# Segmentation and Classification of Magnetic Resonance Images


*Author:*
Stuart Bowyer

*Supervisor:*
Irina Voiculescu

Submitted in partial fulfillment of the degree of
*Master of Science in Computer Science*

September 2010

# Abstract

There are many effective medical systems in use which process a patient's anatomical information in some way so that the output can be used to assist doctors and surgeons. Almost all of these systems require an electronic representation of part of the patient's anatomy. This electronic representation can be considered as a patient model which is typically constructed from the data produced by a standard medical imaging technique. This document details the research carried out into possible methods for creating such a patient model, and goes on to describe the implementation of the techniques which were considered most suited for this application.

The program implemented operates on magnetic resonance images as stored in the DICOM file format. It parses each series of images into an internal three dimensional representation and then preprocesses them to limit the occurrences of imaging errors such as noise and inhomogeneities. After preprocessing, a method called split and merge is used to divide the images into regions, such that each region in the image represents a distinct tissue structure within the patient. The system then classifies each segment produced using fuzzy inference. This classification process is used so that each segment within the image is labelled with its anatomical name as would commonly be performed by a radiographer.

The program was tested on several sets of images and it was successfully able to identify kidneys and liver by applying a set of robust and generally applicable techniques. An assessment of the patient models produced by this program was made by a medical doctor and it was decided that they are both accurate and precise enough for medical use.

# Acknowledgements

I would like to thank my supervisor, Dr. Irina Voiculescu, for her input throughout the course of this project. Her advice and direction have been invaluable in keeping it well focused and ensuring that I was able to create a program as successful as I did.

I am also very grateful to Stuart Golodetz for his assistance during the implementation of my segmentation and classification program as well as for adapting his visualisation tool so that I could use it.

Finally, I would like to thank Dr. James Crossingham for giving up his time to validate my results and my family for their assistance and support throughout the course of this project.

Stuart Bowyer
September 2, 2010

# Contents

# List of Figures

# List of Algorithms

# Nomenclature

| | |
|---|---|
| ADF | Anisotropic Diffusion Filtering |
| CT | Computed Tomography |
| DICOM | Digital Imaging and Communications in Medicine |
| DICOMDIR | DICOM Directory |
| FCL | Fuzzy Control Logic (Language) |
| GUI | Graphical User Interface |
| IE | Image Element (pixel in two dimensions or voxel in three dimensions) |
| IOD | Information Object Definition |
| LoG | Laplacian of Gaussian |
| MAP | Maximum a Posteriori |
| MR | Magnetic Resonance |
| MRF | Markov Random Field |
| MRI | Magnetic Resonance Imaging |
| NFL | No Free Lunch |
| NL | Non-Local |
| RAG | Region Adjacency Graph |
| RF | Radio Frequency |
| SNR | Signal-to-Noise Ratio |
| SOP | Service-Object Pair |
| VR | Value Representation |

# Chapter 1

# Introduction

## 1.1 Motivation

There are many medical systems in use and development which process a patient's medical information so that they can either act independently or assist human medical personnel. Patient information can be suspected symptoms (as in diagnostic expert systems [114]) or vital signs such as heart rate (as in intelligent monitoring systems [4]); but an interesting set of systems are based on processing and interpreting a patient's physical anatomy. Some examples of medical systems which utilise such information are:

- Surgical Robotics - where a robotic manipulator performs or assists in surgical procedures [6];
- Surgical simulation - where a surgeon can practise a procedure on a representation of the real patient's anatomy [121];
- Preoperative planning - where a surgeon can inspect a patient's anatomy to plan a procedure [126];
- Disease monitoring - where a doctor can record the exact progress of diseased structures for assessing treatment [21].

In each of these applications the information used can be considered as an electronic, labelled patient model which represents the position, geometry and other properties of different tissue structures and organs within the patient.

Internal tissue structures tend to have very complicated and intricate geometries. This means that constructing a model by hand would require the setting of so many parameters that it would be impractical. Instead, the method employed by systems which require such models is to generate them automatically or semi-automatically from the data acquired through standard medical imaging techniques such as *Magnetic Resonance Imaging* (MRI), *Computed Tomography* (CT) and *Ultrasound*.

## 1.2 Aims

The aim of this project has been to research and implement a program which can construct an electronic patient model from raw medical images. The intention was that the models created could be used in applications such as those described previously.

