Automatic 2D and 3D Segmentation of Liver from Computerised Tomography

by

Alun Evans

A THESIS SUBMITTED TO THE UNIVERSITY OF LONDON

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MEDICAL PHYSICS AND BIOENGINEERING

UNIVERSITY COLLEGE LONDON

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

Statement of originality

I, Alun Thomas Evans, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Signed: . .....

Alun Evans

Acknowledgements

This thesis, and the work presented within it, would not have seen the light of day without the assistance of many people. I am thankful to my supervisor, Professor Andrew Todd-Pokropek, for allowing me to start the research and for his assistance, support and corrections. My thanks also to the MIAS-IRC, who funded this research.

Special thanks must go to Dr. Tryphon Lambrou, who sat next to me over the years and who I turned to repeatedly for help or advice. Similarly, I would like to thank Cliff Ruff for his frequent assistance, and Professor Alf Linney for his help and proof-reading.

Diolch o galon i fy rhieni, roedd eich cymorth cyson yn werth y byd.

Y también quiero agradecer mi querida, por tu ayuda y paciencia.

List of publications

Journal Papers currently under review:

Evans, A., Lambrou, T., Linney, A.D., Todd-Pokropek, A., (2006) "Automatic 3D segmentation of the liver from CT, using a discrete deformable model" under review at *Journal of Computerized Medical Imaging and Graphics.*

Journal Papers currently under preparation:

Evans, A., Lambrou, T., Linney, A.D., Todd-Pokropek, A., (2006) "Automatic 2D and 3D segmentation of liver from CT, using deformable models" to be submitted to *Medical Physics.*

Invited conferences:

Evans, A., Lambrou, T., Linney, A.D., Todd-Pokropek, A., (2006) "Automatic 3D Segmentation of the Liver from Computerised Tomography Images, a Discrete Deformable Model Approach" to appear in *Proceedings IEEE 9th International Conference on Control, Automation, Robotics and Vision (ICARV 2006)*, 5-8 December 2006, Singapore.

Peer-reviewed conferences:

Evans, A., Lambrou, T., Linney, A. and Todd-Pokropek, A., (2004) "Automatic Segmentation of the Liver from CT Images" *Proceedings 2nd International Conference on Biomedical Engineering*, pp. 205-208, Innsbruck, Austria, 16-18 February 2004.

Evans, A., Lambrou, T., Linney, A. and Todd-Pokropek, A., (2004) "Automatic Segmentation of Liver from Computerised Tomography (CT) Images" *Proceedings of Medical Image Understanding and Analysis*, pp 57-60, London, 23-24 September 2004.

Evans, A., Lambrou, T., Linney, A. and Todd-Pokropek, A., (2005) "Automatic 3D Segmentation of Liver Tissue from CT Datasets" *Proceedings of Medical Image Understanding and Analysis*, Bristol, 19-20 July 2005.

Abstract

As part of the diagnosis of liver disease, a Computerised Tomography (CT) scan is taken of the patient, which the clinician then uses for assistance in determining the presence and extent of the disease. This thesis presents the background, methodology, results and future work of a project that employs automated methods to segment liver tissue. The clinical motivation behind this work is the desire to facilitate the diagnosis of liver disease such as cirrhosis or cancer, assist in volume determination for liver transplantation, and possibly assist in measuring the effect of any treatment given to the liver.

Previous attempts at automatic segmentation of liver tissue have relied on 2D, low-level segmentation techniques, such as thresholding and mathematical morphology, to obtain the basic liver structure. The derived boundary can then be smoothed or refined using more advanced methods. The 2D results presented in this thesis improve greatly on this previous work by using a topology adaptive active contour model to accurately segment liver tissue from CT images. The use of conventional snakes for liver segmentation is difficult due to the presence of other organs closely surrounding the liver; this new technique avoids this problem by adding an inflationary force to the basic snake equation, and initialising the snake inside the liver.

The concepts underlying the 2D technique are extended to 3D, and results of full 3D segmentation of the liver are presented. The 3D technique makes use of an inflationary active surface model which is adaptively reparameterised, according to its size and local curvature, in order that it may more accurately segment the organ. Statistical analysis of the accuracy of the segmentation is presented for 18 healthy liver datasets, and results of the segmentation of unhealthy livers are also shown. The novel work developed during the course of this project has possibilities for use in other areas of medical imaging research, for example the segmentation of internal liver structures, and the segmentation and classification of unhealthy tissue. The possibilities of this future work are discussed towards the end of the report.

Contents

List of figures

List of tables

1. Introduction

This chapter gives an overview of the research presented in the thesis. Section 1.1 provides an overview of the issues facing liver disease and the technology used in its diagnosis and treatment. Section 1.2 details the motivation and purpose behind the project. Section 1.3 lists the goals of the project, defining its scope, and Section 1.4 provides an outline for the structure to the thesis, along with a brief introduction to the content of every chapter.

1.1. Context

The liver is the body's largest internal organ and it is involved with almost all of the biochemical pathways that allow growth, fight disease, supply nutrients, provide energy, and aid reproduction. It is not surprising, therefore, that several different diseases can affect the liver. Cirrhosis is a serious disease, with causes that include alcohol consumption and viral hepatitis; while the many different classes of cells that form the structure of the liver means that several different cancers can affect the organ, primary liver cancer being associated with cirrhosis 60-80% of the time (www.livertumour.org)

The incidence of liver disease differs greatly between countries and regions (Seo and Park, 2005), and is linked to the consumption of alcohol and other carcinogens. In Britain, despite liver cancer not being among the most prevalent of cancers, the incidence rate has been steadily increasing in the last 30 years, as Figure 1.1 demonstrates.

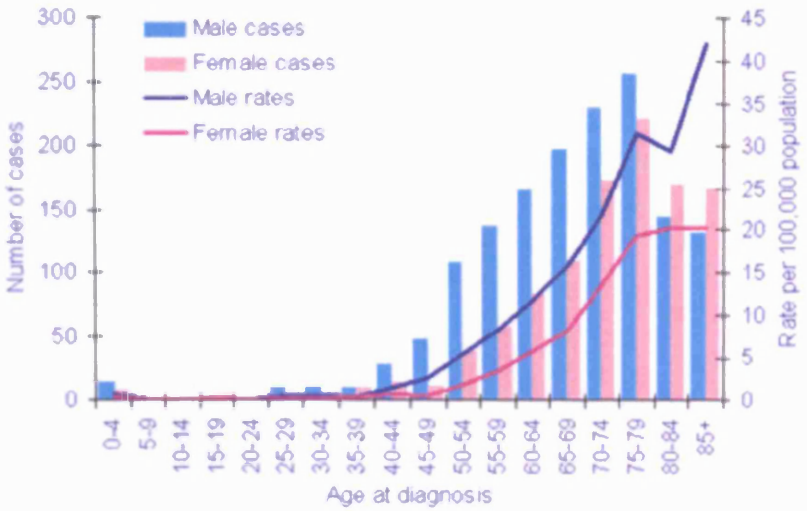


Figure 1.1: *Cancer incidence. Number of new cases and age specific incidence rate per 100,000 population, liver cancer, by sex, UK, 2001 (http://info.cancerresearchuk.org/cancerstats/types/liver/incidence/)*

As part of the diagnosis process, a Computerised Tomography (CT) or Magnetic Resonance Imaging (MRI) scan may be taken of the patient, which the clinician then uses to assist in determining the presence or extent of the disease. CT and MRI are imaging techniques that are particularly useful for the abdomen as they produce a series of images representing cross-sectional slices of a patient's body. CT scans of the abdomen are more common due to the increased resolution of the resulting images, compared with the equivalent MRI images. A further advantage of CT (with respect to image processing) is that the grey values of the images produced are standardised (the unit being the Hounsfield unit) so that the same organs/tissues in images taken from different patients and different machines have similar grey level values. A disadvantage of CT is that the patient is exposed to x-ray radiation during each scan.

Based on CT images, computer aided diagnostic (CAD) systems have been developed that assist clinicians in the diagnosis and treatment of liver disease. However, these systems still require a large amount of user input as the liver needs to be manually highlighted, usually by tracing around the boundary using a mouse or mouse-pen. Given

that 150 image slices is a normal amount for a detailed scan, this procedure is invariably slow and laborious. For this reason an automated tool to achieve the same goal would be very useful as it would save a clinician significant amounts of time – time that could be spent more productively elsewhere.

1.2. Motivation

1.2.1. Liver segmentation

Image *segmentation* is the process of delineating an image into two or more distinct regions, which are uniform with respect to certain desired properties. Liver segmentation refers to the process of isolating and highlighting the areas within an image (e.g. taken from a CT scan) that represent liver tissue. While segmentation is usually carried out in 2D, a series of 2D slices may be amalgamated into a 3D dataset, thus the entire region that represents the liver in a CT dataset is segmented.

Liver segmentation is useful for several reasons (Seo and Park, 2005; Hermoye, Laamari-Azjal *et al.*, 2005):

- Assistance in the diagnosis of disease.
- Planning of the pre-surgical operations for hepatic resection.
- Assessment of suitability for transplantation.
- Assessing therapy response.
- Automatic detection and definition of focal lesions.

As mentioned above, manual segmentation of the liver is a slow process, with a mean interaction time of 25 minutes (Hermoye, Laamari-Azjal, *et al.*, 2005) per dataset. As a result, an automatic liver segmentation system has immediate benefits in terms of saving time. Furthermore, there is some evidence to suggest that computer processing

may even assist in the detection of lesions that manual operators miss (Soler, Delingette *et al.*, 2001).

For many of the issues listed above, knowledge of the *volume* of the liver is an important factor. As such, an automatic segmentation tool must be able to accurately measure liver volume, and this can be considered an important goal for this project.

Previous research has shown that liver segmentation, whether manual or automatic, is difficult to carry out with pinpoint accuracy (Lim, Jeong and Ho, 2005). This is largely due to three reasons:

- The proximity of the liver to other abdominal organs that can have similar intensity values in a CT image, such as the pancreas, spleen, heart and muscles.
- The partial volume effect, where voxels at the boundary between tissues do not clearly belong to the region of either tissue.
- The lack of a 'true' gold-standard for segmentation accuracy. An actual volume value is not available - even if an entire liver removed from the patient, the loss of fluid and transfer to a different environment will greatly affect any measurement of volume. Furthermore, the amorphous structure of the liver means that the precise location of its boundary changes with the movement of the patient.

The issues regarding the validation of automatic segmentation are discussed in further detail in Chapter 5; however it is worth noting at this point that it may be misleading to rely on manual segmentation by a single operator as the gold standard for the validation

of any automatic technique, as there is no guarantee of the accuracy of the manual segmentation.

The specific motivation for the work conducted in this thesis arose from discussion with clinicians at the Radiology department of University College Hospital (UCH), who expressed an interest in the development of an algorithm for the automatic segmentation of liver. The specific requirement in this case is a system that can measure the volume of functioning liver, as this information can be used to assist both diagnosis and treatment of disease, and be of use in other situations such as when assessing the suitability of different livers for their use in transplant operations. After conducting literature reviews and discussing the situation with radiologists at UCH, there does not appear to be agreement as to what level of accuracy is required for an automatic segmentation algorithm to be of clinical use – though the figure of ±10% volume has been suggested by UCH clinicians.

While it is possible that such a system of automatic segmentation may not be deemed suitable to be used directly in terms of patient diagnosis and treatment, it may be of use as a part of clinical research into liver disease, as it can reduce the time taken to analyse large amounts of data which are used in clinical studies. Other possible uses of an automatic segmentation algorithm include the ability to recreate the structure of the liver using 3D computer graphics, and the use of this structure information to assist with treatment planning (for example to plan RF ablation of cancerous tissue) and possibly even image guided surgery.

1.2.2. Automatic segmentation

The bulk of the work in this thesis is concerned with medical image segmentation, and Section 2.1 below comprises an in-depth review of the literature concerning the subject. Pure segmentation techniques usually make no assumptions of shape or use any *a priori* knowledge of the object to be segmented, however it is worth briefly discussing other image processing techniques that have been extensively used in medical imaging to achieve segmentation.

Image *registration* is defined as the determination of a geometrical transform that aligns two or more images of the same or similar object, occasionally using a series of reference points on that object. Registration can be carried out in either 2D, or more usually in 3D using volumetric datasets. Registration algorithms can be broadly classified into two groups, *rigid* and *non-rigid*. Rigid-registration maintains all the distances between points, and usually consists of a two-stage process involving a translation and a rotation of the dataset. Non-rigid registration techniques allow changes in the ratios of distances between points. They are more complex, therefore, yet they have greater functionality, especially in the field of medical imaging. For example, registration of non-rigid anatomy both inter- and intra-patient may be impossible using only rigid registration, and non-rigid transformations can be useful to rectify images that have been distorted in the acquisition process.

Another method of isolating structures within an image is to use a *statistical shape model*. These enable segmentation of the image with reference to a-priori knowledge of the shape of the object(s) that is (are) to be segmented. They are constructed by establishing a set of labelled landmark features using a class of images to be processed. This set is known as the 'training set' and points are manually selected for each image.

Following this selection, the set of points for each image is aligned to one another with respect to translation, rotation and scaling, using an iterative algorithm. The variability in this model is described using a *Point Distribution Model* (PDM) (Cootes, Cooper *et al.*, 1992), which is used to constrain the behaviour of deformable models that are used for segmentation of the object in a new dataset.

During the planning phases of the project, it was decided to focus research on pure segmentation of the liver without the use of registration or statistical shape modelling. The liver is a soft tissue, amorphous, organ whose structure varies greatly both inter-patient and even intra-patient, depending on the scanning conditions. As a result, the correct labelling of significant points of the liver can be difficult even for a trained manual operator. This lack of shape information propelled early research into seeking a method of automatic liver segmentation that does not rely on using *a priori* structural knowledge, nor the location of significant points either within or on the boundary of the structure. It is important to note however, that there have been efforts to apply both registration and shape modelling algorithms to liver data, and these are discussed in Section 2.3.

Pure automatic segmentation of the liver avoids the inherent problems that face registration and shape modelling approaches, but in turn it has several issues that require consideration:

- Irregular boundary intra patient
- Irregular shape inter-patient
- The presence of contrast enhanced vascular structure (see Section 5.2)

Thus, for an automatic segmentation technique to be successful, it is required to accurately find the complex boundary of the liver, irrespective of its actual shape.

1.2.3. Validation

A key factor to be considered alongside the development of any segmentation algorithm is that of the validation of the algorithm's performance. The 'performance' of an algorithm can be split into three aspects: its *accuracy*, *reliability* and *efficiency*. The accuracy of an algorithm is a measure of how successfully it has segmented the desired region of the image/dataset, though this is entirely dependent on the reference used to determine the accuracy, the *gold standard*. In many segmentation procedures, the gold standard can be defined as 'manual segmentation by an expert', yet this leaves the possibility that a manual operator will perform an inaccurate segmentation, thus affecting the accuracy results of any automatic technique. Comparing automatic techniques to *multiple* manual segmentations makes some effort to reduce this error, and it then becomes possible to use statistical measures of rater reliability (see Chapter 5).

The reliability of an algorithm is equally as important as its accuracy; a segmentation technique may give highly accurate results on one dataset yet fail on all others. As a result, the more datasets that the algorithm has been tested on, the better the validation. In addition, if an algorithm contains adjustable parameters, any effort should be made to demonstrate the effect of the changing those parameters can increase its reliability.

Finally, the efficiency of the algorithm should be considered as part of the validation. Efficiency can be measured by calculating the time taken and computational resources required to obtain segmentation results. While efficiency may be of lesser importance

when compared with accuracy and reliability, it is important to take it into consideration and it can provide an additional measure to compare segmentation algorithms.

1.3. Computerised tomography

In traditional x-ray imaging, the linear attenuation coefficient between the x-ray tube and a detector is the measure of the fraction of x-ray that is absorbed by the intervening material. At its most basic level, CT involves rotating a tube and detector around an object in a fixed plane and sampling the coefficient for each rotation angle, thus acquiring a series of projections. By 'smearing' each projection back along it's path, it is possible to reconstruct the structure of the material in the plane of the x-rays, and generate an image of that slice. CT technology has undergone several generational changes since its inception 30 years ago, with the latest spiral-CT techniques capable of sampling multiple slices simultaneously (up to 64 on the most modern machines), vastly reducing acquisition time and capable of producing clearer images.

Sahani and Kalva (2004), in their review of liver imaging, conclude that "CT suffices for most clinical indications", indicating that the use of CT over MRI in liver imaging is a compromise between scan resolution, radiation and lesion characterisation. MRI has particular advantages over CT; the radiation dose for a CT varies with the mass of the patient, but the typical dose of an abdominal scan of a person weighing 75kg is 4mSv (Huda, Scalzetti, Roskopf; 2000), whereas the radiation dose of MRI is zero. Furthermore, while contrast-enhanced CT (see below) allows the characterisation of lesions, it has a lower sensitivity than MRI in characterising those smaller than 1cm (Sahani and Kalva, 2004). Yet the main advantage CT has over MRI is the best spatial resolution, a lower cost, shorter procedure time, and allowing the patient to hold their

breath for shorter periods, leading Sahani and Kalva to conclude that "it serves as an ideal screening examination for the entire abdomen and pelvis".

Intravenous iodinated contrast media are routinely used in the imaging of the liver. In the CT modality, by injecting a contrast medium into the blood-stream the linear attenuation coefficient of the vascular system is temporarily increased, and a series of scans can be carried out during the time that the contrast-enhanced blood flows through the body. During these *contrast phases* (the time-windows in which contrast is present in different areas of the vascular system) the intensity of the liver parenchyma in CT images differs. For example, in the *arterial* phase of contrast, the contrast-enhanced blood is imaged while still within the arteries of the body, enabling the vascular tree of the liver to be visualised while the liver parenchyma remains indistinguishable from much of the surrounding connective tissue. During the *venous* contrast phase, the contrast-enhanced blood has perfused into the smaller vessels and capillaries of the liver, so raising the mean intensity value of the liver parenchyma. The difference in appearance of the liver in the different contrast phases has obvious effects on the segmentation of the organ, and further discussion on which of the contrast phases is used for segmentation in this work can be found in Chapter 5.

All the data used within this project was CT in modality and acquired from the University College Hospital Radiology department. At the time of each scan, the resulting data is tagged with the physical dimensions of the voxels in the three major axes (in x, y and z, measured in millimetres), the duration of the scan, the voltage (kVp), the machine model, and other metadata. For each of the datasets used in the project, this data is presented in Table 5.3 in Chapter 5. It should be noted at this point that this data was different for all of the datasets used; thus, with the notable exception of the choice

of contrast phase, the source of the data was independent from the segmentation procedure i.e. the information presented in Table 5.3 was not used to affect the results of the segmentation.

1.4. Liver anatomy

The liver is the body's largest internal organ and largest gland, its actual size depending on each individual's height, weight and health. It possesses three surfaces, superior, inferior and posterior – the anterior side represents a sharp boundary dividing the superior from the inferior surface. The liver is divided into two major parts, the right and left lobe, by the falciform ligament which connects the organ to the diaphragm and abdominal wall. The right lobe is up to six times larger than the left lobe, and its left inferior part is separated into two 'sub-lobes', the quadrate and caudate, in the area of the gall-bladder and inferior vena cava.

The major vessels connected to the liver are the hepatic artery, the portal vein, and the hepatic veins. The hepatic artery supplies the liver with oxygenated blood and the hepatic veins carry it away, while the hepatic portal vein supplies blood directly from the digestive tract (thus allowing the liver to metabolise both nitrogenous and carbohydrate materials absorbed from the intestine before they enter the main circulation). The arterial tree of the liver is highly complex, as the artery splits into several branches, (vaginal, capsular and interlobular) to supply oxygenated blood to the various sections and lobes of the organ. The other major vessel pathway in the liver is the secretory pathway, which removes secreted bile to either the gall-bladder via the cystic duct, or the duodenum via the bile duct. Figure 1.2 shows two drawing of the liver.

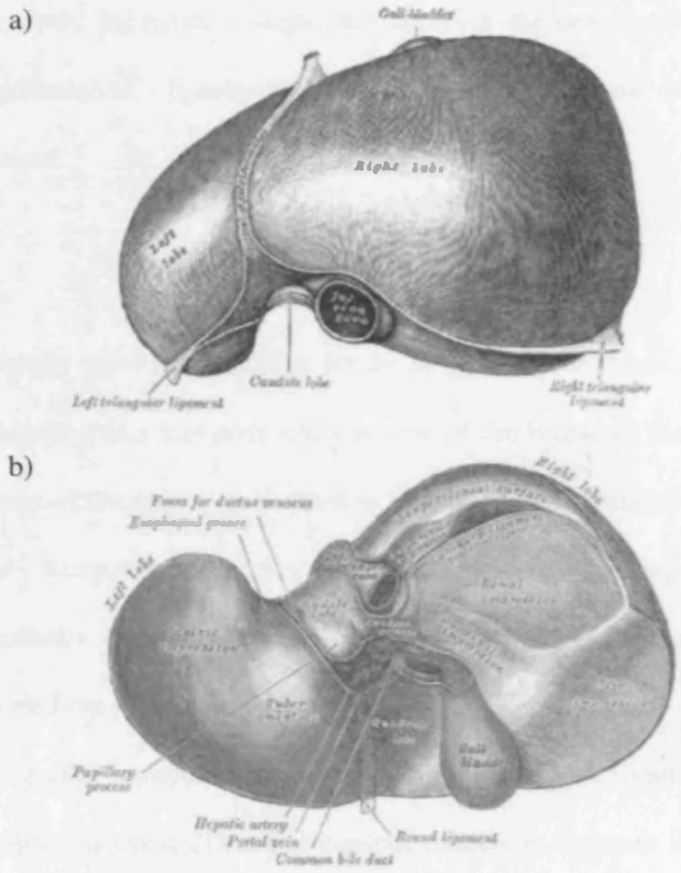


Figure 1.2: Anatomy of the liver. Figure (a) shows the superior surface of the liver, figure (b) shows the inferior surface of the liver. (http://www.bartleby.com/107/250.html)

1.5. Scope and aims

The goals of the projects are defined as:

- To develop a 2D automatic segmentation algorithm that segments healthy liver tissue from CT image slices with minimum user interaction.
- To expand the 2D technique so that it may deal with series' of images that form the dataset of one patient.
- To develop a fully 3D automatic segmentation procedure that segments healthy liver tissue from 3D CT datasets, again with minimal user interaction.
- To develop validation techniques that allow the comparison of the accuracy of the 2D and 3D techniques.

- To make preliminary steps into applying the developed algorithms to the segmentation of unhealthy or abnormal liver tissue such as tumours or lesions.

1.6. Outline

This section briefly provides an outline for the structure of the thesis.

- Chapter 2 is a comprehensive review of the literature that is relevant to the scope of the project as defined in Section 1.3. It commences with analysis of the literature concerning image processing and segmentation, before gradually increasing focus towards areas that are relevant to the research carried out in this project.
- Chapter 3 introduces the methodology and research carried out during this project on using 2D active contour models to segment the liver. The basic formation of the active contour is described, along with several novel research elements that allow greater functionality for the purposes of the segmentation of liver and other structures.
- Chapter 4 discusses the methodology of the development of a 3D active surface model. The model is a partial extension of the 2D model, yet several aspects of novel research work were implemented to ensure its ability to accurately segment objects represented in 3D data arrays.
- Chapter 5 presents results and discussion of the use of both the 2D and 3D techniques to segment both healthy and unhealthy liver tissue. It contains details of the techniques used to validate the accuracy of the segmentation, as well as numerous charts and figures to demonstrate the success of both 2D and 3D algorithms in segmenting the liver. The results are comparable with those obtained by other researchers, and several aspects regarding validation

(an important issue which has been largely ignored by other researchers) are discussed.

- Chapter 6 concludes the thesis by discussing the results and the possible use of the developed techniques in future projects and applications.

2. Related Work

This chapter provides a comprehensive review of previous research that has bearing upon the work carried out in this thesis. It begins by reviewing the literature concerning image processing in general, gradually focusing in on the areas that are more relevant to this project. There then follows an introduction to the previous research into liver segmentation, including recent state-of-the-art publications. The chapter then concludes with a brief review of segmentation work carried out on other organs of the body.

2.1. Image segmentation

One of the most important stages in the analysis of any sort of image is the intermediary step of segmentation. The aim of image segmentation is the domain-independent partition of the image into a set of regions, which are visually distinct and uniform with respect to certain properties, such as grey level, texture or colour (Sonka and Fitzpatrick, 2000). When analysing medical images, such as CT or MRI scans, segmentation techniques are used to isolate specific organs, or regions of organs, within the image. Accurate segmentation is of fundamental importance in such applications, as any errors to could lead to misdiagnosis and complications in a patient's treatment.

Segmenting images manually is the most basic method of image segmentation, and is an extremely laborious process which requires a trained individual to isolate the features of interest by hand. Therefore automatic or semi-automatic segmentation is an important goal in medical imaging; research into the subject has seen much activity in the past decade, on which this section will elaborate.

The literature on image segmentation (both for medical images and general, non-medical images) is extensive, and different techniques have been proposed and proved to be effective. The most commonly used segmentation techniques can be broadly classified into two groups:

(1) *Region-based segmentation* techniques use the homogeneity of features in an image to classify them as one region or another.

(2) *Edge-based segmentation* techniques attempt to highlight the boundaries between regions with different characteristics.

2.1.1. Region based segmentation

2.1.1.1. Thresholding

The most basic example of region-based segmentation is that of global thresholding (Jain, 1988). This technique, given a value somewhere within the range of intensity values of an image, simply separates the intensity values of that image into two sets, based on whether the intensity is above or below the given value (the *threshold*). This technique, though simple, can be remarkably effective if the image histogram is dominantly bimodal. It can also be used to isolate a 'peak' of values within an image, by applying an upper and lower threshold.

Thresholding is a fast and simple way to segment an image, and it is particularly useful in such images where the intensity range of objects that are to be segmented are known beforehand. However in most imaging fields (not just medical imaging) the situation whereby an entire group of images can be segmented using the same global threshold is very rare. For example while, in general, livers from different images will have *similar* Hounsfield unit values (Gao, Heath *et al.* 1996), applying a standard global threshold does not result in effective segmentation of the liver (see Section 5.3.1).

As a result, if global thresholding is to be used, the threshold values need to be set adaptively based on the properties of the image. An early technique proposed involves setting the threshold based on a classification model that minimises the probability of error (Sonka and Fitzpatrick, 2000); more recent techniques are based on fuzzy logic, which assigns a particular object to a particular set based on the object's *similarity* to a set. A fuzzy set is a class of points that possesses a continuum of membership grades, where there is no sharp boundary between elements that belong to a class and those that do not. The membership grade is usually expressed by a *membership* or *characteristic* function, which assigns to each element in the set a membership grade in the interval [0,1] (Li, Zhoa and Cheng, 1995; Tobias and Seara, 2002). In the case of setting global thresholds for images, the pixels of the images are the elements in a fuzzy set, and the sets themselves are ranges of grey level values. By applying different membership functions, different characteristics of an image can be used to assign grey level values to one set or another, and thus segment the image into regions based upon the grey level value of individual pixels.

While global thresholding is simple and computationally fast, it fails when there is low contrast between objects and the surrounding background, if the image is noisy, or if background intensity varies significantly across the image. There are a variety of ways of combating these effects: applying local thresholds across an image or applying some form of pre-processing to the image to improve segmentation (for example applying a Gaussian smoothing filter and attempting to reduce image noise).

Ultimately, no matter how carefully a threshold is set it is unlikely that the segmentation will be consistently reliable for medical applications, and therefore must be augmented

with other, higher level techniques. In imaging modalities such as MRI and ultrasound, thresholding becomes even less reliable due to shading artifacts and issues of noise. However it can provide a quick, simple 'first guess' at segmenting an image, which in some cases can be used as a base for further analysis.

2.1.1.2. Region growing

Region growing is the opposite of thresholding in that, rather than isolating distinct regions immediately, it focuses on adding pixels to a region based on a homogeneity criterion (Sonka and Fitzpatrick, 2000). The basic algorithm starts as the user selects a *seed* point for a region, to which neighbouring pixels are added depending on their *similarity* to that region. Therefore, the key point is the criterion by which pixels are added to the region – too weak a criterion and regions will 'leak' out and result in incorrect segmentation, too strong a criterion and the regions may not grow to their 'full potential'. The basic region growing technique is expanded on by 'split and merge' algorithms. Firstly an image is taken as the frame of interest and analysed to decide whether all the pixels satisfy a region-similarity constraint. If there are differences, the frame of interest is split into (usually) four equal sub-frames, which are then analysed recursively using the same techniques, until all of the pixels in a region satisfy the similarity test. As the splitting of the frames is arbitrary it is not unusual to find several homogenous neighbouring regions; to counter this a merging algorithm is applied after each split. Section 2.1.3.2 below describes how region growing can be quite effectively combined with edge detection techniques to create very powerful segmentation algorithms.

Watershed algorithms (Jain, 1988) simulate a flooding process. An image is identified with a topological surface in which the altitude of each point corresponds to the gradient

value (see Section 2.1.2 below) of the pixel in the image. Holes are pierced in the regional minima, and the algorithm simulates the gradual submerging of this relief into a body of water. Water flows into the valleys of the image from the holes, and where separate areas of water meet a solid boundary is imposed on the image. Once the relief is completely covered with water, this comprises the watershed image. The watershed is a powerful segmentation tool but its overall performance relies greatly on the algorithm used to compute the gradient of the image (Munoz, Freixenet *et al.*, 2003). A conventional gradient operator usually produces an over-segmented image, so a region merging algorithm must then be deployed to correct the image. Like the thresholding technique, watershed algorithms are useful when combined with other algorithms, but on their own they require a significant amount of user interaction to obtain consistently good results.

2.1.1.3. Texture

Image segmentation by texture is usually a region based procedure that, compared to the previous techniques described, is newer, more complex but generally more effective in segmenting images that are particularly difficult to segment otherwise (Reed and Hans du Buf, 1993). For example, the algorithms are particularly effective in dealing with cloud fields in meteorological images, which have long been a source of frustration in image processing because of their difficulty to segment. The main purpose of texture feature extraction techniques is to map differences in image structures, either stochastic or geometric, into differences in grey level value – these can then be segmented in order to extract homogenous regions. The methods to achieve this can be arbitrarily classified as region-based and boundary-based, or some a combination of the two.

2.1.2. Edge based segmentation

The key difference in segmenting an image by its edges is that the *gradient* of pixel intensity is usually the deciding factor in the segmentation. A gradient is an approximation of the first order derivative of the image function, and both the magnitude and direction of a gradient can be displayed as images.

The simplest method of highlighting the edges in an image, based on the pixel gradient, is by applying a convolution filter across the image. In the time (or spatial) domain, convolution filters simply apply weighted summations of the pixel intensities in local neighbourhoods, and are frequently represented as a numerical array representing a kernel, mask, or window. The frequency response of a convolution filter i.e. its effect on different spatial frequencies, can be seen by taking the Fourier transform of the filter.

As an example, the Sobel edge detector (Jain, 1988) employs two such 3x3 convolution filters, one each in the direction of the x and y axes, see Figure 2.1.

$$Gx = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad Gy = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

***Figure 2.1**: The Sobel edge detector. 3x3 convolution kernels are used, one is simply the other rotated by 90°.*

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these Gx and Gy). These can then be combined together to find the absolute magnitude of the gradient at each point. The gradient magnitude is given by

$$|G| = \sqrt{Gx^2 + Gy^2} \tag{2.1}$$

which is typically approximated by

$$|G| = |Gx| + |Gy| \tag{2.2}$$

which is much faster to compute.

The result of the edge detector is a largely dark image with the edges highlighted in increasing intensity, based upon their strength (i.e. the strength of the image gradient) - in frequency terms, the filter has the effect of magnifying high frequencies relative to low frequencies.

A similar method to the Sobel operator is the Kirsch compass operator, which applies a gradient filter in eight different directions for each pixel and selecting the largest result (Russ, 1999). This allows gradients in all eight directions to be used in generating the edged image, which improves on the Sobel operator as it better represents edges that are not perpendicular to the x and y axes. Another technique is to use the Laplacian operator, where an approximation of the second order derivative can also be used to detect edges, since the peaks in the first order derivative correspond to the zeros in the second order derivative.

Unfortunately the use of edge detectors has its drawbacks. They are extremely sensitive to noise, so it is often necessary to apply some sort of smoothing to the image prior to detecting edges. There are several methods of reducing image noise; the Marr-Hildreth filter (Marr and Hildreth, 1980) smoothes the image with a Gaussian mask before

calculating the second derivative to detect edges. More advanced smoothing techniques are particularly effective at retaining the boundaries of the structure of interest, such as wavelets (Mallat, 1999) and anisotropic diffusion (McCool, 1999). However, even with non-noisy images, detected edges may not link up into contours, and extending these edges to match correctly has proved a difficult problem (Munoz, Freixenet *et al.*, 2003). Section 2.4 below describes how combining simple edge-detection with region growing has provided more robust segmentation techniques.

Other edge detectors, such as the Canny technique (Jain, 1988) exist, yet the greatest problem with simple edge detectors is that they are entirely *local* and make no assumptions about the shape of interest. To this extent, much work has been done on higher level, more mathematical techniques.

2.1.2.1. Explicit contour models

Kass *et al.* first proposed Active Contour Models, or *snakes*, in 1987 (Kass, Witkin and Terzopoulos, 1987). A snake consists of a curve, defined within an image domain, and represented by a set of interconnected vertices. It can move under the influence of internal forces, derived from within the curve itself, and external forces derived from the image data. Research involving snakes has been vigorously pursued since they were first proposed and they are used in a wide variety of image processing applications.

Explicitly defined, *parametric* snakes are the most commonly used active contour models, and they allow parametric curves to move towards certain features in an image, for example edges. The forces acting on the snake can be split into two varieties:

1) *Internal Forces* – consist of elasticity forces to keep the curve together, and bending forces to add stiffness to the curve.
2) *External Forces* – can vary greatly depending on the implementation of the snake. The most basic force that draws the snake to an edge is a potential force, derived from the local pixel gradient.

These energies can be represented by:

$$E_{total} = E_{\text{int}} + E_{ext} \tag{2.3}$$

A third type of force exists in the form of *attractor* and *repulsor* forces that have the effect of pulling the contour towards pre-detected edges within the image, and can be calculated as a function of the distance of the curve to an edge. However these forces can be cumbersome to implement and as a result are rarely used in modern snake algorithms. Further discussion of the implementational details of active contour models can be found in Chapter 3.

Despite being a major development in image segmentation and edge detection, the basic snake algorithm has several well-documented drawbacks (McInerney and Terzopoulos, 2000).

i. It is quite sensitive to noise, in that a snake vertex may get 'trapped' on a noisy local edge, which is not the edge of the desired object.
ii. For this reason, traditional snakes have a small capture range i.e. they must be initialised quite closely to the boundary of the target object to capture it correctly, especially in images that have large amounts of unwanted edges.

iii. A snake element can move too far across the desired minimum and never come back *i.e.* the snake passes straight through a desired edge.

iv. The fixed geometric parameterisation of a standard snake, in conjunction with the internal deformation energy constraints, limits flexibility. This prevents the snake from conforming to long tubular shapes, or shapes with significant branches.

As a result, several researchers have tried to overcome these limitations. Cohen (1991) introduced an inflationary force (or a balloon effect) which, to some extent, reduces problems (i), (ii) and (iii) above. The algorithm introduces a second external force, which pushes outwards in the direction of the normal vector to the curve at each snake element. The parameter of the force I set so that a strong edge can still stop the inflation force, but the existence of a pressure force means that the snake can move through weak, noisy edges. This force inherently balances the elasticity internal energy term, which has the effect of drawing control points closer together and thus 'deflating' the contour.

Though Cohen's technique was an improvement on the basic snake, a more robust solution was proposed by Xu and Prince (1998), almost a decade after the original snake was published. They introduced a new class of external forces called *Gradient Vector Flow* (GVF) fields, dense vector fields that are derived from images by minimising an energy functional in a variational framework.

The GVF field essentially consists of a field ∇f that has vectors pointing to gradient-derived edges in an image, with the magnitude of the vector increasing as it nears the edge. Thus the field points strongly towards object boundaries when very near to those

boundaries, but dissipates smoothly over homogenous image regions, extending to the image border. The GVF field is then used to replace the potential force in the traditional snake.

This method successfully addresses several of the problems associated with traditional snakes, especially problems (ii) and (iii) above. The GVF field can capture a snake a long way from an object boundary and crucially from *either side* of the object boundary, so a GVF snake will never 'overshoot' an edge.

While the field enables a snake to be able to deal with concave regions in a much improved fashion, it still falls short of solving problem (iv) above. Research by McInerney and Terzopoulos (2000) proposes a class of deformable contour that addresses the topological inflexibility highlighted in problem (iv), while still retaining an easily modifiable, explicit parametric framework. Called *T-snakes* (for 'topological adaptive' snakes), the model is a discrete approximation to a conventional parametric snake model, with internal spring forces, an inflationary balloon-type force and external image forces all acting upon control points of the contour. The key difference is that the set of vertices and interconnecting elements that describe a T-snake does not remain constant during its evolution. As the snake moves under energy forces and after each iteration, it is *reparameterised* with a new set of vertices.

This reparameterisation is achieved first by separating an image into a set of cells by using a grid of regular size and shape. At specified stages of the contour's evolution, the algorithm removes all references to its current vertices, and creates new vertices at each intersection of the contour with the boundaries of the cells. Regional information is further enhanced by 'activating' grid intersection points as the snake contour passes

them – thus the cell data structure maintains a reduced-resolution area representation of the region bounded by the contour. This regional information is then used further to split and merge two contours by detecting if two separate areas of a contour are sharing the same cell. Thus, the T-snake technique directly addresses some of the intrinsic internal problems that affect active contour models.

2.1.2.2. Implicit contour models

Implicit active contours have been developed by applying Osher and Sethian's level set evolution technique (Osher and Sethian, 1988) to the segmentation problem. Level set models are deformable implicit surfaces where the deformation of a curve is controlled by embedding the curve inside a higher dimensional function. For example, in 2D, rather than follow the propagation of the curve itself, a level set approach instead takes the original curve and builds it into a 3D hyper-surface, which intersects the x-y plane at the exact location of the curve. A formulation of the temporal evolution of the level set equation is given by

$$\frac{d\psi}{dt} = \phi|\nabla\psi|\left(div\left(\frac{\nabla\psi}{|\nabla\psi|}\right) + \underline{F}\right) \tag{2.4}$$

where $\psi(x,y,t)$ is a multidimensional function of the curve in time, $\underline{F}$ is a function that can depend on external factors such as curvature, normal direction etc., and ϕ (in this equation) is an image-based speed function that slows the curve at salient edges. Figure 2.2 shows a diagrammatic representation of a standard stationary level set function.

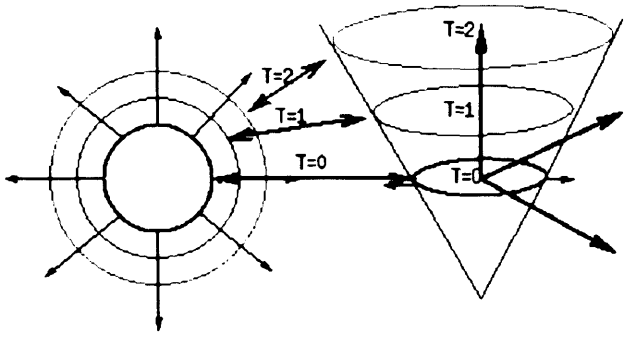


***Figure 2.2**: Level sets. The original (green) front moves outwards and it's position at time T is determined by slicing the surface at time T.*
(http://math.berkeley.edu/~sethian/Semiconductors/ieee_level_set_explain.html)

The main advantage of the level set method is that topology of the curve can change, because the curve merely represents the intersection of the plane with the level set function at a given time. The main disadvantage to this method is that to actually move the curve, it requires the tracking of *all* the level sets, not just the zero level set corresponding to the intersection which requires a high level of computational power. To combat this, the *narrow band* method (Sethian, 1996) focuses only on those grid points which are located in a narrow band around the zero level set. So-called *fast marching* methods (Sethian, 1996) reduce the computational cost even further by only considering a front that moves in one direction, and build the level set function one section at a time, as the curve propagates.

There are several advantages and disadvantages to using a level set formulation to model a curve that could be used as a segmentation tool (Museth, Breen *et al.*, 2002). They always produce closed, non-self intersecting surfaces, and as such are free of the contour or mesh connectivity issues that can plague explicit models. However, the computational power required to model the level set function, even using the narrow banding method, is higher than for explicitly defined contours. Also in some situations the ability to take exact control over the topology of the curve is preferable – for

example to reverse the direction of movement if over-segmentation has been detected. Furthermore, level sets are also disadvantaged by their inability to represent fine, sharp features, or long narrow features, such as blood vessels (Museth, Breen *et al.*, 2002).

Another implicit model that can be used for segmentation is the m-rep. M-reps describe 2D or 3D geometrical shapes through a more unconventional method, by using a multiscale medial technique for modelling and rendering. Pizer *et al.* (2003) claim their method is particularly well suited to modelling anatomical objects, and capturing geometric information effectively. The basic premise of an m-rep is that an object is not described by a set of points on its surface, but by a set of *atoms* located along the medial axis of a shape, each of which is associated with parameters that describe the surface of the object in relation to the medial axis. Segmentation of objects, in both 2D and 3D, is achieved by creating the m-rep manually to fit a small scale representation of the image dataset. The segmentation process then follows a number of stages at successively larger levels of scale, the m-rep deforming to more accurately track the object's boundary at each level of scale.

While m-reps are an interesting method of object representation, the advantages over using this method over more conventional methods of representation (such as explicit definitions of a surface, or implicit level sets) are not clear. When the object to be segmented has clear and defined edges that facilitate the discovery of its medial axis, the technique has the advantage of simplicity; however for more complex objects, where more than one medial axis is taken into consideration, the complexity increases as multiple medial axis branches must be used.

One clear advantage of m-rep segmentation is that, due to the multiscale nature of the technique, internal structures can be effectively dismissed, as the boundary of the m-rep remains near the boundary of the object to be segmented. However this applies to many (if not all) multi-scale segmentation techniques (including other contour models), and m-reps maintain a distinct disadvantage of having to be manually, and relatively accurately, initialised before segmentation can begin.

2.1.2.3. 3D edge detection

In a parametric 3D model, the internal energy of the contour, which is now a surface, has to be calculated in a slightly different way to 2D (Bulpitt and Efford, 1996). Yet the basic principles, whereby the movement of a set of control points, representing a surface, is determined by a set of equations defining internal and external energies, remain the same.

There are a variety of different methods of estimating the energies used to move the surface (Cohen and Cohen; 1993; McInerney and Terzopoulos, 1996), many of which move the surface in a manner that can be encapsulated by equation (2.5).

$$E = \sum_{i=1}^{N} (\alpha_i E_{elast} + \beta_i E_{rigid} + \gamma_i E_{img} + E_{ext}) \tag{2.5}$$

where E is the force moving the contour, the α and β parameters control the internal elasticity and rigidity (E_{elast} and E_{rigid}) forces, γ controls the strength of forces derived from the image dataset (for example, energy derived from the grey value gradient), E_{ext} takes the form of an inflationary force that is calculated using the normal vector for each vertex on the surface, and N is the number of vertices that describe the surface. Its

internal energies are calculated using the set of vertices surrounding the vertex in question (see Section 3.4.2 for more information as to how the energies are calculated at each point). It should be noted that in some implementations, E_{ext} may be calculated in a different manner, or may not be present at all, depending on the desired application of the surface.

Non-parametric models, such as level set-based models and m-reps, can also be extended to a third dimension due to their implicit nature (Osher and Sethian, 1988; Pizer *et al.*, 2003), and in this case the surface of the model does not need to be explicitly described.

2.1.3. Combining region-based and edge-based methods

Both region-based and edge-based methods have advantages and disadvantages and, as a result, there have been significant efforts to combine the two and make use of the advantages of both techniques. The integration of both groups of segmentation techniques can itself be split into two separate strategies (Munoz et. al., 2003):

i. Post-processing integration, where an image is subjected to both region-based and edge-based processing separately, and *a posteriori* attempts are made to fuse or integrate the results.

ii. Embedded integration, which attempts to integrate segmentation through the definition of new parameters or new decision criterion. For example, previously extracted edge information can be used as an active part of a region growing technique, thus using boundary information as a means of avoiding the problems inherent to region-based techniques.

2.1.3.1. Post-processing integration

There are three separate approaches to post-processing integration, *over-segmentation*, *boundary refinement* and *selection-evaluation*.

