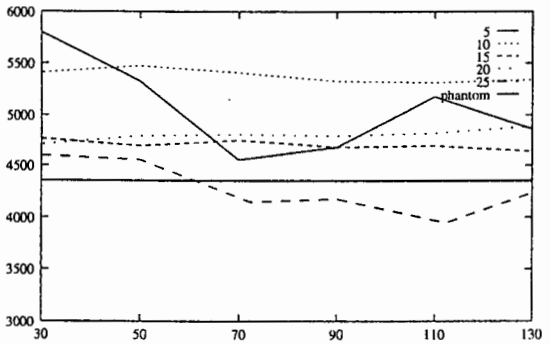
Figure C.2: SRG3D: Phantom size versus Noise for varying phantom mean distances from main tissue mean. The x-axis and $y-axis$ represents the noise variance σ^2 and SRG volume (given in voxels- multiply by voxel volume to give size in mm^3), respectively.



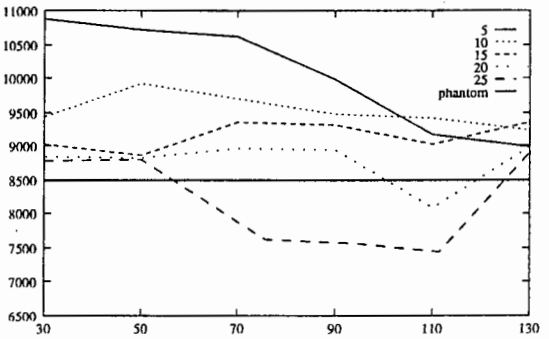
(a) SRG3D: Input phantom radius = 8.



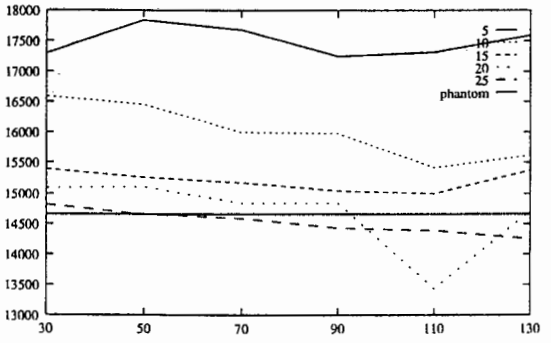
(b) SRG3D: Input phantom radius = 12.



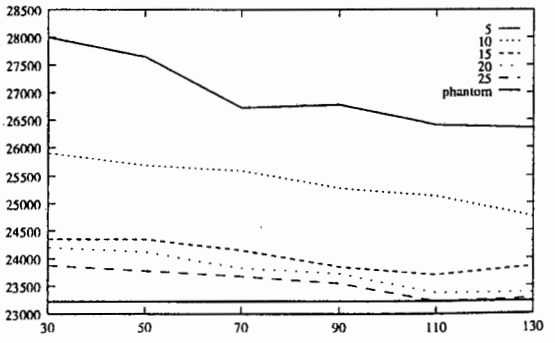
(c) SRG3D: Input phantom radius = 16.



(d) SRG3D: Input phantom radius = 20.



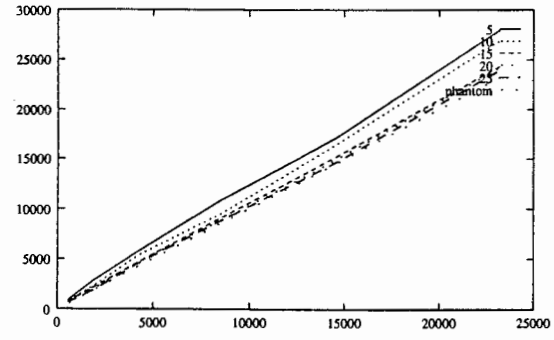
(e) SRG3D: Input phantom radius = 24.