The primary aim has been to process raw medical data and identify the position and geometry of known internal organs within it. Each of the applications for medical patient models is interested in a different area or tissue structure within the patient; therefore I endeavoured to develop a program which was *generally applicable* and could be used for many different functions. This meant that there was a requirement for a high degree of robustness within the process so that it could interpret different tissue structures in one patient as well as matching tissue structures in multiple patients. This second point is relevant because patient anatomy is not fixed and there can be a great deal of variation in size and shape between different patients [81, 115]. Though it was intended to be robust enough to be applicable anywhere, I decided to focus my investigation on the body/torso because there are a range of very different organs and structures and these pose an interesting set of challenges.

As the models were intended for medical use they were required to define the shape and position of each structure with a reasonable degree of accuracy and precision. The program was not designed to operate for any specific application and therefore quantitative limits could not be set. Instead, the program was developed with the intention that the models produced would be accurate and precise enough that a medically trained person would deem them suitable for some practical use.

To make the model useful for as many applications as possible it was decided that it should identify each tissue structure that it defines with a label or name which could be used by a medical system to derive some further information. Each of the labels was required to name the structure with the same anatomical name that a radiographer would use.

MRI was selected as the modality from which the data was taken to build the model. The main reason for this selection was that it is well suited to highlighting definition between soft tissue structures which have similar densities. Other modalities such as CT are less suited to these situations because their process is based purely on density values. In addition to this, MRI is popular medically because it is non-ionising and therefore causes no harm to the patient [76].

Many of the applications for medical models are designed purely to simplify techniques which are already carried out in some way. For example, surgical planning is typically carried out by a surgeon who looks through multiple two dimensional medical images and attempts to interpret them in three dimensions. This method is suitable for most procedures but it can be made much simpler with a manipulatable three dimensional electronic model. For applications such as this it would not make sense to construct a patient model if a large amount of user interaction were required because the advantage of the model would not be worth the time invested. Due to this, an objective of this

project was that the generation of the model should be fully automated and no user input should be required during 'runtime'.

All of the applications above operate on medical models which have been pre-generated. They make the assumption that the patient's anatomy will not change between the time that the image is taken and the time that the model is used. In addition to this, acquiring MR images is a relatively slow process and can take up to several hours. Both of these facts mean that the model does not need to be generated from data in real-time and it is very unlikely that any application will place a particular limit on the time which can be taken. Rather than having no requirement for processing time it was decided that there should be at least a loose requirement to ensure that models could be generated and tested practically.

To summarise, the aim of this project has been to implement a system which is able to construct anatomical patient models with the following constraints:

- The models shall identify the position and geometry of a range of tissue structures;
- The models shall have labels assigned to tissue structures identifying their anatomical name;
- The models shall be precise and accurate enough for use in some medical situations;
- The models shall be constructed from MRI data;
- The system shall be capable of operating in a robust manner on a range of tissue structures;
- The system shall operate without a requirement for any human user interaction;
- The models shall be produced within a reasonable processing time.

## 1.3 Methodology

In the literature there is one common framework which is employed for the construction of patient models such as was required for this application [2, 23, 61, 101, 117]. This framework is based around two key stages: *segmentation* (the division of image data into distinct regions, each of which defines a separate and complete tissue structure) and *classification* (the labelling of segments). To implement these processes some additional stages must be employed and the detail of these is as follows.

Each MRI machine operates with different software and manipulates the imaging data in a different format. For this reason MRI images are commonly transferred and stored in a neutral file format called *Digital Imaging and Communications in Medicine* (DICOM) [102]. By constructing a program which can interpret and use imaging data in the DICOM format it will be useful for images generated on almost any apparatus.

The first stage in developing a program to construct patient models was therefore to investigate the processes used in converting MRI data into images as well as the DICOM storage format so that this imaging data could be utilised. Once this research had been carried out I implemented a parser which converted DICOM files produced by a MRI

machine into an internal format which could be analysed and modified by the remaining steps in the model construction process.

It became apparent while researching the MRI modality that there are several sources of error inherent within it which can cause defects potentially compromising the segmentation process. I therefore decided that the MRI data should be preprocessed to remove or limit any such errors. The segmentation process is influenced by the preprocessing stage to such a great extent that the two were considered in parallel. Once a suitable set of preprocessing techniques had been identified they were implemented to operate on the MRI data.