Pavlidis and Liow (1990) suggest that the major reason that the region-growing technique produces false boundaries is that the definition of region uniformity is too strict. Thus they conclude that the results could be significantly improved by checking all the region boundaries that could qualify as *edges*. To achieve this, an image is over-segmented by setting parameters to specific values that increase the strength of the segmentation; both region-based and edge-based techniques are used separately to create a large number of boundaries/edges. The results can then be compared and, where correspondence exists between the two methods, an edge is preserved; edges where no correspondence is recorded are discarded.

The boundary refinement approach considers region-based segmentation as an initial approximation – an initial boundary is obtained usually by region growing or thresholding, which is then modified and refined by a higher-level technique. One method of doing this is by analysing the image at different resolutions or scales, using a pyramid or quad-tree structure, or Gaussian scale space (Spann and Wilson, 1985). The basic algorithm usually consists of an upward path which smoothes the image at the expense of reducing spatial resolution. This is then counted-balanced by a downward path that attempts to increase the resolution while preserving any information obtained at the lower resolution. Important work by Spann and Wilson (1985) in the late eighties was done in this field, and it has been extended by other researchers including Kim and Kim (2003), who use wavelets to lower the resolution of the image in the hope that noisy edges are discarded while valid edges are maintained. A watershed algorithm is

applied to segment the images at low resolution, before wavelets are again used to increase the resolution of the image (and segmented region) back to it's native size.

One method of boundary refinement is to use active contour models. Using a low-level region technique to obtain a starting contour for a snake avoids one of the technique's major problems, that of initial placement. Work has been done on this technique in the medical field; for example, several researchers use active contour models to refine initially obtained boundaries for segmentation of the liver (Gao, Heath, Kuszyk and Fishman, 1996; Qatarneh, Noz, Hyodynmaa, Maguire, Kramer and Crafoord, 2003).

The third approach to post-processing integration is selection-evaluation, which involves using different parameters to several distinct region-based results, and using an evaluation criterion based on edge-based results to select the best segmentation (Munoz et. al., 2003).

2.1.3.2. Embedded integration

There are many ways to integrate edge information into region-based segmentation. At the most basic level, edge information can be integrated into a region growing algorithm by stopping it from growing when it reaches a pixel that has been previously defined as an edge, for example using an edged-image derived from a Sobel or Kirsch filter. However there are more advanced techniques to decide which pixel is an edge. Xiaohan *et al.* (1992) propose a homogeneity criterion consisting of the weighted sum of the contrast between region and pixel, and the value of the modulus of the gradient of the pixel. If the result of this gradient (therefore edge)-based calculation is below a threshold, the pixel is included in the growing region. Another method developed by

Steudel and Glesner (1999) is to use fuzzy logic to detect edges and as a rule for region growing.

Similar criterion-based techniques can be applied to watershed algorithms, which due their gradient-based nature are a natural combination of region and edge segmentation (Munoz, Freixenet *et al.*, 2003). However the major problem with watershed techniques remains the exact algorithm that computes the gradient of the image, and the fact that noisy images are very difficult to segment correctly - additional noise in the image creates incorrect edges which results in the creation of too many separate regions.

One of the most effective embedded integration techniques involves combining active contour models with region growing, which results in *active region models*. The external energy term of the snake equation is replaced by a term derived from local region information, and the snake elements are allowed to expand or contract according to the match between *local* region information and *global* model of the region. This region-based energy term is derived from all the pixels enclosed by the snake contour, and is defined by some evaluation function that measures the 'goodness' of the image data. Ivins and Porrill (1994) state that the 'goodness' function can be any function that can be used to assess the pixels within the snake region, which determines the value of the function for each particular snake element. One example is a simple binary function that is calculated using the mean intensity value of each pixel within the region:

$$G(I(x,y)) = +1 \qquad (|(I(x,y) - \mu| \leq k\sigma) \tag{2.6}$$

$$G(I(x,y)) = -1 \qquad (|(I(x,y) - \mu| > k\sigma) \tag{2.7}$$

Where $G(I(x,y))$ is a parameter to control the region force at pixel (x,y) in the image I; μ is the mean intensity and σ the standard deviation of pixel values computed from a seed region that can be selected manually, k is a constant. This functional links a pressure, or inflationary, force to the image data, and reverses it to make the region model contract if statistical limits are violated.

The result of this is a model that retains the desirable features of both region growing and active contour techniques, and is particularly effective at ensuring an active contour does not get 'trapped' on weak edges which prevent the correct edge from being found. Alexander and Buxton (1997) furthered this work with several implementational improvements, and compared the performance of several active region model implementations which differed with respect the measurements of goodness and the method of energy minimisation.

2.2. Meshing and collision detection

For the correct implementation of explicitly defined active surfaces in 3D (see Section 2.2.2.3), a surface mesh can be used to connect the vertices that describe the surface. A *mesh* can be defined as a set of vertices that are inter-connected by a set of *edges*, and can exist either in:

- 2D. All vertices and edges lying on the same plane.
- 3D (surface). The vertices are located at any point in 3D Euclidean space, but connecting edges are arranged to form triangular faces to represent a surface.
- 3D (volume). The vertices are located at any point in 3D Euclidean space, and edges connect each vertex to form tetrahedra. Thus the inner volume of the structure is represented as well as its surface.

A mesh can exist in two basic forms, structured and unstructured (Paloc, 2003), see Figure 4.1 in Chapter 4. A structured mesh has a relatively rigid vertex-edge connectivity paradigm, where each vertex has the same number of connecting edges attaching it to its surrounding vertices. Structured meshes are beneficial in situations where the mesh is unlikely to deform greatly (such as describing a cylindrical shape or other regular, rigid body), as the constraints on their structure facilitate many calculations that may involve the mesh. In other situations, however, (such as describing the shape of a highly irregular structure, which may alter its shape over time) the inability to alter the structure of mesh becomes a hindrance. In the latter situation an unstructured mesh is preferable, where the list of connecting edges is unique to each vertex, and a wide variety of connectivity paradigms is allowed.

2.2.1. Segmentation using adaptive remeshing

During the movement and evolution (through time) of a surface mesh, such as the movement of the vertices used to implement an active surface model, the topology of the mesh may change to an extent where it is hindering any further correct movement of the vertices. Alternatively, aliasing problems may arise as the distance between the vertices of the mesh is so large that the representation of the actual surface is inaccurate. These factors can be encapsulated in a concept termed *mesh quality*, where a good quality mesh is one that has an arbitrary (depending on the application) maximum distance between each of its vertices, and a minimum angle (usually 30°) between each of its edges.

To ensure good quality, a surface may be *remeshed*, where the vertices and edges of the mesh are reorganised and reparameterised. There are several examples of self-reparameterising surface meshes being used for the purposes of segmentation.

McInerney and Terzopoulos (2000) extend their T-snake into 3D by using T-surfaces. The cellular image decomposition is extended into 3D, and intersection points of the surface and this grid are deduced. By 'turning on' grid intersection vertices as the surface boundary passes over them, they use the regional information thus obtained to determine locations where the surface intersects, and at each reparameterisation a completely new surface is constructed using the structure of the cellular grid.

Park *et al.* (2001) propose a 3D deformable mesh that employs a non-self-intersection force which increases with inverse proportion as the vertices of a mesh approach each other, which effectively avoids all problems regarding remeshing. Finally Lachaud and Montanvert (1999) implement a fully topologically adaptive model for surface intersection and remeshing, which is able to segment medical image datasets in 3D.

2.3. Liver segmentation

Segmentation of the liver is of particular use when computing a three-dimensional (3D) rendering of the organ, which has been shown to be helpful for surgical planning prior to hepatic resection (Woodhouse, Ney *et al.*, 1994; Soyer, 1991). Currently, commercially available 3D rendering packages require a significant amount of manual input (Ney, Fishman *et al.*, 1990, Hermoye, Laamari-Azjal *et al*, 2005) (which is time consuming as described above).

Some of the first research done on liver segmentation was by Bae *et al.* (1993). Their technique used simple thresholding to isolate the liver from CT images, using a scheme to automatically detect threshold values. Their method was to first detect the abdomen boundary using thresholding, and then create a 20x20 pixel region of interest (ROI), placing it by essentially guessing where the liver was in relation to the abdomen. Once

the ROI had been detected, the liver was segmented by applying thresholds within 95% of the ROI average value, and the contours obtained were smoothed using B-splines. To segment the entire liver, the operator has to choose a slice from the middle of the liver so that the ROI will fall within the correct area. From this central slice it is possible to segment other slices without operator interaction.

In 1996 there was a further breakthrough in actual segmentation of the liver, presented by Gao *et al.* (1996). They used thresholding, along with morphological opening and closing to create an initial contour, which was then modified by a snake-like active contour model, based on Fourier ellipses (Staib and Duncan, 1992). The technique appears to work fairly well on clear liver images, though no results are presented where the liver boundary is less marked. Though the Fourier-based active contour model is effective, it is also a more complicated and less adaptable model than the traditional energy minimising snake.

More recent work on the segmentation of the liver was published by Qatarneh *et al.* (2003). The specific interest of these researchers is the implementation of a 'whole body atlas' for fast segmentation of future images. In their segmentation scheme, an initial 'first guess' contour for the liver is obtained from a whole body atlas (an average of previous segmentation results). A traditional snake is then used to modify the contour to fit the liver in the new image, and the result of this segmentation is stored in the body atlas.

While this is a usable and robust liver segmentation technique, it suffers from several flaws. The first is that it relies on a complex outside source to initialise the snake near the liver boundary, neatly circumnavigating but not solving one of the major problems

associated with snakes. Secondly the basic snake algorithm is very sensitive to its initial location, and there false contours exist outside the liver boundary that can cause serious problems, especially with an unclear image.

One other effective method of obtaining a rough contour to the liver was developed by Shimizu *et al.* (2003). They used the corresponding CT values from four different input images of the same liver (each at a different stage of contrast treatment) to obtain the rough contour. The main limitation of this technique is that four complete datasets are required for effective segmentation of one liver, computationally it involves four times the memory and processing power that is used when analysing a single dataset, and it is heavily dependent on the timing of the scans as the contrast agent flows through the liver and circulatory system (see Section 5.2).

Hong et. al. (2001) use thresholding and morphology to segment the liver and detect tumours wholly encompassed by liver tissue; however they also propose an interesting technique to cater for situations where a tumour exists on the edge of the liver i.e. where the liver boundary is represented by a tumour. Using their standard thresholding methods, this situation results in incorrect segmentation of liver and tumour, however by using morphological opening and closing techniques to detect and connect the boundaries of the tumour, the boundary of the liver is effectively rebuilt using guesswork. The technique certainly addresses a problem not discussed in similar literature, and should form the basis of more research.

A level-set approach to liver segmentation was used by Pan and Dawant (2001) to segment a total of five livers (a mixture of normal and abnormal) in both 2D and 3D. While the topological adaptivity of the level-set snake avoids problems of

parameterisation, the selection and usage of a suitable speed function proved a difficult problem to overcome, to the extent that *a priori* knowledge (in the form of detection of the skin surface) had to be used to ensure correct segmentation. Although this is the only published work that attempts pure 3D segmentation of the liver, the researchers had difficulty in constraining the propagation of the front in the third dimension, despite the use of *a priori* information.

There have been some efforts to use other image processing techniques, such as registration and statistical shape modeling (Lamecker, Lange and Seebaβ, 2002) to segment the liver. Soler *et al.* (2001) first use a thresholding and mathematical morphology step to segment several significant organs and areas of the abdomen, such as the skin, bones and lungs, before registering a liver model with reference to the location of other organs. The work makes further effort to use voxel intensity information to isolate vascular structure and potentially identify abnormalities, yet its reliance on a reference liver model makes it unsuitable for livers that are abnormally shaped.

In 2005, several more papers were published on liver segmentation, yet much of the work did not expand on the previous research described above. Seo and Park (2005) developed several algorithms that allow multiple use of thresholding and masking techniques to segment the liver. Lim, Jeong and Ho (2005) use a 2D deformable contour model, relying again on thresholding and morphological operators for its initial placement. Liu, Zhao and Kijewski (2005) use a Gradient Vector Flow snake to segment the healthy and unhealthy liver in 2D, their technique relying on a preprocessing thresholding step to obtain the initial contour; to date, this is the most comprehensive research on liver segmentation, yet it is still constrained to 2D. In all of

this recent research, the only method used to measure the success of the segmentation is by area/volume comparison of the results of the automatic segmentation and those obtained by manual segmentation by a *single* expert.

Table 2.1 and 2.2 summarise, where quantitative data exists, the liver segmentation results of the methods reviewed in this section, in addition to results from papers published in late 2006 after the completion of the work presented in this thesis. The results must be viewed carefully to ascertain the relative success of each technique, as the validation methods used to judge accuracy are not always the same for each research group. Table 2.1 presents the results of 2D automatic segmentation to a single manual segmentation, and Table 2.2 presents results from 3D automatic segmentation, including those obtained by active shape models. Both tables list the method used to validate the segmentation accuracy for each result. Chapter 6 critically analyses these results and compares them with the results of the work presented in this thesis.

Author	Seo & Park	Liu & Zhao	Bae *et al.*	Lim *et al.*	Pan & Dawant
Number of Datasets	12	20	4	6	5
Measurement method	*Area*	*Area*	*Area*	*Area*	*Overlap*
Result	8.28%	5.30%	6.50%	3.00%	0.95

Table 2.1: *2D segmentation accuracy. The units for the area results are percentage error compared to manual segmentation. The overlap is measured where a value of one equals perfect overlap.*

Author	Pan & Dawant	Soler & Delinguette	Lamecker *et al*		Heimann *et al.*	
Number of Datasets	5	5	33		59	
Measurement method	*Overlap*	*One sided mean surface distance*	*Volume*	*Mean distance*	*Median volume*	*Mean distance*
Result	0.92	2mm	7%	2.3mm	11%	1.3mm

Table 2.2: *3D segmentation accuracy. The units for the volume result are percentage error compared to manual segmentation, the units for distance measurement are mm. The overlap is measured where a value of one equals perfect overlap.*

2.3.1. Relevant work on other organs

There has been significant research into segmentation of other organs from CT images, particularly the brain and bone structure; although most work regarding the brain is done using Magnetic Resonance Imaging (MRI) as this is potentially less damaging to the patient (Bezdek, Hall and Clarke, 1993). In the field of abdominal organ research, much work has been done on the colon, especially on automatic detection of polyps (Nappi, Dachman *et al.*, 2002). Recent research on lung segmentation has been carried out by Hu *et al.* (2001) and again by Qatarneh *et al.* (2003). However developments in the actual *segmentation* techniques used have not proceeded far beyond simple thresholding. This is partially due to the technique achieving satisfactory results as it is (both lung and colon can be cleared and filled with air; this appears black in CT images and as such is relatively straightforward to segment using thresholding), but mostly because there are more interesting challenges to be found post-segmentation (for example polyp detection as mentioned above, mapping the path through the colon, or through the brachea to the lungs etc). As a result there is a disappointing lack of techniques developed that might be directly applied to liver segmentation.

3. Active contour models

Due to the wide variety of object shapes and the variability of image quality, deformable models can consistently produce more accurate results than those that would be obtained using classic image processing techniques, such as thresholding and edge detection. For the segmentation of abdominal tissue from large CT datasets, a parametric deformable model is suitable due to its speed, the flexibility of its implementation, and robustness. Nonetheless, several problems exist if such a model is to be used in such a way as to extract accurate boundary information. This chapter presents details of the implementation of established two-dimensional active contour models as well as describing several modifications and improvements that enhance accuracy and allow greater flexibility of use.

Section 3.1 provides background implementational detail on active contour models, Section 3.2 describes the benefits and details of using topology adaptive models, while Section 3.3 presents novel work that improves upon the performance of existing models for the purposes of segmentation of liver and other abdominal structures. Section 3.4 briefly describes what influence the correct setting of parameters has on the segmentation results; finally, Section 3.5 contains figures and descriptions for examples of results that can be obtained using active contour models. Sections 3.7 summarise the work in the chapter and the novel contribution.

3.1. Background

3.1.1. Contour structure

The basic composition of the active contour model (or *snake*) implemented in the work presented in this thesis is closely related to the *dynamic force* formulation of active

contour movement, which is described in detail by Xu, Pham and Prince in Sonka and Fitzpatrick (2000). The main difference between a force-based model and the classic *energy minimising* formulation (see Chapter 2) is that the latter calculates the movement of each vertex describing the contour by calculating the energy for the entire curve, so that the movement of each vertex can be affected by all the vertices describing the curve. By contrast, the force-based model calculates and applies a force at each location of each *individual* vertex, thus the vertices are more independent.

The decision on whether to use an energy-based or force-based model depends on its intended application - whether it is beneficial for the curve to be treated as a single entity, encompassing an entire region; or whether it is suitable for the curve to move according to a series of forces applied at the points that describe it. Importantly, a force-based method allows the use of external forces that are *not* potential forces i.e. those that cannot be written as the negative gradient of potential energy functions, for example an inflationary force. It is this benefit, along with greater flexibility allowed applying independent forces at each point, which led to the decision to use a force-based model in this work.

The snake is represented as a contour, V, defined as a sequential set of connected vertices

$$V = \{v_i\},\ i = 1, \ldots, I;$$

where each node v_i is associated with time (t) varying locations, $\boldsymbol{x}_i$, in the image plane, where

$$\boldsymbol{x}_i(t) = [x_i(t), y_i(t)]$$

as well as internal elastic forces $\alpha_i(t)$, internal bending forces $\beta_i(t)$, inflationary forces $\gamma_i(t)$ and external forces $\delta_i(t)$. The movement behaviour of each vertex in the contour is governed by equation (3.1):

$$x\vec{X}_i = a\alpha_i + b\beta_i + c\gamma_i + d\delta_i \tag{3.1}$$

where $\vec{X}_i$ is the velocity of vertex v_i, x is a damping coefficient that controls the rate of dissipation of the kinetic energy of the nodes that is calculated by equations (3.2) and (3.3) below, and a, b, c and d are weighting parameters that control the relative strength of each force. The equation is applied iteratively to each vertex, continuing until there is no movement of any vertex in V.

The implementation of the active contour model involves use of a circular double linked-list, where each node represents a vertex and contains references to the vertices immediately adjacent to it. The circular nature of the list allows the vertex at the end of the list to refer to the vertex at the start of the list, as demonstrated in Figure 3.1. Mathematically this concept is expressed by applying a periodic boundary condition to V so that $x_1(t) = x_I(t)$, thus ensuring that V is always produces a closed contour model.

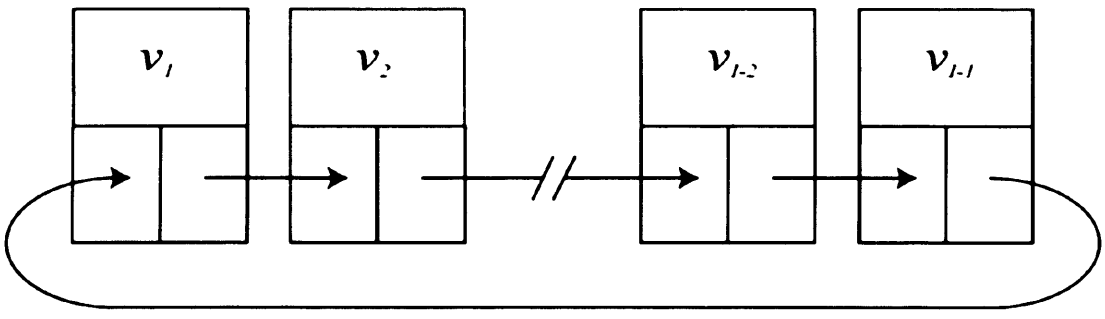


Figure 3.1: *Circular double linked list. The implementation of the active contour involves the use of a circular double linked list, where each vertex object references both the previous and subsequent adjacent vertex.*

3.1.2. Vertex movement

Equation (3.2) controls the movement of each vertex and is a balance of *internal forces* (elastic $\alpha_i(t)$, bending $\beta_i(t)$) and *external forces* (inflationary pressure $\gamma_i(t)$, image $\delta_i(t)$). The internal elastic force is controlled by equation (3.2):

$$\alpha_i(t) = 2x_i(t) - x_{i-1}(t) - x_{i+1}(t) \tag{3.2}$$

and will have larger values when there is a large gap between successive points on the contour (larger still if the gap sizes are uneven). Thus it acts to maintain a uniform spacing between vertices, as demonstrated in Figure 3.2.

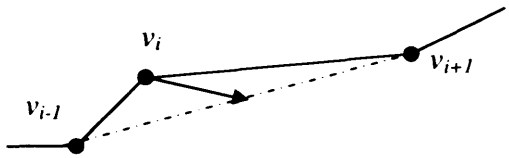


Figure 3.2: *The elasticity force. Drags the control point v_i towards the average location of the control points v_{i-1} and v_{i+1}.*

The internal bending force is a controlled by equation (3.3):

$$\beta_i(t) = x_{i-2} - 4x_{i-1} + 6x_i - 4x_{i+1} + x_{i+2} \tag{3.3}$$

and will have larger values at areas of higher contour curvature, thus it acts to smooth the contour and reduce areas of high curvature, as shown in Figure 3.3.

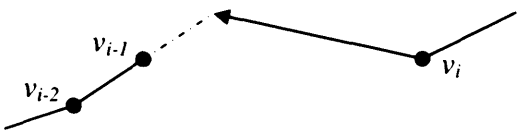


Figure 3.3: *The bending force. Drags the control point v_i towards the position predicted by the control points v_{i-2} and v_{i-1}. The control points v_{i+2} and v_{i+1} create a similar force.*

An inflationary force $\gamma_i(t)$ is used at each vertex to expand the contour from an initial starting point, and is calculated using equation (3.4). The direction and magnitude of the force for each vertex is calculated using the unit Normal vector to the contour at each vertices' location.

$$\gamma_i(t) = f(I(\boldsymbol{x}_i(t)))\, n_i(t) \tag{3.4}$$

where $n_i(t)$ is the unit Normal vector to the contour at vertex $v_i(t)$. The binary function $f(I(v_i(t)))$ provides basic region information which is used to control the deformation of the contour over time. It is based upon image intensity data and is slightly modified from the similar function described by McInerney and Terzopoulos (2000) in that it uses two threshold levels, an upper and a lower threshold, instead of a single value:

$$f(I(\boldsymbol{x}_i(t))) = \begin{cases} +1, \text{ if } T_{lo} \le I(\boldsymbol{x}_i(t)) \le T_{hi}, \\ -1, \qquad\qquad \text{otherwise,} \end{cases} \tag{3.5}$$

where T_{lo} and T_{hi} are upper and lower grey-level thresholds and $I(\boldsymbol{x}_i(t))$ represents the average pixel grey-level in a 3x3 square centred upon $\boldsymbol{x}_i(t)$. This prevents the snake from leaking into other organs in the abdomen at locations where the external image energy is not sufficient to stop the snake. To prevent the normal force from oscillating indefinitely between areas of intensity within/without the threshold levels, the constant c (in equation (3.1)) acts as a relaxation parameter which is gradually lowered towards zero as soon as any oscillation in the sign of f is detected.

The external force image-based force, $\delta_i(t)$, is not calculated as a potential force as it is in the traditional energy minimising active contour model. Instead, the relative strength of edges present within the image is calculated using the Kirsch compass filter (Russ,

1999). The operator works by convolving the image data surrounding a given pixel with an edge detecting kernel (see Figure 3.4), before rotating the kernel and moving each value clockwise one stage. By rotating the kernel eight times, convolving with the image each time, the mean value of the results of these convolutions represents the strength of a gradient-based edge for eight different gradient directions.

$$h_1 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix} \quad h_2 = \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix} \quad h_3 = \begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix} \ldots$$

Figure 3.4: *Kirsch edge detector. The first three convolution filters used in a Kirsch edge detecting algorithm. h represents one instance of the rotated kernel.*

The filter is applied for each pixel in the entire image, resulting in an edge map. The values of the relative strength of edges at each pixel are then used in equation (3.1).

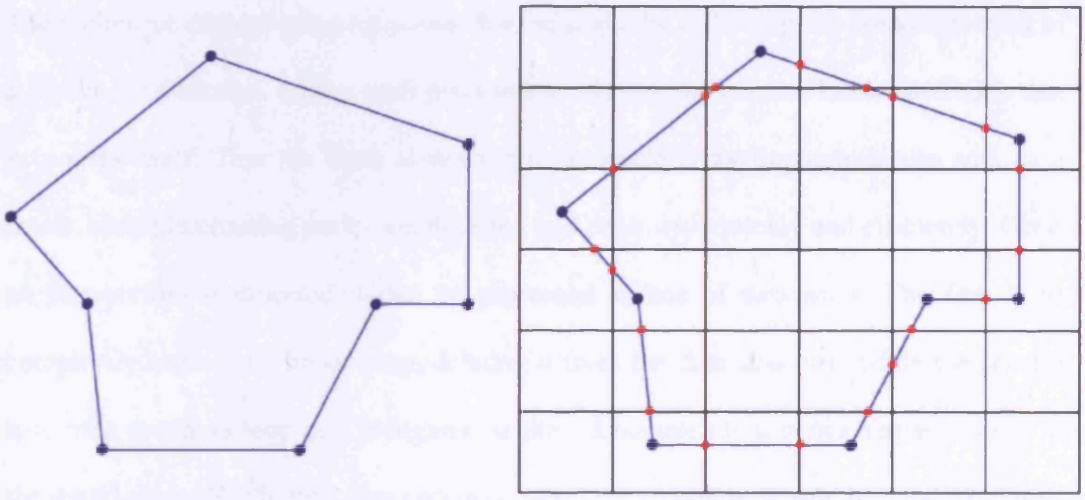
3.2. Topological adaptive active contour models

3.2.1. Reparameterisation

As described in Chapter 2, a topology adaptive snake, or *T-Snake*, is one technique that can be used to avoid the problems associated with traditional active contour models. The key difference between a topology adaptive model and a conventional snake is that the set of vertices in a T-snake does not remain constant. As the contour moves due to internal and external forces, it is *reparameterised*, at regular intervals, to a grid superimposed upon the image. At each reparameterisation step, the previous set of vertices is removed and a new vertex added at each point where the contour intersects with the superimposed grid. Figure 3.5 shows a simple, diagrammatic representation of the concept. Here, the use of a circular double linked-list (illustrated in Figure 3.1) to implement the contour demonstrates its advantages. Addition or subtraction of a vertex from the list merely requires the modification of only two existing references, one each

in the vertices that sandwich the newly created vertex. Unlike a standard array implementation, there is no movement of objects in physical memory, and thus the reparameterisation is fast and memory efficient (see Chapter 4 for more details on algorithm speed).

This reparameterisation overcomes aliasing problems that naturally occur with inflationary contours, as the creation of new points reduces the average distance between each point. At each reparameterisation the resolution of the contour is effectively reset. This method allows the addition of an inflationary force to the basic equations while avoiding any aliasing issues. A contour can now be initialised at any point within the structure that is to be segmented, thus one of the traditional snake's largest problems, that of accurate initial placement, is avoided.



(a) An example contour. The blue dots represent the vertices of V.

(b) A grid (black) is superimposed on the image, and new vertices (red dots) are added to V at the points at which the lines that connect the existing vertices (blue dots) intersect with the grid.

Figure 3.5: *Simple reparameterisation of a contour.*

3.2.2. Splitting and merging

One disadvantage of using an inflationary active contour model is that, unless the image data representing the object that is to be segmented is relatively homogenous, it is

possible for the contour to get trapped on unwanted edges, and thus not find the correct boundary. While the tissue of the liver is relatively homogenous (see Chapter 5 for more details on specific issues regarding liver segmentation) a true topological adaptive model should be able to split (separate into multiple distinct, contours), and merge (join two separate regions of the contour that have moved to contact each other). McInerney and Terzopoulos (2000) incorporate region information (based on the grid cells within/outside the snake) to determine whether the snake has merged with itself, and use this to split the contour into two separate contours, the original and a new, separate, set of vertices. Yet the details of this technique was deemed unsuitable during the development of the model described in this work, due a more efficient method of contour merging and splitting being developed.

The technique detects 'crossing points' for the snake by following the contour around in a clockwise direction, testing each pixel to see whether the contour has looped such that it crosses itself. This has been abstracted to a simple array implementation and, as a result, multiple crossing points are detected and dealt with quickly and efficiently. Once an intersection is detected it can be processed in one of two ways. The first is to completely remove the inner loop, deleting it from the data structure; while the second is to treat the inner loop as a 'daughter' snake – a completely separate active contour to the initial snake. While the latter option is useful for wrapping around internal structures that the user may not desire to be part of the final segmentation, it also raised the possibility of the creation of unwanted 'noisy' daughter snakes, as noisy edges within the image prevent the contour from expanding correctly. Figure 3.9 shows an example of the two possible methods of dealing with intersections.

3.3. Novel improvements

3.3.1. Reparameterisation based on curvature

A novel method of reparameterisation has been developed during the course of this project which improves upon both the final accuracy and efficiency of the segmentation procedure. The reparameterisation of the snake is different from that presented by McInerney and Terzopoulos (2000) in that a rectangular grid is used for reparameterisation (as opposed to a triangulated simplicial cell structure) and the resolution of this grid changes *depending on the local curvature* of the contour at each individual vertex. In fact, the data structure consists of three separate grids of decreasing cell size (increasing resolution). These are deduced experimentally and can be set to any values; in tests, however, it was found that resolutions of 8, 5 and 2 pixels provided accurate results. Depending on the curvature of the contour, it is reparameterised on a specific grid; if the curvature is high, the contour is reparameterised on a smaller grid size, otherwise it is reparameterised on a larger grid size. The technique is illustrated hypothetically in Figure 3.6. The method used to deduce the curvature $\kappa_i(t)$ at each vertex $v_i(t)$ is a simple analysis of the average magnitude of the inner angles of the contour between the Euclidean lines connecting the set of four vertices either side of v_i. The angle between two lines is calculated by converting the lines into vectors and taking the inverse cosine of the dot product of the normalised vectors. Thus $\kappa_i(t)$ can be calculated using equation 3.6:

$$\kappa_i(t) = \sum_{v_{i-4}}^{v_{i+4}} \cos^{-1}(\mathbf{u} \cdot \mathbf{w}) \tag{3.6}$$

where **u** and **w** represent the normalised vectors of the two lines forming the angle at each vertex.

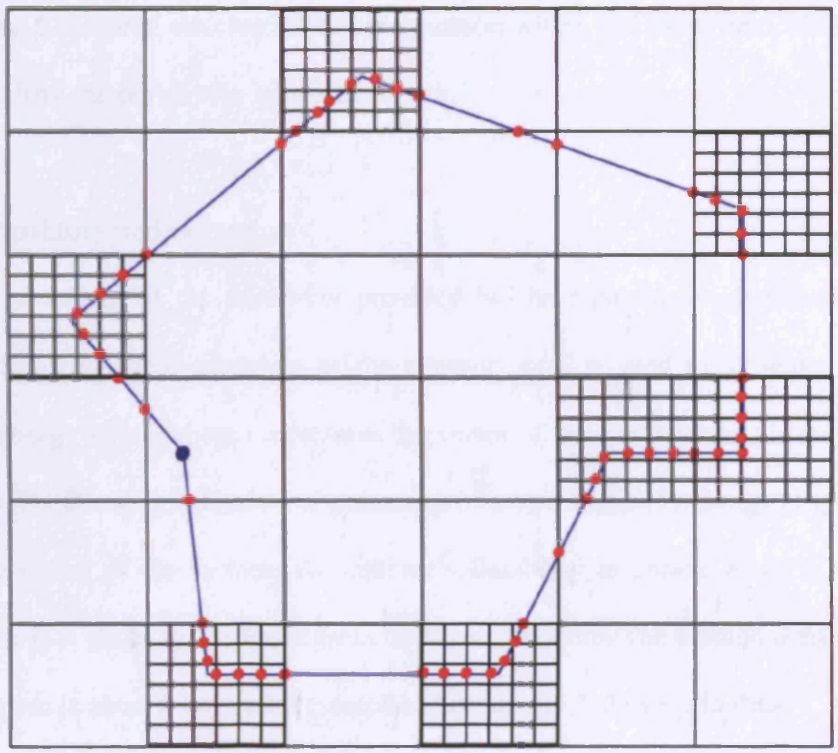


Figure 3.6: *Changing grid resolution. A modified version Figure 3.5, showing the effect of increasing the resolution of the grid in areas of higher curvature. The red dots represent new controls points that are created where the contour intersects the grid.*

The major advantage of this original technique is that the resolution of the snake increases at complex and highly irregular areas of the shape to be segmented, thus enabling the inflating contour to push itself into sharp corners and avoid aliasing effects that might otherwise cause a false segmentation result. In areas where the contour is relatively straight a larger grid size is used for reparameterisation and less points are required to accurately represent the shape, reducing the number of unnecessary calculations and improving the performance of the snake. Furthermore, the increased inherent stiffness in areas of lower resolution decreases the probability of the contour 'leaking' through weak edges (see Section 3.3.2 below).

This method sees improved segmentation results compared to those obtained with a single, uniform grid (see Chapter 5), as the snake is able to push into sharp corners and

wrap around complex structures that are common within the liver, particularly around the area where the portal vein leaves the organ.

3.3.2. Flexibility and curvature

To take advantage of the flexibility provided by the reparameterisation scheme, the magnitude of the local curvature of the contour, $\kappa_i(t)$, is used to influence $\beta_i(t)$, the bending energy force, where i represents the vertex of the contour, and t is the time step as before. By allowing the value of parameter b in equation (3.1) change proportionally to the curvature of the surface, the contour's flexibility in curved areas is increased; conversely it is set to act stiffer in areas of lower curvature. The method used to deduce the curvature is identical to that presented in Section 3.3.1 above, and thus

$$b_i(t) \propto k \frac{1}{\kappa_i(t)} \tag{3.7}$$

where k is a constant used to control the influence of b. The result of this modification is particularly useful in liver segmentation –in a typical image of one of the central slices from a CT abdominal scan focusing on the liver, one of the areas in which the Kirsch edge detector frequently does not highlight a desired edge is the boundary between the liver and the intercostal muscles, as demonstrated in Figure 3.7. As the curvature of the boundary is relatively low in this area, the added stiffness of a segmenting contour (due to the use of equation (3.7) in deciding the value of parameter b_i) minimises the chances of the contour 'leaking' outwards through the weak edge (thus over-segmenting the image and including intercostal muscle in the segmentation of the liver). However, on the opposite side of the image the liver has a sharp, acute boundary with well defined edges; in this situation the higher curvature of a contour ensures that the value of b is

lower, and thus it has extra flexibility to enable it to better segment this highly curved area.

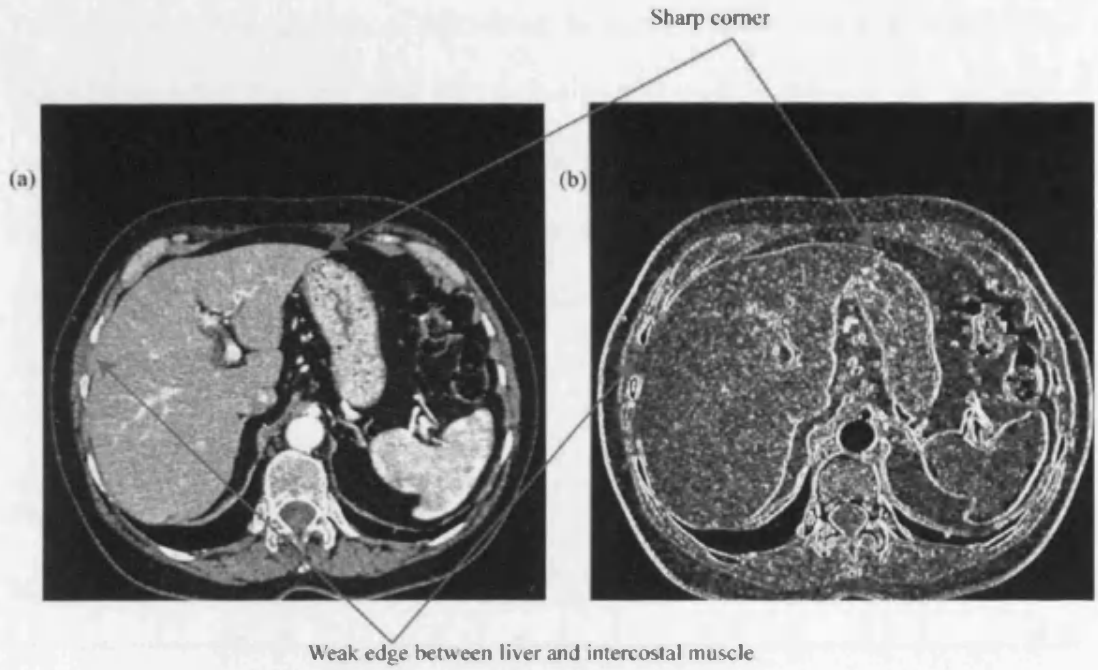


Figure 3.7: *Liver segmentation challenges. A CT image of the abdomen, (a), that has been convolved with the Kirsch compass edge detector, (b). The weakness of the boundary between the liver and intercostal muscles is highlighted on the right of the image, as is the sharp corner of the upper left area of the liver.*

3.4. Parameters

The movement of the contour is affected by the values of the parameters used to control the relative effect of each term of equation 3.1. The values of the parameters were set by analysing the results of applying the algorithm to subset of images from the datasets used in Chapter 5 (5-10 images each from a subset of 8 datasets). The images for this subset were chosen at random from the middle 25% of slices in the datasets as, in general, it is in this area that the liver's boundary is more complex, thus presenting a more difficult segmentation challenge. The decision as to whether a given set of parameters gave accurate segmentation was determined by visual analysis of the results and by comparison of the area of the automatically segmented region with that obtained by manual segmentation (see Chapter 5 for more details of the manual segmentation procedure).

The values of the parameters used for all the 2D work in this thesis are presented in Table 3.1. Note that parameter c, controlling the strength of the inflationary force, has a much larger value than the other parameters as it is used to scale-up the unit vector value of γ to enable the surface to expand. The parameters T_{hi} and T_{lo} were set to be 'SD' standard deviations above and below the mean intensity value of the pixels enclosed within the contour at the time of its initialisation, thus the value SD directly controls the values of T_{hi} and T_{lo}.

Parameter	x	a	b	c	d	SD
Value	1	0.5	0.3	5.0	0.4	1.5

Table 3.1: Parameter value for 2D algorithm

It should be noted that this method is not a particularly robust way of setting parameters, and while it was not considered practical to undertake a full parameter optimisation study, a small experiment was run to discover which of the six parameters was the most sensitive (i.e. which had the greatest effect on the segmentation results after being changed by the smallest amount). Furthermore, once the two most sensitive parameters were discovered, they were subject to further analysis to see how changing both of them simultaneously affected their sensitivity. Full details and results of this study are presented in Section 5.6.

3.5. Examples and test images

Segmentation tests involving a variety of grey-scale images were carried out during the development of the active contour model, and this section presents several figures relating to those tests that demonstrate the functionality of the model. The signal to noise ratio of the images was calculated by dividing the mean intensity value of the image by the standard deviation of the pixel values. Full results for the segmentation of liver from CT images are presented in Chapter 5.

Figure 3.8 shows the segmentation of the circular object (the back of a clock) in a greyscale photograph. The image was chosen a simple, real-world example of how the inflationary model deals with edges in digital images. It is important to note that, as long as the contour is initialised at some point within the structure to be segmented, functionally identical results are achieved irrespective of the precise starting location of the contour. In natural images such as this one, the presence of directional light creates shadows which appear as strong edges when the image is processed using an edge detector, such as the Kirsch compass filter used in this case. As a result, the 'correct' edge of the object may not always be found. Figure 3.8(c) shows the result of weakening parameter d, which controls the influence of the image-based force on the movement of the contour. If d is lowered, the contour may leak through the desired edge.

Figure 3.9 shows several synthesised images designed to test to functionality of the contour in an artificial environment that is similar to the intended use of the contour on clinical data. They were synthesised by drawing a foreground shape of consistent grey-level value against a background that varies in intensity according to a gradient mask over the image; this non-uniform background results in foreground-background edges

that vary in their prominence over the image. The benefit of a self-reparameterising, inflationary contour is evident here as most areas of the shape are well segmented, although the final result on the 'torture test' star-shaped structure on the right of the image presents too great a challenge for contour to segment with 100% accuracy. One important point to note is that, because the forces that act upon each vertex are entirely *local*, the segmentation of complex boundaries in one area of the image has no effect on the segmentation of more simple boundaries in other areas of the image. Subsequent images in Figure 3.9 show the effect that adding increasing levels of artificial noise has on the accuracy of the segmentation. Gaussian noise was added, using a standard algorithm presented by Press *et al.* (2002), in increasing amounts. The greatest effect that increased noise has is the proliferation of small, noisy edges, which prevent the contour from reaching the 'correct' edges. Figure 3.9(b) shows that a small level of noise does not greatly affect the final result, while (c) and (d) show that as noise increases, the contour is increasingly unable to detect the correct boundaries.