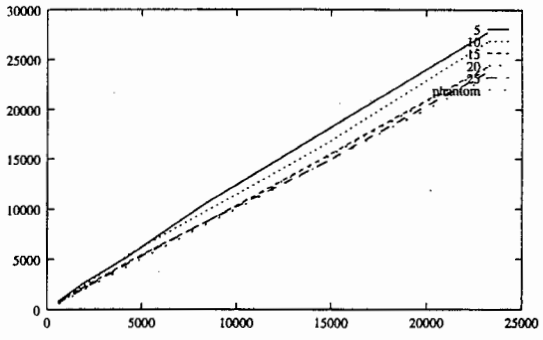
(f) SRG3D: Input phantom radius = 28.

C.3 Effect of phantom grey level distance from tissue mean - SRG3D

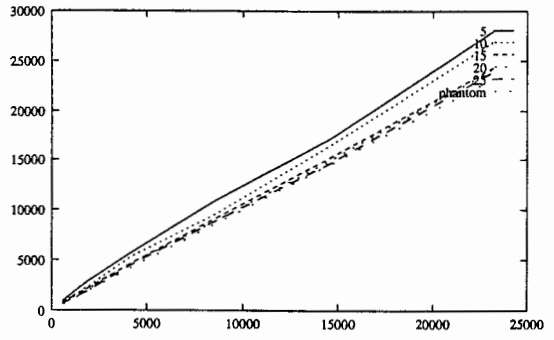
Figure C.3: SRG3D: Input versus Ouput Phantom sizes for varying phantom mean distances from main tissue mean. Each plot represnts a different noise value. - The x-axis and $y-axis$ represents the input Phantom Volume and SRG Output volume (given in vixels - multiply by voxel volume to give size in mm^3), respectively.