There is a very wide range of segmentation techniques available, each of which has certain attributes which make them more suited to some applications than others. For this reason a thorough investigation of a broad selection of these techniques was carried out so that the most appropriate could be selected. After this investigation each of the techniques was compared with a range of features required for MRI data and the most applicable was selected.

The selected segmentation technique was initially implemented in its most basic form, but testing led to a series of modifications which incorporated aspects from other techniques to achieve optimal results.

After preprocessing and segmentation had been researched and implemented a set of possible classification techniques were analysed so that one could be selected to allocate anatomical labels to each of the segments identified. The classification methods were measured against the aims of this project and the best matched method was selected and implemented.

A final stage was employed to validate the results and ensure that the models produced met the aims specified above. This validation was carried out qualitatively by a doctor performing a visual examination. For this a user interface was constructed which allowed the doctor to view the model from different perspectives at any point. In addition to this user interface, the models were visualised in three dimensional rendered form to ensure the surfaces appeared correct.

With the exception of the three dimensional visualisation[1] all of the above stages were implemented using the Java programming language. Java was selected because it is object oriented and this allowed for the creation of a range of objects used throughout the process. Java also has a large library of image manipulation classes which it was thought might be of use.

[1]Which was produced by S. Golodetz in C++ [37].

## 1.4 Document Structure

Chapter 2 describes the research that was carried out into the MRI modality and the DICOM format in which MRI data is stored. Also in this chapter is a description of the method used to extract the data from these files.

Chapter 3 includes an description and reasoning behind the preprocessing technique selection and also explains the method used to implement them.

Chapter 4 contains a thorough investigation and analysis of assorted segmentation techniques which could be used in this application.

Chapter 5 then describes the implementation of the selected segmentation technique as well as containing a study and comments on its performance.

Chapter 6 has an analysis of the classification required for this application and an explanation of the selection made. This chapter also details the method used to implement this technique.

Chapter 7 illustrates the methods that were used to validate the results from the previous stages as well as the feedback from the medical assessment.

Finally, Chapter 8 presents a set of conclusions including the major contributions of my research and a description of areas which could be further investigated.

# Chapter 2

# Image Interpretation

## 2.1 Magnetic Resonance Imaging

A description of the process used to produce images in MRI is given in Appendix A. The most relevant aspect of MR imaging to this application is the set of imaging errors which are commonly produced. The potential sources of error are as follows [16, 56, 76, 79]:

### 2.1.1 Signal, Noise and Inhomogeneities

As with other imaging modalities, MRI is susceptible to both random and systematic errors which cause noise in the output image. Mitchell [79] states that there are three sources for error: imperfections in the imaging system, imperfections in the acquisition process and patient factors.

With MRI technology there are three image attributes which must be balanced to form an image. Increasing one attribute can often have the effect of decreasing one or both of the others. These three factors are: signal to noise ratio, spatial resolution and acquisition time.

#### Signal to Noise Ratio

The three sources of noise stated previously can all affect the value recorded by the MRI unit for a given spatial location. Generally the amount of noise generated by each of these factors is fixed, whereas the value generated by the actual tissue of interest is a function of the apparatus' settings. This means that the proportion of the total signal resulting from noise can vary. For the purposes of image analysis and interpretation the noise is harmful and therefore the proportion of noise in the total value should be kept as low as possible by increasing the signal from the tissue. The balance between noise and signal is referred to a the signal to noise ratio (SNR). The signal from tissue can be increased by methods such as increasing magnetic field strength, but this is often at the

cost of spatial resolution and acquisition time.

#### Spatial Resolution

Spatial resolution refers to the amount of space which is contained within each discretised *image element*[1] (IE) retrieved from the MRI apparatus. An increased spatial resolution means that each discretised element represents a smaller amount of the tissue within the patient, and therefore finer detail can be extracted. Theoretically the size of each element can be reduced to a very fine amount, but this is not carried out due to the resulting increase in SNR caused by the reduction in sample (and hence signal) size.

#### Acquisition Time

From the perspective of image analysis, acquisition time would appear less important than the other two image factors, but it is relevant as a result of the problems caused by long acquisition times. It is possible to adjust pulse sequences and take repeated values at each spatial location to improve SNR and spatial resolution, but these methods often increase the time taken to acquire the image as well as the time to convert the raw data into an image. The downside of this increase in time for image analysis is that patients are unable to stay still for prolonged periods, so motion artifacts (explained below) can be introduced.