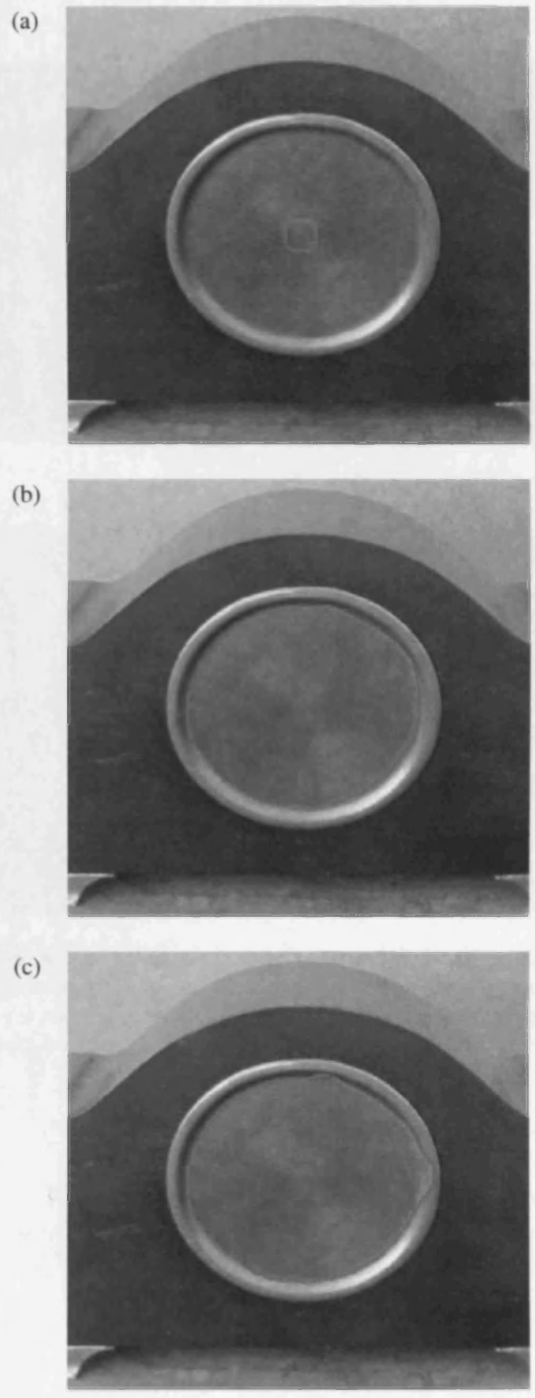


Figure 3.8: *Segmentation of an object from a natural photograph. Figure (a) shows the contour at the point of its initialisation, Figure (b) shows the result of the segmentation procedure, and Figure (c) shows 'leakage' through a weak edge which occurs when external force parameter d is lowered. The signal-to-noise ratio of the image is 1.95.*

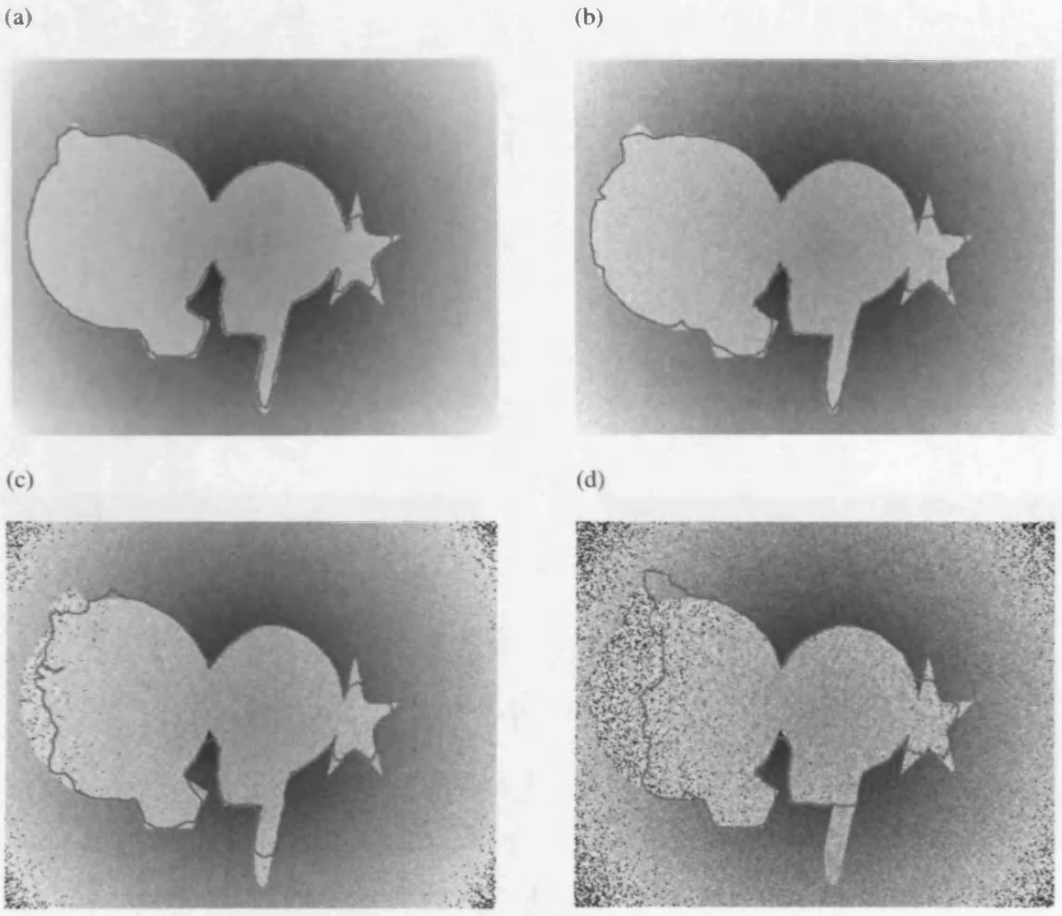


Figure 3.9: *Effect of noise. Four images showing the effect that adding Gaussian noise has on the accuracy of the segmentation of a complex shape. The image (a) contains no noise, whereas the images in figures (b), (c) and (d) contain increasing levels of noise. The signal to noise ratios of the images are as follows: (a) 3.66; (b) 3.65; (c) 3.42; (d) 2.80.*

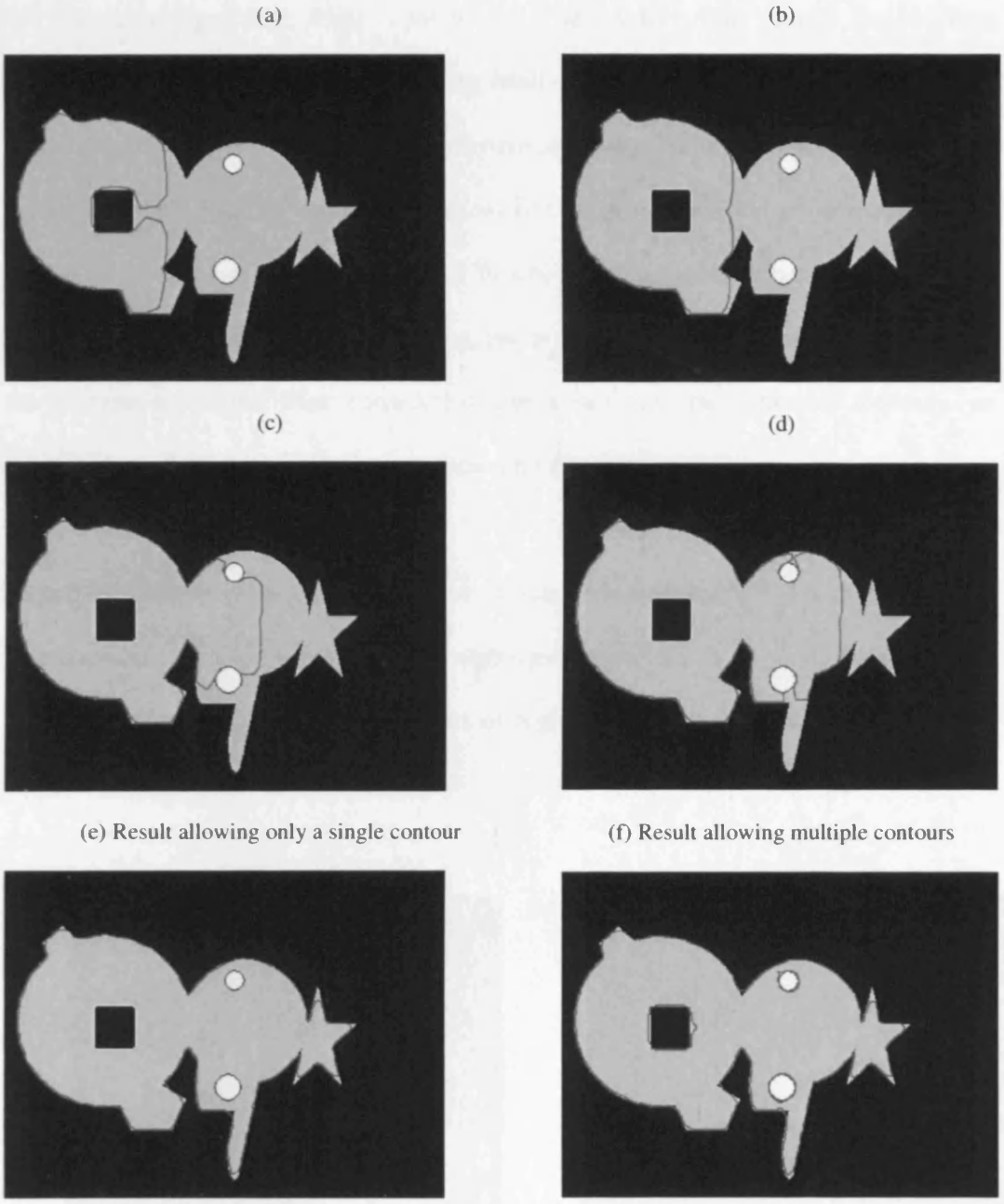


Figure 3.10: *Split and merge. Six images demonstrating the model's ability to split and merge. No noise was added to the image and thus the contrast-to-noise ratio between the grey area (to be segmented) and the black/white areas is infinite.*

Figure 3.10 demonstrates the behaviour of the contour when faced with objects fully contained within the area that is to be segmented. The figure shows images of the evolution of the contour during its movement, and demonstrates its capability to merge with itself after wrapping around extraneous structures. The first five images ((a) to (e)) show the segmentation proceeding with the option of creating daughter snakes – when

the contour merges with itself – set to *off*. The model wraps around the blocking structures as it inflates, and merges with itself as soon as it self-intersects. The final image (f) shows the result of the segmentation, under identical conditions, with the option to create daughter snakes turned *on*. In this case, when the initial contour self-intersects, it executes a split-and-merge function to create two separate snakes – the 'original' contour that continues inflating, and a daughter snake which constricts around the extraneous object. Not considering the boundaries of computer memory, an unlimited number of such daughter snakes can be created.

Figure 3.11 shows example segmentation results of two abdominal structures using the active contour. Figure 3.11(a) shows segmentation of the liver, and 3.11(b) of the kidney. Further examples and full results of segmentation of such abdominal structures are shown in Chapter 5.

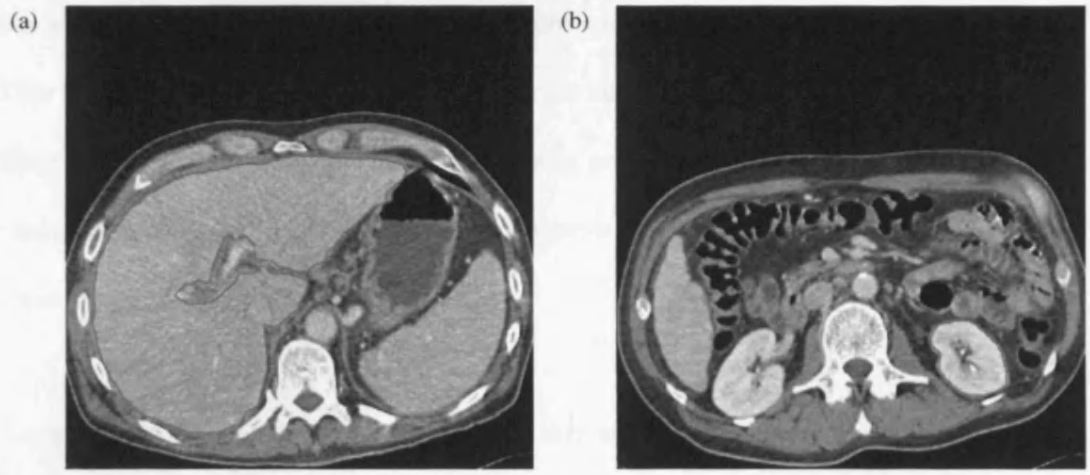


Figure 3.11: *Successful segmentation. Two sample images of successful segmentation of abdominal organs from a CT image slice. Figure (a) shows liver segmentation, (b) shows kidney segmentation.*

3.6. Summary

This chapter has described the methodology and implementation of an active contour model that allows the automatic detection of boundaries of objects that exist in a two-dimensional image. The model described is an inflationary parametric model that is similar in its basic details to a *T-snake* developed by McInerney and Terzopoulos (2000), however several unique and novel improvements have been proposed; in particular the ability to reparameterise the contour to a higher resolution at areas of higher curvature, the linking of contour flexibility to local curvature, and a more efficient manner of detecting and dealing with self-intersection.

Section 3.1 described the basic concept of an active contour and the data structure used for its implementation. It demonstrated that the basic principles of movement of the developed model are similar to the *dynamic force formulation* type of model.

Section 3.2 introduced the concept of a topologically adaptive active contour model. This technique avoids three problems that face standard active contour models (initial placement, topological adaptivity, and aliasing problems using inflationary forces) by reparameterising the contour, at specific time-steps during its evolution, to a grid superimposed upon the image.

Section 3.3 describes a series of novel techniques that improve upon topologically adaptive models previously described in the literature (McInerney and Terzopoulos, 2000; Giraldi, Strauss and Oliveira, 2003). The two main novel contributions of this chapter are the implementation of a model that reparameterises to a higher resolution grid at areas of higher contour curvature, thus allowing greater accuracy of the segmentation of objects with complex boundaries while maintaining efficiency of the

segmentation procedure; and the linking of the contour's flexibility to its local curvature, ensuring increased flexibility to correctly locate a complex boundary, yet increased stiffness to prevent the contour from leaking through perceived 'holes' in a relatively even boundary.

Section 3.4 briefly describes the effects that the correct or incorrect setting of parameters can have on the accuracy on the segmentation, and this topic is further discussed in Chapters 5 and 7. Finally, Section 3.5 shows some examples of the topologically adaptive active contour model in action.

3.7. Novel contributions

- Reparameterising an inflationary active contour to a greater level of detail in areas where the local curvature of the contour is greater. As a result, objects with complex boundaries can be efficiently segmented with greater accuracy.
- Linking an inflationary active contour's local flexibility to its local curvature, allowing the contour to deform correctly into complex structures, yet increasing its stiffness to prevent it from leaking through weak edges in other areas of the object boundary.

4. Active surface models

Segmentation of two dimensional images is a powerful method of extracting information from the raw data. Yet when the available data is extended to a third dimension, it can be unproductive to treat it as merely a set of unconnected 2D slices. In the case of liver segmentation, for example, the organ is obviously a 3D structure, and a CT dataset showing the liver can be considered as a 3D dataset. By analysing and segmenting the dataset in its entirety in a single instance (as opposed to separating it and segmenting different parts of it over many instances) it is possible to maximise the information acquired from the dataset as a whole, resulting in topologically more robust segmentation.

Section 4.1 describes some background information regarding deformable surfaces. Section 4.2 and 4.3 describe the data structure of the surface implemented in this work, and the equations and information that govern the movement of the vertices of the model. Section 4.4 presents novel work on the reparameterisation of the surface, explaining how the active surface is described by a greater number of vertices at areas of higher curvature. Section 4.5 details how surface self-intersection is detected and handled, while Section 4.6 concludes with an example of the segmentation procedure on an artificial shape. Section 4.7 and 4.8 summarise the work and novel contribution of the chapter.

4.1. Background

The discussion and images in Chapters 3 demonstrate that, in many instances, the described 2D active contour model described in Chapter 3 is effective at segmenting structures from different types of image. However when applied to a series of images

representing a three dimensional dataset, it is immediately apparent that segmenting in one plane, at regular repeated samples of the dataset, merely creates a set of contours that must be layered together to reconstruct the actual shape of the object of interest. Furthermore, each segmentation instance in 2D inherently ignores the adjacent data in the third dimension.

To overcome these disadvantages, one method of true 3D segmentation extends the concept of a contour to a third dimension by creating a *surface*. For segmentation purposes, the surface can be initialised within the 3D object to be segmented, and inflated (in a manner analogous to air filling a balloon) until it reaches the boundaries of that object. This overcomes problems that affect 2D segmentation, such as the effects of local noise in a single slice, segmentation of intricate shapes that vary continuously in the third (z) axis of the dataset, and 3D reconstruction of objects following segmentation. Nonetheless it also generates its own set of problems, such as surface movement, the accurate description of the structure of the surface, and collision detection.

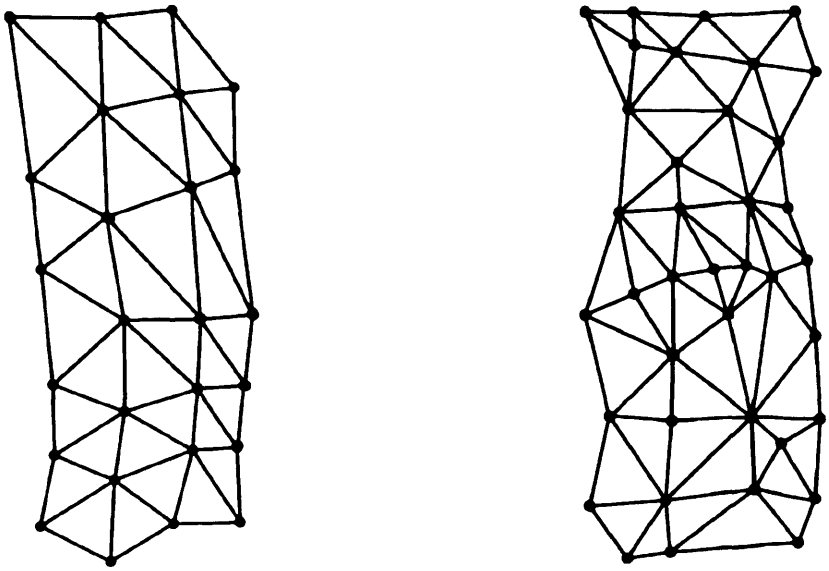
Chapter 2 described several approaches to segmentation using 3D surfaces, including active surface models, 3D levels sets, and 3D deformable m-reps. The 3D segmentation method developed during this project is closely related to an active surface model. Following the development of the 2D segmentation technique, there existed several possibilities to extend the novel schemes established during the development of that work to 3D, and for this reason work on active surface models was begun in favour of studying the two methods of implicit surface segmentation cited above.

Chapter 2 also reviewed previous work in the active surface field, in particular that which has been done using discrete deformable models. Bulpitt and Effort (1996) and Ahlberg (1996) present an extension of the basic active contour model energies to three dimensions, while Park *et al.* (2001), and Lachaud and Montanvert (1999) describe using similar energy formulations in conjunction with adaptive remeshing systems that increasing surface flexibility. The work presented in this chapter draws on aspects of this previous work while implementing several novel initiatives to improve segmentation accuracy, in particular the use of curvature to affect surface movement and mesh detail, and making use of region information to assist in correct boundary detection. As such the developed model can be associated with a 3D version of the Active Region Model developed by Ivins and Porrell (1994).

4.2. Mesh data structure and movement

4.2.1. Basic mesh structure

The model developed in this work for segmentation is based on a closed, orientated, triangular surface mesh. Meshes can be classified as either structured or unstructured. A structured mesh has a uniform topological structure and can be defined by the fact that the indices of the neighbours of any vertex can be calculated using simple addition. An unstructured mesh, however, requires each vertex to store a list of its neighbours, and the size of this list may vary in between vertices. Figure 4.1 shows and example of the two types of mesh, showing how links between vertices are represented by a set of *edges*. Note how, in a structured mesh, each vertex is associated with a fixed number of edges (disregarding boundary vertices), whereas vertices in an unstructured mesh may be associated with varying quantities of edges.



(a) A structured mesh (b) An unstructured mesh

Figure 4.1: *Structured and unstructured meshes. The black dots represent the vertices of each mesh, while the black lines show the edges connecting them.*

Structured meshes have the advantage of being straightforward and efficient to use, however their inherent inflexibility is a limiting factor if the mesh is to be modified due to the movement of its vertices. In this work, an unstructured mesh is used and the reasons for this are elaborated upon in Section 4.4.

4.2.2. Mesh representation

The data structure for the surface comprises a non-directional graph-like structure, comprising two complementary lists. The first consists of a base set of vertices

$$V = \{v_i\},\ i = 1, \ldots, I,$$

representing the surface itself, where each vertex v_i is associated with time varying locations, $\boldsymbol{x}_i(t)$, in three-dimensional space.

$$\boldsymbol{x}_i(t) = [x_i(t), y_i(t), z_i(t)] \tag{4.1}$$

The second list consists of a set of *faces*, triangles in 3D space. Each face consists of references to three vertices, and each vertex has associated with it references to the faces which it is a part of. The list of faces is an important modification to the basic graph structure of the surface. Not only does it facilitate display of the surface (the rendering algorithm requires a list of triangles, and their normal vectors, to create the display on the screen), but also facilitates implementation of 3D meshing algorithms, such as reparameterisation and collision detection, which will be described in detail in Sections 4.4 and 4.5.

To create an initial surface with a suitable amount of points for accurate representation of a contour, a quasi-regular polyhedron is created, connected by edges to form triangles. Using a 20-vertex polyhedron as a base structure, with equal distances between each vertex, a new vertex is placed at the mid-point between every two vertices. This process can be iterated until the required number of starting vertices is obtained.

4.2.3. Vertex movement

The movement of the vertices of the surface is governed by the same basic equation as the 2D active contour model, with internal elastic forces $\alpha_i(t)$, internal bending forces $\beta_i(t)$, inflationary forces $\gamma_i(t)$ and external forces $\delta_i(t)$ all governing the velocity, $\vec{X}_i$, of each vertex v_i, which is weighted by parameter x.

$$x\vec{X}_i = a\alpha_i + b\beta_i + c\gamma_i + d\delta_i \tag{4.2}$$

The parameters a, b, c, and d again control the effect that each of these forces has on the movement of each vertex. The equation is applied iteratively to the vertices of the

surface, each iteration representing a single time step. Once a situation has been reached where no vertices are moving ($\vec{X}_i \leq 0$ for each vertex), the algorithm terminates.

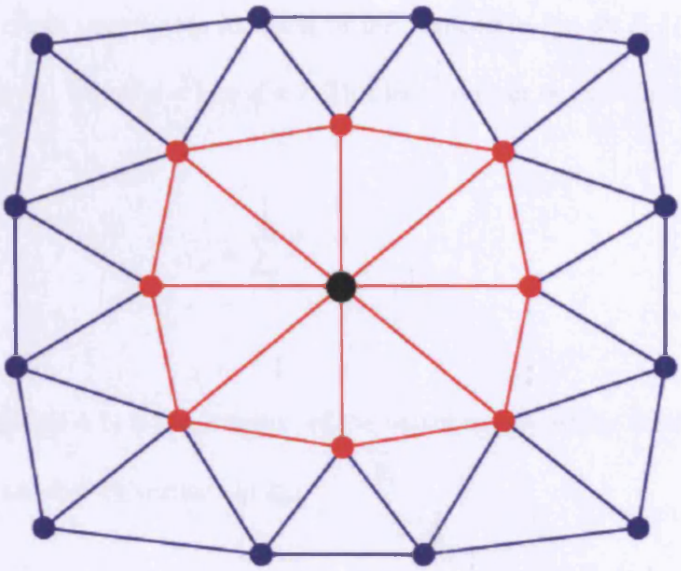
The calculations for each of the components of equation (4.2), however, are very different to their equivalent in 2D. As demonstrated in Figure 3.2, elasticity can be regarded as a force pulling a vertex to the average location of its neighbouring vertices, while resistance to surface bending can be regarded as the force from a vertex to a point linearly 'predicted' by surrounding points (see Figure 3.3). In 2-D these calculations only involve the two surrounding points on the contour, in 3D the *set* of surrounding points has to be considered to calculate the average.

Consider a vertex v_i within V, and let

$$S_{i,d} = \{S_{i,d,j}\}, j = 1,\ldots, J,$$

be the set of J vertices connected at a distance d edges from v_i. For example, if $d = 1$, S would be the set of vertices connected by one edge to v_i; if $d = n$, S is the set of vertices connected to v_i by n edges. Figure 4.2 shows a diagrammatic representation of which vertices are considered part of which set. It should be reiterated that the mesh is a *surface mesh*, and thus edges are only present *across* the surface; edges that would form tetrahedral are not permitted.

(a)



(b)

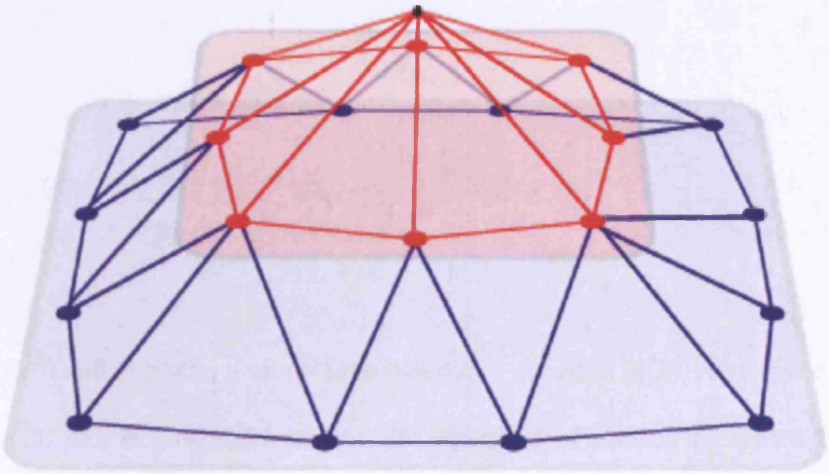


Figure 4.2: *A diagrammatical representation of a surface mesh. The black dot represents vertex* v_i*, the red dots the set of vertices at distance* $d = 1$*, and the blue dots the set of vertices at distance* $d = 2$*. Figure (a) shows a representation of the surface in 2D, whereas Figure (b) shows a pseudo-3D recreation of the surface. The shaded rectangles represent the average plane of the* $d=1$ *vertices (red) and* $d=2$ *vertices.*

Let $c_{i,d}$ be the *average coordinate location* of the vertices in the set $S_{i,d}$ (the vertices at distance d around v_i), where $d = 1$ or $d = 2$. This location can be calculated using:

$$c_{i,d} = \sum_{j=0}^{J-1} \frac{\boldsymbol{x}_j}{J} \tag{4.3}$$

where $\boldsymbol{x}_j$ (see equation 4.1) is the location of the vertex v_j (one of the vertices of set $S_{i,d}$), and J is the total number of vertices in $S_{i,d}$.

The elasticity, $\boldsymbol{\alpha}_i(t)$, and bending, $\boldsymbol{\beta}_i(t)$, forces in the point v_j can now be expressed as:

$$\boldsymbol{\alpha}_i(t) = \boldsymbol{x}_i(t) - c_{i,1} \tag{4.4}$$

$$\boldsymbol{\beta}_i(t) = 3\boldsymbol{x}_i(t) + c_{i,2} - 4c_{i,1} \tag{4.5}$$

Equations (4.4) and (4.5) can now be substituted into equation (4.2). Note that equations (4.4) and (4.5) can be derived from the 2D equations of motion presented by Kass, Witkin and Terzopoulos (1987), using equation (4.3).

As in 2D, parameter b in equation (4.2), which controls the bending force $\boldsymbol{\beta}_i$, is set to be inversely proportional to the local curvature of the surface, $\kappa_i(t)$, at each vertex location (see Section 3.3.2 and equation 3.5)). Stokely and Wu (1992) present five different methods of estimating the curvature of an arbitrary 3D surface, and evaluate each method's performance by calculating the surface curvature of a variety of spheres of known dimensions. The majority of the methods they present are based on building local coordinate systems on the surface at the point of analysis i.e. transform all surface points (x, y, z), within a set distance of the central point of interest, to a local (u, v)

coordinate system, and estimate the curvature by applying equations in the local coordinate dimension. Yet one of their methods, termed the *surface triangulation* method, computes curvature, using (x, y, z) coordinates, from a series of adjacent flat triangles adjacent that project from the inspection point. Stokely and Wu's analysis shows that, for a noise free sphere, the graphs for the calculated sphere radii using the surface triangulation method closely match those for other techniques.

Due to the fact that an adaptation of this triangulation technique can be rapidly applied to the vertices in the surface described in this work (which are all associated with a set of triangular faces by default) and that this removes the requirement for the complicated and time-consuming processes of transforming (x, y, z) coordinates to local geometry coordinates for each vertex, an adaptation of the surface triangulation method was used to calculate the curvature for the points of the surface. The curvature is calculated for each vertex, v_i, by first calculating the sum of the angles, $\Sigma\theta$ (expressed in radians), between the edges projecting from v_i. This is then used to calculate the curvature, κ, according to the following equation (Stokely and Wu, 1992):

$$\kappa = 2\pi - \Sigma\theta \tag{4.6}$$

The curvature will be zero when the surface is completely flat (i.e. when $\Sigma\theta = 2\pi$ radians).

4.3. External forces

Two external forces also affect the movement of each vertex of the surface, an inflationary or pressure force (γ_i), and an image force (δ_i). Both are affected by basic statistical region information obtained from the surrounding voxels.

4.3.1. Inflationary force

This force used to 'inflate' the surface from its starting location and shape acts in the direction of the normal vector for each vertex, v_i. The normal is calculated using the sum of the cross products of the vectors that exist to form each *face* of the surface which v_i is a part of. It is normalised to unit length and used in equation (4.6) to calculate the inflationary force.

$$\gamma_i(t) = f(I(\boldsymbol{x}_i(t)))\, n_i(t) \tag{4.7}$$

where n_i is the unit normal at v_i. The binary function $f(I(\boldsymbol{x}_i(t)))$ is based upon region information and is used to prevent the surface from 'leaking' into area that surround the object that is to be segmented, where the boundary between these regions is not clearly defined. In its most basic form this function is identical to the one presented in equation (3.5) (reprinted below), considering that in this case $I(\boldsymbol{x}_i(t))$ is the mean voxel intensity in a 5x5x5 (see Section 4.3.2) voxel cube surrounding $\boldsymbol{x}_i(t)$; again T_{lo} and T_{hi} are low and high voxel threshold values respectively.

$$f(I(\boldsymbol{x}_i(t))) = \begin{cases} +1, \text{ if } T_{lo} \le I(\boldsymbol{x}_i(t)) \le T_{hi}, \\ -1, \qquad \text{otherwise,} \end{cases} \tag{3.5}$$

In tests during the implementation of the surface for the purpose of the segmentation of liver, a more sophisticated version of equation (3.5) was found to prevent the surface 'getting caught' on noisy voxel areas that fell outside the threshold values. It was found during tests that, on certain datasets, the threshold values T_{lo} and T_{hi} could not be set to any value that would prevent the surface leaking into unwanted regions *and* allow the surface not to get caught on anomalous areas within the region to be segmented. In liver segmentation, if T_{hi} was set at a level low enough to prevent leakage into the heart or the

kidneys, vascular structures within the bulk of the liver would prevent the accurate segmentation of the organ.

To counter this effect, a modified version of equation (3.5), using the standard deviation of voxels surrounding a vertex, is used to govern $f(I(\boldsymbol{x}_i))$, and thus $\gamma_i(t)$.

$$f(I(\boldsymbol{x}_i(t))) = \begin{cases} +1, \text{ if } T_{lo} \leq I(\boldsymbol{x}_i(t)) \leq T_{hi}, \\ +1, \quad \text{if } \sigma_{\boldsymbol{x}_i(t)} > \sigma_{orig}, \\ -1, \quad \text{otherwise}, \end{cases} \tag{4.8}$$

$\sigma_{\boldsymbol{x}_i(t)}$ is the standard deviation of the voxel intensity values in the area surrounding $\boldsymbol{x}_i(t)$, and σ_{orig} is the standard deviation of the voxel intensity of the volume of the dataset that is contained 'inside' the surface at the time of its initialisation.

The effect of applying equation (4.7) can be summarised as follows. If the mean voxel intensity surrounding a given vertex, $I(\boldsymbol{x}_i(t))$, is within the threshold values, the surface expands. If $I(\boldsymbol{x}_i(t))$ is outside the threshold values, but the surrounding voxels have a relatively high intensity variation, the surface will still expand. Yet if $I(\boldsymbol{x}_i(t))$ is outside the threshold values and the surrounding voxels have a relatively low intensity variation, the normal force is reversed, and the surface contracts. 'High' and 'low' local voxel variation is determined by comparison with the voxels that were contained 'inside' the surface at the time of its initialisation.

It should be noted that the actual size of the voxels, recorded as metadata when the CT scan is conducted, will affect the results of using local region information. In this implementation, local region information is gathered by using a volume of voxels of a pre-specified size (a 5x5x5 cube, as noted earlier). Thus, as the voxel dimensions differ,

so does the actual volume used to obtain region information. To guarantee that the same volume is used for each dataset, the voxel size metadata could be used at the time of segmentation to calculate the precise size of the local region. However, all the results presented in this thesis make use of a region measured in voxels, thus the actual region size used differs between datasets.

Figure 4.3 shows three examples of liver tissue where the value of the function $f(I(\boldsymbol{x}_i))$ can affect the movement of the surface (the images are 2D but the concept extends to 3D). Table 4.1 shows the average values and the standard deviation of the pixels in each sample image.

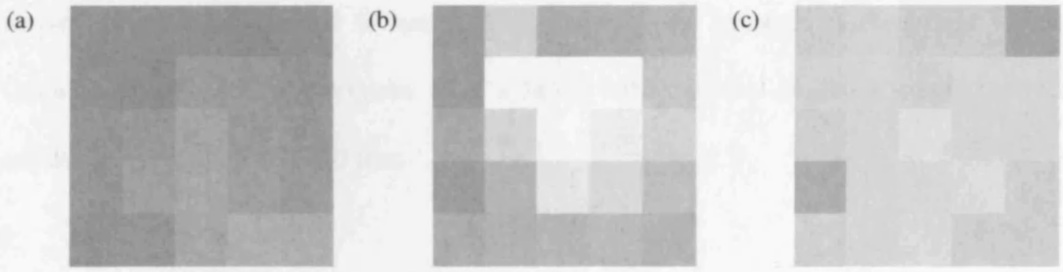


Figure 4.3: *Three 5x5 squares taken from 2D image slices of a CT dataset of the abdomen. Figure (a) shows an area of parenchymal liver tissue, figure (b) shows an area of liver tissue containing vascular structure, and figure (c) shows an area of parenchymal kidney tissue. All figures are taken from a single 256-level greyscale image slice, in the post-arterial contrast phase.*

Image from Figure 4.3	(a)	(b)	(c)
Average voxel value	169	213	212
Standard Deviation	15.74	30.2	12.71

Table 4.1: *Pixel and SD values. The average pixel values and standard deviation of the pixel values of the images in Figure 4.3.*

Figures 4.3(b) and (c) show that regions that possess similar average intensity values can have very different standard deviation values, and equation (4.7) exploits this to allow thresholds T_{lo} and T_{hi} to be set without the worry that irregular regions of an

object to be segmented (such as in Figure 4.3(b)) will force the incorrect movement of the surface.

4.3.2. Image forces

As in the 2D case, a Kirsch compass filter is used to calculate the strength of edges within the dataset, represented in equation (4.2) by term δ_i. The convolution filter is a 3D extension of the one presented in section 3.1.2 (see also Figure 3.4). The extension to 3D involves rotating a kernel in the three primary axes of the dataset, thus the filter calculates edge gradient in 24 separate 3D vectors. As in 2D, the gradient of the largest magnitude is used for the equations of vertex movement.

Region information is used to automatically control the strength of the image force. Once again the standard deviation, σ, of a 5x5x5 cube of voxel intensity values centred around $\boldsymbol{x}_i(t)$ is calculated and thus

$$\delta_i(t) = \varepsilon_{\boldsymbol{x}_i(t)}\, \sigma_{\boldsymbol{x}_i(t)} \tag{4.9}$$

where ε represents the result of convolution with the 3D Kirsch compass edge detector centred upon the $\boldsymbol{x}_i$. The size of the cube used to consider voxel intensity is chosen as a balance between obtaining local information while ensuring sufficient size to obtain meaningful region statistics. For example, if the region is too large then individual voxels could affect the movement of more than one vertex, if the vertices are closely spaced. Yet too small a region may not yield meaningful statistics.

The effect of equation (4.8) is to enable the vertices of the surface to 'push through' weak or noisy edges that may be given undue prominence by simply using the detector

alone. This enables more flexibility when setting parameter d in equation (4.1) and decreases the likelihood of the surface folding in upon itself and self intersecting.

4.4. Parameters

The parameters controlling the algorithm are different to those used for the 2D algorithm, due to the difference in the computation of several of the forces (for example, the local region data is used to greater effect in the 3D algorithm) and the differences that arise from using a greater number of surrounding vertices to calculate the internal forces. The method used to deduce the parameters, however, was analogous to the 2D case, in that the accuracy of using different parameter values was measured in a subset of 8 datasets by visual analysis of the segmentation and by comparison of automatically measured volume with that estimated by manual segmentation.

The values of the parameters used for all the 3D work in this thesis are presented in Table 4.2. Note again that parameter c, controlling the strength of the force in the normal direction, has a much larger value than the other parameters as it is used to scale-up the unit vector value of γ to enable the surface to expand. The parameters T_{hi} and T_{lo} were set to be 'SD' standard deviations above and below the mean intensity value of the pixels enclosed within the contour at the time of its initialisation, thus the value SD directly controls the values of T_{hi} and T_{lo}.

Parameter	x	a	b	c	D	SD
Value	1	0.1	0.3	6.0	0.1	1.5

Table 4.2: Parameter value for 3D algorithm

As for the 2D case, a study was carried out to investigate which of the six parameters has the most effect on the segmentation when changed, and the two most sensitive parameters were subject to further analysis to establish how their sensitivity was altered when both were changed simultaneously. Further details and results of this study are presented in Section 5.6.

4.5. Reparameterisation

The net effect of applying equation (4.1) iteratively to all vertices is the expansion, or inflation, of the surface, in a manner analogous to air filling a balloon. This creates two immediate problems. The first is that, as the spacing between each vertex increases, the strength of the internal forces, α_i and β_i, also increases; once the vertex spacing reaches a certain point, α_i and β_i will override the other terms of the equation and prevent the surface from expanding further. The second difficulty is the appearance of aliasing issues - the accuracy with which the vertices of the model represent the *actual* boundary of the object that it is segmenting decreases as the distances between vertices increase.

To combat these effects it is clear that some form of reparameterisation of the surface is required. Reparameterisation of inflationary active surface models has been previously addressed in different manners. Park *et al.* (2001), and Lachaud and Montanvert (1999), both use techniques such as direct point insertion, edge melting, splitting and flipping; whereas McInerney and Terzopoulos (2000) track the movement of the surface over a simplicial decomposition of the dataset, partitioning the dataset into a set of tetrahedra, and extracting the 'surface' at each new time step from the set of tetrahedra that are covered by the existing surface.

The reparameterisation scheme developed for this work involves the monitoring of the size of the faces of the mesh that represents the surface, and adding new points to the mesh when their sizes exceed a threshold. The main novel contribution of this technique is that it is possible to link the threshold directly to local curvature information and thus reparameterise the surface to a greater level of detail at higher areas of local curvature, thus ensuring better segmentation of objects with complex boundaries.

4.5.1. Local mesh refinement

A 2D Delaunay triangulation, illustrated in Figure 4.4, has the property that the circumcircle of every triangle in a 2D mesh does not contain any other points of the triangulation (Lawson, 1972).

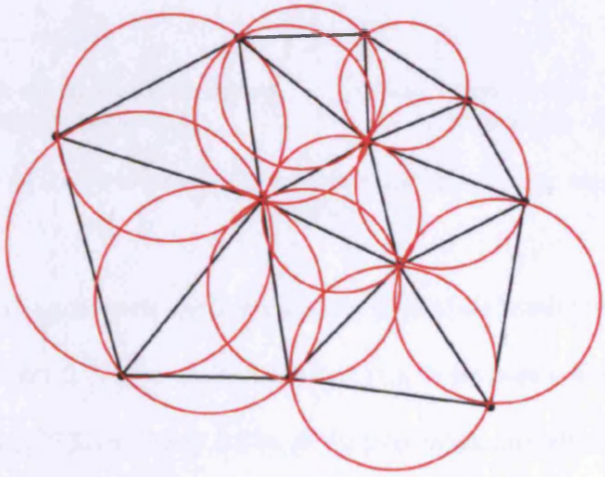
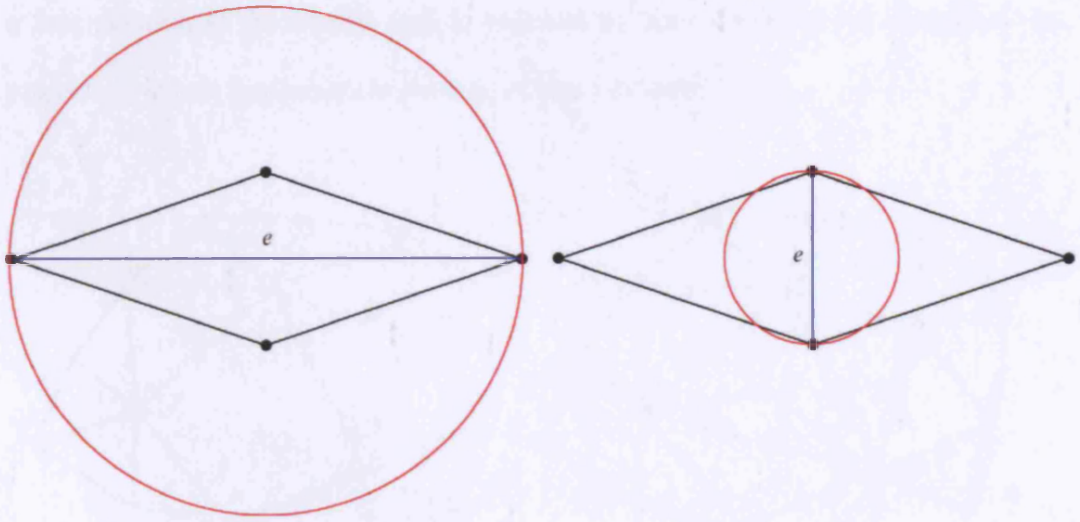


Figure 4.4: *Delaunay triangulation. The circumcircle (red) of every triangle (black) does not contain any other points of the mesh.*

The *quality* of a mesh, or how regular the shape of a mesh's constituent faces are, is a judgement on the shape and structure of a 3D mesh's maximum triangle area and minimum angle between edges. A mesh can be described as 'good quality' if the areas of its triangles and angles between its edges fall between specific, user-specified values. Usually these values specify that a triangle should not have any angle smaller than 30°,

and that the area of the triangles should be as close to a user defined value as possible. One of the most valuable properties of using the Delaunay criterion is that it can guarantees mesh quality (Paloc, 2003). To achieve Delaunay triangulation in a mesh, Lawson's (1972) flip algorithm, which is illustrated in Figure 4.5, can be used.



(a) The circumcircle of edge *e* encloses vertices other than those part of the edge

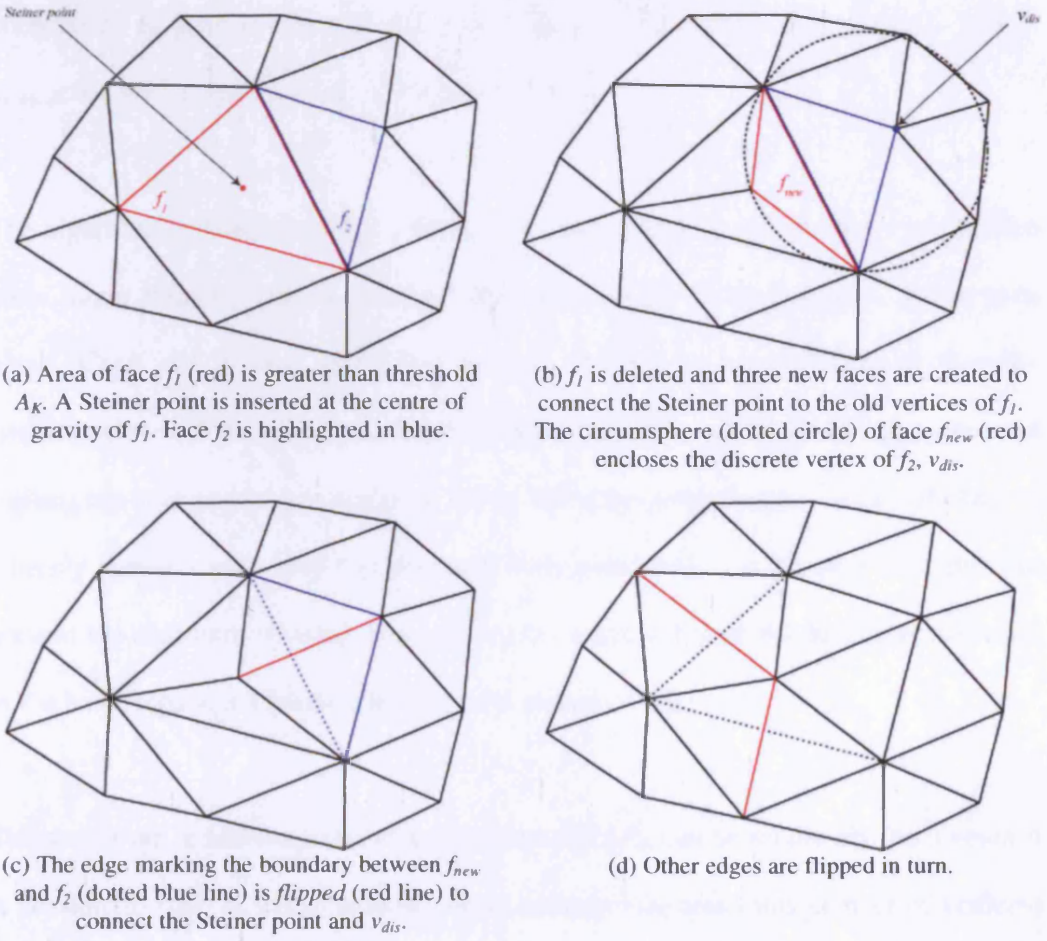
(b) *e* is 'flipped' and its circumcircle no longer encloses any other vertices.

Figure 4.5: *Edge flips. Two triangulations of the same shape, after flipping edge e becomes locally Delaunay*

Several procedures to guarantee mesh quality of 3D surface mesh have been developed, as reviewed in Chapter 2. The work presented in this thesis uses a similar method to that proposed by Chew (1993) to extend the edge flip algorithm directly into 3D. The benefit of using this technique is that it inserts new points into the mesh (known as *Steiner* points) where faces have areas that are greater than a certain threshold; this ensures that the distance between each point in the mesh will never exceed a certain value and thus problems of aliasing and force balancing outlined above are eliminated.

The algorithm, illustrated in Figure 4.6, works as follows. An area threshold A_K is set as the maximum allowed area for any face of the surface mesh. The area for each face is

trivially calculated due to the existence of the separate list of faces, as detailed in section 4.2.2 above. For each face that has an area greater than A_K, a Steiner point is inserted and connected at the location of the centre of gravity of that face, creating three new faces in place of the original. Once all faces have been tested for area, an edge-flipping algorithm is run, whereby an edge is flipped if the location of the discrete vertex, v_{dis}, of a face adjacent to the control face is enclosed by the control face's circumcircle *as projected onto the surface of the mesh* (see Figure 4.6(b)).