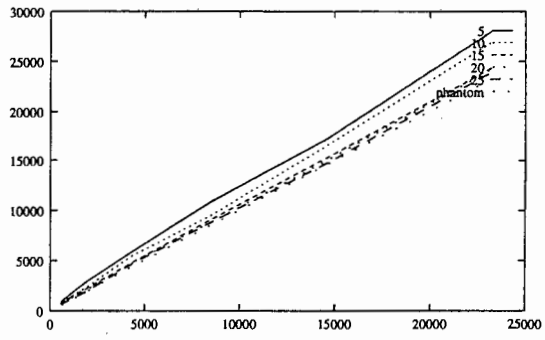
(a) SRG 3-D Input Noise level = 30



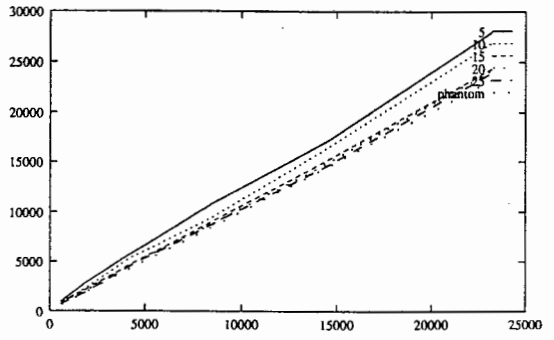
(b) SRG 3-D Input Noise level = 50



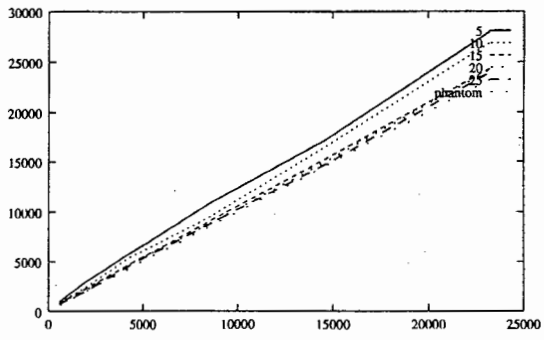
(c) SRG 3-D Input Noise level = 70



(d) SRG 3-D Input Noise level = 90



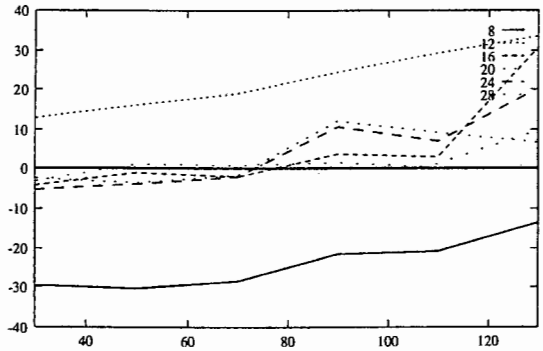
(e) SRG 3-D Input Noise level = 110



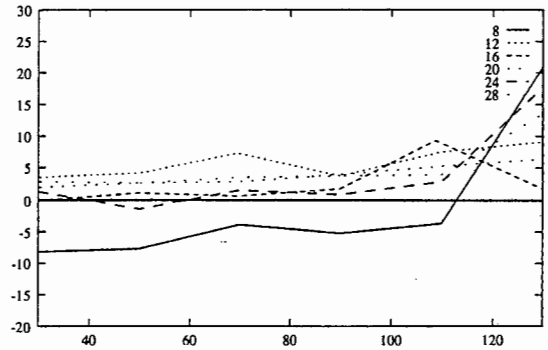
(f) SRG 3-D Input Noise level = 130

C.4 Percentage Error for various a fixed mean and varying Phantom Size - SRG 2-D

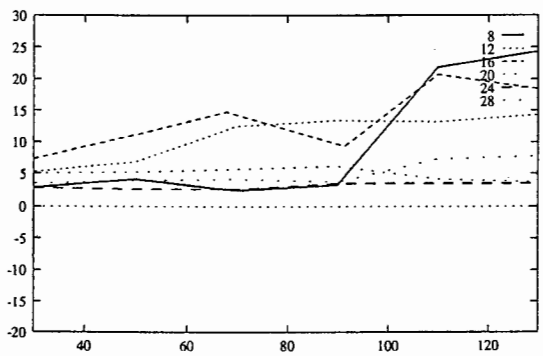
Figure C.4: SRG2D : Percentage error versus Noise for phantom sizes. Each plot represents a fixed grey level distance from the main tissue mean. The x-axis and $y-axis$ represents the noise variance σ^2 and SRG percentage volume error, respectively.



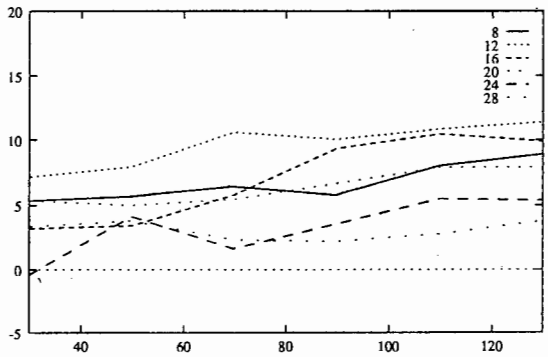
(a) SRG2D: Grey level distance from Tissue mean = 10.



(b) SRG2D: Grey level distance from Tissue mean = 15.



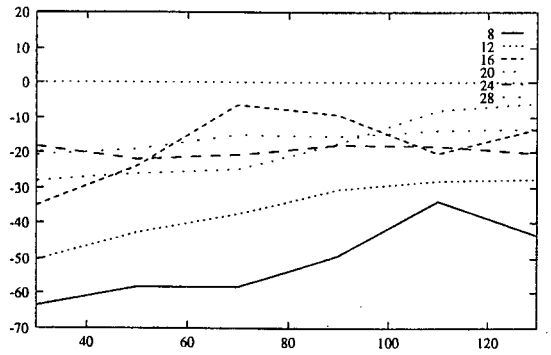
(c) SRG2D: Grey level distance from Tissue mean = 20.