### 2.1.2 Artifacts

Artifacts in MR images are structures which are caused by various errors, but do not represent accurately and consistently any actual tissue structure within the patient. There are many types of artifact described in the literature, but the most common and relevant are 'Motion Artifacts' and 'Partial Volume Artifacts'.

#### Motion Artifacts

Motion artifacts are cause by patient tissue movement between samples taken by the MRI apparatus. The movement means that certain tissue structures are detected in more than one spatial location and the signal for the structure is blurred across the area in which it has moved. When the motion is periodic, such as breathing or cardiac beating, the sample interval can form a regular pattern in the image giving the appearance of multiple copies of the same tissue structure ('ghost' artifacts).

[1] Image element (IE) is used within this document to identify either pixels in two dimensional images or voxels in three dimensional images.

#### Partial Volume Artifacts

Due to the discretisation of digital imaging techniques such as MRI, for a single IE to be formed, an average value for the signal across the region is taken. If a single IE covers two tissue structures which have different values, then it will take a value in between the two. This can cause problems because the boundary pixels can appear to represent a separate volume to each neighbour which does not correspond to any actual tissue structure.

### 2.1.3 Radio Frequency Non-Uniformity

MRI units are constructed with a high degree of accuracy and precision but small inhomogeneities in components such as the radio frequency (RF) transmitters and receivers can have an effect on the quality of images produced. These inhomogeneities mean that the signal value detected for a given location will not depend solely on the tissue structure. Instead, each value will also be a function of its proximity to the different radio frequency transmitters and receivers. As there are few transmitters and receivers within a unit the change in value is quite gradual and the effect can be thought of as a gradient function. Clarke et al. [23], state that radio frequency non-uniformity is the greatest single limitation to the quality of data which can be retrieved from a MRI unit.

### 2.1.4 Implications of MRI Research

From this investigation and that shown in Appendix A it is clear that MR imaging is very suited to constructing patient models due to its ability to detect and represent a range of different structures. In addition MRI is particularly adept at depicting varied tissue boundaries even when the neighbouring structures have similar densities. It is important to note that the research into this modality has highlighted that project aim for robustness is likely to be difficult to achieve. Each image weighting and parameter setting can be used to highlight certain aspects of the patients anatomy, but this is at the cost of degrading others. I therefore believe that acquiring a single set of images which clearly depict a large range of structures is likely to be difficult.

Additionally, this research has highlighted several sources of error within MRI which can corrupt the images produced. Each of these sources has been shown in the literature to affect various segmentation processes and therefore it was decided that they should be removed or limited wherever possible by preprocessing the image data.

## 2.2 Digital Imaging and Communications in Medicine

The Digital Imaging and Communications in Medicine (DICOM) standard is a fundamental standard employed in many aspects of medical imaging. The DICOM standard was created to allow medical imaging data to be transferred, stored and displayed in a neutral

manner which is independent of application, modality and equipment manufacturer. This is of use for medical imaging because there are numerous different medical image acquisition, processing and visualisation applications; if these cannot communicate with each other in a well defined way then the process of using more than one application can become complicated or even impossible.

The DICOM standard is a collection of documents which define a precise method by which real world medical data can be encoded as an *Information Object Definition* (IOD). These IODs can be considered as a framework for a list of attributes which define the data. For MR images the definition includes a set of MR image pixel values, the image dimensions as well as anything else that the equipment deems necessary. Once an IOD has been created from a specific set of data it is referred to as an IOD instance. Each application typically produces an IOD instance for a specific purpose such as data storage or instruction transfer and each purpose is called a service. This means that each IOD instance is coupled with a service to create a *Service-Object Pair* (SOP) which is the format for communication between applications and which the program for this application must be able to interpret [15, 102].

A full description of the DICOM file format is given in Appendix B. The following is a description of the parser which was implemented to convert the raw DICOM files into an internal representation which could be used in the model processing stages.

## 2.3 Implementation of a DICOM MR Image Series Parser

For this application, three dimensional patient images will be analysed and processed at the segmentation stage. Three dimensional images are considered as a collection of two dimensional images with a known offset between them. Though the DICOM layered structure would allow for a complex three dimensional representation to be stored, this is not commonly done and instead each two dimensional slice is stored in a separate DICOM file within a single series. A parser must therefore be able to interpret this series of images and reconstruct them into a three dimensional representation of the patient.