(a) Area of face f_1 (red) is greater than threshold A_K. A Steiner point is inserted at the centre of gravity of f_1. Face f_2 is highlighted in blue.

(b) f_1 is deleted and three new faces are created to connect the Steiner point to the old vertices of f_1. The circumsphere (dotted circle) of face f_{new} (red) encloses the discrete vertex of f_2, v_{dis}.

(c) The edge marking the boundary between f_{new} and f_2 (dotted blue line) is *flipped* (red line) to connect the Steiner point and v_{dis}.

(d) Other edges are flipped in turn.

Figure 4.6: *New point insertion and edge flipping.*

This last point is a significant theoretical stumbling block to any such surface meshing algorithm, as the projection and warping of the circumcircle of a planar face onto the (likely different) planes of several surrounding faces is not trivial.

The method used to achieve this goal in this work is to find the smallest circumsphere who's boundary intersects all three vertices of a given face, and use the intersection of the circumsphere with the surrounding faces of the surface to mark the 'boundary' of the circumcircle used in the algorithm illustrated in Figure 4.6. Note that for this calculation, all that is required is to find the coordinates of the circumcentre of the control face, and calculate the Euclidean distance, d_{vert} from this point to any of the three vertices of the control face. To test whether a given point, p, is within the boundary of the circumsphere, all that is needed is to measure the distance from p to the coordinates of the face circumcentre and compare this distance, d_p, with d_{vert}. If $d_p < d_{vert}$, p lies within the boundary of the circumsphere.

The algorithm iterates repetitively through the faces of the mesh until there are no face areas larger than A_K, and no circumcircles that contain discrete vertices. It has been noted (Chen and Bishop, 1997) that the use of smallest circumspheres to map the circumcircles and the use of an iterative procedure may result in the algorithm not converging in highly curved surfaces. While this is theoretically true – a pair of edges in a highly curved region may flip back and forth continually - in practice it is trivial to prevent the algorithm from oscillating in such a manner. Figure 4.7 illustrates the result of the basic reparameterisation algorithm on a simple shape.

This technique is advantageous in that the threshold A_K can be set *locally*. As a result it is possible to have an irregular mesh where average face area (thus number of vertices) in some regions of the mesh is higher than in others. For segmentation purposes this can be used to *increase accuracy* at areas of higher curvature, as illustrated in Figure 4.8. This concept, which is novel in its application to active surface models, is of great benefit to 3D segmentation, as it allows the computing resources required to calculate

the movement of the vertices of the mesh to be concentrated in regions where a high level of detail is required. Furthermore, the greater distance between vertices in areas of lower curvature inherently increases stiffness and therefore reduces the probability of a surface 'leaking' through weak edges.

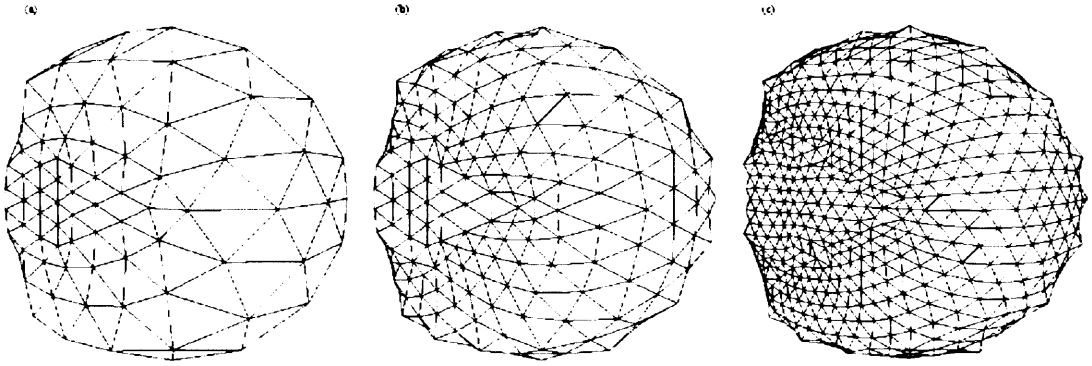


Figure 4.7: *The reparameterisation algorithm. Figure (a) shows an unreparameterised 3D surface mesh. Figure (b) shows the same mesh reparameterised with A_K =250 voxels. Figure (c) shows reparameterisation with A_K =100 voxels.*

The algorithm used to calculate the curvature is the surface triangulation algorithm presented by Stokely and Wu (1992), which is also used to calculate the curvature to control the bending energy (see section 4.2.3). Section 4.6 and Chapter 5 show examples of how this procedure improves segmentation accuracy, while also increasing computing efficiency.

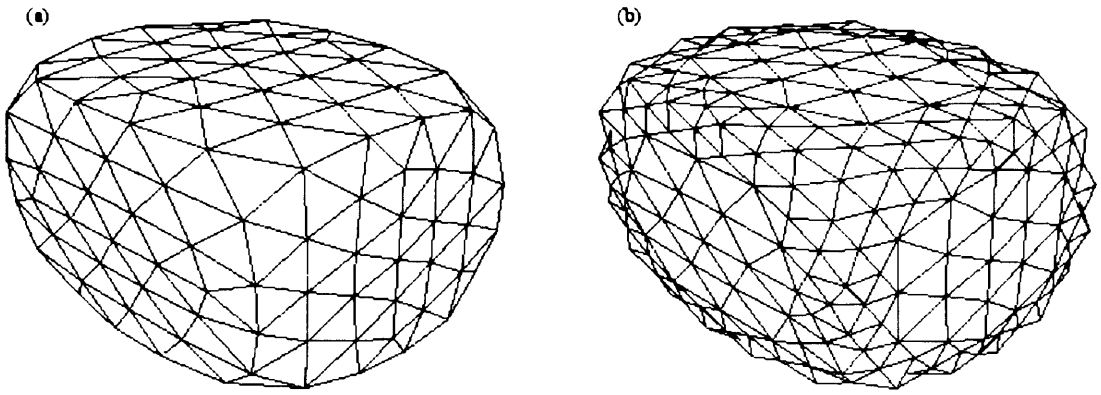


Figure 4.8: *Curvature at higher resolution. The ability of a sample mesh (a) to reparameterise to a higher resolution at areas of greater curvature (b).*

4.6. Collision detection

As the surface expands and moulds itself to the boundaries of the object that it is segmenting, there exists a strong possibility that the surface will intersect with itself. This may occur naturally if the object to be segmented has a complex topology, or otherwise if noise within the bulk of the object 'catches' the vertices of the surface, may cause the surface to fold in upon itself. As a result, a procedure must be set in place to detect any surface collisions, and deal with them in an effective manner.

The detection of surface collision can be separated into two distinct challenges. The first is to actually detect any intersection between the primitives (the faces, edges, and vertices) of the surface, and the second is to abstract this technique to avoid attempting to detect intersections between primitives that are so far apart that it would be impossible for them to intersect at a given time of testing. It should be noted that the techniques used for collision detection are well-established and are widely used in several polygon-based graphics applications, both in the academic and commercial domains. The details of their implementation in this work are presented here for clarity.

4.6.1. Primitive intersection

Ericson (2005) presents a comprehensive discussion of the standard methods of detecting the intersection of primitives. The structure of the surface described in section 4.2 consists of a set of vertices organised by triangular faces, and thus a triangle-triangle intersection test was considered to be suitable to use in this work.

The most straightforward method of detecting triangle-triangle intersection is based on the fact that, in general, two triangles are intersecting when either two edges of one triangle pierce the interior of the other, or one edge from each triangle pierces the

interior of the other triangle. By testing the edges of each triangle with the face of the other (i.e. testing whether a line (representing an edge) intersects the plane of a triangle at a point located within the region enclosed by its constituent vertices) it is possible to deduce whether the triangles are intersecting. If all six possible edge-triangle tests fail, there is no intersection.

A much faster method of intersection-testing, known as the *interval overlap method* was developed by Möller (1997). The technique finds, where it exists, a line, *L*, that represents the intersection of the two planes of the triangles. It then proceeds to calculate the scalar intersection intervals between each triangle and *L*, if the intervals for both triangles overlap, then the triangles are intersecting. Figure 4.9 illustrates the method, which is used in this work to detect self-intersection of the surface at primitive level.

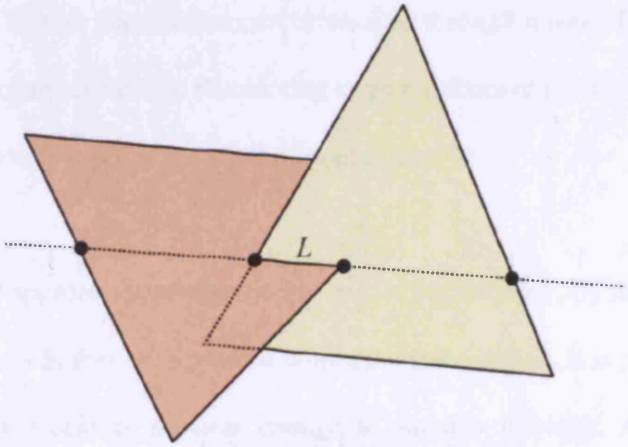


Figure 4.9: *The interval overlap method (adapted from Ericson, 2005)*

4.6.2. Abstraction

The Möller fast triangle-triangle intersection algorithm is one of the fastest tests for the detection of collisions of surfaces that consist of triangular primitives. However, depending on the resolution and accuracy of the required segmentation, surfaces may exceed 3000 separate faces, necessitating over 4.5 million face-pair tests per movement

iteration. This creates a significant time penalty for any surface movement, much of which is wasted as it is unnecessary to test a pair of faces if they are separated by a significant Euclidean distance. Fortunately, several methods exist to abstract the collision detection procedure and minimise the number of primitive tests, only testing where there is a reasonable possibility of intersection.

There are two broad approaches to the abstraction of primitive collision detection, although they are not mutually exclusive. The first is to use *bounding volumes* – a single simple volume used to encapsulate objects of a more complex nature. The intersection tests for simple volumes, such as boxes or spheres, are easy and fast to compute, thus crucially the *non-intersection* of two large (and possibly very complex) areas of primitives can be calculated in a fraction of the time that it would take to achieve the same result using pair-wise primitive tests alone. Bounding volumes can be organised hierarchically, so that an intersection can be tracked through a *tree* of several bounding volumes of increasing resolution, eliminating large numbers of primitives, before finally reaching the stage where primitive testing is necessary.

The second broad approach to abstraction is *spatial partitioning*. By dividing space into regions and testing whether objects exist within the same region, it is possible to rapidly deduce whether two objects are near enough to possibly intersect. As with bounding volumes, *trees* can be used to hierarchically partition space into regions of increasing resolution, drastically reducing the number of pair-wise primitive tests. Note that is perfectly possible, and indeed highly beneficial, to combine bounding volumes with spatial partitioning to create very powerful collision detection algorithms, and such systems are used in highly sophisticated commercial 3D graphics applications.

A key factor in deciding a suitable method to abstract the collision detection of the surface described in this thesis was that, in the majority of cases, the surface exists as a single, closed, object; thus the only collisions that occur would arise due to *self-intersection*. For this reason, the use of bounding volumes was not considered suitable as it would require complex and static methods to define which faces of the surface belonged to which bounding volume.

By contrast, partitioning the space in which the surface sits can be considered an ideal method of abstracting collision detection in this specific case, as it avoids artificially partitioning the surface itself. By classifying the location of the primitives of the surface using a hierarchical tree structure it is possible to rapidly eliminate large areas of primitives that will not be intersecting. The archetypal tree-based spatial partitioning system is the *octree* (Ericson, 2005), an axis-aligned hierarchical partitioning of the volume of a 3D world space. Each parent node of the tree is associated with a finite volume of space, and each is subdivided into eight equal child nodes, created by simultaneously dividing the volume of the parent node in half along the x, y, and z axes. Figure 4.10 illustrates the spatial partitioning of the octree; in the figure only the nodes encompassing the prominent corner are shown.

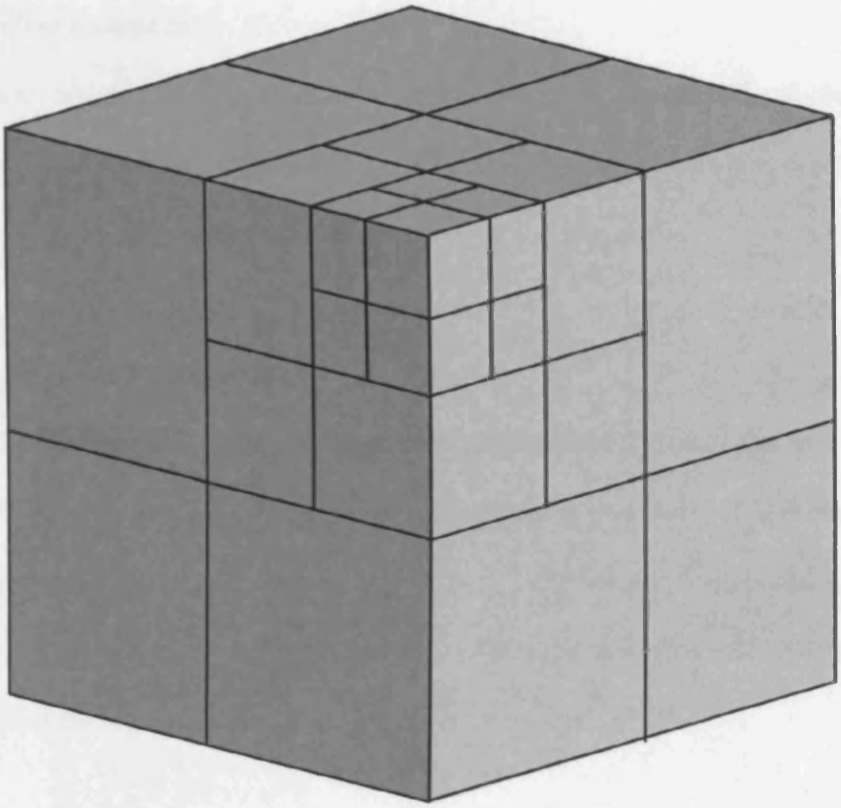


Figure 4.10: *Octree spatial partitioning*

It is the octree system which is used in the work presented in this thesis. The tree is constructed at the same time as the initialisation of the surface, and each face object is added to the octree hierarchy. As the vertices of the surface move, the locations of the faces within the octree are updated, and during reparameterisation faces are inserted or deleted as appropriate. After each movement iteration of the surface, a collision detection algorithm is run that uses the octree to ensure that only faces that are located within the same volume at the bottom node of the octree are tested using the triangle-triangle intersection test described in section 4.5.1.

4.6.3. Collision response

The surface responds to detected collisions by halting movement and collided locking vertices in place. This ensures that there is no possibility of the surface ever intersecting. Further collision response possibilities are discussed in section 6.

4.7. Examples and test datasets

Figure 4.11 demonstrates the segmentation of an artificial 3D liver shape in a binary 3D dataset. The surface is initialised at any point within the bulk of the object to be segmented, and immediately begins expanding. The first four screenshots show the expansion of the surface at different iterations, while the final two screenshots show the final result of the segmentation from two different angles.

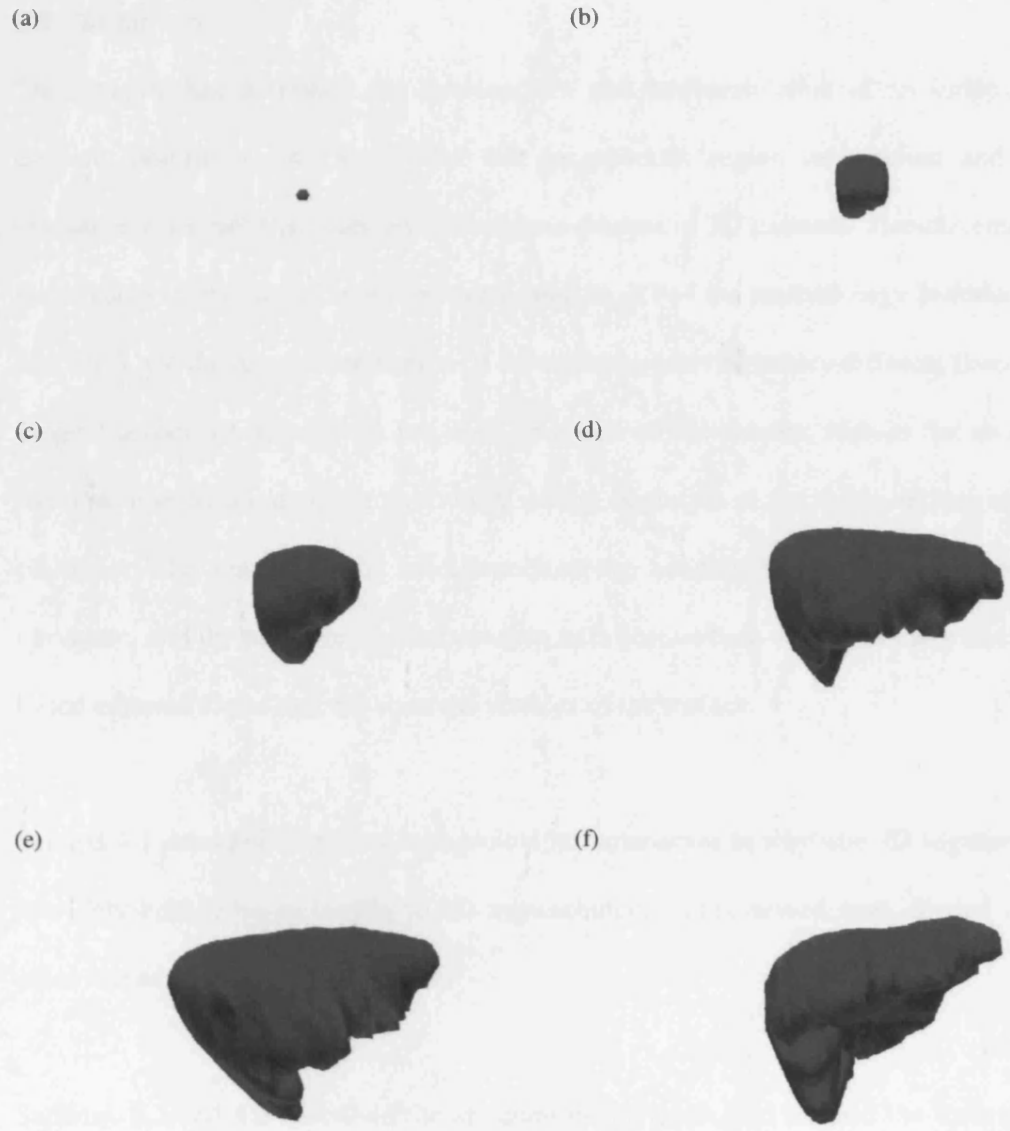


Figure 4.11: Screenshots of the segmentation of an artificial liver shape. Figures (e) and (f) show the final segmentation from two different angles.

4.8. Summary

This chapter has described the development and implementation of an inflationary, discrete, deformable surface model that incorporates region information and local curvature to extract the boundaries of objects present in 3D datasets. The movement of the vertices of the model is a direct extension to 3D of the methodology introduced in chapter 3, yet the reparameterisation of the surface occurs by a very different procedure. Several aspects of novel work has been proposed in this chapter, such as the ability of the reparameterisation algorithm to increase the resolution of the active surface at areas of higher local curvature, the modification of the bending force in response to local curvature, and the use of region information to influence both the inflationary and edge-based external forces that act upon the vertices of the surface.

Section 4.1 provided important background information as to why true 3D segmentation is widely held to be preferable to 2D segmentation, and reviewed work carried out by other researchers in the field.

Sections 4.2 and 4.3 described the structure of the mesh, and defined the forces, both internal and external, that affect the movement of the vertices. Novel work included the use of local curvature to influence the inflationary force, and use of regional information to influence both the inflationary force and the image force derived from significant edges.

Section 4.4 detailed the requirements and procedures for reparameterisation of the surface. As this surface expands the area of the faces of the surface is used to determine whether extra vertices should be added to the data structure. The novel contribution in

this area was to use increase the number of vertices and faces at areas of higher local curvature, thus ensuring more accurate and more efficient segmentation.

Section 4.5 described the work that was required to ensure that any self-intersection of the surface is effectively and efficiently dealt with, and the chapter concludes with Section 4.6 showing some examples of the segmentation of sample 3D shape.

4.9. Novel contributions

- Reparameterising an inflationary active surface to a greater level of detail in areas of high local surface curvature. This ensures that the surface is capable of finding complex boundaries of 3D structures, yet does so in an efficient manner. Furthermore, the reduced resolution in areas of low curvature prevents the surface from leaking through weak edges, and minimises the effect of noise.
- Linking the local curvature of a surface to the internal bending force that affects the movement of the vertices, reducing the probability of the surface leaking yet allowing it to deform correctly where an object's boundary is more complex.
- The combination of the two factors above with the use of region-based image information, which influences both inflationary and edge-based forces, to create a flexible, fast, accurate discrete deformable model that is capable of segmenting complex 3D structures with accuracy.

5. Liver segmentation results

In this chapter, the results of the automatic segmentation of liver from CT datasets are presented, along with the specific procedures used to achieve the results. The relevance of these results is discussed in Chapter 6. 18 healthy liver datasets were used to validate the segmentation procedure, and the final sections of the chapter discuss the segmentation of abnormal liver tissue, and of other abdominal organs.

Section 5.1 introduces the methods and techniques by which the segmentation results are validated to ascertain their accuracy. Section 5.2 briefly discusses dataset pre-processing and ideal contrast agent phase. Sections 5.3 and 5.4 present results from the automatic segmentation of healthy liver tissue, using 2D and 3D techniques respectively. Section 5.5 compares and contrasts the results obtained by the 2D and 3D techniques. Section 5.6 presents the result of preliminary investigation into abnormal liver segmentation, and Section 5.7 discusses the possibilities of using the developed algorithms to segment abdominal structures other than the liver. Sections 5.8 and 5.9 summarise the work and novel contribution of the chapter.

5.1. Validation of segmentation

5.1.1. Gold standard

One of the most important issues facing any automatic segmentation procedure is the validation of its accuracy. Ideally, a variety of phantoms of known dimensions and volume should be scanned, and the segmentation procedure carried out on the resulting data. The results of the segmentation can then be compared with the actual physical measurements of the phantom, which can be considered as the gold standard.

Unfortunately, during the course of this project it was not possible to conduct such a study, nor was it possible to obtain real liver volume data (from transplantation, for example) and thus the only method available to validate both the 2D and 3D segmentation algorithms was to compare the segmented regions with those obtained from manual segmentation.

Using manual segmentation as the sole method of validation is not ideal due the differences between the result of segmentation by different human operators. Hermoye *et al.* (2005) conducted a study comparing the actual volume (obtained during the course of transplantation) of 18 livers with the volumes estimated by both manual and semi-automatic segmentation. The differences between the actual volume and the estimated volume ranged from -15% to +8% for manual segmentation, and from -15% to +6% for semiautomatic segmentation. The mean difference was greater for manual segmentation than for semiautomatic segmentation. It should be noted also that even volume measurements obtained from transplanted livers are not perfect measurements of *in vivo* liver volume, as the loss of blood and other fluids from a liver during its transplantation will affect its volume.

The likely causes for such inaccuracies in the manual segmentation are varied. Firstly, the nature of images produced by CT scanning forces the operator to segment the dataset as a series of 2D slices, thus minimising any useful information that is present in the third dimension. Secondly, the partial volume effect may cause one operator to exclude an area in a particular slice, which is included by a different operator. Finally, major differences may appear between operators when segmenting the vascular structure of the liver, a problem compounded by the use of intravenous contrast agent during the scan.

Taking this into consideration, the results presented in this thesis make use of manual segmentation carried out independently by two separate clinical or trained medical imaging experts, rather than relying on a single manual segmentation. While a greater number of manual operators would be desirable, using more than one makes some effort to minimise the error inherent in using manual segmentation as a gold standard. This issue of validation via manual segmentation is discussed further in Chapter 6.

5.1.2. Volumetric comparison

By comparing the liver volume obtained by automatic segmentation with the volume of the same liver segmented manually, it is possible to obtain a simple, fast estimate of the accuracy of the automatic segmentation. Volumetric analysis is appropriate as this is one of the most important results of segmentation that is desired by clinicians, as discussed in Chapter 1. The *actual* volume of each liver, as measured in mL, can be calculated from the scaling information stored with each DICOM (http://medical.nema.org) dataset. While this information is clinically useful, it is misleading to use it as judge of algorithm accuracy, as the scaling data for the datasets used to test the algorithms have different scaling values. Both the 2D and 3D algorithms were run on the raw data of each scan, and so the volume as measured in voxels is used to judge segmentation accuracy.

2D and 3D automatic segmentation require different methods of analysis. In 2D the *area* of corresponding image slices are compared; these areas may be combined to obtain a value for the volume of the whole liver, yet it is important to remember that each slice has been treated as an individual instance in two dimensions. In 3D, the entire liver dataset is compared at once, and thus it is appropriate to compare the overall volume directly.

A variety of statistical techniques can be used to ascertain the accuracy of the segmentation, yet it is important to bear in mind each test's suitability when assessing the significance of its result. The goal of the statistical analysis in this work is to find a method to measure how reliable, or how valid the results are from both (2D and 3D) automatic segmentation algorithms, by comparison with manual segmentation. There is little consensus about which statistical methods are best to analyse such data (Uebersax, 2006), and several techniques exist, varying from established methods such as the ANOVA (Hayslett, 1974) to tests of marginal homogeneity, tetrachoric and polychoric correlation, latent trait/class models, and kappa statistics (Uebersax, 2006). For this case, where the task is to assess the *validity* of a test (automatic segmentation) where there is no true gold standard, the accuracy of the measurement is assessed by comparing its results with existing raters (the manual segmentations).

The method chosen as a suitable means of discovering the validity of the segmentation procedures is to use *inter-rater-reliability*, which assesses unsystematic variation due simply to which 'rater' carries out the test (Shrout and Fleiss, 1979). In this case, there are four raters: two manual segmentation operators (see Section 5.1.1 above), the 2D algorithm, and the 3D algorithm. To measure the inter-rater-reliability, the *intraclass correlation coefficient* (*ICC*) is used, as suggested by Gerig, Jomier and Chakos (2001):

$$ICC = \frac{ms_B - ms_W}{ms_B + (n-1)ms_W} \tag{5.1}$$

where ms_B is the mean squares between groups, and ms_W is the mean squares within groups, both developed during an ANOVA. The *ICC* assesses rater reliability by comparing the variability of different ratings of the same subject to the total variation across all ratings and all subjects. It will approach 1.0 when there is no variance within

target variables (*i.e.* each individual target is given the same value by all raters), indicating that total variation in measurements is due solely to the natural variation of the target variable. Applied to this situation, the *ICC* will be high when the respective liver volumes (the target variable) tend to have the same value estimated by each of the four segmenting operators. This indicates that the total variation in the measure of the volume depends *solely on the natural variability in liver volumes across a population*, thus there is perfect intra-rater reliability.

An alternative to the ICC for this situation is to calculate the Pearson correlation between all pairs of rater (Uebersax, 2006). The Pearson correlation measures association between raters, but is insensitive to rater mean differences (bias). Yet the ICC decreases in response to both lower correlation between raters and larger rater mean differences, and thus it is used in preference.

Along with the *ICC*, an ANOVA test is also carried out of the volume data. The ANOVA measures the difference between the means of two or more groups and, unlike the *ICC*, makes no attempt to compare the variability of different ratings of the same subject to the total variation across all ratings and all subjects. Nonetheless the F values developed by an ANOVA for each group of volumes are presented for completeness.

Figure 5.1 shows a Bland-Altman plot comparing the two manual segmentations. The Bland-Altman plot is a statistical method to compare two measurement techniques, the difference between them are plotted against their averages. Horizontal lines are drawn at the mean difference and at mean difference ±1.96 standard deviations. Differences plotted within the ±1.96SD boundary suggest that the two methods may be used interchangeably. The Bland-Altman plot is useful to check for systematic bias and to

identify outliers. Figure 5.1 shows that, while all but one of the datasets lie within the ±1.96SD boundary, there is systematic bias between the observers; the fact that the mean line of the graph does not go through zero (its value is -21675.2) shows that one observer consistently gives lower values than the other. Figure 5.9 and 5.18 below show the Bland-Altman plots for the 2D and 3D segmentation results respectively, and it is possible to see the difference in the magnitude of the bias between the two manual observers and the automatic segmentation techniques.

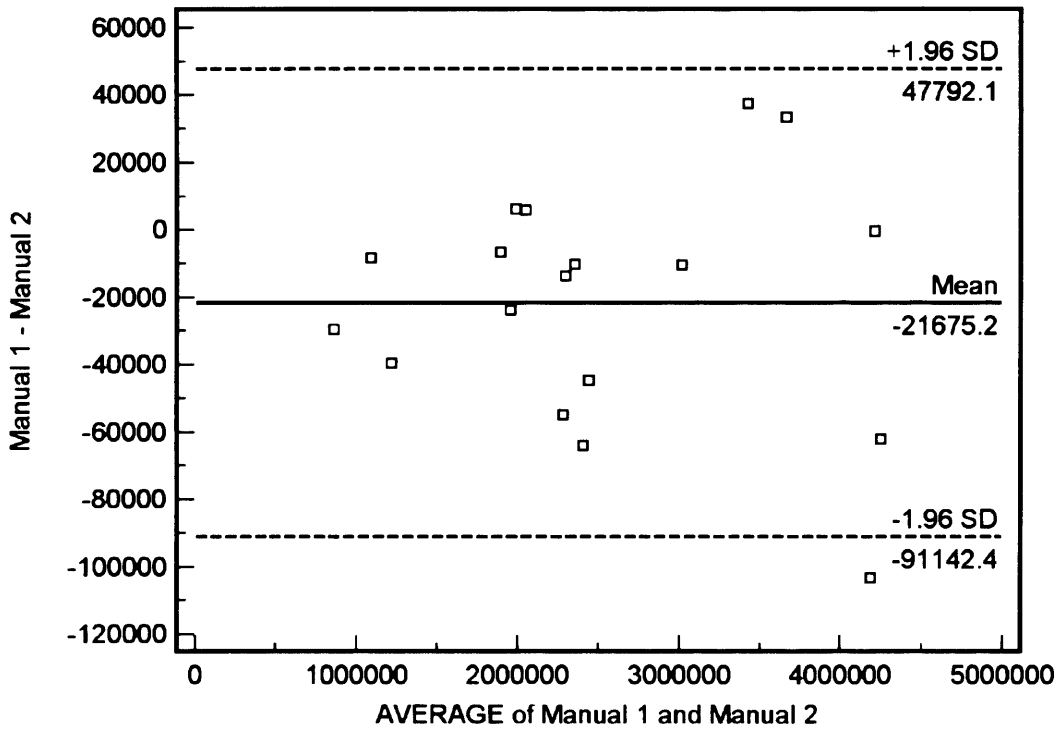


Figure 5.1: Bland-Altman plot comparing the manually segmented liver datasets.

5.1.3. Volumetric overlap

The simple comparison of area or volume described above, while providing a good overview, is unacceptable as a sole measure of difference between segmentations. It is possible to obtain a false positive volume result if the automatic segmentation has over-segmented in one region of the liver, yet under-segmented in another. Measurement of the *overlap* of the segmented volumes *is* affected by such an over/under-segmentation, and as such is an excellent measure of segmentation accuracy. Datasets can be analysed

voxel by voxel to calculate false positive, false negative, true positive and true negative voxels. Gerig, Jomier and Chakos (2001) suggest two methods for estimating the overlap, the intersection (V_I) of the volume of the automatic segmentation (A) and manual segmentation (M) divided by the manual segmentation,

$$V_I^1 = (A \cap M) / M \tag{5.2}$$

and the intersection divided by the union,

$$V_I^2 = (A \cap M) / (A \cup M) \tag{5.3}$$

Both measures give scores of 1 for perfect alignment and 0 for complete disagreement. Equation (5.2) penalises under-segmentation, but over-segmentation is effectively ignored; if A is greater than M (i.e. if over-segmentation has occurred), $(A \cap M)$ cannot be less than M alone. Equation (5.3) penalises both under-segmentation and over-segmentation, but is more sensitive to any differences since both denominator and numerator change with increasing or decreasing overlap. In practice, neither equation is ideal; a high score using equation (5.2) is effectively meaningless (as it is unaffected by oversegmentation), and while a high score using equation (5.3) signifies an excellent match, poorer matches may achieve disproportionately lower scores as any mismatch is doubly penalised.

Lamecker, Lange and Seebaβ (2002) define the volumetric overlap in a manner that takes into account more information than equation (5.2), yet does not penalise errors so harshly as equation (5.3). Here, their definition is modified to fit in with the scoring paradigm used for equations (5.2) and (5.3), to give a third measure (V_O) of volume

overlap. It is this equation that is used to score the volumetric overlap for all the results obtained in this work.

$$V_O = \frac{A \cap M}{(A+M)/2} \tag{5.4}$$

In the 2D case, comparing segmented areas is trivial as there is a direct correspondence with slices that have been manually segmented in two dimensions. In 3D, comparison is more complicated because the liver is not directly represented by a set of points lying on the same regular planes as the manually-segmented slices. As a result, it is necessary to interpolate between the vertices of the surface to obtain the boundary location for each slice in the z-axis (*i.e.* the x-y plane, the usual plane of manual segmentation). Due to the large number of vertices, the fact that the distances between vertices are rarely large, and the fact that the inter-vertex space is already interpolated in a linear fashion of sorts due to the triangular organisation of the surface structure, it was decided that straightforward linear interpolation would be the most appropriate approach. The technique used to obtain the boundary is a scan conversion on each slice in the z-axis. The location at which the scan lines intersect a face of the surface is marked as a boundary, resulting in a binary image containing the boundary of the surface in that plane. Thus, a set of binary images is created with direct correspondence to the manual segmentation images.

Due to the nature of the results, the overlap between any automatic segmentation must be measured separately against the two manual segmentations. Yet it is important to note that the two manual segmentations do not overlap perfectly, as Table 5.1 shows. The table shows the result of applying equation (5.4) to the two sets of manually segmented datasets.

Dataset	V_O
1	0.98
2	0.97
3	0.97
4	0.98
5	0.97
6	0.98
7	0.96
8	0.98
9	0.97
10	0.97
11	0.96
12	0.97
13	0.98
14	0.98
15	0.97
16	0.98
17	0.98
18	0.97
Mean	**0.97**
SD	0.006

Table 5.1: *Overlap between the manual segmentations*

As expected, the results indicate a very close alignment between each dataset, yet it is important to note that the mean value of 0.97 indicates that there is a 3% mean difference in the overlap measures between the manual segmentations, and this must be taken into consideration when analysing the results of the automatic segmentation.

5.1.4. Boundary distance measures

A further method of comparing segmentation results is direct measurement of the distance between boundaries. The distance in this case can be defined as the number of voxels between any given point on one surface and the nearest boundary region of a second surface (see further discussion below). This distance is relatively easy to measure, as both the 2D active contour and 3D active surface consist of a set of vertices at defined points in space, thus it is trivial to measure the Euclidean distance to the nearest area of 'liver' labelled as such in a manual segmentation of the dataset. Nonetheless, boundary distance measurement should be treated with caution if it is to be

used to judge segmentation accuracy. The reasons for this are two-fold; firstly, in this situation there exists no correspondence between measurement points – the measurement records merely the location of the *nearest point* of the equivalent surface, thus false positive measurements can be common. Secondly, the two diagrams in Figure 5.2 show how it is possible for distance measurements to incorrectly measure the accuracy of the segmentation; for example, a volumetric overlap test would rate Figure 5.2(b) as a more accurate segmentation than Figure 5.2(a), yet a distance measurement using the vertices of the automatic segmentation might rate them as similar in standard.

As a result, when measuring contour distances it is necessary to consider the reverse case *i.e.* measure the distance from the manual contour to the automatic contour. As the manual contours are not described by sets of vertices (in the same manner as the automatic contours), a scan grid with a resolution of 10 pixels is used to obtain a set of points on each (2D) manually segmented boundary. The distance between these points and the nearest voxel of the equivalent automatically segmented volume is used as a distance measure.

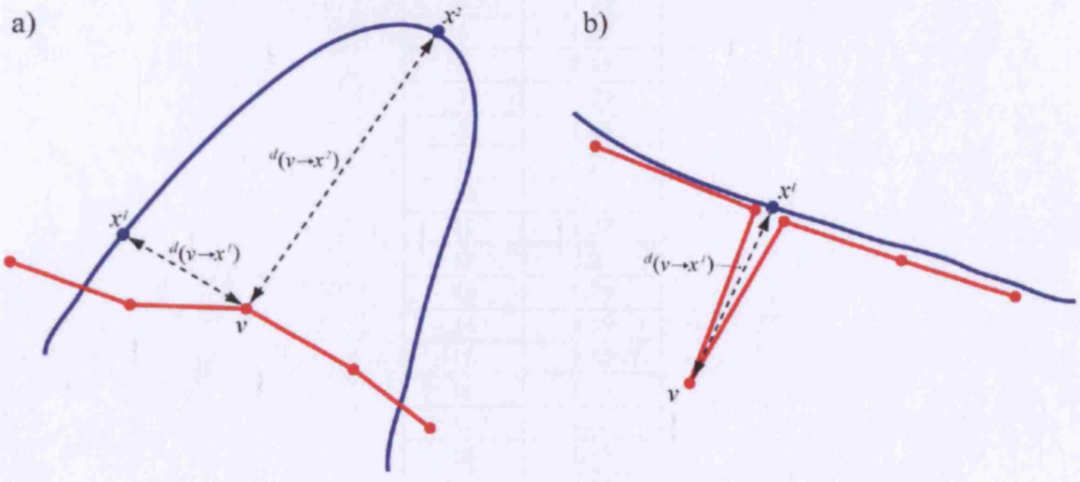


Figure 5.2: *How surface distance can give misleading results. In both figures, the blue continuous line represents the boundary of a manual segmentation, the red line represents the boundary of an automatic segmentation. In figure (a), the nearest vertex to point* x^2 *is vertex* v. *Yet the nearest point on the manual boundary to* v *is point* x^1*; thus the blue line could represent an incorrect segmentation, yet this will not be accurately reflected in any measure of distance. The reverse case exists in Figure (b), where vertex* v *is some distance from the manual boundary and thus any measurement of distance will be high, and may not truly represent the accuracy of the segmentation.*

It is also necessary to consider the differences between 2D and 3D; in 2D only a single image slice can be used to obtain distance information (as the segmentations for each slice occurred as separate instances), yet in 3D the data from surrounding slices can be used as each segmentation instance considers the full 3D dataset at once.

The *actual* distances between contours (*i.e.* measured in millimetres) can be calculated using the scaling data stored in the DICOM file format at the time the scan was carried out. These distances are unique for each scan, though they usually lie between 0.5mm and 0.8mm per voxel for the x and y dimensions, and varying between 1 and 5 voxels in the z dimension. These scaled distances between the contours for each dataset may be of clinical interest, yet the voxel-distance (d) is considered the more relevant value to measure the accuracy of the algorithm as it is independent of individual scan scaling values.

Dataset	d
1	1.18
2	1.28
3	1.16
4	1.25
5	1.27
6	1.15
7	1.29
8	1.26
9	1.20
10	1.31
11	1.32
12	1.25
13	1.29
14	1.19
15	1.15
16	1.13
17	1.23
18	1.19
Mean	**1.22**
SD	0.060

Table 5.2: *Distances between the manually segmented datasets*

In a similar manner to the volumetric overlap comparison method, it is important to recall that there are likely to be differences between manually segmented contours. Table 5.2 demonstrates this difference as measured in boundary distance. The mean distance between the contours is 1.22 pixels.

5.2. Data source, pre-processing and contrast phase

Data from CT scans is stored in the DICOM file format. Prior to its use with any of the procedures detailed within this report, the OSIRIS medical image processing software (http://www.sim.hcuge.ch/uin) was used to window the appearance of the each image slice, which was then directly saved as 8-bit uncompressed greyscale image data. This pre-processing was useful, both to facilitate the handling of the data, and to maximise the differences in appearance of soft tissue while eliminating surplus data at the higher and lower end of the Hounsfield scale. The windowing of the data was identical for each dataset and was set to the standard CT Abdomen window of the OSIRIS software (a freeware DICOM/Papyrus viewer developed by the Digital Imaging Unit of the Radiology department, University Hospitals of Geneva http://www.sim.hcuge.ch/uin/). This is a linear window between 30-300 Hounsfield Units.

Table 5.3 shows the metadata for all the CT datasets used in this work. It can be seen from the table that a variety of different datasets were used, scanned on different machines, using different reconstruction software, and with different voxel dimensions. The x, y and z dimensions are given in mm, kVp is the voltage measured in volts and the tube current is measured in milliamps. The acquisition time stated in the DICOM header files, for all scans of all datasets and regardless of contrast phase, was 500ms.

Dataset	x	y	z	kVp	X-Ray Tube Current	Make	Model	Software
1	0.63	0.63	1	120	300	SIEMENS	VOLUME ZOOM	VA20Q
2	0.50	0.50	1	120	300	SIEMENS	VOLUME ZOOM	VA20Q
3	0.74	0.74	1.5	120	280	SIEMENS	VOLUME ZOOM	VA40C
4	0.49	0.49	1	120	300	SIEMENS	VOLUME ZOOM	VA40C
5	0.71	0.71	1.5	120	350	SIEMENS	VOLUME ZOOM	VA40C
6	0.72	0.72	1.5	120	500	SIEMENS	VOLUME ZOOM	VA47C
7	0.66	0.66	1.5	120	350	SIEMENS	VOLUME ZOOM	VA47C
8	0.74	0.74	1.5	140	337	SIEMENS	SENSATION 4	VA47C
9	0.75	0.75	1.5	120	367	SIEMENS	SENSATION 5	VA47C
10	0.62	0.62	1	120	350	SIEMENS	VOLUME ZOOM	VA47C
11	0.68	0.68	1.5	120	325	SIEMENS	VOLUME ZOOM	VA47C
12	0.65	0.65	1.5	120	325	SIEMENS	VOLUME ZOOM	VA47C
13	0.61	0.61	1	120	196	SIEMENS	SENSATION 64	SYNGO CT 2006 A
14	0.78	0.78	1	120	350	SIEMENS	VOLUME ZOOM	VA47C
15	0.86	0.86	0.75	120	523	SIEMENS	SENSATION 64	SYNGO CT 2006 A
16	0.65	0.65	3	120	286	SIEMENS	SENSATION 64	SYNGO CT 2006 A
17	0.65	0.65	1.5	120	327	SIEMENS	SENSATION 4	VA47C
18	0.74	0.74	1	120	480	SIEMENS	SENSATION 64	SYNGO CT 2006 A

* see note in text

Table 5.3: Metadata for all CT datasets used in the thesis.

Due to the relatively short total acquisition time of spiral CT, imaging of the liver is possible in different contrast enhancement phases. These multi-phase studies offer clinicians the chance to view the interaction of the liver with its blood supply, and thus aid in the detection of abnormalities. With no contrast enhancement, the liver has the same appearance as much of its surrounding tissue, which creates difficulty for any segmentation algorithm. During the *arterial phase* of contrast enhancement, the heart and blood vessels appear as bright white in the images, while the liver usually remains relatively dark. Again, this is awkward for segmentation purposes, as the bulk of liver remains the same shade as surrounding tissue, yet its blood supply is highlighted. The *venous phase* is most suitable for use with segmentation, as the contrast has perfused into the tissue of the liver, raising the mean pixel value of the liver to a greater value than much of the surrounding tissues.

Figure 5.3 contains two images showing examples of the different stages of contrast, where parenchymal liver tissue in 5.2(a) has a lower average pixel value than parenchymal tissue in 5.2(b).