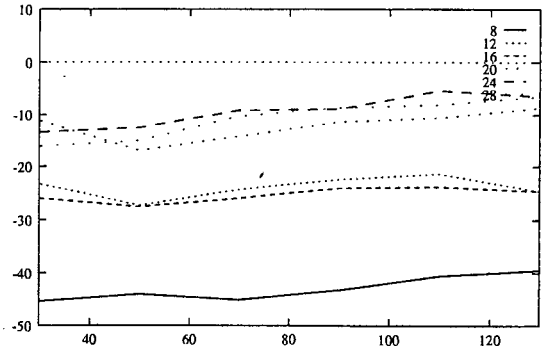
(d) SRG2D: Grey level distance from Tissue mean = 25.

C.5 Percentage Error for various a fixed mean and varying Phantom Size- SRG 3-D

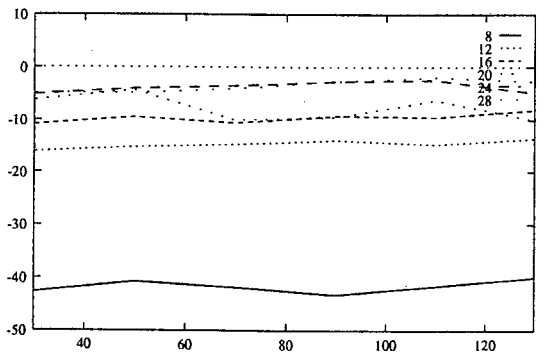
Figure C.5: SRG3D : Percentage error versus Noise for phantom sizes. Each plot represents a fixed grey level distance from the main tissue mean. The x-axis and $y-axis$ represents the noise variance σ^2 and SRG percentage volume error, respectively.