### 2.3.1 Image Volume Data Structure

To store a three dimensional volume I created a Java data structure, `voxelArray`. Its main body is a three dimensional array of integer values, and it also has some additional parameters for features such as spatial resolution and value extrema.

This structure is very simplistic, and as a result is takes up quite a large amount of memory ($\approx$ 100 MB for a $512\times512\times100$ image volume). In a raw MR image there are very few areas with good homogeneity and I therefore believe that using a more intelligent compressed or mesh based structure would not achieve a large improvement. Moreover, the values at each IE are likely to be retrieved multiple times during the segmentation

stage and this can be done efficiently with an array based encoding.

The actual implementation of this data structure includes a secondary three dimensional array which is used to store the region identities of each voxel for use by the segmentation algorithms.

### 2.3.2 Parsing Algorithm

In order to parse a set of DICOM images into a voxel array the DICOMDIR directory file must first be analysed so that the images which compose a series can be identified and retrieved. Algorithm 2.1 shows a pseudo-code representation of this top level parsing approach.

**Algorithm 2.1** Image Series Parser

**Input:** *dicomdir* The directory file referencing the required image series

```
1: dicomdir-data ← DATA-ELEMENT-PARSER(dicomdir)
2: record-structure ← EXTRACT-RECORD-STRUCTURE(dicomdir-data)
3: required-series ← USER-SELECTS-SERIES(record-structure)
4: image-file-array ← EXTRACT-IMAGE-SERIES-FILES(required-series, dicomdir)
5: for each image-file in image-file-array do
6:     image-data ← DATA-ELEMENT-PARSER(image-file)
7:     pixel-data ← EXTRACT-PIXEL-DATA(image-data)
8:     voxel-array ← ADD-TO-ARRAY(pixel-data, voxel-array)
9: end for
10: return voxel-array
```

This algorithm first parses the directory file into its composite data elements so that the record structure can be extracted and offered to the user, allowing them to select which series they would like to process. Once the image series is selected by the user, the function extracts the locations of the DICOM files which contain the required images from the directory file. The program then loops through each of the DICOM image files and parses the data elements from each. The pixel data can be retrieved from parsed DICOM file allowing it to be added into the full image array. The actual DICOM file manipulation was carried out using the DCM4CHE Java toolkit [32]. Once this process is complete for every image in the series the voxel array is full. This means that it has an integer value at every element which defines to the MR value of a corresponding point in the patient.

# Chapter 3

# MR Image Preprocessing

It was stated in the previous chapter that MRI is susceptible to various imaging errors. Clarke et al. [23] state that each of these errors can affect the success of segmentation and therefore require removal or reduction.

There are many preprocessing methods described in the literature which have been developed to remove or reduce errors for various images and applications. Whilst I acknowledge that preprocessing is an important stage in preparing for segmentation, I felt that there would be a greater benefit for this project of a more thorough investigation into suitable segmentation techniques. The reason for this is that I consider it to have a greater influence on the application's overall performance and the quality of the patient models produced. There is a review and analysis of several preprocessing techniques in Appendix C. Each of the techniques covered was selected for their relevance to MRI and for their ability in: the suppression of noise; removal of artifacts or compensation of RF non-uniformities. However, this review is not the focus of this project and is supplied as background information for those unfamiliar with digital image processing.

This chapter details some factors which affected the selection of preprocessing techniques, and goes on to describe the reasons for each method's selection as well as their implementation.

## 3.1 Factors Important in Preprocessing MR Images

Clarke et al. [23] state that preprocessing of MR images is required because there are several common sources of error in the modality which are harmful to effectiveness of segmentation. They consider three different sources of error:

- Random image noise;
- Image artifacts;
- Signal intensity variation (also referred to as RF non-uniformity).

They also state that there are five image aspects to consider when preprocessing images:

- Information (i.e. correctly recorded pixel intensities);
- Image contrast;
- Signal to Noise Ratio (SNR);
- Spatial resolution;
- Computational complexity and operation time

where improving any one aspect often has a detrimental effect on one or more of the others.