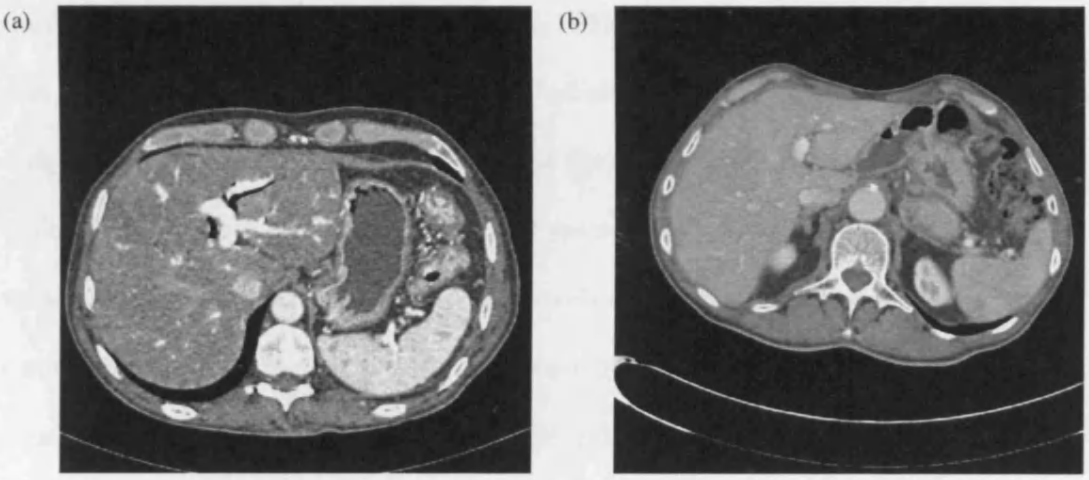


Figure 5.3: *Contrast phases of liver during CT scans. Figure (a) shows arterial phase, Figure (b) shows venous phase.*

5.3. Two dimensions

5.3.1. Initial studies

Prior to using an active contour model, attempts were made to repeat previous work and segment the liver using low level techniques. Thresholding is used for liver segmentation by several authors (Bae *et al.*, 1993; Gao *et al.*, 1996; Hong *et al.* 2001; Seo and Park, 2005), yet the results of initial efforts to repeat such techniques suggested that, as a segmentation tool for the liver, it is not at all robust. Automatic segmentation is possible by manually selecting an area of parenchymal tissue and using some simple measures such as mean pixel value and standard deviation to obtain threshold level estimates. Yet, while there is significant amount of research into automatic thresholding (Sonka and Fitzpatrick, 2000), initial tests revealed that, for the liver, it is difficult to achieve reproducible results even when setting thresholds manually. Thus, after these initial studies, no further research into automatic thresholding assignment was carried out.

5.3.2. Active contour model

The results in Sections 5.3.3 and 5.5 were obtained by segmentation with the active contour model described in detail in Chapter 3. The model was developed to be initialised within the bulk of the liver and expand outwards until the liver's boundary is met. There are two major benefits of using an inflationary model to segment the liver. The first is that the interior of a healthy liver is more uniform than the exterior, thus there are less noisy edges that can trap vertices as they move outwards towards the edge of the liver. The second is that the snake can be initialised at almost any point within the liver, without greatly affecting the final segmentation result. The importance of the these two points should not be underestimated as they completely avoid the major stumbling block of previous liver segmentation algorithms, that of correct initialisation, and thus provide a faster and more robust segmentation.

One major factor that affected the accuracy of the final segmentation was the setting of the parameters. To recap the information presented in Chapter 3, there are 6 parameters that affect the movement of the contour: four of these control the relative strength of the elasticity force, bending force, inflationary force and image force; and two further parameters represent low and high pixel threshold values. For all segmentation instances, the contour was reparameterised after every five movement iterations; this value was again arbitrarily determined with reference to the literature (McInerney and Terzopoulos, 2000).

The two threshold parameters were set via a semi-automatic method. The contour was initialised in a 10x10 pixel area of parenchymal liver tissue chosen by an operator, and the mean and standard deviation of the pixel values in this region were measured. The threshold values T_{lo} and T_{hi} (see equation (3.5)) were set to be 1.5 standard deviations (SD) below and above the mean value respectively. In practice the mean grey level value of the parenchymal liver for each dataset varied between 160 and 200, and the mean SD varied between 10 and 20, thus T_{lo} and T_{hi} varied accordingly.

Chapter 3 discussed and demonstrated the developed active contour model's capability to split and merge, allowing the contour to fold around structures within the bulk of the liver. In healthy liver, there are no interior structures that require segmentation, yet 'noisy' edges within the dataset (frequently caused by contrast enhanced vascular structure) occasionally cause the contour to fold in on itself. For this reason, the 'merge' capability of the contour was enabled when segmenting the liver, but the 'split' function was disabled to prevent incorrect looping and the proliferation of 'noisy' daughter contours.

The criterion used to terminate the algorithm (*i.e.* the final segmentation) is deduced with reference to the reparameterisation algorithm. The primary termination condition is: '*if, at a reparameterisation stage, no further vertices have been added to the contour (thus the contour has not expanded since the last reparameterisation) the algorithm should terminate*'.

The biggest disadvantage to the 2D segmentation technique is its inherent inability to deal with areas of the liver that are completely separate from the main bulk of the organ. Figure 5.4 shows an example of such a situation. As the segmentation algorithm is entirely 2D, a single contour has no way of extending beyond the boundaries of the area that it has segmented. Thus, if separate areas are to be included in the segmentation, it is necessary for multiple contours to be manually initialised within a single slice.

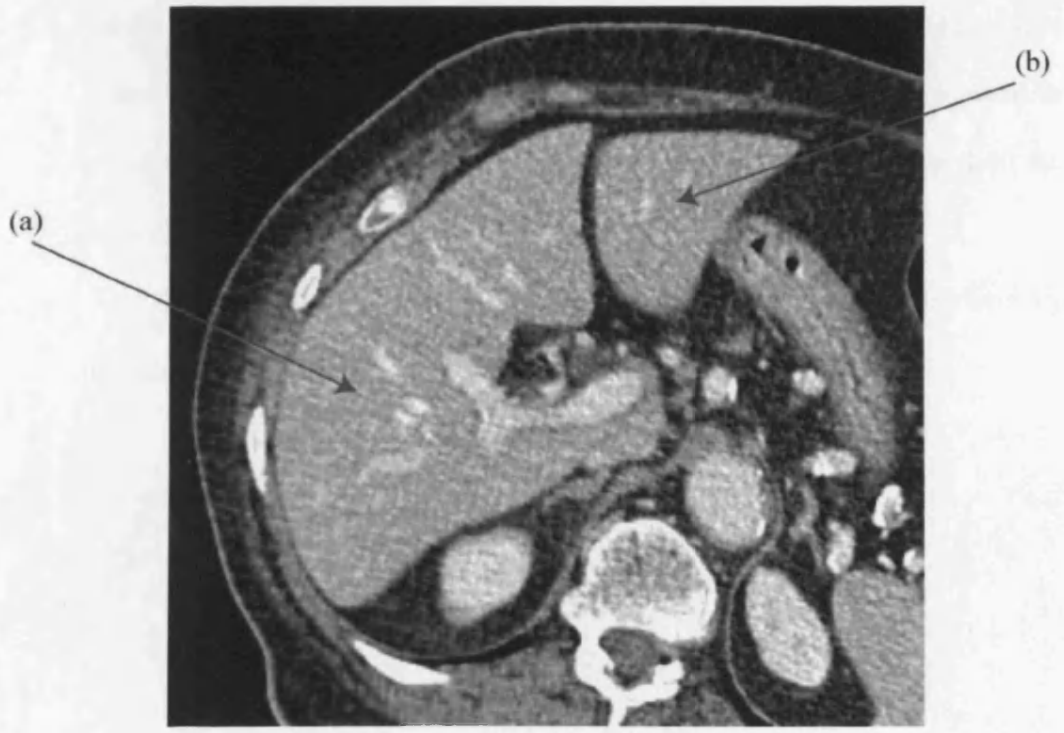


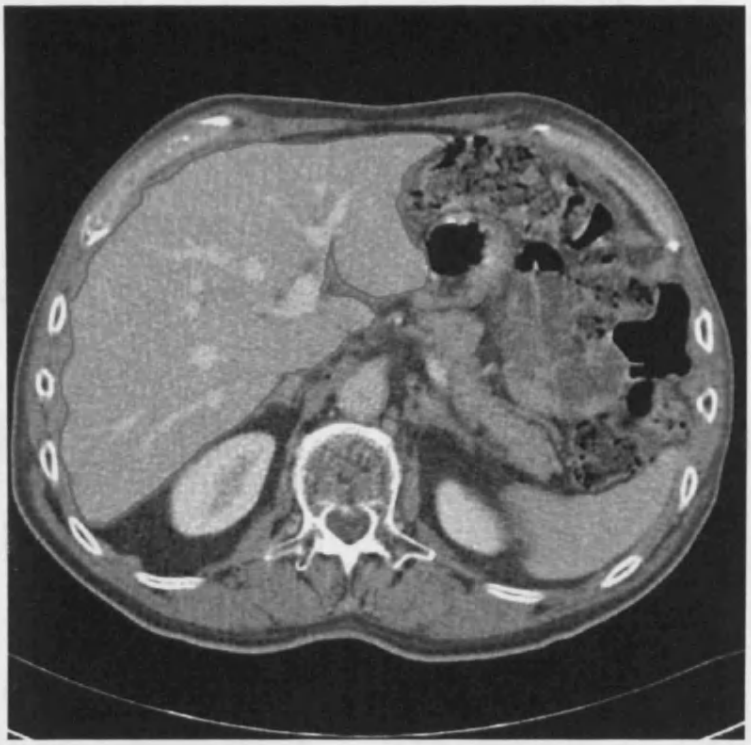
Figure 5.4: *Separate areas of the liver in a single slice. The area indicated by arrow (a) is the main bulk of the liver, yet area (b) is completely separate.*

For the results presented below, the segmentation procedure for an entire liver dataset proceeded as follows:

- Operator selects an image slice from the middle of the dataset.
- Operator selects 10x10 area of parenchymal liver. This area determines threshold values for all the images in the liver dataset, and acts as the initialisation point for the active contour model.
- The segmentation algorithm is started. The contour inflates to segment the selected liver slice. Once the termination condition has been met, the centre of gravity of the area enclosed by the final position of the contour is calculated and stored, and this is used as the seed point for the subsequent slice in the dataset. This process is iterated (the centre of gravity for each segmentation acting as a seed point for the next slice) until the end of the dataset.
- The focus moves to the image slice that precedes the initial slice. The stored centre of gravity from the initial segmentation area is recalled and used as a seed point for the segmentation of the preceding slice. This process is iterated in reverse order up the dataset in a mirrored fashion to the previous step, until the start of the dataset is reached.
- The operator checks the segmented area and begins similar iterations by initialising a contour within the bulk of any separate lobes of the liver.

5.3.3. Two dimensional results

a)



b)

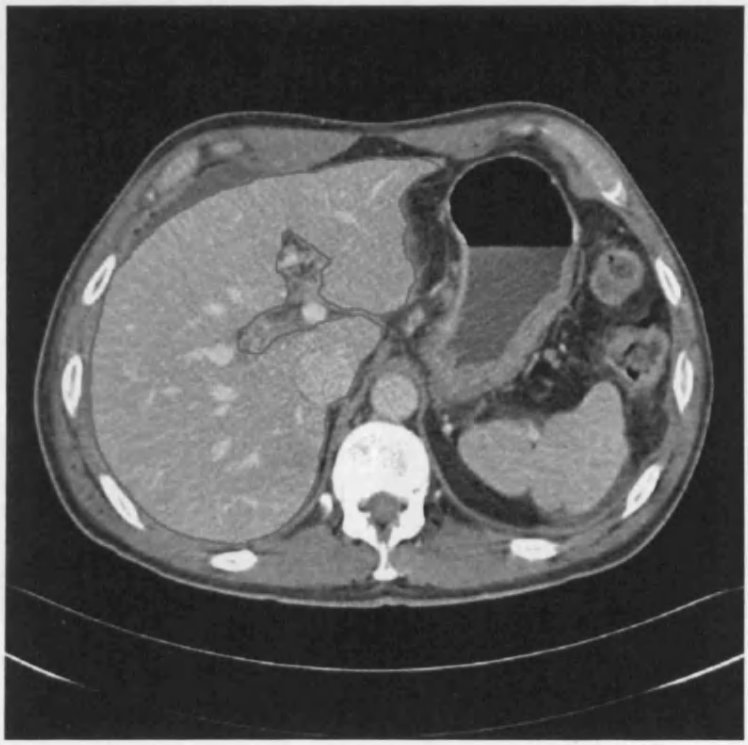


Figure 5.5: *Sample images of automatic 2D segmentation*

Figure 5.5 shows two sample images of results obtained by automatic segmentation of the liver by the 2D active contour model. The images are from two separate datasets, and the contour model has found successfully the boundary of the liver in all areas, except for minor errors in the sharp upper right corner of the liver in the image. Figure 5.6 consists of several images showing the progression of the active contour model, from its initialisation to the final boundary.

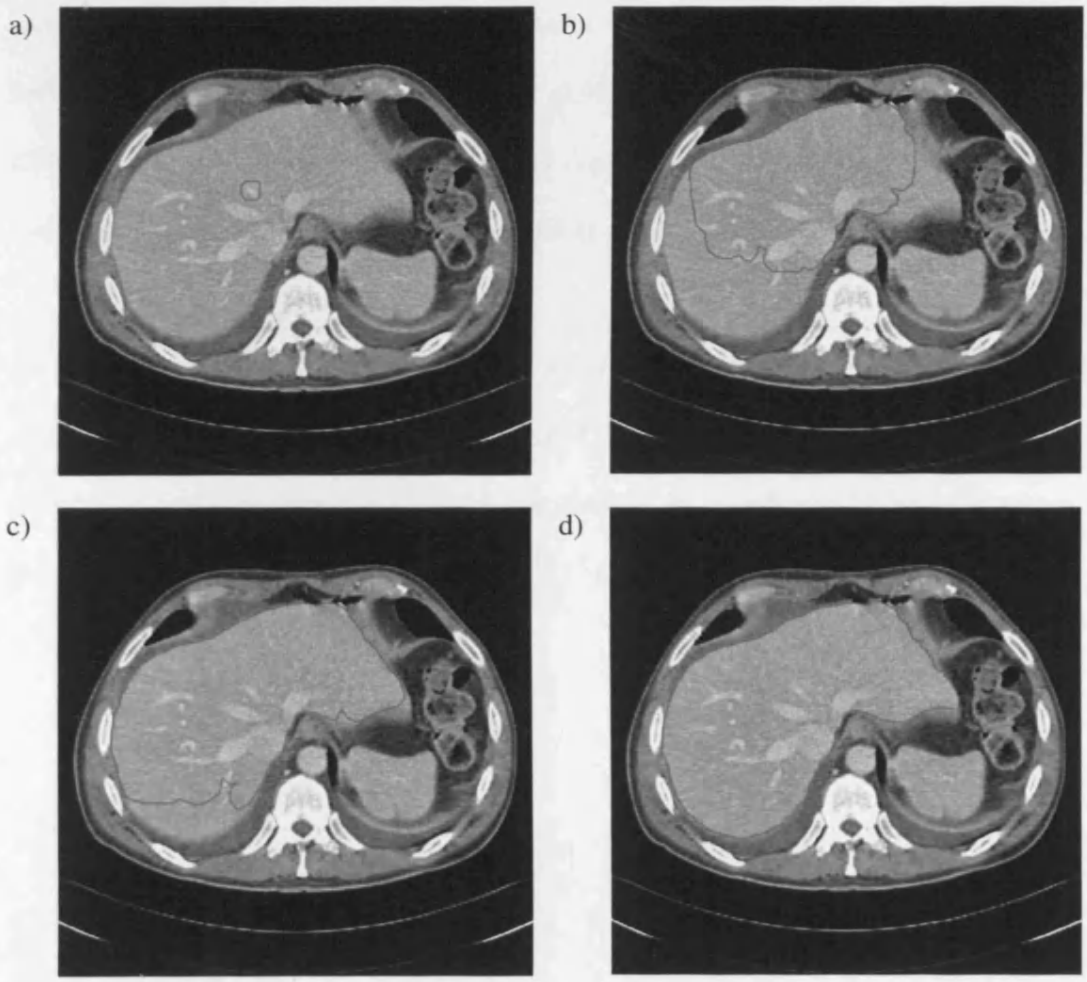


Figure 5.6: *Four images showing the development of the contour during segmentation*

5.3.3.1. Volume estimation

Figure 5.7 is a bar chart that compares the measured volumes of two manual segmentations and the automatic 2D segmentation of 18 liver datasets. Table 5.4 shows the raw data. The units used are voxels, as explained in Section 5.1.2 above.

Using equation (5.1), the relevant intraclass correlation (*ICC*) value calculated for the data presented in Table 5.4 is 0.992 (to three significant figures). As presented in section 5.1.2 above, an ICC value of 1 means there is no difference in the variance between raters, and thus with a value of 0.992 it is possible to conclude that the differing values of the volumes in Table 5.4 depend solely on the natural variability in liver volumes across a population (Gerig, Jomier and Chakos, 2001).

An ANOVA test on the data produces an F value of 0.036 (to three significant figures), which is less than the *F-critical* value of 3.178, as given by the data tables. Thus, it is possible to conclude from the ANOVA that there is no significant difference between the groups.

Dataset	Manual 1	Manual 2	2D
1	3438874	3401530	3692620
2	3676325	3643133	3921864
3	2047119	2041405	2179866
4	4220236	4282390	4220236
5	1942143	1965896	2173264
6	1986030	1979839	2175173
7	2253491	2308419	2582051
8	4211000	4211511	4322497
9	2342679	2353009	2508978
10	3005896	3016371	3041173
11	1088956	1097179	1079586
12	2376157	2440348	2512034
13	4133606	4237067	4094168
14	1197437	1237001	1269585
15	1884221	1891053	1786347
16	848182	877761	876575
17	2282859	2296545	2302583
18	2420211	2465118	2253719

Table 5.4: *Raw data of volumes measured by manual and 2D automatic segmentation. The ICC value applied to this data is 0.992.*

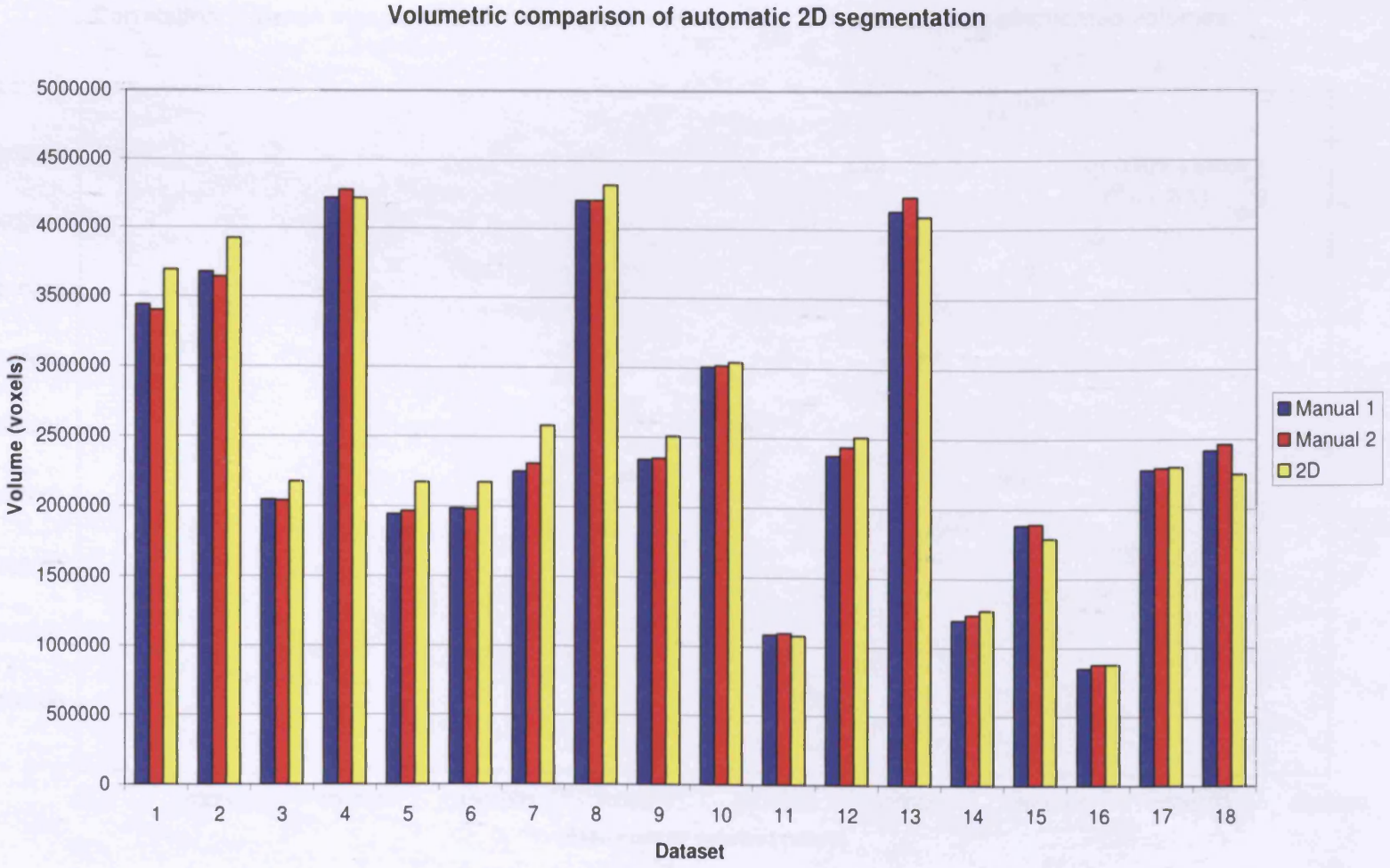


Figure 5.7: Measured volumes of manual and 2D automatic segmentation.

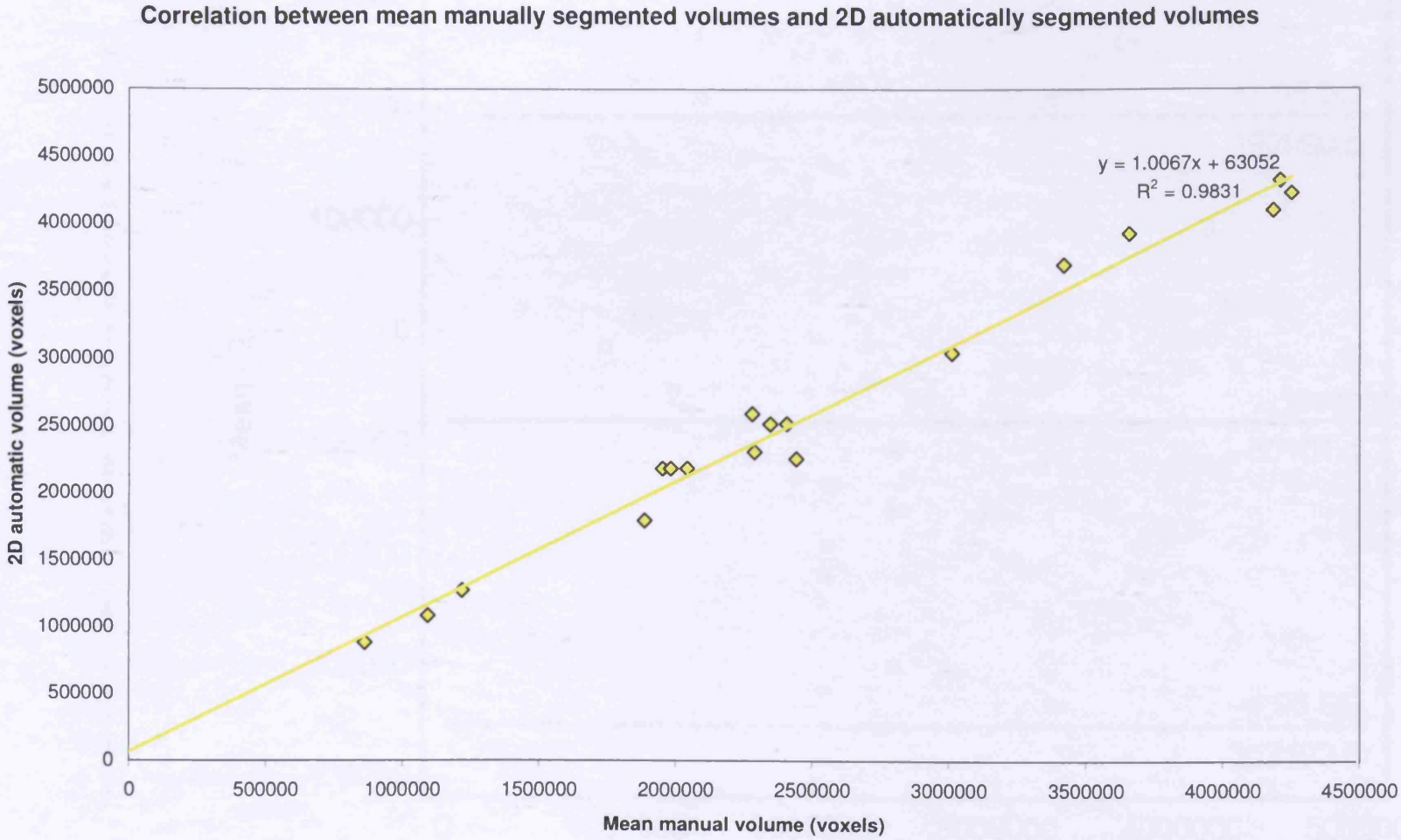


Figure 5.8: Correlation between volumes of manual and 2D automatic segmentations

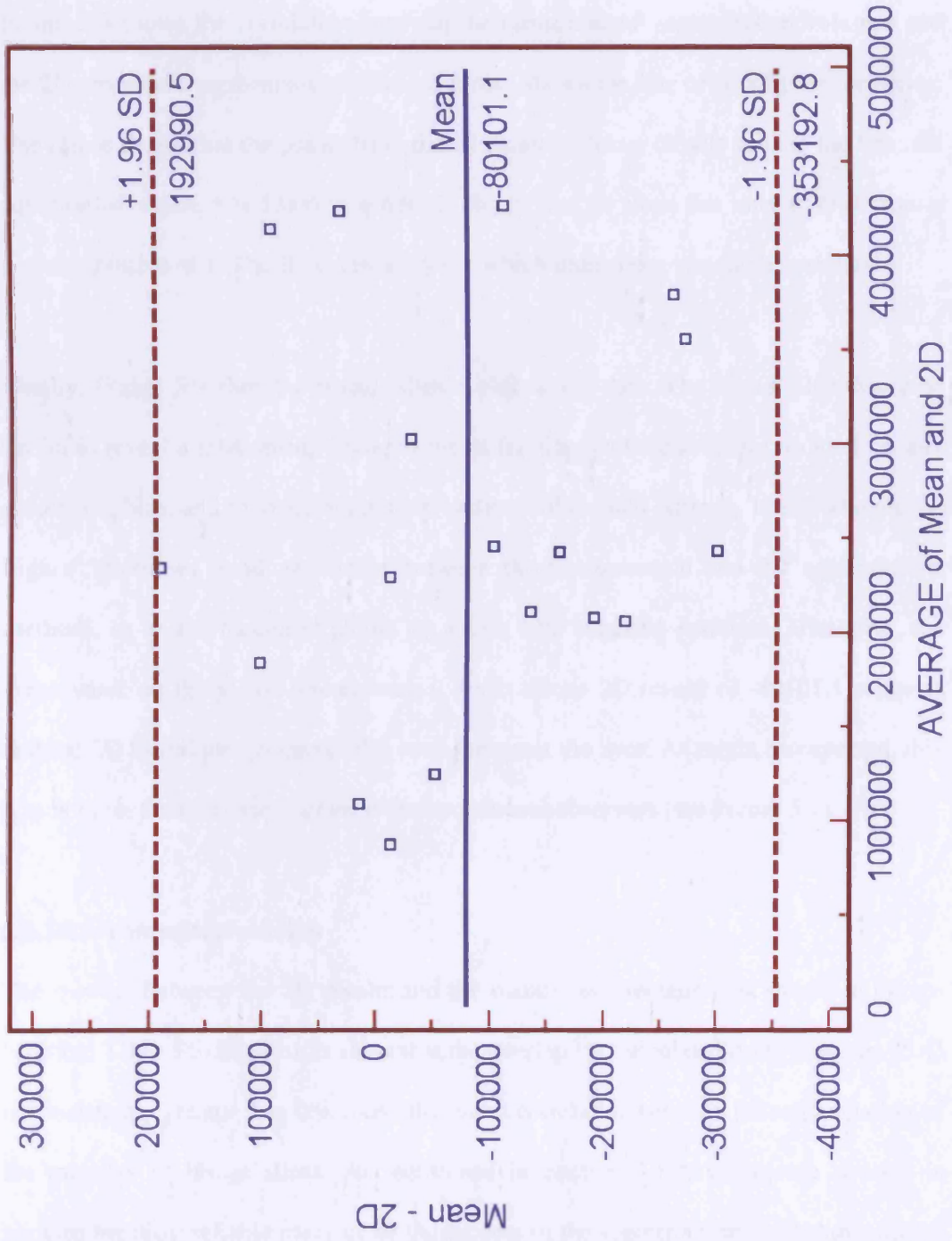


Figure 5.9: *Bland-Altman plot between mean manual and 2D automatic segmentations*

Figure 5.8 shows the correlation between the mean manual segmentation volumes and the 2D automatic segmentation volumes; it also shows the line of best fit for the series. The figure shows that the points from the 18 datasets cluster closely around the line; the equation of which, y = 1.0067x + 63052, shows that its slope lies only 0.0067 from a perfect gradient of 1. The R^2 value is 0.9831 which indicates a very high correlation.

Finally, Figure 5.9 shows a Bland-Altman plot of the data. The Bland Altman plot is useful to reveal a relationship between the differences and the averages, to look for any systematic bias, and to identify possible outliers (Bland and Altman, 1986). The plot in Figure 5.9 shows good agreement between the mean-manual and 2D segmentation methods, as all the measured points lie within 1.96 standard deviations. However, the mean value on the y-axis (mean manual result minus 2D result) of -80101.1 suggests that the 2D technique systematically over-segments the liver. As might be expected, this bias is more than that seen between the two manual observers (see Figure 5.1).

5.3.3.2. Volumetric overlap

The overlap between the 2D results and the manual segmentations is shown in Figure 5.10 and Table 5.5. The results show that the overlap V_O, calculated using equation (5.4) is consistently greater than 0.9, indicating good correlation between the segmentation of the majority of image slices. As mentioned in Section 5.1.3, volumetric overlap is perhaps the most reliable measure of the success of the segmentation algorithm, and the mean value of 0.93 is only 0.04 less than the mean value for the overlap between the two sets of manually segmented livers (see Table 5.1).

Dataset	V_O
1	0.96
2	0.91
3	0.94
4	0.93
5	0.93
6	0.94
7	0.91
8	0.93
9	0.92
10	0.94
11	0.93
12	0.94
13	0.95
14	0.95
15	0.93
16	0.96
17	0.95
18	0.89
Mean	**0.93**
SD	0.018

Table 5.5: *Raw data of the volumetric overlap of 2D automatic segmentation with manual segmentation*

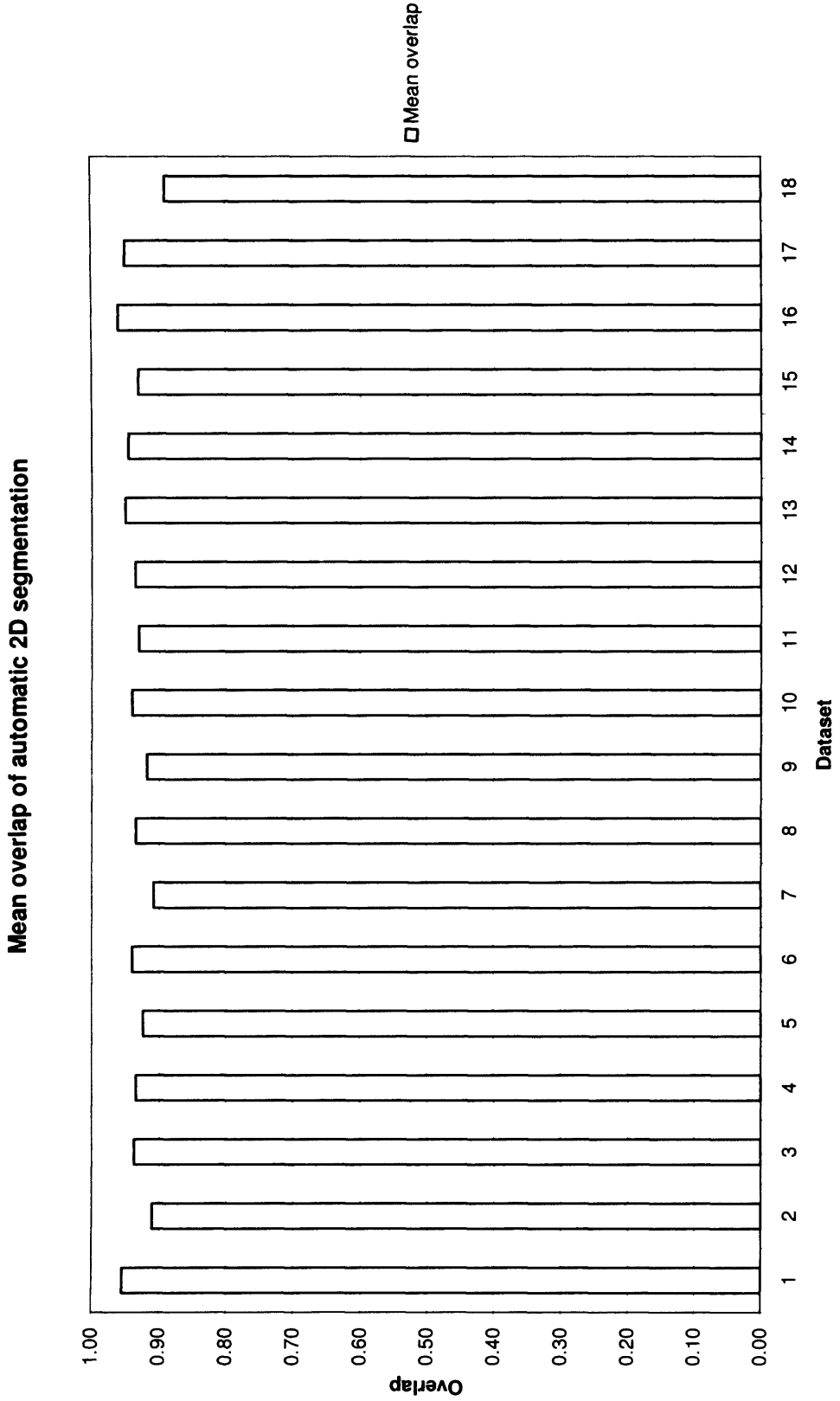


Figure 5.10: *Mean overlap of 2D automatic liver segmentation with two separate manual segmentations*

5.3.3.3. Contour distance

The mean results for the average distance between the automatic segmented contour, *A*, and the manually segmented contours, *M1* and *M2*, are presented below. As discussed above, the distances $d(A, M1|M2)$ are not the same as $d(M1|M2, A)$. In order to present results in the clearest fashion, means have been taken of results so that the actual accuracy can be better judged (Lamecker, Lange and Seebaβ, 2004). Thus, Table 5.6 shows the raw data, where the values represent the average of $d(A, M1|M2)$ and $d(M1|M2, A)$. Figure 5.11 presents the results graphically.

Dataset	Mean distance (pixels)
1	2.85
2	3.15
3	3.22
4	2.98
5	4.72
6	2.96
7	3.72
8	3.02
9	3.36
10	2.88
11	2.65
12	2.74
13	2.93
14	2.62
15	2.73
16	2.40
17	2.73
18	3.21
Mean	**3.05**
SD	0.518

Table 5.6: *Raw data of 2D segmentation distances between contours*

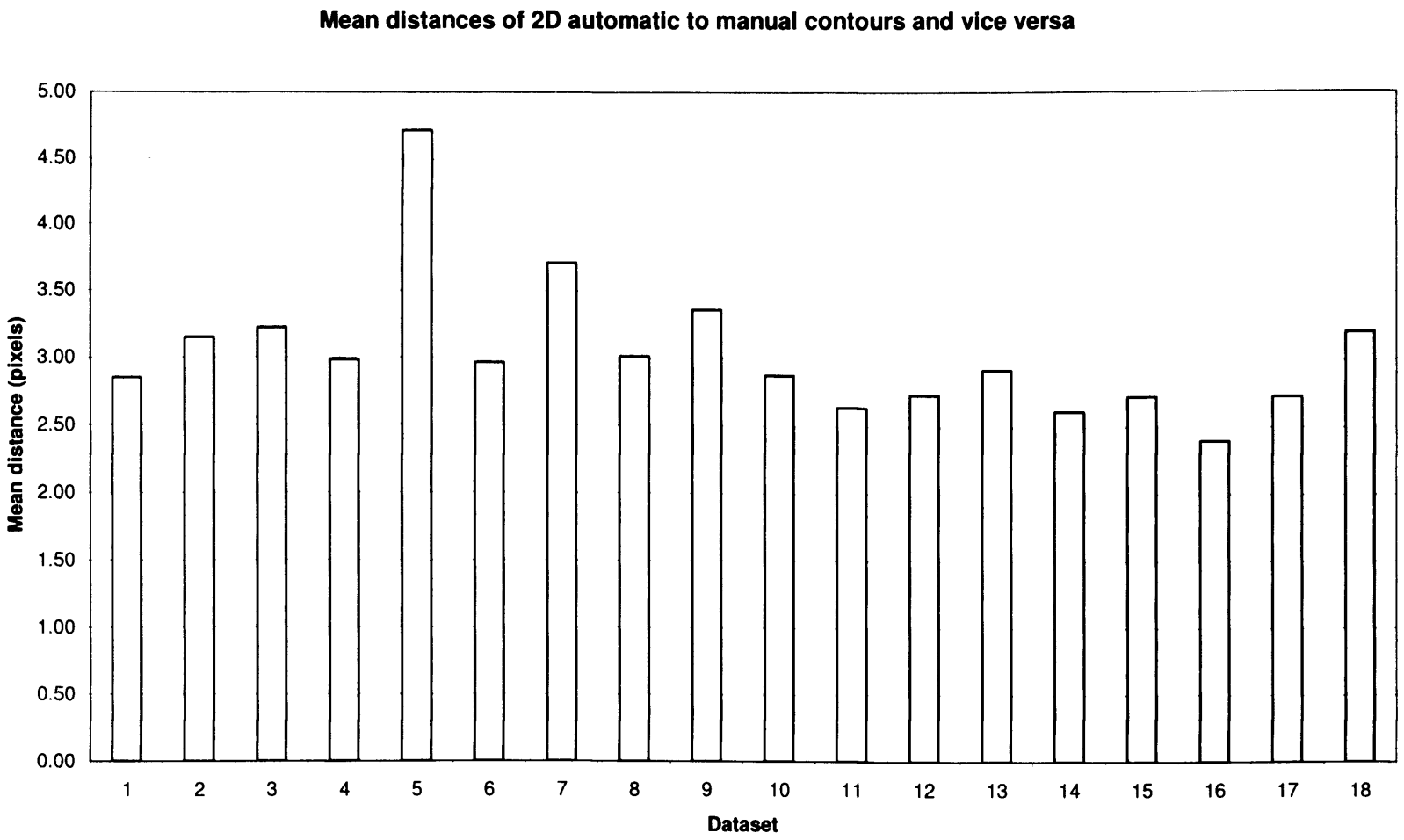


Figure 5.11: 2D segmentation distances between contours.

5.3.3.4. Effects of increased resolution at areas of higher curvature

One of the major novel contributions of the developed 2D segmentation is the ability to increase the accuracy of the segmentation result by increasing the resolution of the contour at areas of higher curvature. Figure 5.12 demonstrates how using a smaller grid resolution can prevent significant segmentation errors. In both the images in the figure, the active contour was initialised within the bulk of the liver. In Figure 5.12(a), the resolution of the grid has been set to 8 pixels, and it is clear that the contour has not segmented the liver correctly as it has not inflated into the lobe in the upper right hand side of the liver in the image. In Figure 5.12(b) the resolution of the grid is set to increase at areas of higher curvature (to a smallest grid resolution of two pixels), and the image shows more accurate segmentation of the liver.

(a) (b)

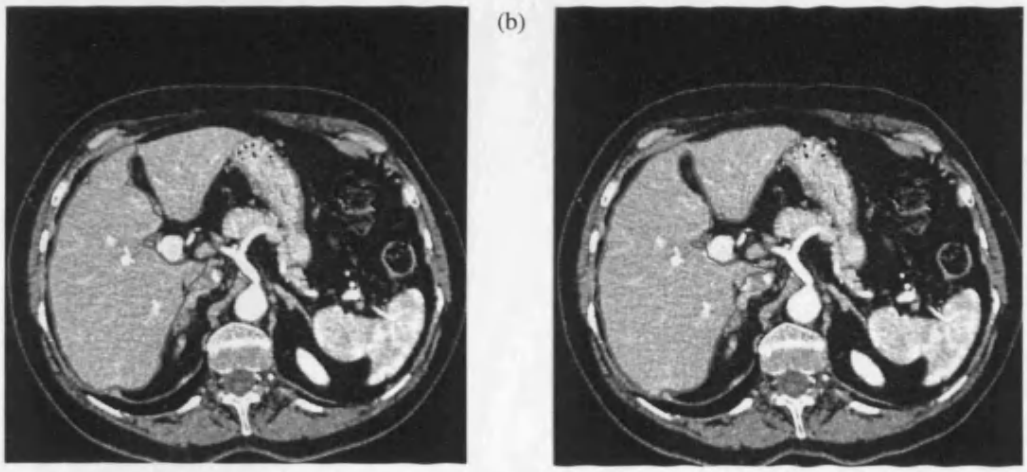


Figure 5.12: *Demonstration of increased segmentation accuracy with increased curvature.*

It is possible merely to set the resolution of the grid to a constant smaller size (e.g. two pixels). There are several disadvantages to doing this, however; one is that the computational power required for algorithm increases along with number of control points, which naturally decreases the algorithm's speed. Another is the fact that the presence of more control points greatly increases the frequency of the contour looping

around itself, as individual elements get trapped on 'noisy' edges, and the strength of the internal forces between closely spaced elements is insufficient to pull the trapped element 'through' the edge. Finally, a constant smaller grid size may allow the contour to 'leak' and cause incorrect segmentation, especially where there are breaks in significant edges, as demonstrated in Figure 3.7 in Chapter 3. These leakages and their direct causes are difficult to quantify as they may differ in their exact location and frequency. The robustness and repeatability of the main bulk of the results presented above are a testament to the developed algorithm's ability to change the contour's resolution and behaviour in areas of greater or lesser curvature.

Further statistics, discussion, and comparison with the 3D results are presented in Section 5.5 below.

5.4. Three dimensions

5.4.1. Active surface model

The 3D segmentation procedure addresses, in theory, several of the weaknesses of the 2D technique. In particular, the inherent nature of 3D segmentation is that each instance uses all the available data in a dataset, whereas in 2D only one slice of data can be used at a time. This means that 3D segmentation would be considered to be topologically more robust, and does not require additional segmentation procedures (which are required in 2D to segment liver tissue that appears separate from the main bulk of the liver in individual slices).

Nevertheless, 3D liver segmentation also has its disadvantages with respect to 2D segmentation. The desired topological robustness becomes a hindrance as the inflationary surface can 'leak' in a third dimension, and this is a particular problem in the liver. The *inferior vena cava* is a vein that returns deoxygenated blood from the lower half of the body to the heart. It travels very closely alongside the liver and, for a short distance, it is usually almost completely surrounded by liver tissue. During this distance it becomes visually indistinct from parenchymal liver tissue, and as a result it is invariably included in both manual and automatic segmentation – in such image slices it is frequently impossible for manual observers to identify its location within the liver.

At the superior and inferior ends of the liver, the inferior vena cava eventually becomes visually distinct, yet its average Hounsfield values (and thus appearance in images) and texture remain similar to those of the liver. Figure 5.13 contains four images, from the same dataset, that illustrate the locational relationship of the inferior vena cava and the liver. In Figure 5.13(a) and (d), the inferior vena cava is visually distinct from the liver. In Figure 5.13(c) the vein is beginning to become more apparent (when viewing slices

sequentially in a superior to inferior direction), yet in Figure 5.13(b) the vein is very difficult to distinguish.