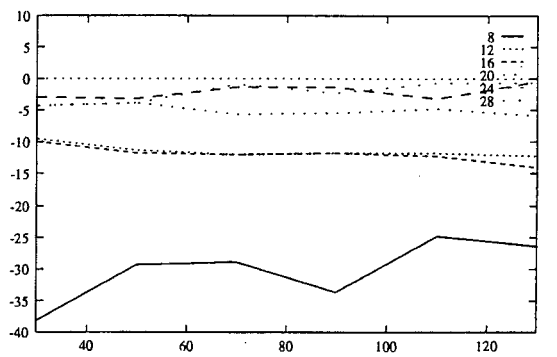
(a) SRG3D: Distance from Tissue mean=5



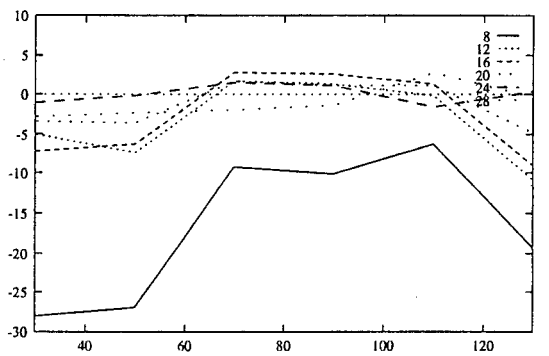
(b) SRG3D: Distance from Tissue mean=10



(c) SRG3D: Distance from Tissue mean=15



(d) SRG3D: Distance from Tissue mean=20



(e) SRG3D: Distance from Tissue mean=25

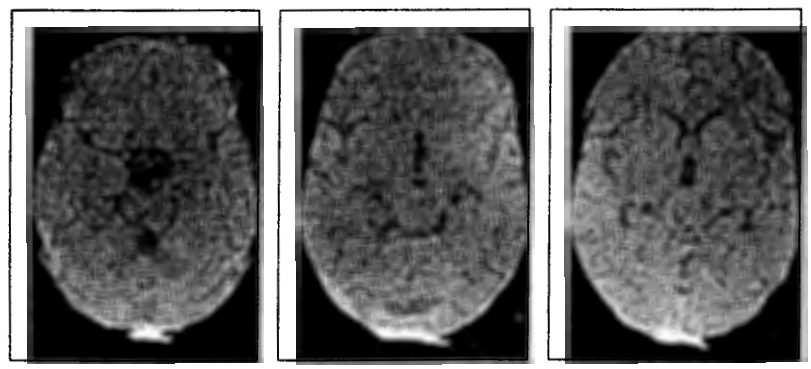
Appendix D

Image Sets used in dissertation

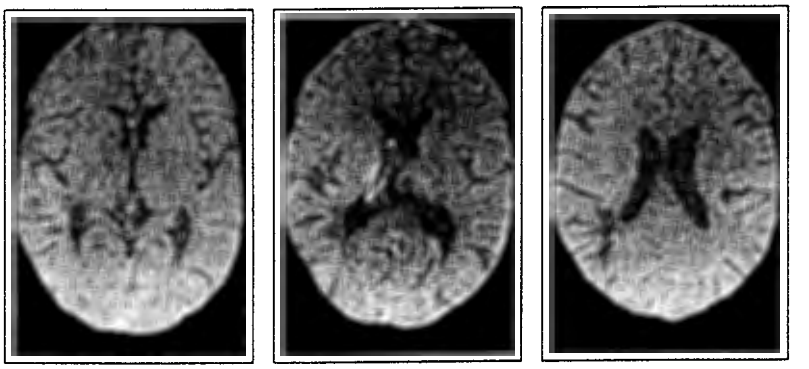
The images used were 512×512 with a voxel resolution of $0.46875mm \times 0.46875mm \times 2mm$. The $0.46875mm \times 0.46875mm$ corresponds to the $x - y$ dimensions and the $2mm$ is the slice thickness. The photon energy used was $120keV$.

D.1 Image Set 1

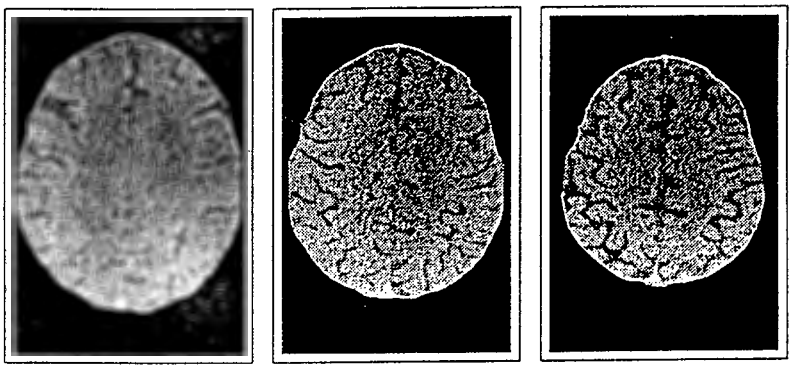
Figure D.1: Image Set 1 - Good Definition of ventricular system, useful for discriminating CSF from rest of tissue.