These five image aspects are important for effective segmentation and therefore finding a preprocessing technique which can balance each of them is important. Information loss will mean incorrect pixel values are used in segmentation, making it more likely that segments will be created in the wrong shape. Image contrast is key to many segmentation techniques for separation of different tissue structures. SNR is important as comparatively large amounts of noise will cause high variance in tissue regions making their recognition technically demanding. Good spatial resolution is vital in image segmentation as many techniques require strong edge definition to detect tissue boundaries.

## 3.2 Technique Selection and Implementation

The literature reviewed shows that there are a range of preprocessing techniques available for MR images, none of which are perfect, but each of which finds a different balance between the five image factors identified above [17, 23, 38, 56]. Each of the segmentation techniques described in Chapter 4 process and interpret the MR image in different ways. This difference of approach means that each segmentation technique is affected by the balance of image factors in different ways and an optimal preprocessing technique for one segmentation method might perform poorly on another.

Once split and merge was selected as the segmentation technique it identified a range of attributes which the ideal input image should have, as well as attributes which are less important. The reader may want to look forward to the segmentation implementation in §5.1 to fully understand the selected process before continuing with this section.

Using the absolute difference uniformity criterion for split and merge means that if any single IE within a region is very different from the remainder then it will have a large effect on the ability of the region to be joined with others. Random noise, common in MR images, is seen as scattered high frequency points and can occur within otherwise homogeneous regions causing the split and merge process to fail. Reducing random noise is an important aim for the preprocessing stage as a result. Another shortcoming of the absolute difference uniformity criterion is that it sets a maximum difference between values across a whole region and it is highly inflexible towards regions affected by intensity inhomogeneities as a result.

### 3.2.1 Gaussian Low-Pass Filtering

High frequency noise within the image can be removed by many noise suppression techniques, such as: anisotropic diffusion filtering [99]; non-local means filtering [17] and wavelet based filtering [38]. As well as noise suppression, each of these methods also have other attributes which must be balanced. Gaussian low-pass filtering [86] was selected for this task because it is known to remove common high frequency noise and can be applied very efficiently with a convolution mask. The other examples above achieve a similar smoothing in homogeneous areas, and their advantage is their ability to preserve contrast and spatial resolution at edges. Split and merge is not based on edge identification and therefore these two factors are not important and increase complexity needlessly.

The Java Advanced Imaging toolkit [119] was researched to ascertain if it could be used to implement the Gaussian filter, but it did not include a three dimensional method. Implementing a Gaussian filter in three dimensions has several advantages over two dimensions, but the main advantage is that a greater number of neighbours can be considered within the same radius. Using more neighbours in the filter means that that a more informed value can be selected which is less affected be noise.

The Gaussian low-pass filter was manually implemented with a standard convolution matrix as shown in algorithm 3.1. The matrix used was three dimensional, and each point was calculated using the standard Gaussian distribution.

**Algorithm 3.1** Gaussian Filter Application

**Input:** $V$ The input voxel array
**Input:** $\sigma^2$ The Gaussian distribution's variance
**Input:** $S$ The convolution matrix's size
1: $C \leftarrow$ CREATE-CONVOLUTION-MATRIX($\sigma^2$, $S$)
2: $F \leftarrow V$
3: **for** each *position* in $V$ **do**
4: $\quad voxel \leftarrow$ GET-VOXEL-AT(*position*, $V$)
5: $\quad result \leftarrow$ APPLY-CONVOLUTION-MATRIX-AT(*voxel*, $C$, $V$)
6: $\quad F \leftarrow$ SET-VALUE-AT(*position*, *result*, $F$)
7: **end for**
8: **return** $F$

The matrix's size and the distribution's variance were tested in combination with the segmentation implementation and a matrix of size 5×5×5 and a standard deviation of 1.2 were found to be give the best balance between smoothing and edge degradation. The result of applying this filter to a test image is shown in figure 3.1.

Once the Gaussian filter and split and merge processes were implemented the complete system was tested and visually checked. There were two areas where the quality of segmentation was deemed to be unsuitable and required further attention. The first error common in segmented images was that thin, shell-like segments were being identified surrounding 'real' organs. Their appearance was similar to what would be expected from partial volume effects, but some of them had thicknesses of up to 10 voxels implying that