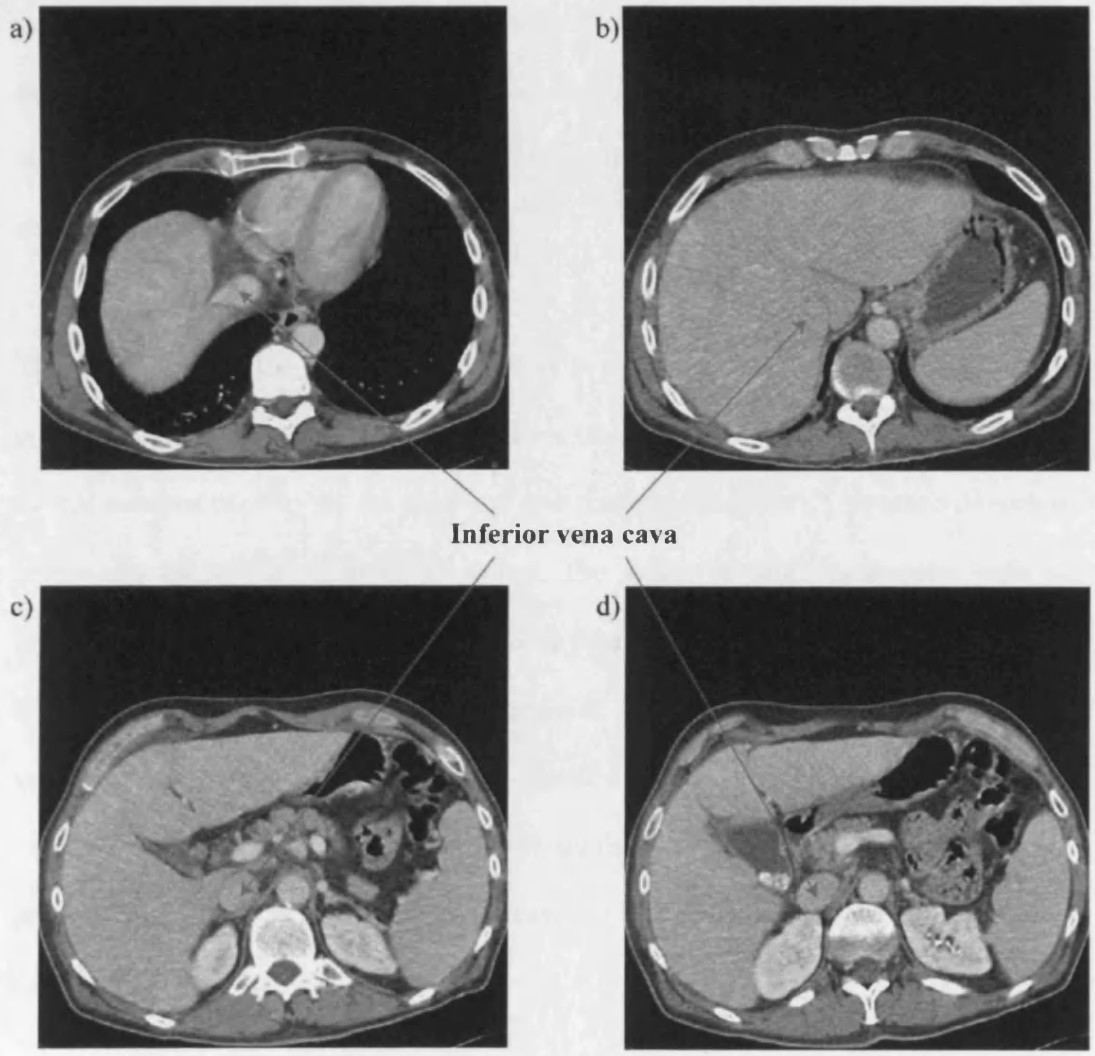


Figure 5.13: *The passage of the inferior vena cava through the liver, as viewed in CT images. Figures (a) to (d) are image slices from the same CT dataset and progress from the superior to inferior body direction. The arrow for figure (b) shows only the likely area of the vena cava.*

The net effect is that, as it inflates in some cases, the active surface may leak along the vena cava, in both superior and inferior directions. In the superior direction in particular, any leakage is catastrophic as the surface begins to expand into the heart. This problem is completely avoided in 2D due to the dataset being treated in only two dimensions – once the vessel is distinct from the liver, there is no probability that the 2D contour will

segment it. To counter this unwanted effect, unfortunately the level of automation must be reduced. Prior to initialising the algorithm, the operator may scroll through the images of the dataset and mark the points (superior and inferior of the liver) at which the vena cava becomes obviously separate from the liver. These markers are then used as restrictions for the algorithm, and the surface may not pass these markers. It is unfortunate that such steps are necessary, yet without them the segmentation results are not sufficiently robust.

The parameters for the surface model are set in a manner analogous to the 2D case. An operator selects an area of parenchymal liver tissue and the statistics of the data values in this area are used to set the high and low threshold values (1.5 standard deviations above and below the mean voxel value). The values of other parameters were set empirically and remained constant for each dataset. The surface was reparameterised after each 25 iterations of movement, the value is greater than that used in 2D as the volume of data being considered by the algorithm is much larger. Again, as in the 2D case, it was considered outside the scope of the project to investigate any possible parameter optimisation algorithms. The scope for future work in this area is discussed in Chapter 6.

The termination condition used for the segmentation was identical to that used in the 2D case: '*if, at a reparameterisation stage, no further vertices have been added to the surface data structure (thus the surface has not expanded since the last reparameterisation) the algorithm should terminate*'.

5.4.2. Three dimensional results

a)

b)

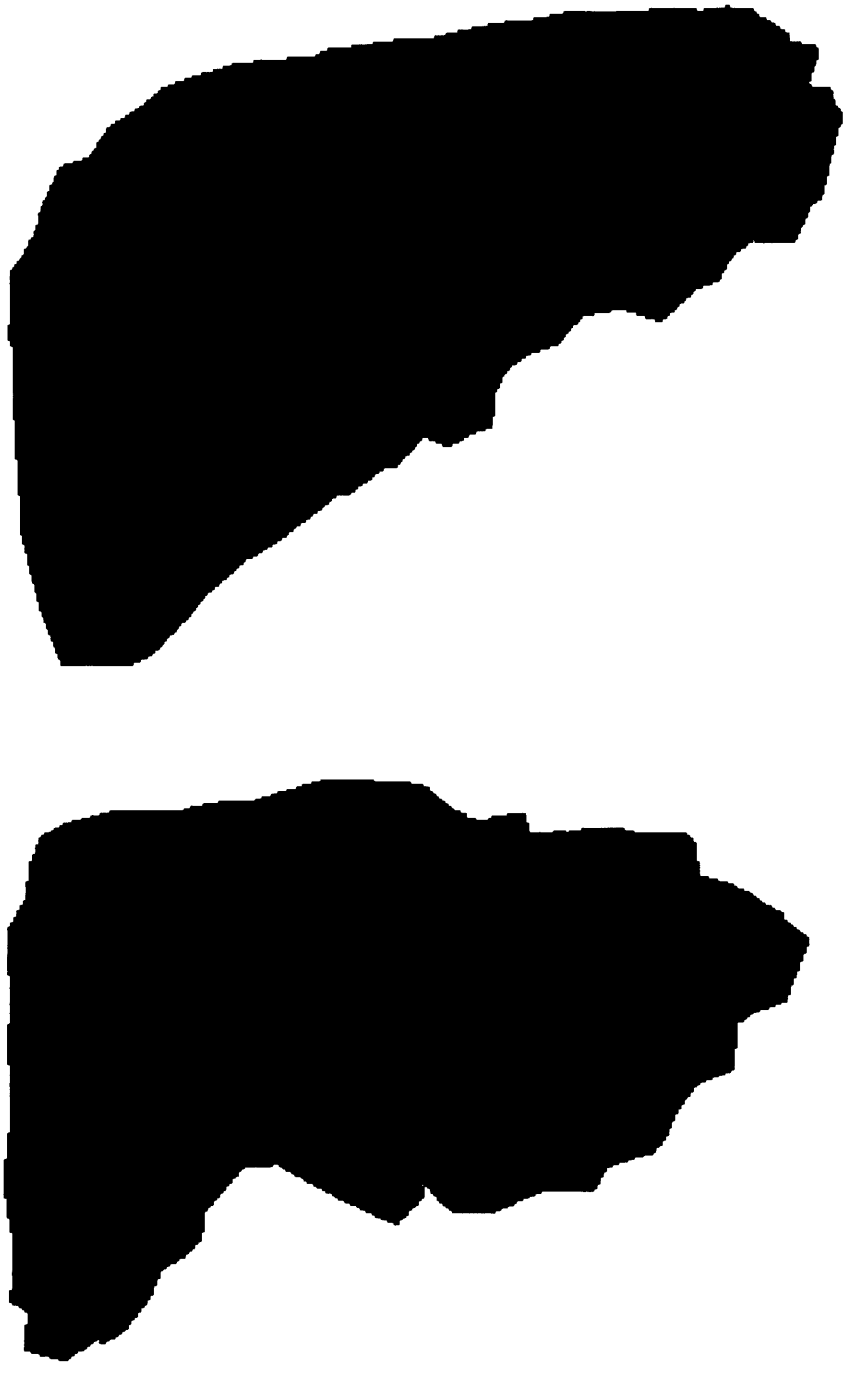


Figure 5.14: *Rendering of two separate livers segmented by the 3D active surface model*

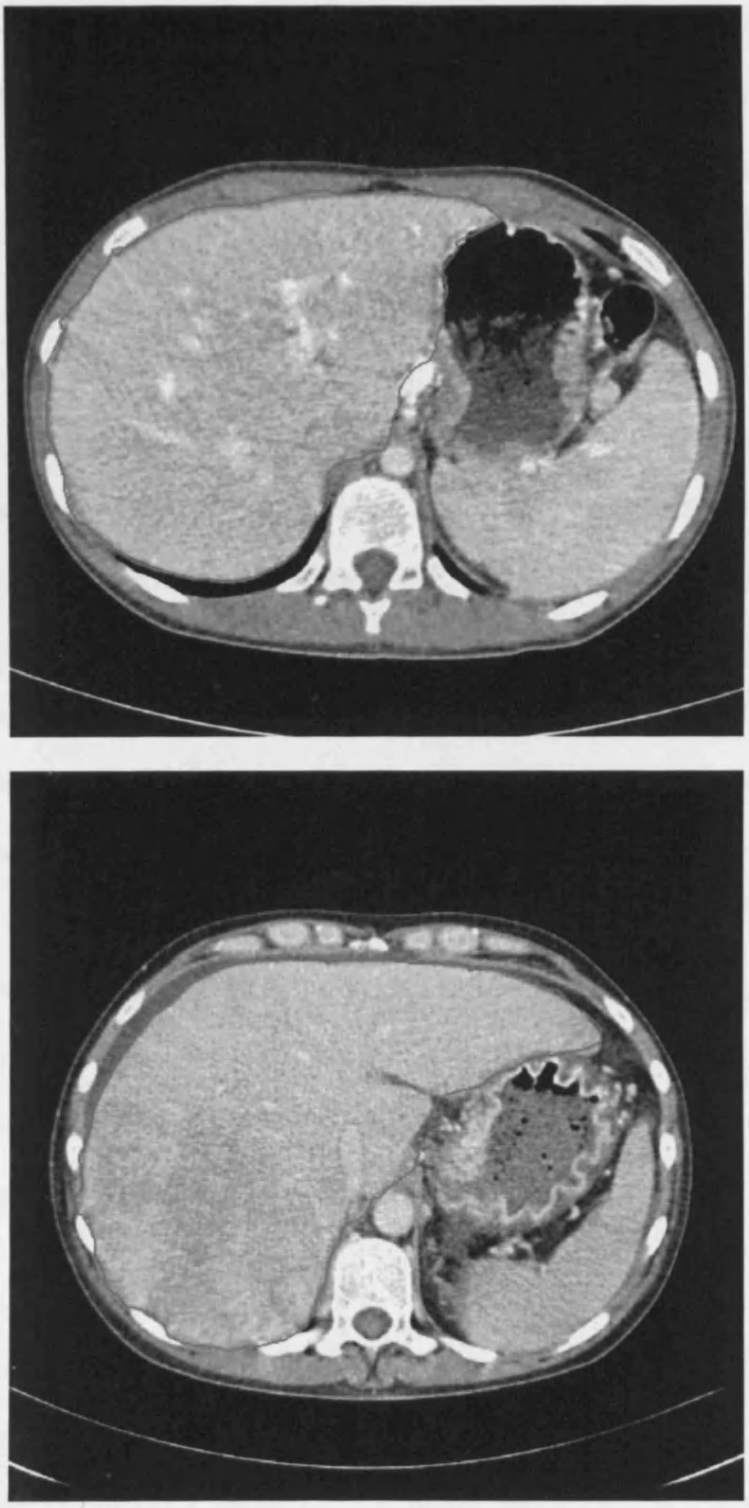


Figure 5.15: *Sample images of automatic 3D liver segmentation. The red contour represents the intersection of the surface with the plane of the image.*

Figure 5.14 contains two images showing 3D rendering examples of the final segmentation of two livers, using the active surface model. Figure 5.15 shows sample overlays of the segmentation results in 2D, where the 3D surface has been artificially 'sliced' by scan conversion.

5.4.2.1. Volume estimation

Figure 5.16 shows a graph comparing the measured volumes with segmentation carried out by two manual operators. Table 5.7 shows the raw data used for the graph. Using equation (5.1) the relevant intraclass correlation (*ICC*) value is calculated from the data presented in Table 5.7 is 0.995 (to three significant figures). According to this result is it possible to conclude (Gerig, Jomier and Chakos, 2001) that the variability in the data is solely due to the natural variation of liver volume among the population, and thus there is no significant difference between the values assigned by each rater.

The *F* value produced by an ANOVA for the data is 0.002 (to three significant figures). This is less than the relevant *F-critical* value of 3.178 that is presented in the data tables, and we can conclude from the ANOVA that there is no significant difference between the groups.

Dataset	Manual 1	Manual 2	3D
1	3438874	3401530	3335657
2	3676325	3643133	3892620
3	2047119	2041405	2093829
4	4220236	4282390	4217115
5	1942143	1965896	2115042
6	1986030	1979839	1841427
7	2253491	2308419	2381895
8	4211000	4211511	4323284
9	2342679	2353009	2451048
10	3005896	3016371	2979038
11	1088956	1097179	1037034
12	2376157	2440348	2360554
13	4133606	4237067	3908039
14	1197437	1237001	1100764
15	1884221	1891053	1878408
16	848182	877761	779870
17	2282859	2296545	2282626
18	2420211	2465118	2378448

Table 5.7: *Raw data of volumes measured by manual and 3D automatic segmentation*

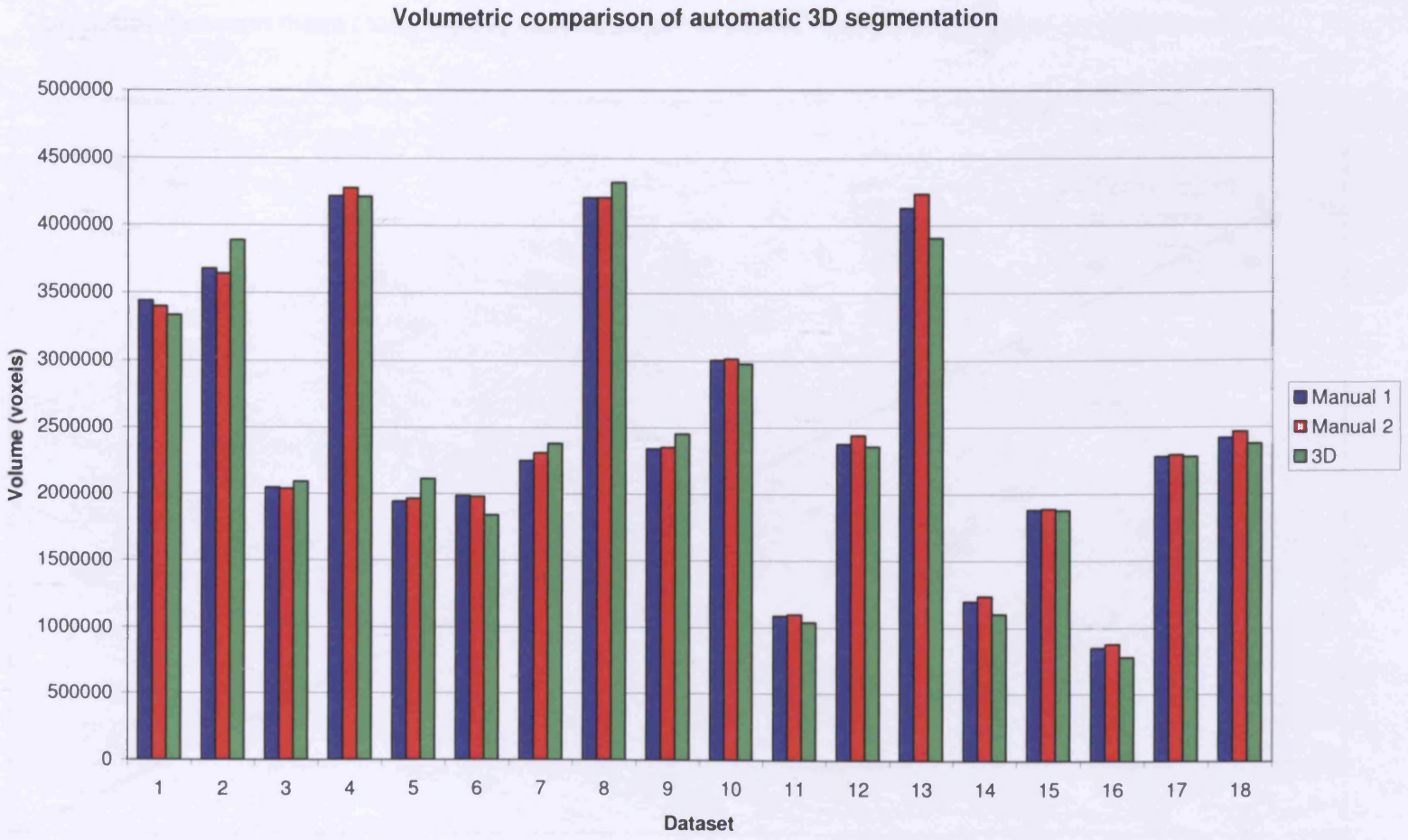


Figure 5.16: Measured volumes of manual and 3D automatic segmentation

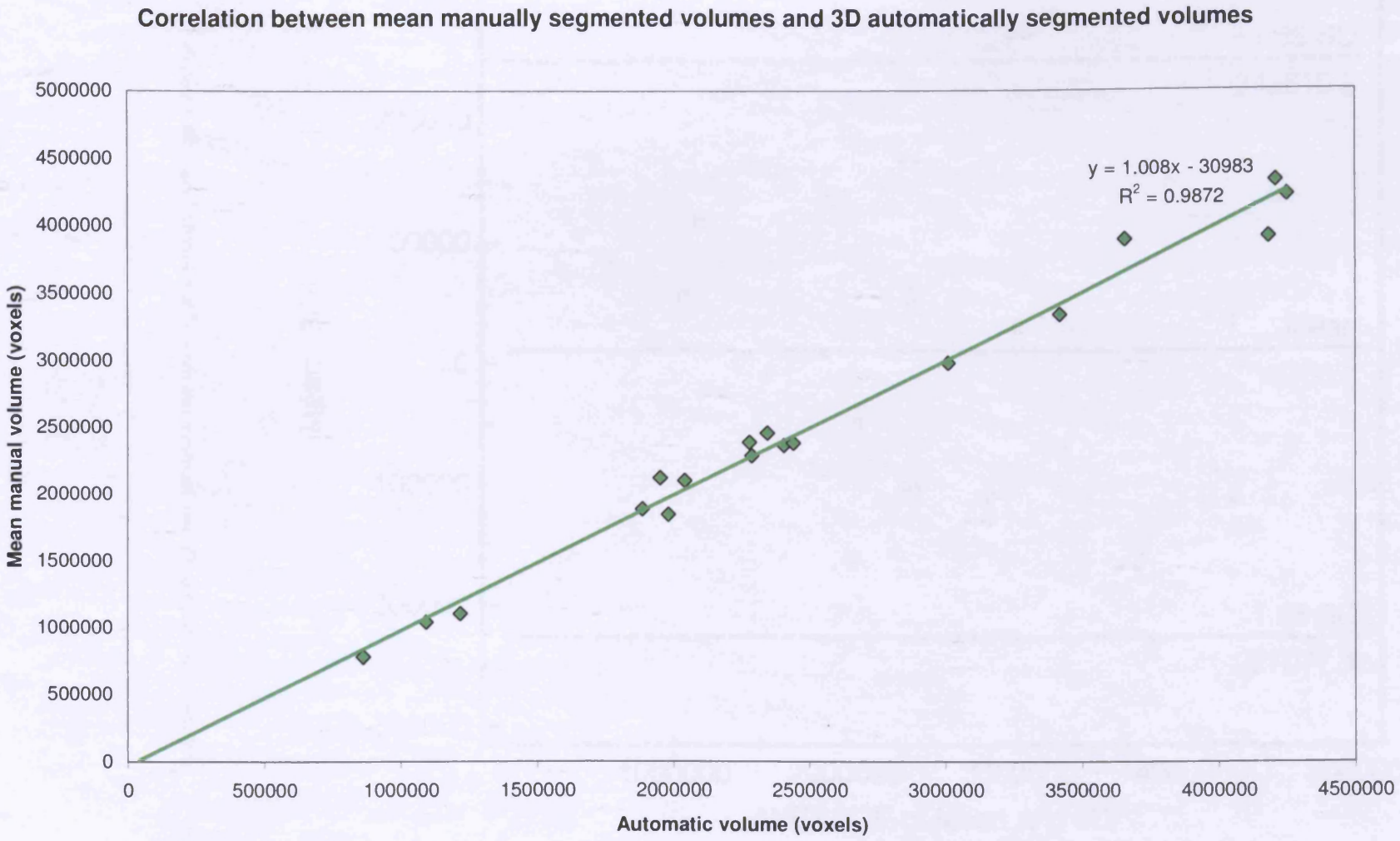


Figure 5.17: Correlation between volumes of manual and 3D automatic segmentations

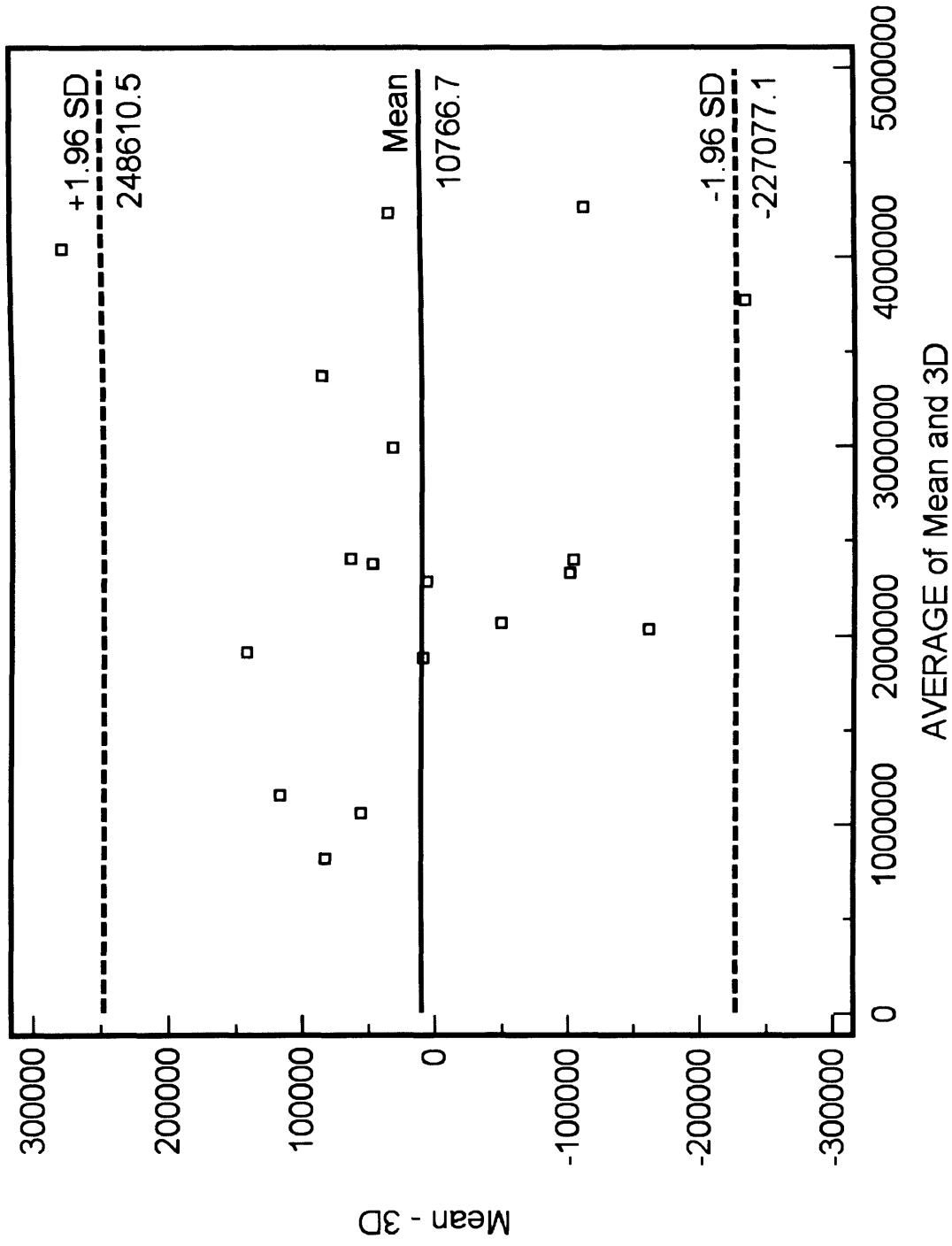


Figure 5.18: *Bland-Altman plot between mean manual and 3D automatic segmentations*

Figure 5.17 shows the correlation between the mean manual segmentation volumes and the 3D automatic segmentation volumes; it also shows the line of best fit for the series. The figure shows that the points from the 18 datasets cluster closely around the line; the equation of which, y = 1.008x - 30983, shows that its slope lies only 0.008 from a perfect gradient of 1. The R^2 value is 0.9872 which indicates a very high correlation.

Finally Figure 5.18 shows a Bland-Altman plot of the data, described above in Section 5.3.3.1 (Bland and Altman, 1986). The plot shows good agreement between the mean-manual and 3D segmentation methods, as all but two of the measured points lie within 1.96 standard deviations. The mean value on the y-axis (mean manual result minus 3D result) of +10766.7 suggests that the 3D technique systematically under-segments the liver. In contrast to the 2D case, this value is *less* than the bias observed in between the two manual observers (see Figure 5.1).

5.4.2.2. Volumetric overlap

The overlap between the 3D results and the manual segmentations is shown in Figure 5.19 and Table 5.8. The results show that the mean overlap, V_O, calculated using equation (5.4) is consistently greater than 0.9, indicating good correlation between the segmentation of the majority of image slices. Further discussion of the overlap results is present in section 5.5 below.

Dataset	V_O
1	0.93
2	0.92
3	0.93
4	0.91
5	0.91
6	0.93
7	0.91
8	0.91
9	0.92
10	0.92
11	0.94
12	0.93
13	0.91
14	0.87
15	0.94
16	0.89
17	0.94
18	0.92
Mean	**0.91**
SD	0.018

Table 5.8: *Raw data of the volumetric overlap of 3D automatic segmentation with manual segmentation*

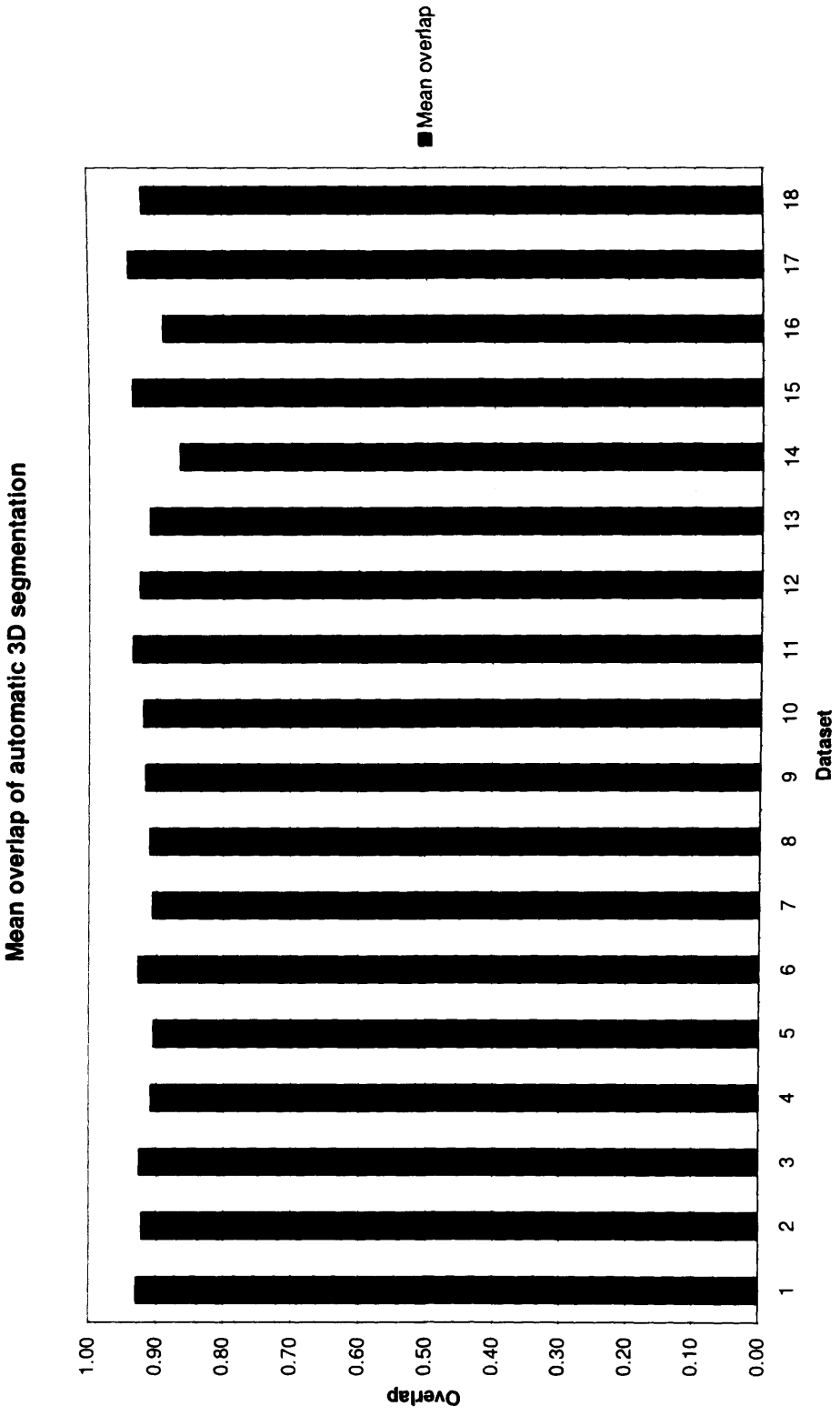


Figure 5.19: *Mean overlap of 3D automatic liver segmentation with two separate manual segmentations*

5.4.2.3. Contour distance

As for the 2D results, the means of the results for the average distance, *d*, between the automatic segmented contour, *A*, and the manually segmented contours, *M1* and *M2*, are presented below. Figure 5.20 presents the result graphically, while Table 5.9 shows the raw data.

Dataset	*d*
1	2.69
2	3.60
3	2.65
4	3.41
5	2.67
6	2.44
7	3.46
8	3.02
9	3.36
10	3.27
11	2.92
12	3.09
13	4.05
14	3.81
15	2.71
16	3.53
17	2.98
18	2.98
Mean	**3.15**
SD	0.443

Table 5.9: *Raw data of 3D segmentation distances between surfaces*

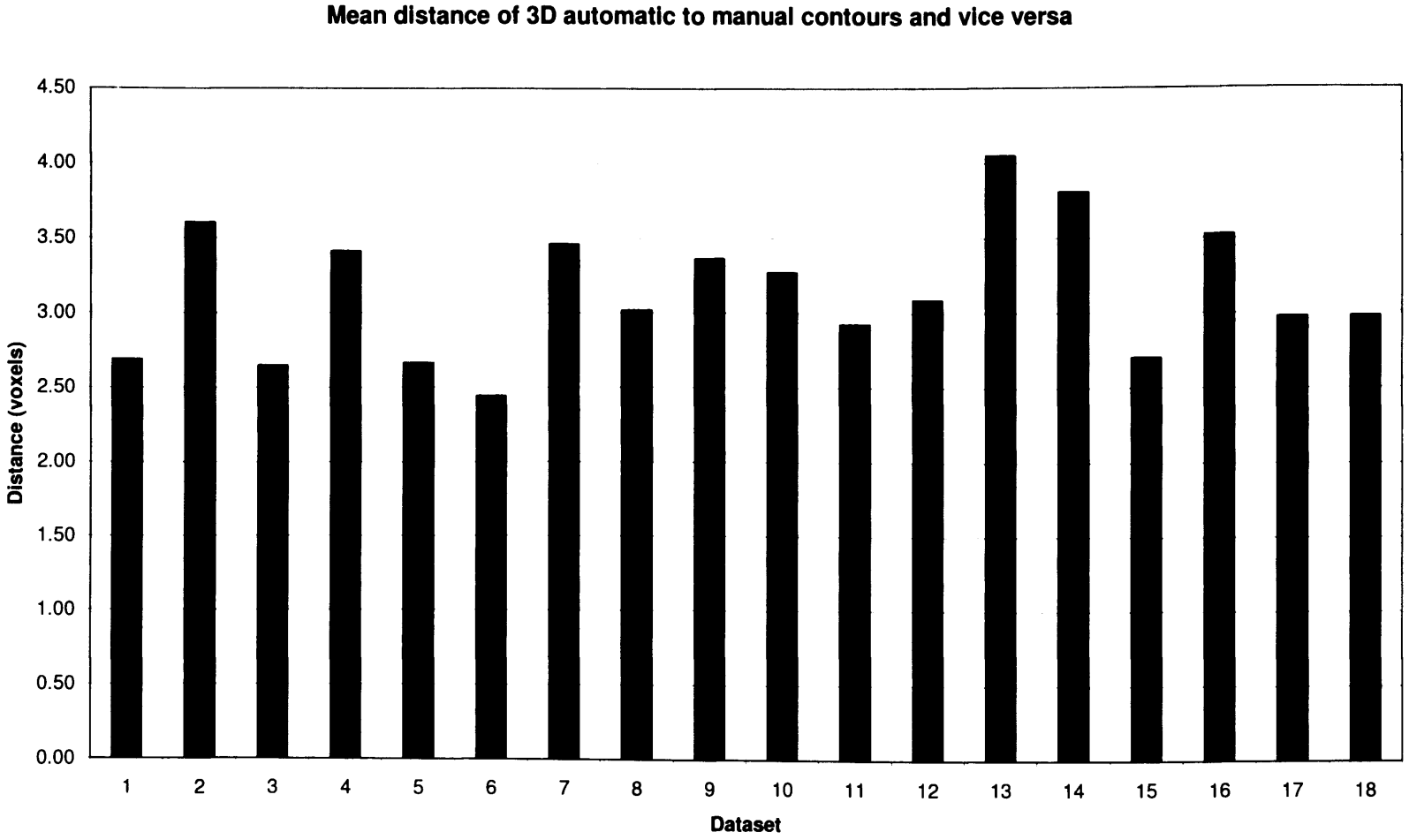


Figure 5.20: 3D segmentation distances between surfaces

Figure 5.21 shows 3D renderings of liver segmentation of a single dataset. The surface has been colour coded in order to show the location of the greatest segmentation error; blue indicates a surface distance less than the mean contour distance (presented in Table 5.9 as 3.15 voxels), green indicates a greater distance that is still less than twice the mean contour distance (6.30 voxels) and red indicates a distance value greater than twice the mean contour distance. Visual analysis of the results of segmenting other datasets shows that error distribution in the figure demonstrates the typical pattern of error that is observed for the segmentation results. The images show that, as might be expected, the greatest distance values occur in complex areas of the liver, whereas the distance measure in more uniform areas of the organ is below the mean distance.

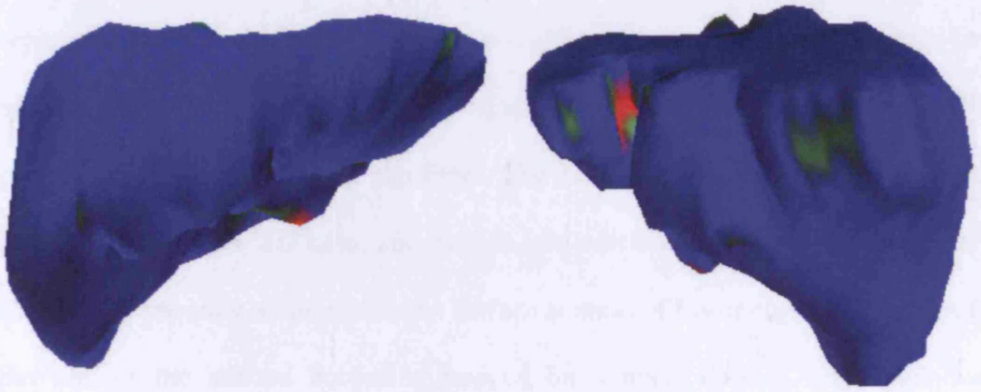


Figure 5.21: *Two 3D rendered views of the same 3D segmented liver, colour coded to show the areas of largest segmentation error. Blue indicates a distance value ≤ mean contour distance for all datasets, green indicates a distance ≤ twice mean contour distance, and red indicates a distance > twice mean contour distance.*

5.4.2.4. Effects of increased resolution at areas of higher curvature

As with the 2D algorithm, the curvature of the surface determines the resolution at which it is reparameterised, and Figure 5.22 shows three screenshots of a surface in the process of segmenting the liver, which has been colour coded so that blue-green-red represents increasing levels of curvature.

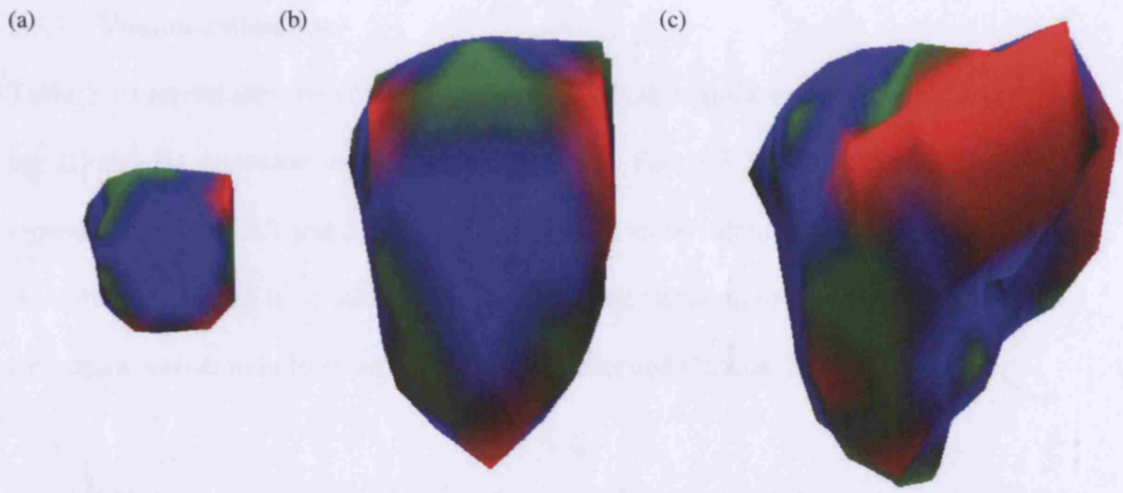


Figure 5.22: *Curvature of the 3D surface in its evolution during liver segmentation. The surface has been colour coded blue-green-red to represent increasing levels of curvature.*

As discussed in Section 5.3.3, it is possible merely to keep the resolution of the reparameterisation algorithm as high as possible, thus ensuring small distances between each vertex of the surface. The same arguments against implementing this in 2D bear particular relevance to the 3D algorithm. The 3D surface cannot deal with self-looping as effectively as the 2D case, and despite implementation efforts to prevent this (see Chapter 4), ensuring a low resolution surface at areas of lower curvature reduces further the risk of the surface becoming trapped on a noisy dataset 'edge' and looping. Furthermore, a small decrease in the value of A_k (the value controlling the maximum allowed face area in the surface, see Chapter 4) has an exponential increase in the number of vertices that comprise the model, resulting in the model becoming extremely slow and unwieldy to use. Maintaining high levels of surface accuracy only at the locations where it is required drastically reduces the computational power required to segment a single liver dataset.

5.5. Comparison of 2D and 3D

The purpose of this section is present the result of sections 5.3 and 5.4 side by side, so that the 2D and 3D techniques may be judged in comparison with each other.

5.5.1. Volume estimation

Table 5.10 reproduces the volumes estimated by both manual segmentations, and both the 2D and 3D automatic segmentation algorithms. Figure 5.23 contains a bar chart that represents the data in Table 5.10. The relevant *ICC* value calculated using equation (5.1) is 0.992, from which is possible to conclude that the variation in the data is solely due to the natural variation in liver volume (Gerig, Jomier and Chakos, 2001).

An ANOVA analysis of the data produces an F ratio of 0.030 (to three significant figures). This is less than the *F-critical* value of 2.740, as given in the data tables, a result that states that the differences between the sets of data are not statistically significant.

Dataset	Manual 1	Manual 2	2D	3D
1	3438874	3401530	3692620	3335657
2	3676325	3643133	3921864	3892620
3	2047119	2041405	2179866	2093829
4	4220236	4282390	4220236	4217115
5	1942143	1965896	2173264	2115042
6	1986030	1979839	2175173	1841427
7	2253491	2308419	2582051	2381895
8	4211000	4211511	4322497	4323284
9	2342679	2353009	2508978	2451048
10	3005896	3016371	3041173	2979038
11	1088956	1097179	1079586	1037034
12	2376157	2440348	2512034	2360554
13	4133606	4237067	4094168	3908039
14	1197437	1237001	1269585	1100764
15	1884221	1891053	1786347	1878408
16	848182	877761	876575	779870
17	2282859	2296545	2302583	2282626
18	2420211	2465118	2253719	2378448

Table 5.10: *Raw data of volumes measured by manual, 2D automatic and 3D automatic segmentation*

Figure 5.24 contains a bar chart that demonstrates the absolute percentage difference between the automatic segmentation volumes and the *mean* of the two manually segmented volumes. Due to the absence of an absolute gold standard, this graph is not wholly representative of the actual accuracy of either algorithm; however it is reproduced here as a method of comparing the performance of the 2D and 3D algorithms.

The pale red-band in the figure demonstrates the ± mean difference between the results of the two manual operators. For three datasets (4, 10 and 17) the automatic error is within the mean distance boundaries, the vast majority of automatic segmentation results like outside this boundary. The graph suggests that neither of the automatic segmentation techniques are as robust (i.e. they cannot repeatably provide accurate results) as manual segmentation.