(a) slice 0 (b) slice 2 (c) slice 4



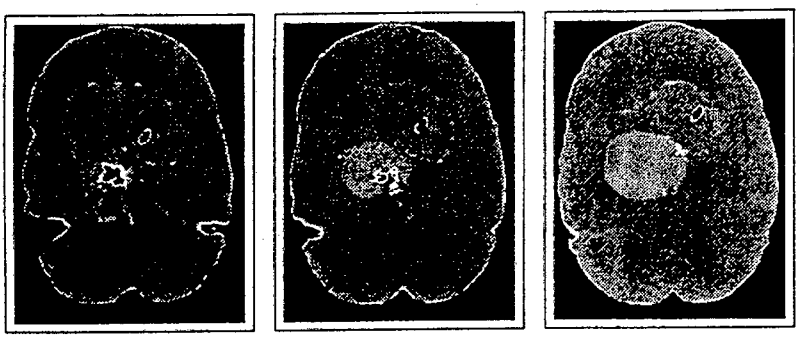
(d) slice 6 (e) slice 8 (f) slice 10



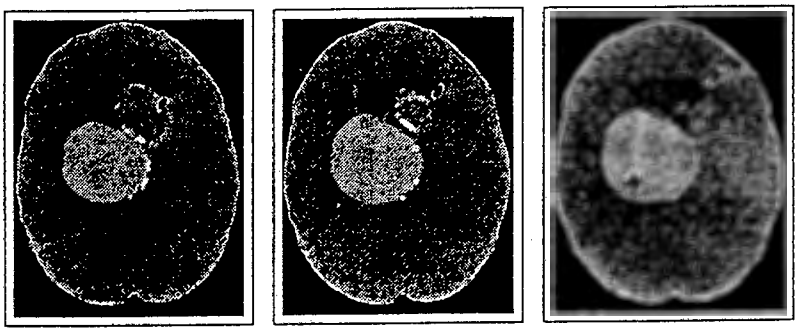
(g) slice 12 (h) slice 14 (i) slice 16

D.2 Image Set 2

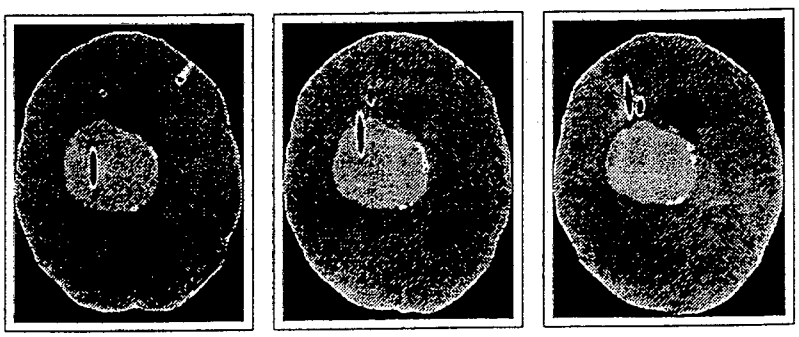
Figure D.2: Image Set 2 - Patient has large cyst, useful for comparing hand and SRG 2-D / 3-D volumes.



(a) slice 0 (b) slice 2 (c) slice 4

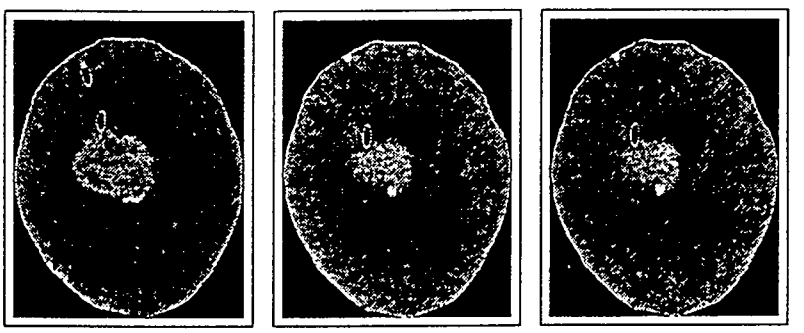


(d) slice 6 (e) slice 8 (f) slice 10

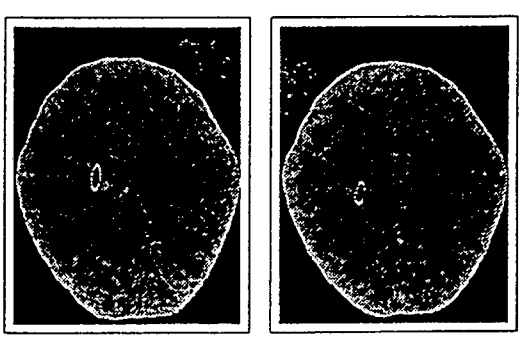


(g) slice 12 (h) slice 14 (i) slice 16

Section D.2: Image Set 2



(j) slice 18 (k) slice 20 (l) slice 22



(m) slice 24 (n) slice 26