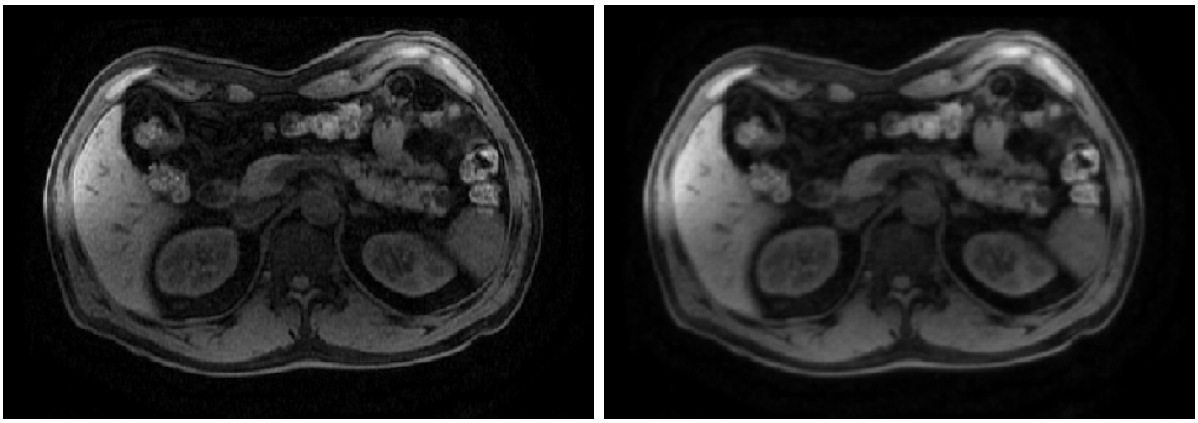

Figure 3.1: Comparison between raw MR image (left) and Gaussian filtered MR image (right) showing noticeable reduction in random noise, but also blurred edges.

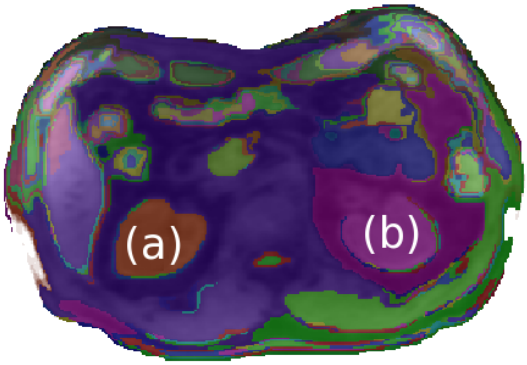


Figure 3.2: Screen capture from split and merge segmentation with thin regions at organ boundaries, (a), and a missing kidney which has been joined with the main cavity, (b).

this was not the case[1]. These shell structures are illustrated in figure 3.2(a). I consider that these structures are the result of edge blurring caused by the Gaussian filter. At the middle of the blurred edge, where the sharp edge had previously been, the Gaussian filter changed the grey-scale values so that they were midway between the values at either side. The 'midway' values were far enough from both 'real' regions that the split and merge process considered these as completely separate regions.

The shell regions could be identified and discarded at the classification stage and therefore do not pose a problem themselves, but they do have the effect of moving the segment boundaries away from where the true edge is, by up to 5 pixels.

The second error caused by preprocessing was that the boundary elimination process tended to merge segments which defined valid structures. A segmentation which has

[1]It these were caused by partial volume effects I would expect them to only be 1 or 2 voxels wide as this is the likely limit of sample averaging.