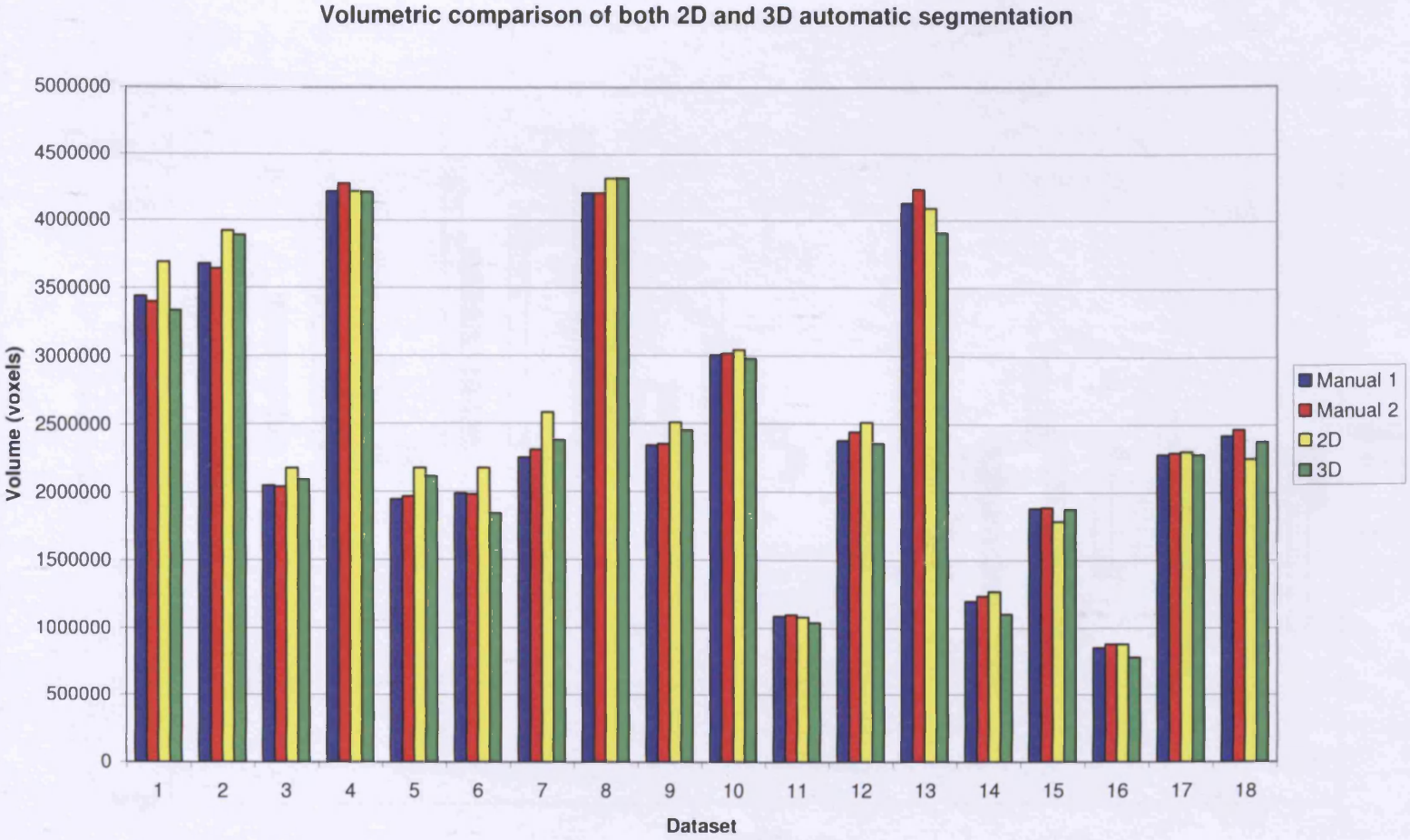


Figure 5.23: Measured volumes of manual, 2D and 3D automatic segmentation

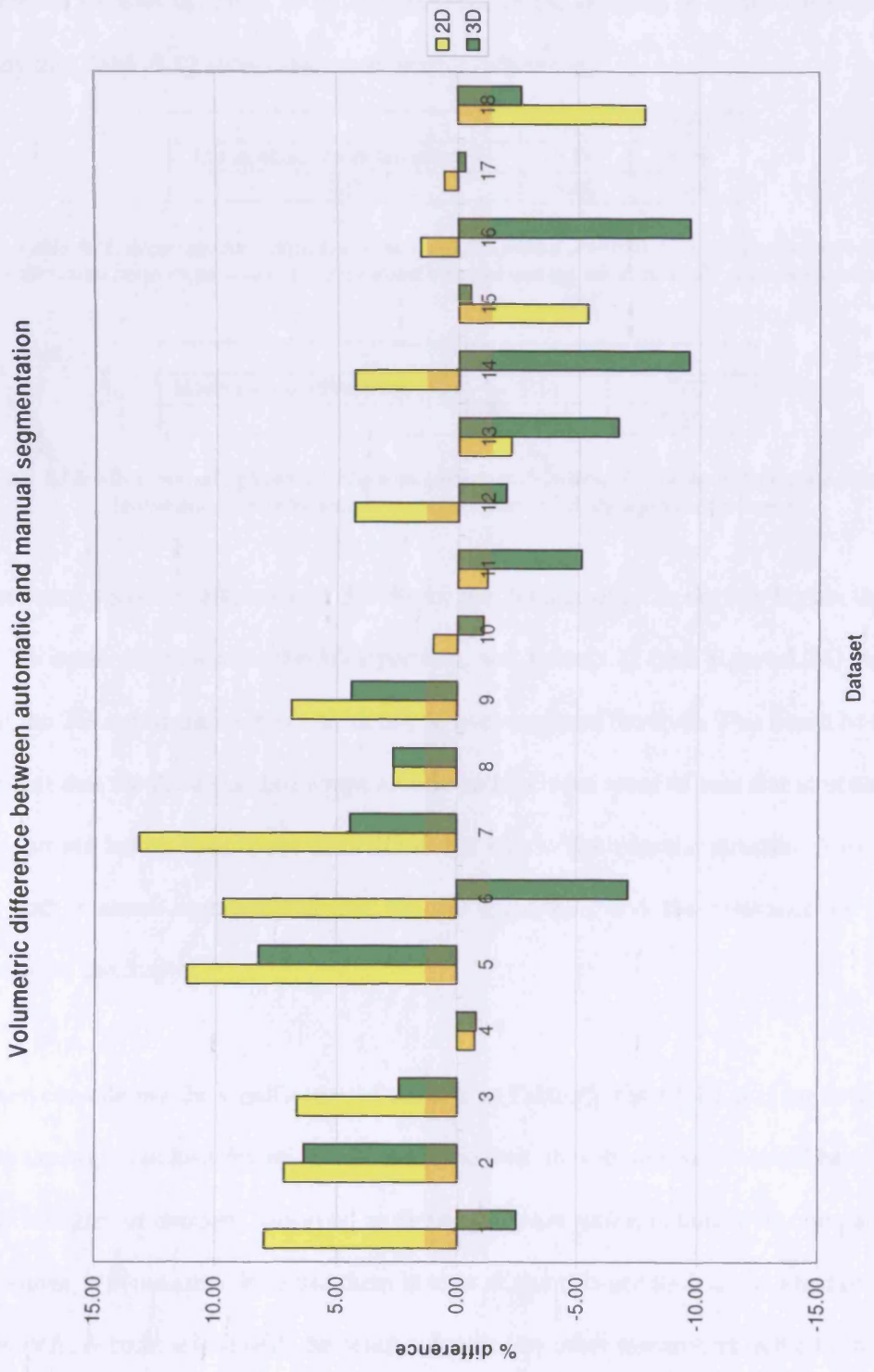


Figure 5.24: *Volume differences. Percentage difference between the automatic segmented volumes and the mean manually segmented volume. The region shaded red indicates the ± 1.34% absolute error between the two manual segmentations*

Table 5.11 shows the mean *absolute* difference of the volumes of all the datasets in the study and Table 5.12 shows the mean *overall* difference.

	2D	3D
Mean absolute difference (%)	5.24	4.24
SD	3.82	3.08

Table 5.11: *Mean absolute differences. Mean and standard deviation (SD) of the absolute value differences between the automatic segmented volumes and the mean manually segmented volumes*

	2D	3D
Mean overall difference (%)	3.33	-1.07
SD	5.66	5.22

Table 5.12: *Mean overall differences. Mean and standard deviation (SD) of the differences between the automatic segmented volumes and the mean manually segmented volumes*

The mean absolute difference of 5.24% for the 2D algorithm is slightly higher than the 4.13% mean difference for the 3D algorithm, and Table 5.12 (and Figure 5.24) suggests that the 2D segmentation has a tendency to over-segment the liver. This could be due to the fact that the 2D algorithm wraps around and includes areas of vascular structure that are surrounded by liver tissue (in a 2D image slice). The vascular structure is excluded by both manual segmentation and the 3D algorithm, and the relevance of this is discussed in Chapter 6.

When considering the significance of the data in Tables 5.9 and 5.10, it is important to note the high standard deviation values for the data; thus these results should be treated with a degree of caution. However, as these values are *ratios*, obtained via comparative measures, it is not suitable to use them in tests of significance such as the *t*-test or ANOVA. A comparison with the results obtained by other researchers in the field is presented in Chapter 6.

As a final volume comparison measure, Figure 5.25 summarises the data in Figures 5.7 and 5.16 showing the correlation between the mean manual segmentation volumes and both the 2D and 3D automatic segmentation volumes. The graph contains the two separate lines of best fit for both the 2D and 3D series'. The equations for the lines are displayed both in the figure and in Table 5.13 below, which are both very near to a perfect gradient of 1. The R^2 values for both sets of points are above 0.98, which indicates a very high level of correlation.

Dimension of segmentation algorithm	Equation of line of best-fit
2D	y = 1.0067x + 63052
3D	y = 1.008x - 30983

Table 5.13: *Equations of lines of best-fit – estimate through the volume correlation graph shown in Figure 5.25*

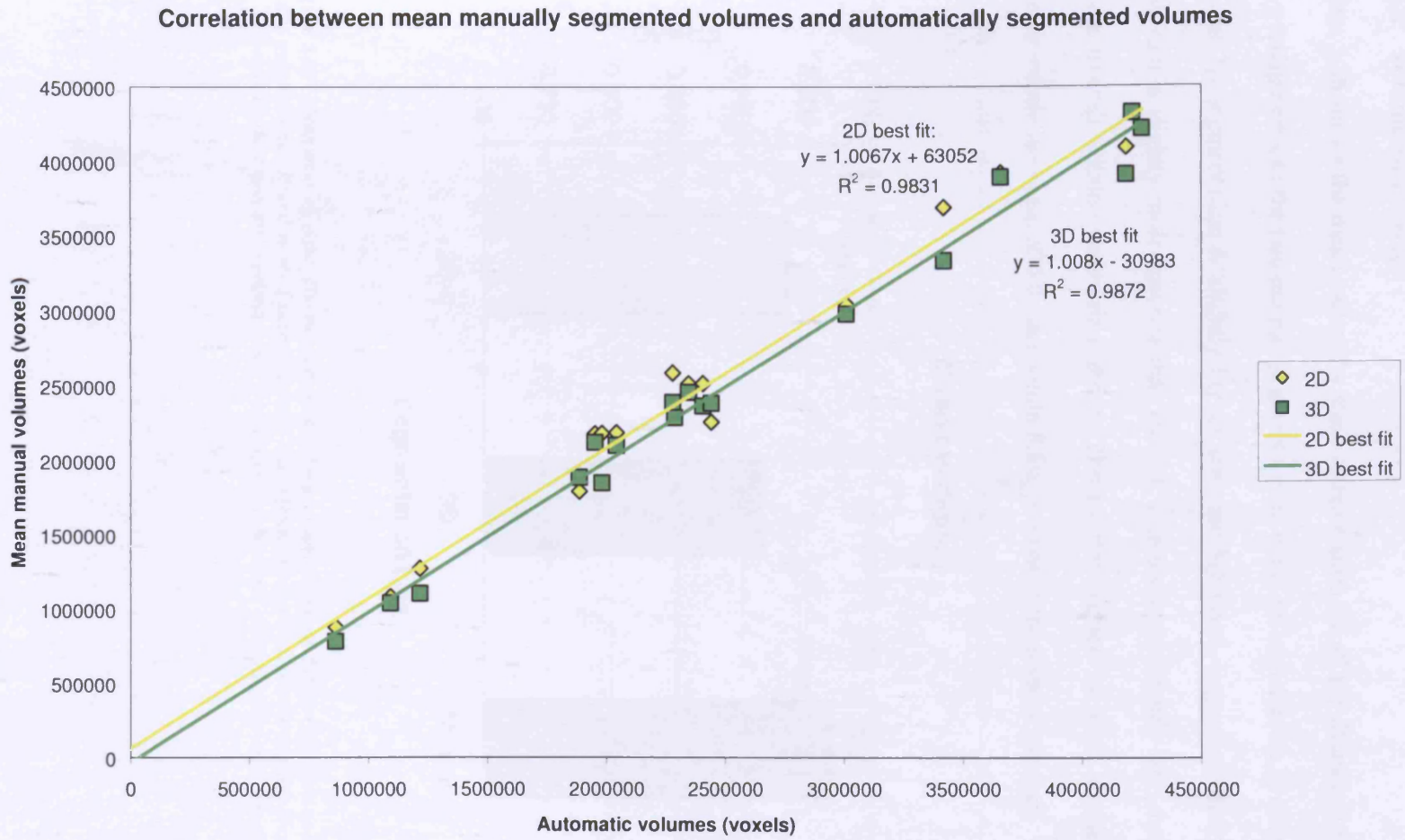


Figure 5.25: Correlation between volumes of manual and automatic segmentations

5.5.2. Volumetric overlap

Figure 5.26 shows the mean values for the overlap results for all the datasets, including the overlap between the two manual segmentations. It can be seen that the overlap value for the 2D segmentation is slightly higher than the 3D value, suggesting that the 2D algorithm is slightly more accurate than the 3D. This disagrees slightly with the results of the overall volume estimation above. The overlap values for both segmentation algorithms are in excess of 0.9, and within 5.6% percent of the manual overlap.

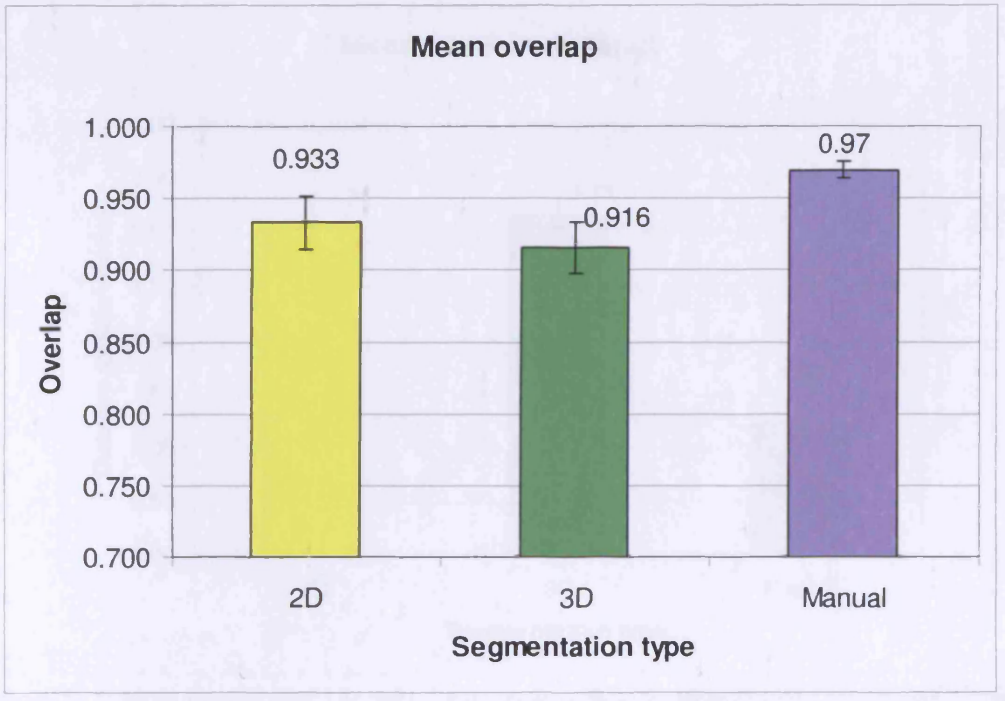


Figure 5.26: *Mean overlap data. The two left-most columns show the mean data for the overlap between 2D and manual, and 3D and manual segmentations. The rightmost column shows the overlap between the two individual manual segmentations (y-axis range is shortened to highlight to differences)*

5.5.3. Contour distance

Figure 5.27 shows the mean distances between the automatic contours and the manual contours. The mean distances for the both the 2D and 3D segmentation are very similar, and nearly two pixels greater than the distance between the two manually segmented contours. From this data it can be seen that there is a < 2 pixel mean overall distance difference between both 2D and 3D automatic segmentation and mean manual segmentation.

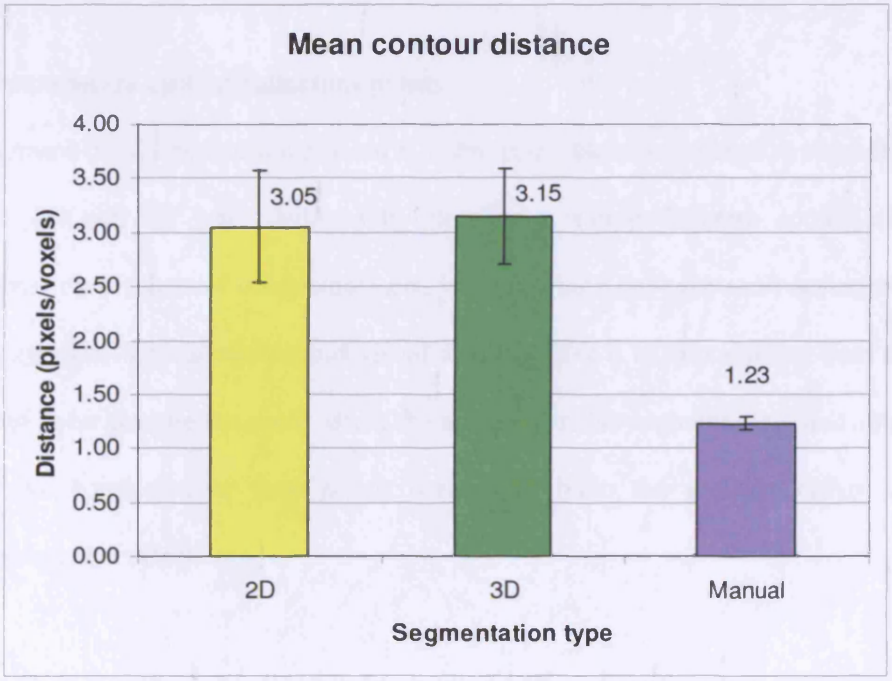


Figure 5.27: *Mean distance data. The two leftmost columns show the mean distance between the 2D and 3D segmentations with the manual segmentation, and the remaining column shows the mean distance between the two individual manual segmentations.*

It is apparent here that there is a surprisingly low mean distance difference, given the magnitude of the differences of volume estimation and volume overlap. For example, in dataset 7, there is a 13% difference between the 2D automatic segmentation volume and the mean of the manual segmentations, yet only a 3.72 average pixel difference between the surfaces (for all the image slices). The possible reasons for such a result are discussed in Section 6.3 below.

5.5.4. Algorithm speed

No in-depth speed studies were conducted, although the speed of the algorithms varied greatly depending on dataset size and liver shape complexity. As a rough guide, each individual 2D slice would take ~3-5 seconds to segment, and an entire liver (containing 150 slices, for example) would take ~10 minutes. By contrast, the 3D technique was much faster, and a typical 150 slice dataset would take ~5 minutes. All time estimates are based on algorithms running on a PC with a 2.6Ghz Pentium Xeon processor.

5.6. Parameters and initialisation points

As discussed briefly in Section 3.4 and 4.4 above, the parameters used to segment livers in both 2D and 3D were deduced by iteratively testing different combinations of parameters on a subset of images/datasets, judging which gave the most accurate results by comparison of area/volume and visual analysis. Yet it is important to note that the values of these parameters could affect the accuracy of the segmentation, and so a small study was conducted to find which parameters have the greatest effect, and to demonstrate that effect.

The study was conducted, separately for both the 2D and 3D algorithms, in two phases. The first phase attempted to find the two most sensitive parameters i.e. the parameters that created the largest segmentation error when their values were changed by the smallest amount from those presented in Chapter 3 and 4. To achieve this, each of the parameters was altered independently until *catastrophic segmentation failure* occurred, defined as an area/volume error of ± 25% from mean manual segmentation. Volume was chosen as the measure of segmentation error as it allows the results to more easily demonstrate whether the automatic algorithm has over- or under-segmented the data;

this is useful when analysing the results and when discussing the reasons for why altering a parameter produces failed segmentation.

Segmentation was carried out in increasing 'bands' of difference, ± 5, 10, 25, 50, 75, 100, and 200% e.g. each parameter was changed (independently) by ±5 and segmentation carried out; then each changed by ±10 etc. This approach makes it possible to eliminate surplus tests – once the two most sensitive parameters have been found, it is unnecessary to keep testing other parameters at higher difference values.

The two parameters which produced catastrophic error due to the smallest percentage change in their value were put forward to the second phase of the study. This phase measured the volume using parameter values varying at regular intervals in between the values which produced catastrophic failure. In addition to measuring the effects of changing the parameters individually, the results of changing the parameters *simultaneously* were also measured. Again, the method used to measure the accuracy of the segmentation was comparison of area/volume with mean manual segmentation.

All tests were carried on a subset on six datasets, chosen at random from the 18 used for the main study. The mean of the results is used in the graphs below, though due to the large volume of data and complexity of the three-dimensional graphs, standard deviation values are omitted. The 'starting' values of the parameters for 2D and 3D are listed in Chapter 3 and 4 respectively. It should be noted that, under the study design, it would be possible to change the sign of a parameter once the percentage difference reaches a large enough value. As the value of the parameter (with the exception of the standard deviation value used to set thresholds T_{lo} and T_{hi}) is designed to control the *magnitude* of each force, this is nonsensical. As a result, negative parameters values

were not allowed, and parameter changes > -100% were dealt with so that the parameter remained equal to zero.

5.6.1. 2D parameters

The data in Table 5.14 shows the number of datasets that failed at each parameter value (zero values are left as blank cells for clarity). In the -50% row, the table shows all six datasets failed for the SD parameters (controlling the threshold values). Four segmentations failed for parameter *c* (controlling the strength of the inflationary force), with the remaining two failing at -75%. Two suffered catastrophic failure when the image-based parameter, *d*, was increased by 75, but the most sensitive parameters are clearly *c* and SD. Furthermore it can be seen that it is a reduction in these values that causes failure – the normal force becomes so weak that it cannot overcome either the elasticity term of the surface or the external, image-based force; and the threshold values come increasingly close together such that the normal force is reversed more frequently.

	x	a	b	c	d	SD
-75				2		X
-50				4		6
-25						
-10						
+10						
+25						
+50						
+75					2	

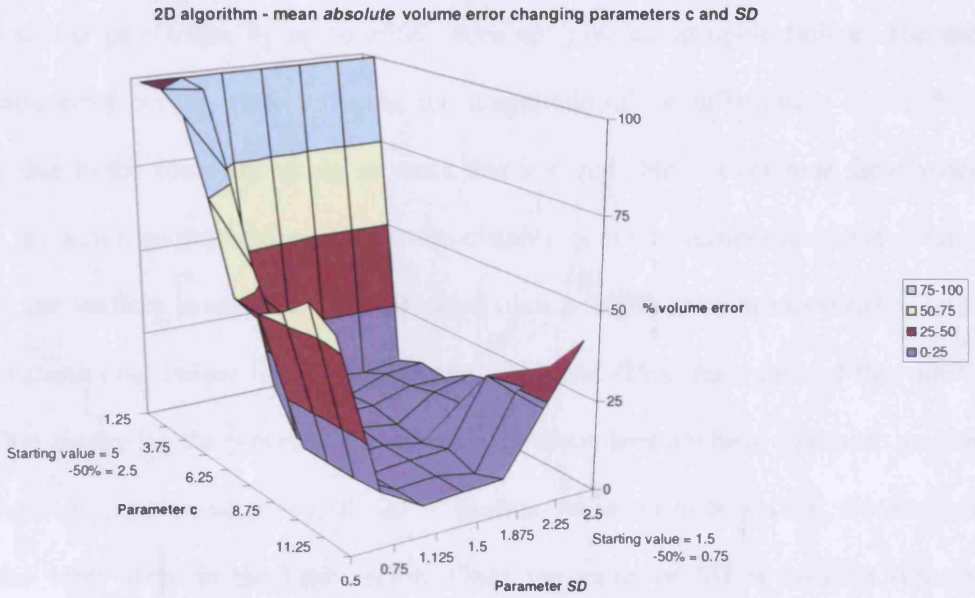
Table 5.14: *Number of liver datasets achieving catastrophic segmentation failure at each parameter alteration. 'X' indicates no test was carried out, due to all datasets having failed at a lower absolute percentage difference.*

Parameters *c* and SD were put forward for the second phase of the study where their effect on the segmentation accuracy was measured over a range of values. Segmentation was carried out while changing the parameters simultaneously, thus allowing a two

dimensional grid of parameter values to be tested. Figure 5.28 contains two graphs that show the segmentation error (as measured by percentage volume difference between automatic and manual segmentations). Figure 5.28(a) is a three-dimensional chart where the x and y axes represent the value of the parameter, while Figure 5.28(b) is a contour chart, where the different colours represent the different levels of percentage difference to manual segmentation. The colour code of the graphs is set so that light blue represents a segmentation result below the catastrophic failure threshold of 25% difference.

It should be noted that parameters c and SD are directly linked, in that they both control the inflationary force, γ, of the contour/surface. The value of c controls the magnitude of the force, whereas the value of SD assigns the upper and lower threshold values – where intensity values above or below the respective upper and lower thresholds reverse the direction of the inflationary force and make it an inflationary force.

(a)



(b)

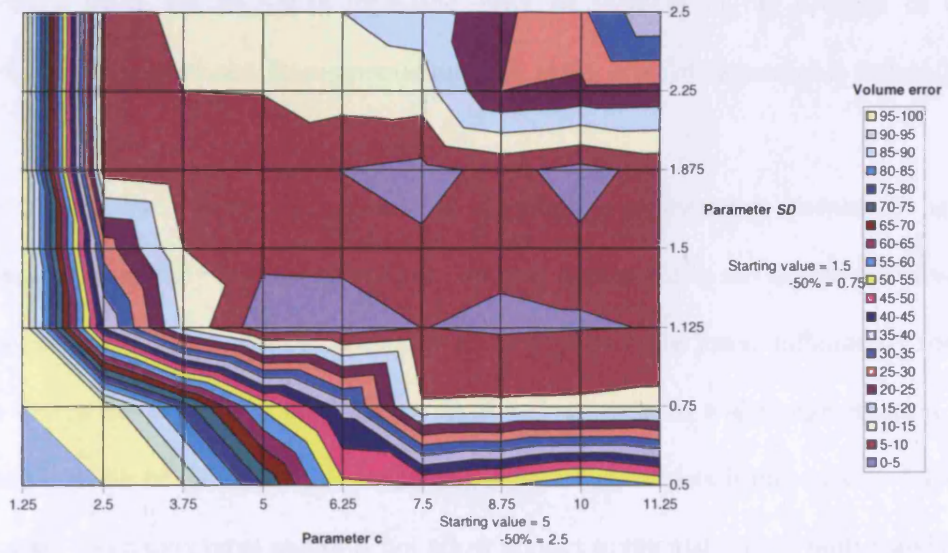


Figure 5.28: *Absolute volume difference of 2D segmentation accuracy changing parameters c and SD. (a) shows a 3D graph, (b) shows a contour graph with increased resolution.*

The shape of the graphs show a distinctive flat 'base', indicating the alterations of the value of the parameters by up to ±50% does not give catastrophic failure. The most dramatic error occurs while reducing the magnitude of the inflationary force; this is likely due to the force becoming so weak that it is not able to overcome the elasticity force, α, which attempts to regularise the distance between vertices and thus naturally draws the vertices inwards to a single point (thus a 100% error in measured volume). This catastrophic failure is seen at between -50% and -75%, regardless of the value of SD. Yet increasing the power of the inflationary force does not have so drastic an effect on the segmentation accuracy; if SD is unchanged, even at > +100% the measured volume error stays in the base region. Once the value of SD is lowered (thus the threshold width is smaller) segmentation fails regardless of the strength of the inflationary force; yet as SD is increased, only an increase in the strength of the inflationary force increases the segmentation error to the level of catastrophic failure.

Figure 5.29 shows the mean volume of automatic segmentation, measured as a percentage of manually segmented volume, where it is possible to see whether the livers have been over- or under-segmented. As might be expected, the lower inflationary force causes severe under-segmentation. Lowering SD also results in under-segmentation, as the narrow width of the upper and lower threshold values makes it more likely for the inflationary force to reverse and thus not allow correct segmentation. As both c and SD increase, the resulting increased strength of the inflationary force coupled with the lower propensity for it to be reversed results in over-segmentation.

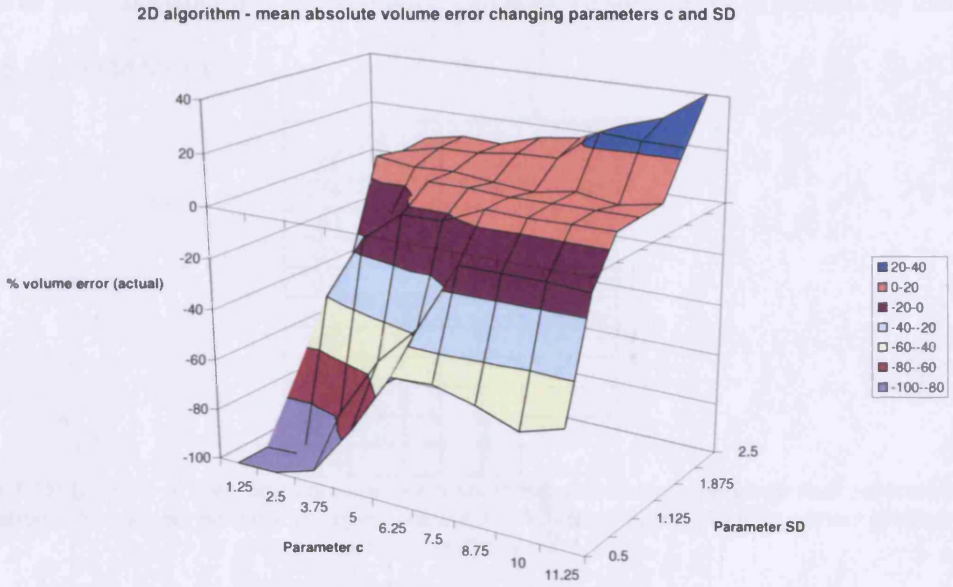


Figure 5.29: *Mean volume of 2D segmentation, measured as percentage of manually segmented volume, changing parameters c and SD.*

5.6.2. 3D parameters

The data in Table 5.15 shows the number of datasets that failed at each parameter value (zero values are left are blank cells for clarity). The table shows that most sensitive parameter is SD, with five datasets failing at -50% and one at -75%. The second most sensitive parameter is *d* (controlling the strength of the image force), with four datasets failing at +75 and two at +100%. Thus parameters *d* and SD were moved forward to the second phase of the study.

One point worthy of note is the fact that when parameters *a*, *b* are reduced to zero (-100%) catastrophic failure does not occur. This is because there is some overlap between the effects of the internal forces α and β – analysis of equation (4.5) shows that a proportion of the same data is used to calculate α is also used for β. Thus removing the influence of either α or β does not have a catastrophic effect on segmentation. Furthermore, failure only occurs in three datasets when *d* (the edge-based strength) is

reduced to zero, showing that segmentation can achieve some level of success by using only the threshold levels.

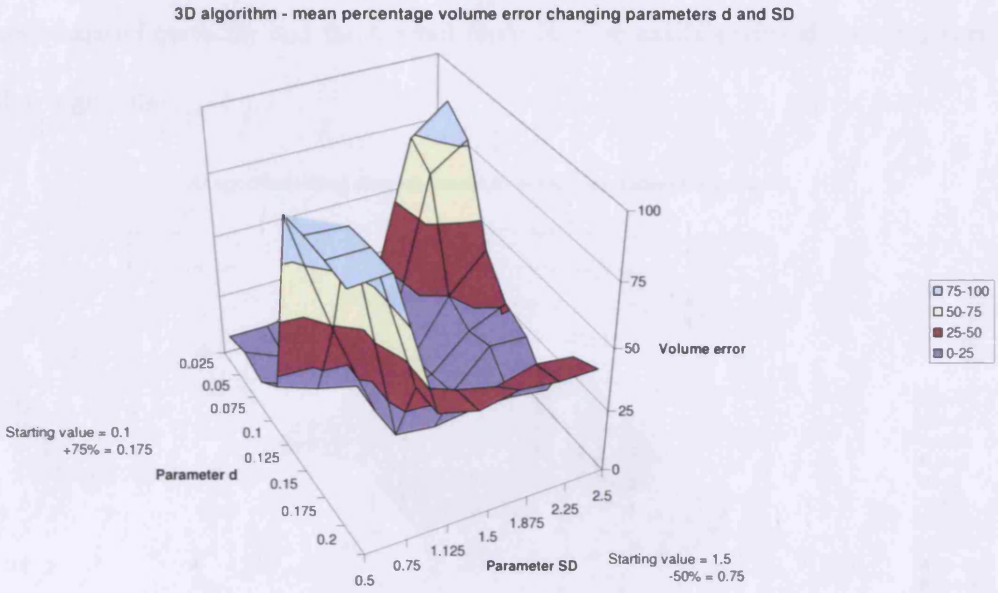
	x	a	b	c	d	SD
-100	6			4	3	X
-75				2		1
-50						5
-25						
-10						
+10						
+25						
+50						
+75					4	
+100					2	

Table 5.15: *Quantity of liver datasets achieving catastrophic segmentation failure at each parameter alteration.. 'X' indicates no test was carried out, due to all datasets having failed at a lower absolute percentage difference.*

As for the 2D algorithm, segmentation was carried out while changing the parameters simultaneously, allowing a grid of parameter values to be tested. Figure 5.30 contains two graphs that show the segmentation error (as measured by percentage volume difference between automatic and manual segmentations). Figure 5.30(a) is a three-dimensional chart where the *x* and y axes represent the values of the parameters, while Figure 5.30(b) is a contour chart, where the different colours represent the different levels of percentage difference to manual segmentation. The colour code of the graphs is set so that light blue represents a segmentation result below the catastrophic failure threshold of 25% difference.

In contrast to the 2D study, parameters *d* and SD are completely independent, the former controlling the strength of the external, image based force (represented by edge detection), and SD representing the width of the upper lower and thresholds as above. The graphs in Figure 5.30 show a bimodal graph which again has a flat 'base', showing that there is some flexibility in setting of the parameters to obtain results comparable with those presented above for the main study of the thesis.

(a)



(b)

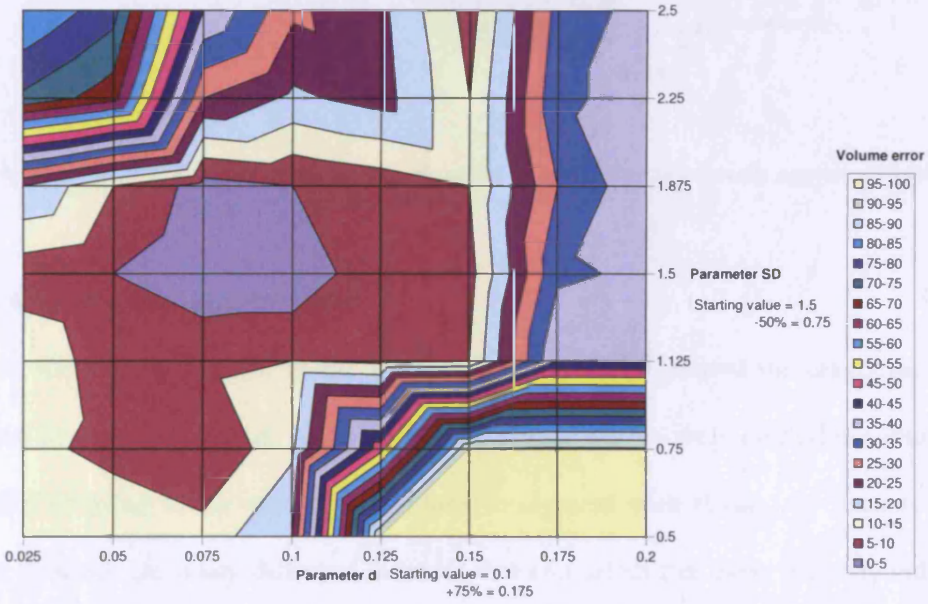


Figure 5.30: *Effect of changing parameters c and SD on 3D segmentation accuracy, measured by volume difference to manual segmentation. Figure (b) shows a contour graph with increased resolution.*

Figure 5.31 shows the mean volume of 3D automatic segmentation, measured as a percentage of manually segmented volume. It shows that as both the edge-based force is decreased while the SD (thus threshold width) is increased, the surface tends to over-segment due the surface forcing its way through edges, and the wide threshold width allowing it to continue segmenting surrounding tissue. By contrast, as the strength of

edges in the dataset is increased and the width of the thresholds decreased, the surface cannot expand correctly and the normal force is more easily reversed, thus the surface under-segments.

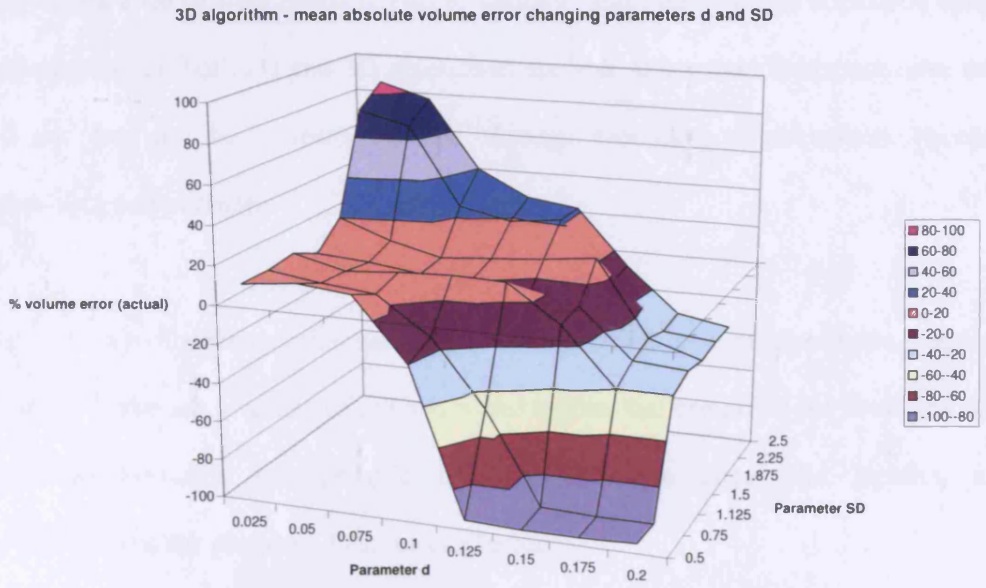


Figure 5.31: *Mean volume of 3D segmentation, measured as percentage of manually segmented volume, changing parameters c and SD.*

5.7. Unhealthy liver segmentation

Much of the clinical interest in the liver naturally revolves around the diagnosis and treatment of unhealthy tissue. As a result, preliminary studies were carried out into the feasibility of using the developed algorithms to segment such tissue. As discussed in Chapter 1, there are many different diseases that can affect the liver; it is beyond the scope of this thesis to deal with the classification of such disease, however the developed algorithms can assist with the quantification of disease, and provide an important basis for future work in the area, which is discussed in Chapter 6.

Two types of abnormalities were used in a small study to gauge the suitability of the developed segmentation algorithms to assist with future work. Firstly, the algorithm was run on livers with varying levels of cirrhosis. It was found both 2D and 3D algorithms

were able to successfully segment mildly cirrhotic livers, though as the levels of cirrhosis increased, the segmentation algorithms struggled to find the correct boundaries. The reason for this is that, as the level of disease increases, the uniformity of the appearance of liver tissue begins to degrade, and 'noisy' edges appear. A simple conclusion is that both 2D and 3D algorithms are able to segment livers that have mild cirrhosis, but as the severity of the disease increases, segmentation becomes increasingly less accurate.

A second class of abnormalities used to test the segmentation algorithms were abscesses or lesions. There are a variety of abscesses and lesions that can affect the liver, and their correct segmentation can provide useful information about the location and quantification of the disease. Lesions can include:

- Primary liver tumour (hepatocellular carcinoma): Discrete solitary or multiple lesions are seen as well-defined low-density areas compared to the surrounding liver.
- Hepatic metastases: hypovascular and therefore hypodense on contrast enhanced scans.
- Benign Cysts: seen as sharply defined homogenous areas, the contents of which have a density nearer to water and are not affected by intravenous injection of contrast medium.

Figure 5.32 contains a selection of images that demonstrate the ability of both 2D and 3D algorithms to segment well defined abnormalities. In each case, the algorithm was initialised within the bulk of the abscess or lesion (as in the case of healthy tissue segmentation), and the threshold parameters set automatically according to the pixel/voxel values contained with the initialised area/volume. Table 5.16 presents

quantitative results from five separate lesions. The volumes of the manually segmented abnormalities are shown, with the two subsequent columns containing the volumes measured by the 2D and 3D segmentation. The final two columns contain the values for the overlap between the automatic and manual segmentations. Figure 5.33 contains a bar chart showing the segmented volumes (the data in the columns two, three and four of Table 5.16).

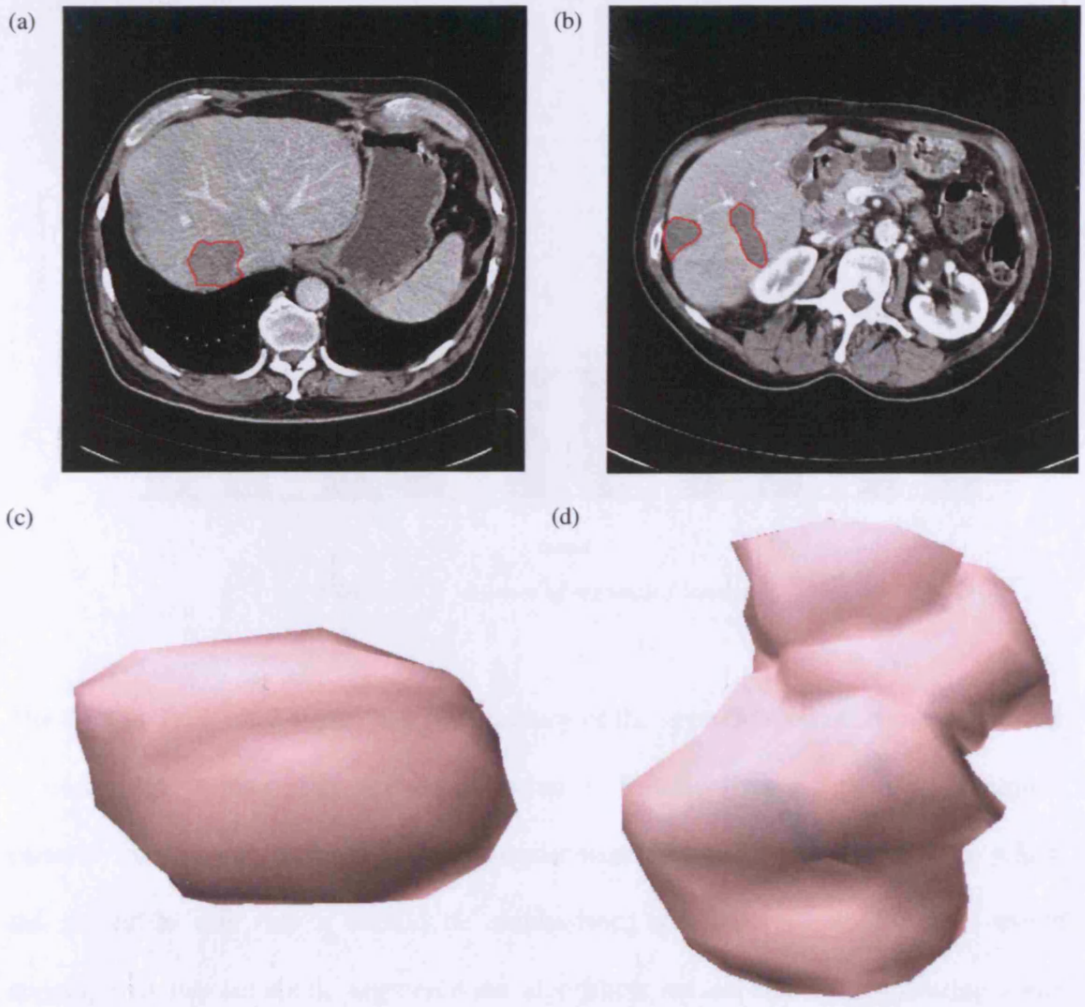


Figure 5.32: *Lesion segmentation. Figures (a) and (b) show the results of 2D segmentation, with two separate contours having been initialised in figure (b). Images (c) and (d) show rendering of the abnormalities segmented using the 3D algorithm.*

Abnormality ID	Manual volume	2D automatic volume	3D automatic volume	2D overlap	3D overlap
1	129568	129014	129365	0.96	0.98
2	86894	87009	87158	0.99	0.98
3	35689	35421	35525	0.98	0.99
4	82791	80126	81542	0.97	0.98
5	102156	101875	102589	0.98	0.98

Table 5.16: *Results of the segmentation of well-defined lesions*

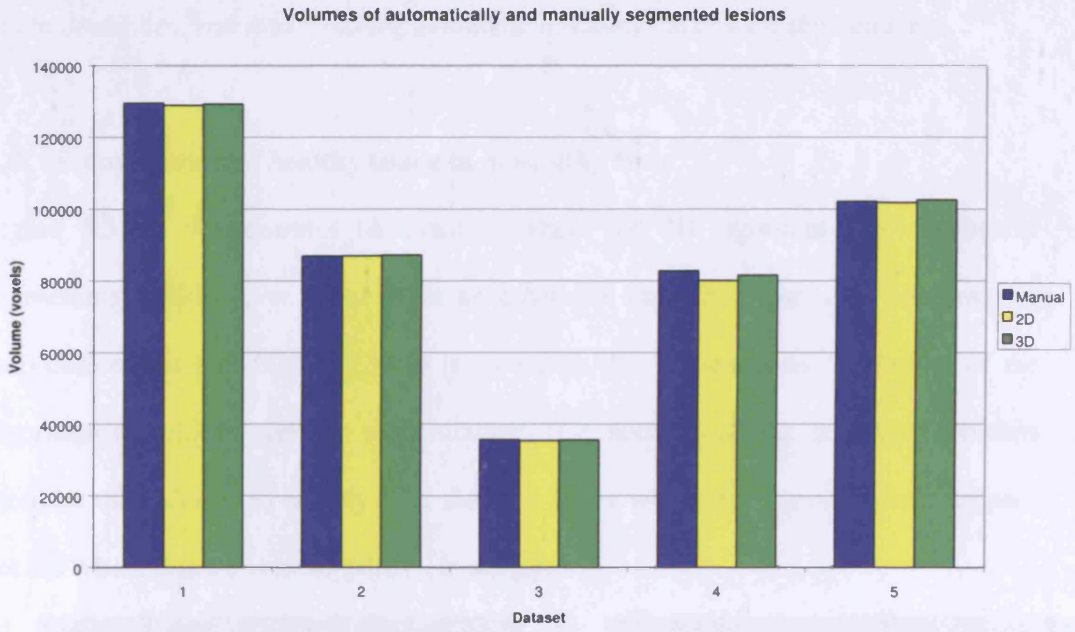


Figure 5.33: *Volumes of segmented lesions.*

The data in Table 5.12 shows that the accuracy of the segmentation of the abnormalities is quite high, higher than the segmentation of healthy liver tissue. This is almost certainly due to the much smaller and simpler structures of the abnormalities on which the algorithm was run. It should be emphasised, however, that while these results suggest that the automatic segmentation algorithms are capable of segmenting some forms of lesions, liver abnormalities vary greatly in their appearance, size, distinction, location and distribution, and a more in-depth study would need to be carried out to assess in which situations the automatic segmentation algorithms can be used to accurately quantify the extent of disease.

It is also important to note that while the automatic segmentation of the lesions was initiated by an operator, the actual location of the lesion does not appear to have an affect on the accuracy of the segmentation. Both 2D and 3D techniques are able to segment abnormalities whether they are wholly surrounded by liver tissue or on the perimeter of the liver. Chapter 6 will discuss several possibilities of using the structure of the automatically segmented healthy liver tissue as a basis for estimating the location of abnormalities, and thus enabling automatic initialisation of their segmentation.

5.8. Segmentation of healthy tissue in unhealthy liver

Figure 5.34(a) demonstrates an example where the 2D algorithm was capable of segmenting healthy liver tissue from an otherwise unhealthy liver that contains two well-defined lesions. Figure 5.34(b) is an image that demonstrates the failure of the algorithm to achieve accurate segmentation. The model is able to segment (and thus measure the volume of) healthy liver tissue in livers where the disease is well defined; yet the where disease is more diffuse, it struggles.

(a) (b)

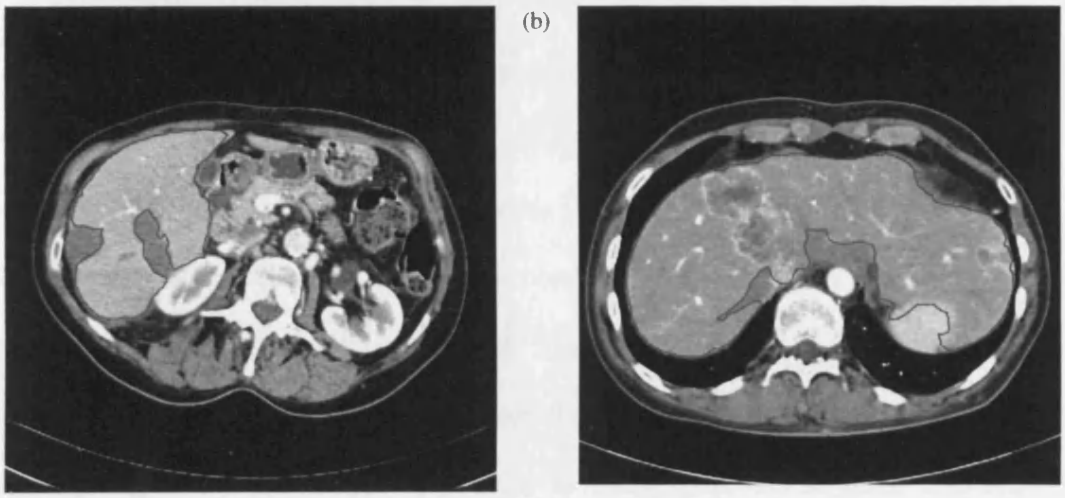


Figure 5.34: *Healthy liver tissue segmentation in an unhealthy liver. Figure (a) shows successful segmentation where well-defined lesions exist. Figure (b) shows unsuccessful segmentation.*

Figure 5.35 contains images showing views from two different angles on the results of segmenting healthy liver tissue in 3D, from an otherwise unhealthy liver. The obvious

indentation indicates the location of a lesion at the edge of the liver. Chapter 6 discusses further possibilities for making use of such a segmentation.

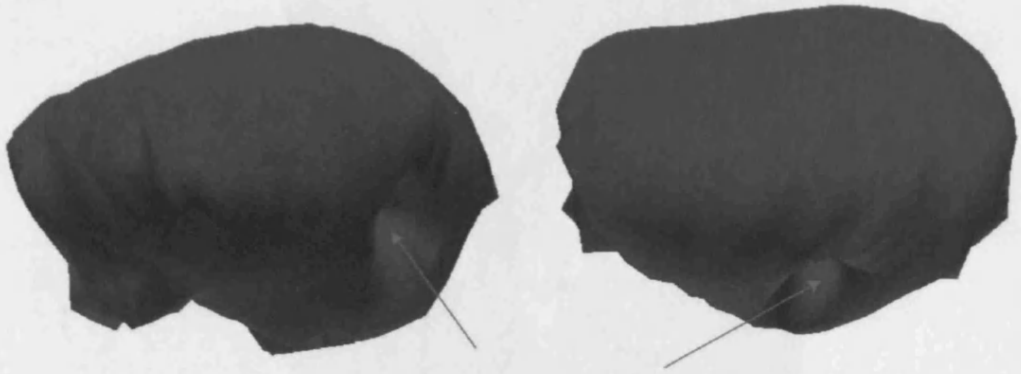


Figure 5.35: *3D Segmentation of healthy tissue from an abnormal liver. Two images of the same liver, viewed from different angles. The obvious indentation indicated by the red arrows marks the location of a lesion that has not been included in the segmentation*

5.9. Other abdominal organs

The nature of the developed algorithms means that they may be used, in theory, to segment other abdominal structures from CT scans. Figure 5.36 shows images that demonstrate both the 2D contour's and 3D surface's ability to segment the kidney. In this scenario careful attention must be paid to the parameters of the models, as the histogram of voxel values of the interior of the kidney has a different shape to that of the liver. It is also possible to segment gas filled structures such as the lungs and the colon, using both the 2D contour and 3D surface. However, the segmentation of these structures can be achieved easily, rapidly, and possibly more accurately using low level techniques (Hu, Hoffman and Reinhardt, 2001; Nappi, Dachman *et al.*, 2002) such as thresholding and 3D region growing; the gas that fills such organs has the lowest Hounsfield value and appears perfectly black in CT images. In the same manner, segmentation of arterial trees is also possible with the models developed in this work (in fact, their inflationary nature is well suited to the branching structure of the arterial tree) yet the use of contrast agent in a patient's blood during a scan allows the arteries to

appear bright white in CT images; again this is relatively simple to segment with lower level techniques.

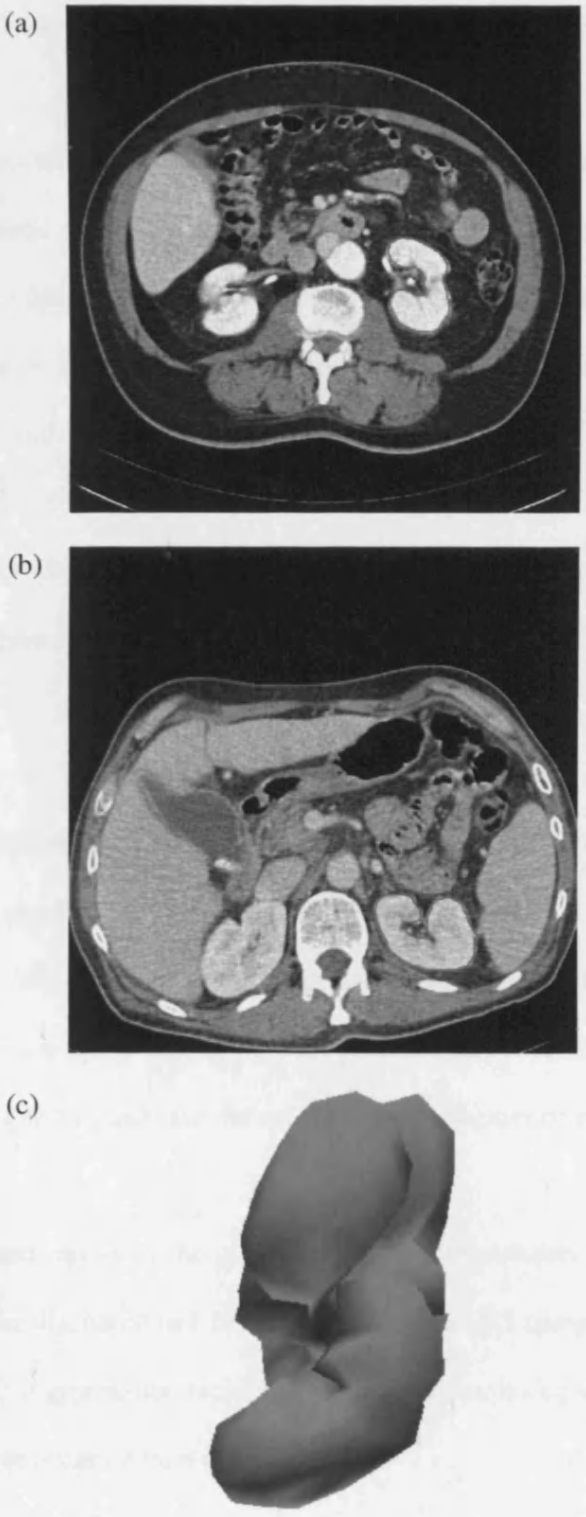


Figure 5.36: *Kidney segmentation. (a) and (b) show examples using the 2D algorithm, (c) is 3D segmentation.*

5.10. Summary

This chapter has presented the procedures and results of the automatic segmentation of liver tissue from CT datasets.

Section 5.1 introduced the methods and techniques that were used to validate the accuracy of automatic segmentation. In the absence of both a realistic phantom and surgical liver data, comparison is made with liver datasets segmented manually by two independent operators. Three measures of validation were described, volume estimation, volumetric overlap and contour/surface distance.

Section 5.2 briefly discussed the pre-processing carried out on all datasets, and explained which phase of contrast enhancement provided preferable segmentation conditions.

Section 5.3 presented results of the efforts made to segment liver datasets as a series of 2D images. Initial results using low-level segmentation techniques were rejected due to lack of robustness. The results of using the active contour model, detailed in Chapter 3, to segment the liver are presented, along with evidence that several of the novel aspects of the developed algorithm increase the accuracy of the segmentation.

Section 5.4 presented results of the segmentation of liver datasets in full 3D, using the active surface model discussed in Chapter 4, and Section 5.5 compared the accuracy of both the 2D and 3D segmentation techniques, with the conclusion that the 2D technique may be slightly more accurate than the 3D.

Section 5.6 shows preliminary results that demonstrate the ability of both the 2D contour and 3D surface to segment abnormal liver tissue.

The chapter concludes with Section 5.7, demonstrating that segmentation using the developed algorithms is not merely confined to liver tissue.

From these results, it is possible to conclude that the developed algorithms achieve good accuracy in liver segmentation. Both the 2D and 3D techniques estimate liver volumes that correlate very closely with those obtained from manual segmentation; the volumetric overlap values are within 5.6% of those obtained from manual-manual overlap, and contour/surface distance measures are within 2 voxels of the mean difference between two manual segmentations.

The novel work carried out in the techniques' development plays a strong role in ensuring this accuracy. In the following chapter, the relevance of the results is discussed, along with comparison to existing and previous work in the field. In addition, the possibilities for incorporating this work into future projects concerning the liver are presented.

5.11. Novel contributions

- Liver segmentation in 3D. To the author's knowledge, this is the first work that presents robust, accurate and repeatable 3D liver segmentation from CT data.
- Liver segmentation using no prior reference model, by initialising the procedures at any location within the organ.
- Segmentation of unhealthy liver, and potential use for segmentation of other organs.

6. Conclusions

This chapter summarises the work described in the thesis emphasising the main contributions to the fields of deformable models and medical image segmentation. It then gives some concluding remarks and discusses some areas for future research.

6.1. Contributions

The main goal of the work presented in this thesis was to develop a technique to automatically segment liver from abdominal CT scans. In pursuit of this goal, both 2D and 3D segmentation techniques were developed; while some of the basic methodology was developed from work carried out by other researchers, several novel techniques were implemented and used effectively to achieve the segmentation results. This section summarises the novel contribution of the work and discusses its significance.

Chapters 3 and 4 explained in detail the development of the inflationary, parametric models that were used to segment the liver. Parametric active contours and surfaces have been used extensively in image processing since their introduction, yet this work demonstrates that there is still scope for them to be further developed to achieve more accurate results, and so that they may be used in further applications.

Central to the many aspects of the novel technical contribution of this thesis is the concept of the utilisation of contour or surface *curvature* to locally modify the behaviour of the model. While curvature in itself has been researched at length previously, both in its definition and utilisation, in this work it is used to directly affect and improve the final segmentation results achieved using parametric models. To the author's current knowledge, no other work on using curvature to affect segmentation

with active contour or surface models in such a manner, either in 2D or in 3D, has been previously published.

Chapter 3 introduced an inflationary active contour model that overcomes the traditional flaws of similar techniques by reparameterising the elements that constitute the model at certain iterations of its movement. This concept was developed initially by McInerney and Terzopoulos (2000) in their work on T-snakes, yet this thesis develops the concept further by reparameterising the contour at greater levels of detail where it is most required, in areas where the curvature and complexity of the contour is at its highest. Functionality is further improved by ensuring lower resolution in areas of lower curvature decreasing the likelihood of the contour leaking through weak edges. While the technique was developed specifically for the liver, preliminary tests suggest that it is suitable for use in a wide variety of segmentation situations.

Chapter 4 introduced the active surface model to achieve full 3D segmentation. Several concepts used in work by other researchers were combined to create a novel and unique segmentation tool. The use of surface curvature information to increase the local resolution of the surface has, at the time of writing, not previously been used in conjunction with active surface models. Further novel contribution has been achieved in the field of using regional information to affect the movement of the vertices of the model. While active region models were proposed some time ago (Ivins and Porrell, 1994) this work used basic region statistics to directly influence the effect of external, image-based, forces. As a result, the vertices of the model do not get 'caught' on insignificant edges, yet the effect of prominent edges is amplified.

Chapter 5 demonstrated the effectiveness of both 2D and 3D techniques to segment the liver. The major novel contribution to the field of liver segmentation is the robust and repeatable segmentation of liver in full 3D. A further aspect of novel contribution is the ability of the segmentation algorithm to be initialised *at any location within the liver*, so long as that location represents an area of parenchymal liver tissue. This avoids several of the major issues that affected early attempts to segment the liver (see Chapter 2), and improves upon the steps required for segmentation with more modern techniques, such as level set theory (Pan and Dawant, 2001). Furthermore, segmentation is achieved using *no prior reference model*, which enables both healthy and unhealthy tissue to be segmented.

6.2. Achievements and discussion

As stated above, the main goal of this project was to develop a technique to automatically segment liver from abdominal CT scans. From the results presented in Chapter 5, it is possible to conclude that this goal of accurate segmentation of the liver was successfully accomplished. The intra-class correlation results suggest that the variation in the estimation of liver volume from segmentation by two manual operators, the automatic 2D algorithm, and the automatic 3D algorithm, is solely due to the natural variation of liver volume between patients. Direct comparison of the results of automatic algorithms with the results of manual segmentation, carried out using volumetric overlap and contour distance measures, suggest that the accuracy of both automatic techniques is high – mean overlap error did not exceed 5.6%, and mean surface distance was less than 2 voxels, when compared to manual assessment.

Tables 6.1 and 6.2 compare the automatic liver segmentation results achieved by previous researchers with the 2D and 3D results presented in this thesis, which are

highlighted in bold type. Several researchers present average distance measurements (measured in mm) as measures of segmentation. For this work measurement of accuracy in mm was not considered suitable (as discussed in Chapter 5), as the algorithms treated each dataset independent of its voxel dimensions, and thus taking the mean of the actual distance conversions would produce results skewed to give greater weight to datasets with larger voxel dimensions.

Author	**Evans**	**Evans**	Seo & Park	Liu & Zhao	Bae *et al.*	Lim *et al.*	Pan & Dawant
Number of Datasets	**18**	**18**	12	20	4	6	5
Measurement method	***Area***	***Overlap***	*Area*	*Area*	*Area*	*Area*	*Overlap*
Result	**5.24%**	**0.93**	8.28%	5.30%	6.50%	3.00%	0.95

Table 6.1: *Comparison of segmentation accuracy in 2D. The units for the area results are percentage error compared to manual segmentation. The overlap is measured where a value of one equals perfect overlap.*

Author	**Evans**	**Evans**	Heimann *et al.*	Lamecker *et al.*	Pan & Dawant
Number of Datasets	**18**	**18**	59	33	5
Measurement method	***Volume***	***Overlap***	*Volume*	*Volume*	*Overlap*
Result	**4.24%**	**0.92**	11%*	7%	0.92

* *median of volumetric errors*

Table 6.2: *Comparison of segmentation accuracy in 3D. The units for the volume result are percentage error compared to manual segmentation, the units for distance measurement are mm. The overlap is measured where a value of one equals perfect overlap.*

The figures presented in the table show that the volumetric errors for both 2D and 3D techniques presented in this thesis are lower than all but one of those obtained by other researchers, the exception being Lim *et al.*. (2006). However, they present results for a small number of datasets (six) and their technique is very slow (taking 1-3 minutes to segment a single slice, compared to < 5 seconds for this work's 2D technique), due to it relying on several thresholding and morphological processing stages. Of the previously presented 2D techniques, the most accurate appears to be Liu and Zhao's (2005)

gradient vector flow technique, the volumetric error being almost identical to the 2D result from this work. Table 6.1 shows the overlap result obtained in this thesis to be 0.02 less than that obtained by Pan & Dawant's (2005) level set technique, which is heavily constrained by marking of external boundaries, and only five datasets are used for validation. From the results presented in the tables it can be seen that the 2D work equals or improves on previously published research.

While only Pan and Dawant have previously made efforts to use data-driven techniques segment the liver in 3D, Heimann *el al.* (2006) and Lamecker *et al.* (2004) use active shape models for segmentation purposes. Table 6.2 shows the accuracy of the 3D active surface in this model to be greater than the active shape model approaches, though it should be noted these results used a larger quantity of datasets, and could be considered more robust. Volume overlap comparison gave an identical result to that of Pan & Dawant (2005), though their technique only treats five datasets.

The results in Table 6.2 raise the question of whether data-driven segmentation techniques (such as the inflationary models presented by this research and that of Pan and Dawant), or techniques relying heavily on prior shape knowledge (such as active shape models) are better suited to the task of liver segmentation. While it is true that assumptions for shape should not be made in liver pathology (as the shape and structure of the liver can be drastically different in unhealthy organs) the fact is that data-driven segmentation is very difficult *without* such prior knowledge. Pan and Dawant (2005) find that the "2D version of [their] algorithm leads to better results than the 3D version...[due to]...the rapid change in liver shape from one slice to the next...where virtually no black boundary between the [liver and heart] exists". As discussed above in Chapter 5, this lack of boundary is caused by the passage of the inferior vena cava

through the liver on its way to the heart, and prior knowledge (in the form of operator interaction) was required to prevent leakage for the 3D algorithm presented in this thesis.

Therefore, the results suggest that prior knowledge *is* a requirement for accurate 3D segmentation, yet the fact remains that the data-driven segmentation results presented in this work give greater accuracy than the active shape model research carried out by other groups; indeed, closer analysis of the results presented by Heimann *et al.* shows that, while the median volumetric error was 11%, the error ranged from 9% to 25% - not a single one of their 59 sample datasets was segmented as successfully (in volumetric terms) as the mean accuracy of the 18 datasets segmented in 3D in this work. A possible reason for this has been mentioned briefly above; in that the shape of the liver can vary greatly, and thus even with a large training set there may be certain livers that the model will not be able to fit.

Thus, from comparison of the results in this thesis and those of previous work, it can be concluded that prior knowledge is required for 3D segmentation, though techniques that retain some form of shape independence achieve better results. Furthermore, it can also be concluded that 2D techniques appear to be better suited to liver segmentation, due to accuracy of results and the lack of requirement for prior knowledge. This conclusion would appear to be counter-intuitive, yet it agrees with the conclusions of the only other published effort to compare 2D and 3D liver segmentation (Pan and Dawant, 2005).

Two particularly interesting results relate to the comparison between 2D and 3D results. The results in Chapter 5 show that the volumes segmented by the 2D algorithm tend to be of larger value than that those segmented in 3D. One explanation for this fact is the

2D algorithm's capability to merge at points of self-intersection, possibly including non-liver areas within the segmented region. Figure 6.1 shows an example of how this occurs. It contains three images showing the 2D contour in the final few iterations of segmenting a liver slice. While it has wrapped around the non-liver tissue, in the final image it has intersected with itself, merged and altered to *include* the non-liver tissue.

While Figure 6.1 demonstrates a rather extreme example, in several situations many small areas of vascular structure on the liver perimeter are included by the 2D algorithm, yet not included by 3D or manual segmentation. These errors accumulate to leave a disproportionately large difference in measured volume, especially compared with the smaller differences in volumetric overlap and mean contour/surface distance. This situation does not arise during 3D segmentation. Although this is partially due to the fact that the 3D surface cannot merge with itself, it is further due to the inherent nature of 3D segmentation, which does not treat each slice as an individual case.

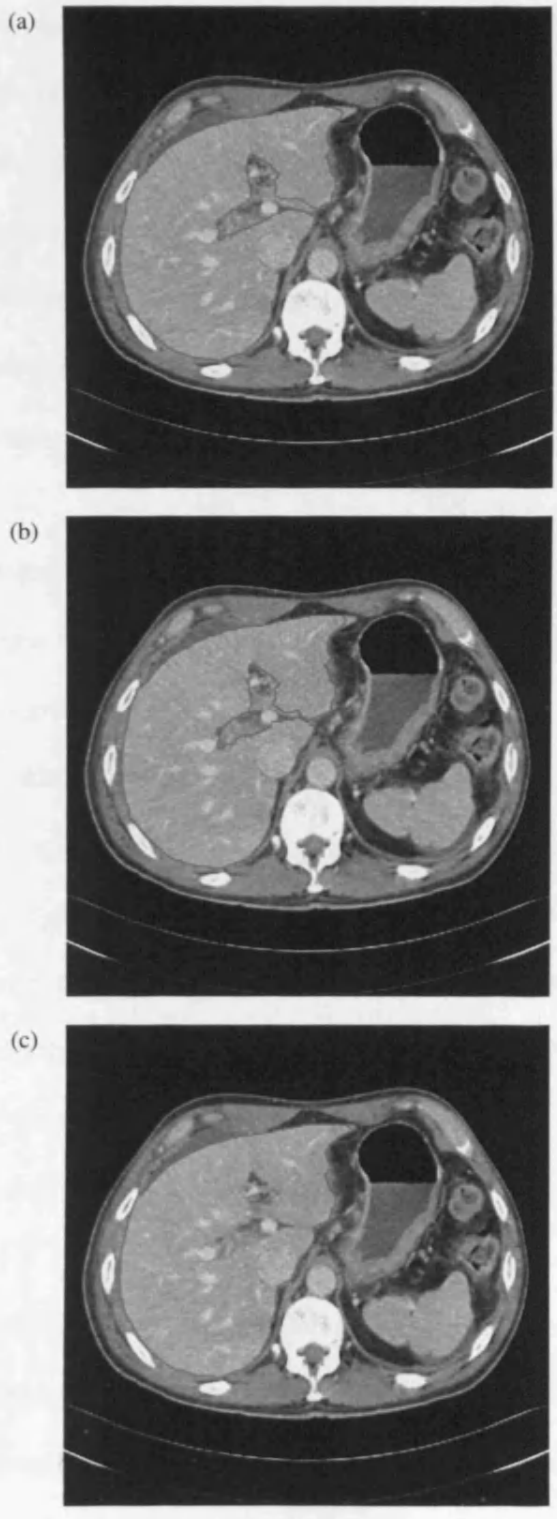


Figure 6.1: *Contour merging increases volume. Figure (a), (b) and (c) show the contour expanding. Figure (c) shows the result of the contour's self-intersection.*

The issue of over-segmentation is also relevant when discussing the unexpected result of relatively low mean distance measurements combined with relatively high differences in volumes, in certain cases. This can be explained by the fact that an error in a small number of points can have a large effect on the value of the volume, but a much smaller effect on the mean pixel distance for the entire liver. This explains how, as highlighted in Chapter 5 above, in dataset 7 there is a 13% difference between the 2D automatic segmentation volume and the mean of the two manual segmentation volumes, yet only 3.72 average pixel difference over the entire organ.

While appearing to be less accurate in terms of overall volume estimation, Figures 5.24, and 5.25 suggest that the 2D algorithm is slightly *more* accurate than the 3D algorithm (although not by a great deal). This concurs with the previous effort at comparing 2D and 3D liver segmentation (Pan and Dawant, 2001), yet the difference between the results in this work is of a lesser magnitude. One possible explanation for the phenomenon is that the 2D algorithm is, in general, more accurate than the 3D algorithm, yet more susceptible to the type of large-scale error as demonstrated in Figure 6.1. Importantly the accuracy of segmentation for both techniques appears to be similar or greater than that presented by other researchers using different techniques, such as level-sets and registration (see Chapter 2).

One important question that is of particular relevance clinically, is what is the required (or accepted) accuracy of the automatic segmentation if it is to be used clinically? This question is not answered easily, as a search of the literature reveals that there have been no clinical studies investigating this matter, and thus there is potential in this area for research.

One of the most important aspects of the developed algorithms is their ability to segment the liver with *no prior reference* to any shape model or other structures of the abdomen, which can be seen as a significant breakthrough from previously published research. It is of great benefit when dealing with an organ such as the liver as its inter-patient shape is irregular, as discussed in Section 1. It is of further benefit when dealing with abnormal livers, as it means both 2D and 3D algorithms can be used not only to segment the abnormal tissue, but to segment the healthy tissue around it, as demonstrated in Figures 5.25 and 5.26.

6.3. Limitations and future work

In addition to succeeding to reach the goal of automatically segmenting the liver, the work developed during this thesis can be both improved and, more promisingly, combined with other aspects of medical image processing to further research into the diagnosis and treatment of liver disease. This section discusses both elements of such future work.

One of the most prominent limitations of the work is that neither the 2D algorithm nor the 3D algorithm is 'fully' automatic. Both require a manual operator to select an area of parenchymal liver tissue to initiate the segmentation algorithm, and further manual intervention is required in 3D to prevent the surface from inflating along in the inferior vena cava. In addition, an obvious area of future work is to implement a 3D mesh merging scheme, in a manner similar to Lachaud and Montanvert (1999), so that the surface may merge with itself at locations of self-intersection. However it is not clear whether this would improve upon the accuracy of healthy liver segmentation, and a study would need to be carried out to test this.

One limitation that affects both 2D and 3D algorithms is the setting of the parameters that control the movement of the models. As Chapter 5 states, the parameters for the models were set empirically. One interesting further study would be to implement some form of parameter optimisation system, whereby an optimisation algorithm compares the results of segmentation obtained using different parameter sets, before outputting the 'optimal' set. If this system were implemented carefully, it is highly likely that it would improve on the presented segmentation results.

The nature of the developed segmentation algorithms, and their inherent lack of reliance on any prior model of shape, makes them particularly suitable to be used *in conjunction* with other image processing techniques that rely heavily on prior shape knowledge. This can be used both to improve upon the accuracy of the segmentation, and to enable automatic identification of abnormal tissue.

For example, the segmentation algorithm could be used in conjunction with a statistical shape model of the liver. In terms of pure segmentation accuracy, the constraints of the model could be used to reduce or eliminate any need for manual intervention e.g. there would be no need for an operator to mark the entry and exit points of the vena cava, as a shape model could automatically constrain the active surface model in these areas. Furthermore, the segmentation results obtained using the active models could be compared with the shape of a statistical model – any regions where the segmentation and model disagreed strongly could be highlighted as potential abnormalities. For example, comparison with a statistical shape model may automatically highlight the obvious indentation in the results of the 3D liver segmentation shown in Figure 5.29.

6.4. Summary

This thesis has presented work on the automatic segmentation of liver from Computerised Tomography, using both 2D and 3D techniques.

The main novel contribution has been the successful implementation of accurate, robust and repeatable automatic segmentation of liver in full 3D which, at this date and to the author's knowledge, has not been previously published. The most recent comprehensive previous liver segmentation research was carried out by Liu, Zhao and Kijewski (2005), which dealt solely with 2D techniques, which are inherently limited as is discussed above.

The thesis has also presented novel contribution in technical areas. Established 2D active contour and 3D active surface algorithms have been adapted for use with differing segmentation tasks, and Chapter 5 shows examples of how both algorithms can be used to segment other abdominal organs as well as the liver. The most important technical novel contribution is the use of curvature to affect the resolution at which both the 2D contour and 3D surface are reparameterised, and the local flexibility, thus allowing more accurate segmentation in areas of high curvature, and increasing the resistance of contour/surface leaking into unwanted areas/volumes in areas of lower curvature.

It is hoped that this step into full 3D liver segmentation may be used as a basis for future research, especially in the fields of the detection and diagnosis of areas of abnormal liver; as a result, it will contribute to efforts to facilitate the diagnosis and treatment of liver disease.

References

Adalsteinsson, D. and Sethian, J. A., (1995) "A fast level set method for propagating interfaces," *Journal of Computational Physics*, vol. 118, no. 2, pp. 269-277.

Ahlberg, J., (1996) Active contours in three dimensions. Report No.: LiTH-ISY-EX-1708. Linkoping university, Sweden.

Alexander, D. and Buxton, B., (1997) "Implementation and analysis of an optimized rainfalling watershed algorithm," *Proceedings of the British Machine Vision Conference.*

Bae, K. T., Giger, M. L., Chen, C. T., and Kahn, C. E., (1993) "Automatic segmentation of liver structure in CT images," *Medical Physics*, vol. 20, no. 1, pp. 71-78.

Bland, J., Altman, D., (1986) "Statistical methods for assessing agreement between two methods of clinical measurement" *Lancet*, vol. 1, pp. 307-310.

Bezdek, J. C., Hall, L. O. and Clarke, L.P., (1993) " Review of MR image segmentation techniques using pattern recognition" *Medical Physics*, vol. 20. no. 4. pp 1033-1048.

Bulpitt, A. J. and Efford, N. D., (1996) "An efficient 3D deformable model with a self-optimising mesh," *Image and Vision Computing*, vol. 14, no. 8, pp. 573-580.

Chen, H. and Bishop, J., (1997) "Delaunay Triangulation for Curved Surfaces" *Proceedings 6th International Meshing Roundtable* pp. 115-127.

Chew, L., (1993) "Guaranteed-quality mesh generation for curved surfaces" *Proceedings 9th Annual Symposium on Computational Geometry* pp. 274-280.

Cohen, L. D., (1991) "On active contour models and balloons," *CVGIP-Image Understanding*, vol. 53, no. 2, pp. 211-218.

Cohen, L., and Cohen, I., (1993) "Finite-element methods for active contour models and balloons for 2-D and 3-D images" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 15, no. 11, pp. 1131-1147.

Cootes, T. F., Cooper, D., Taylor C.J., and Graham, J., (1992) "A Trainable Method of Parametric Shape Description." Image and Vision Computing, vol. 10, pp.289-294.

Coxeter, H. S. M., (2001) *Regular polytopes* Dover Publications.

Ericson, C., (2005) *Real-time Collision Detection*, Morgan Kaufmann.

Gao, L. M., Heath, D. G., Kuszyk, B. S., and Fishman, E. K., (1996) "Automatic Liver Segmentation technique for three-dimensional visualization of CT data," *Radiology*, vol. 201, no. 2, pp. 359-364.

Gerig, G., Jomier, M. and Chakos, M., (2001) "Valmet: A new validation tool for assessing and improving 3D object segmentation" *Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp.516-523.

Giraldi, G., Strauss, E., and Oliveira, A., (2003) "Dual-T-Snakes model for medical imaging segmentation," *Pattern Recognition Letters*, vol. 24, no. 7, pp. 993-1003.

Hayslett, H. T., (1974) *Statistics made simple*, Richard Clay (The Chaucer Press)

Hermoye, L., Laamari-Azjal, I., Cao, Z., Annet, L., Lerut, J., Dawant, B.M., Van Beers, B.E., (2005) "Liver segmentation in living liver transplant donors: comparison of semiautomatic and manual methods" *Radiology*, vol 234, no. 1, pp. 171-178.

Hong, J. S., Kaneko, T., Sekiguchi, R., and Park, K. H., (2001) "Automatic liver tumour detection from CT," *IEICE Transactions on Information and Systems*, vol. E84D, no. 6, pp. 741-748.

Hu, S. Y., Hoffman, E. A., and Reinhardt, J. M., (2001) "Automatic lung segmentation for accurate quantitation of volumetric X-ray CT images," *IEEE Transactions on Medical Imaging*, vol. 20, no. 6, pp. 490-498.

Huda, W., Scalzetti, E., Roskopf., M., "Effective doses to patients undergoing thoracic computed tomography examinations" *Medical Physics*, vol. 27, no. 5, pp. 838-844.

Ivins, J. and Porrill, J., (1994) "Active Region Models for Segmenting Medical Images" *Proceedings 1st IEEE International Conference on Image Processing*, vol. 2 pp. 227-231.

Jain, A., (1988) *Fundamentals of Digital Image Processing*, Prentice Hall.

Jang, D., (1997) "Contour detection of hippocampus using dynamic contour model and region growing," *Annual Conference IEEE Engineering in Medicine and Biology Society*, vol. 19 pp. 763-766.

Kass, M., Witkin, A., and Terzopoulos, D., (1987) "Snakes - active contour models" *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321-331.

Kim, J. B. and Kim, H. J., (2003) "Multiresolution-based watersheds for efficient image segmentation" *Pattern Recognition Letters*, vol. 24, no. 1-3, pp. 473-488.

Lachaud, J.-O. and Montanvert, A., (1999) "Deformable Meshes with Automated Topology Changes for Coarse-to-fine 3D Surface Extraction." *Medical Image Analysis*, vol. 3, no. 2, pp.187-207.

Lamecker, H., Lange, T. and Seebaβ, (2002) "A Statistical Shape Model for the Liver" *Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 421-427.

Lawson, C., (1972) "Transforming Triangulations" *Discrete Math.*, vol. 3, pp 365-372.

Li X. Q., Zhoa. W., and Cheng, H. D., (1995) "Fuzzy entropy threshold approach to breast cancer detection," *Information Sciences-Applications*, vol. 4, no. 1, pp. 49-56.

Lim, S-J., Jeong, Y-Y., Ho, Y-S., (2005) "Segmentation of the Liver Using the Deformable Contour Method on CT Images" *Proc. PCM 2005*, Part I, Lecture Notes in Computer Science 3767, pp. 570-581.

Liu, F., Zhao, B., Kijewski, P.K., (2005) "Liver segmentation for CT images using GVF snake" *Medical Physics*, vol. 32, no.12, pp 3699-3706.

Mallat, S., (1999) *A Wavelet Tour of Signal Processing*. Academic Press.

Marr, D. and Hildreth, E.C., (1980) "Theory of Edge Detection" *Proceedings of the Royal Society, London B*, vol 207, pp 187-217.

McCool, M., (1999) "Anisotropic diffusion for Monte Carlo noise reduction" *ACM Transactions on Graphics*, vol. 18, no. 2, pp. 171-194.

McInerney, T. and Terzopoulos, D., (1996) "Deformable models in medical image analysis: a survey" *Medical Image Analysis*, vol. 1, no. 2, pp. 91-108.

McInerney, T. and Terzopoulos, D., (1999) "Topology adaptive deformable surfaces for medical image volume segmentation" *IEEE Transactions on Medical Imaging*, vol. 18, no. 10, pp. 840-850.

McInerney, T. and Terzopoulos, D., (2000) "T-snakes: Topology adaptive snakes," *Medical Image Analysis*, vol. 4, no. 2, pp. 73-91.

Möller, T., (1997) "A fast triangle-triangle intersection test" *Journal of Graphics Tools* vol. 2, no. 2, pp. 25-30.

Munoz, X., Freixenet, J., Cufi, X., and Marti, J., (2003) "Strategies for image segmentation combining region and boundary information," *Pattern Recognition Letters*, vol. 24, no. 1-3, pp. 375-392.

Museth, K., Breen, D., Whitaker, R., and Barr, A., (2002) "Level Set Surface Editing Operators," *Proceeding of SIGGRAPH*, pp. 330-338.

Nappi, J., Dachman, A. H., MacEneaney, P., and Yoshida, H., (2002) "Automated knowledge-guided segmentation of colonic walls for computerized detection of polyps in CT colonography," *Journal of Computer Assisted Tomography*, vol. 26, no. 4, pp. 493-504.

Ney, D. R., Fishman, E. K., Magid, D., and Drebin, R. A., (1990) "Volumetric rendering of computed-tomography data - principles and techniques," *IEEE Computer Graphics and Applications*, vol. 10, no. 2, pp. 24-32.

Osher, S. and Sethian, J. A., (1988) "Fronts propagating with curvature-dependent speed - algorithms based on hamilton-jacobi formulations," *Journal of Computational Physics*, vol. 79, no. 1, pp. 12-49.

Owen, S., (1998) "A Survey of Unstructured Mesh Generation Technology" *Proc. 7th Int. Meshing Roundtable*, pp. 239-267

Paloc, C., (2003) *Adaptive Deformable Model (allowing Topological Modifications) for Surgical Simulation* PhD thesis, Bioengineering Department, Imperial College London.

Pan, S., and Dawant, B.M., (2001) "Automatic 3D segmentation of the liver from abdominal CT images: a level set approach" *Proc. SPIE* Vol. 4322, p. 128-138.

Park, J.-Y., McInerney, T., Terzopoulos, D. and Kim, M.-H., (2001) "A non self-intersecting adaptive deformable surface for complex boundary extraction from volumetric images" *Computers and Graphics* vol. 25, pp.421-440.

Pavlidis, T. and Liow, Y. T., (1990) "Integrating region growing and edge-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 3, pp. 225-233.

Pizer, S. M., Fletcher, P. T., Joshi, S., Thall, A., Chen, J. Z., Fridman Y., Fritsch, D. S., Gash, A. G., Glotzer, J. M., Jiroutek, M. R., Lu, C. Muller, K. E., Tracton, G., Yushkevich, P. and Chaney, E. L., (2003) "Deformable *M-Reps* for 3D Medical Image Segmentation" *International Journal of Computer Vision*, vol. 55. no. 2/3, pp. 85-106.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., (2002) *Numerical Recipes in C++: The Art of Scientific Computing*, Cambridge University Press.

Qatarneh, S. M., Noz, M. E., Hyodynmaa, S., Maguire, G. Q., Kramer, E. L., and Crafoord, J., (2003) "Evaluation of a segmentation procedure to delineate organs

for use in construction of a radiation therapy planning atlas," *International Journal of Medical Informatics*, vol. 69, no. 1, pp. 39-55.

Reed, T. and Hans du Buf, J. M., (1993) "A Review of Recent Texture Segmentation and Feature Extraction Techniques" *CVGIP: Image Understanding*, vol 57, no. 3, pp 359-372.

Rushing, J. A., Ranganath, H., Hinke, T. H., and Graves, S. J., (2002) "Image segmentation using association rule features," *IEEE Transactions on Image Processing*, vol. 11, no. 5, pp. 558-567.

Russ, J., (1999) *The Image Processing Handbook*, 3rd Edition ed. CRC Press in cooperation with IEEE Press.

Sahani, D., Kalva, S., (2004) "Imaging the Liver" *The Oncologist*, vol. 9, no. 4, pp. 385-397.

Sethian, J. A., (1996) *Level Set Methods*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press.

Seo, K-S., Park, J-A., (2005) "Improved Liver Segmentation of a Contrast Enhanced CT Image" (2005) *Proc. PCM 2005*, Part I, Lecture Notes in Computer Science 3767, pp. 899-909.

Shimizu, A., Hitosugi, J., Nakagawa, J., Tamura, M., and Kobotake, H., (2003) "Development of computer-aided diagnosis system for 3D multi-detector row CT images or livers," *Internation Congress Series*, vol. 1256 pp. 1055-1062.

Shrout, P. E., and Fleiss, J. L., (1979) "Intraclass correlations: Uses in assessing rater reliability" *Psychological Bulletin*, vol. 86, no. 2, pp. 420-428.

Soler, L., Delingette, H., Malandain, G., Montagnat, J., Ayache, N., Koehl, C., Dourthe, O., Malassagne, B., Smith, M., Mutter, D., and Marescaux, J., (2001) "Fully automatic anatomical, pathological, and functional segmentation from ct scans for hepatic surgery" *Computer Aided Surgery*, vol. 6, no. 3, pp. 131-142.

Sonka, M., Fitzpatrick, J.M., (2000) *Handbook of Medical Imaging, Vol. 2 Medical Image Processing and Analysis*. SPIE Press.

Soyer, P., (1991) "Preoperative segmentatal localization of hepatic metastases: utility of three-dimensional CT during arterial portography," *Radiology*, vol. 180 pp. 653-658.

Spann, M. and Wilson, R., (1985) "A quad-tree approach to image segmentation which combines statistical and spatial information," *Pattern Recognition*, vol. 18, no. 3-4, pp. 257-269.

Staib, L. H. and Duncan, J. S., (1992) "Boundary finding with parametrically deformable models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1061-1075.

Steudel, A. and Glesner, M., (1999) "Fuzzy segmented image coding using orthonormal bases and derivative chain coding," *Pattern Recognition*, vol. 32, no. 11, pp. 1827-1841.

Stokely, E. M. and Wu, S. Y., (1992) "Surface parameterization and curvature measurement of arbitrary 3-d objects - 5 practical methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 8, pp. 833-840.

Tobias, O. J. and Seara, R., (2002) "Image segmentation by histogram thresholding using fuzzy sets," *IEEE Transactions on Image Processing*, vol. 11, no. 12, pp. 1457-1465.

Uebersax, J., (2006) "Statistical methods for rater agreement" Retrieved September 3, 2006, from http://ourworld.compuserve.com/homepages/jsuebersax/agree.htm

van Ginneken, B. and ter Haar Romeny, B. M. T., (1999) "Automatic segmentation of lung fields in chest radiographs," *Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI),* , vol. 1679 pp. 184-191.

Volino, P., and Magnenat Thalmann, N., (2000) "Accurate Collision Response on Polygonal Meshes" *Proceedings of the Computer Animation* pp. 154.

Williams, D., and Shah, M., (1992) " A fast algorithm for active contours and curvature estimation" *Computer Vision, Graphics, and Image Processing: Image Understanding,* vol 55 pp.14-26.

Woodhouse, C., Ney, D., Sitzmann, J., and Fishman, E., (1994) "Spiral computed tomography arterial portography with three-dimensional volumetric rendering for oncolological planning; a retrospective analysis," *Investigative Radiology*, vol. 29 pp. 1031-1037.

Xiaohan, Y., Yla-Jaaski, J., Huttunen, O., Vehkomaki, T., Sipild, O., Katila, T., (1992) "Image segmentation combining region growing and edge detection," *Proceeding of the International Conference on Pattern Recognition*, vol. C pp. 481-484.

Xu, C. Y. and Prince, J. L., (1998) "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359-369.