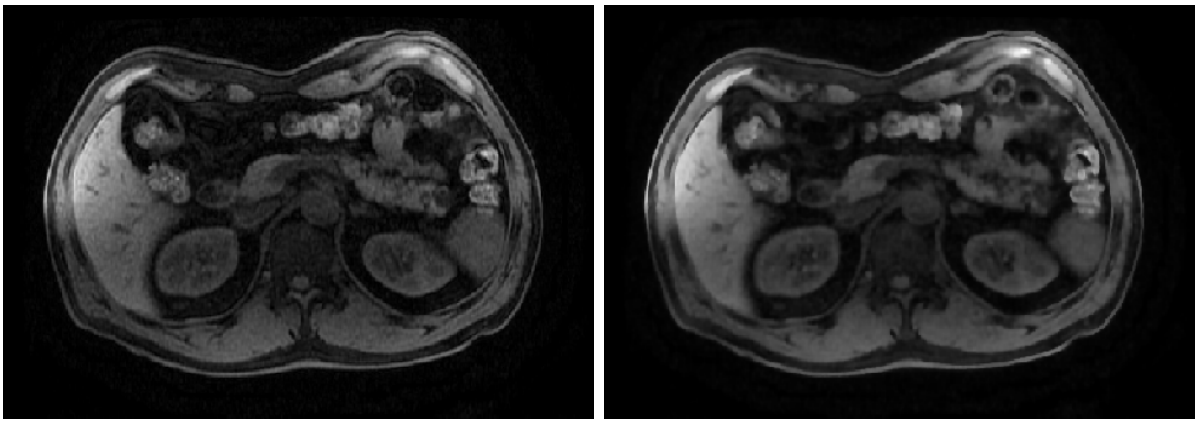

Figure 3.3: Comparison between raw MR image (left) and ADF filtered MR image (right) showing similar noise reduction to Gaussian filtering with reduced contrast loss at boundaries (particularly visible towards the top of the liver and inside the kidneys).

lost a valid kidney segment is shown in figure 3.2(b). Split and merge does not rely on edge definition for its operation and therefore the edge degradation caused by Gaussian filtering was not previously believed to be a problem. Conversely, boundary elimination does asses boundaries based on their gradient and therefore blurred edges may loose contrast and be eliminated erroneously as a result.

### 3.2.2 Anisotropic Diffusion Filtering

Both of the segmentation errors identified using Gaussian filtering were a direct result of degraded edges. Anisotropic Diffusion Filtering (ADF) [99] is able to minimise edge degradation by only applying smoothing parallel to edges rather than across them. This anisotropy means that the smoothing still removes noise in homogeneous areas, but avoids high gradient sections which are likely to represent object edges. ADF is considerably faster in its operation than other edge preserving methods such as NL-means filtering [17] and therefore it was selected for use.

ADF was implemented in three dimensions using the method included in the open-source Insight Toolkit [53]. The result of applying this filter to a test image is shown in figure 3.3. Based on the MRI testing carried out in [54] the optimal settings for ADF on MR images were found to be 20 iteration steps, a time step of 0.05 and a conductance of 3.

Using ADF instead of Gaussian filtering had a large improvement on the segmentation produced and the result is shown in §5.1.10. There are still partial shells in some areas, but these are much rarer and are typically only 3 pixels in thickness. I consider this to be acceptable because in the few areas where they do occur they are thin enough that the affect that they have on boundary positioning is minimal. Also, the boundary elimination process was considerably more stable with ADF and the erroneous removal of correct boundaries was greatly reduced.

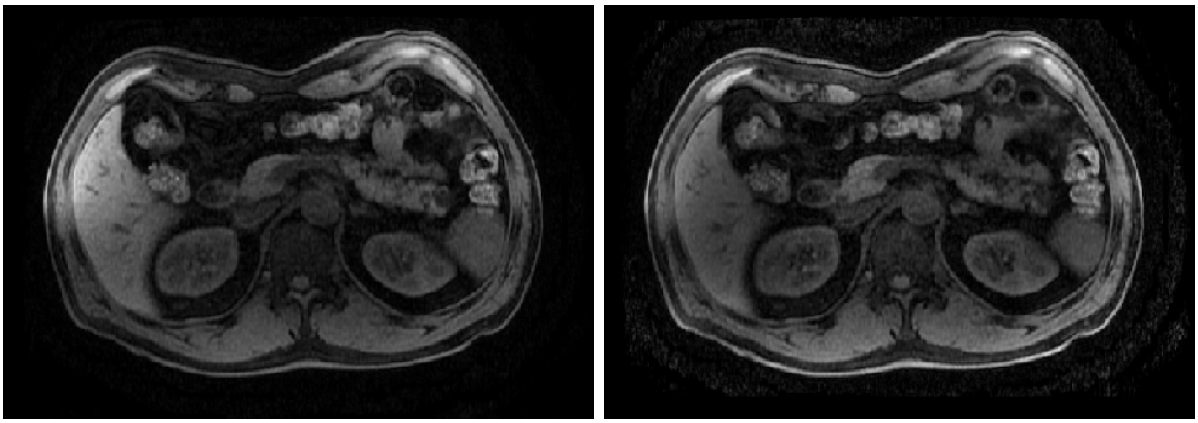

Figure 3.4: Comparison between raw MR image (left) and non-uniformity compensated MR image (right) showing greatly increased uniformity in the grey-scale values of the liver on the left hand side.

### Filter Based Non-Uniformity Compensation

To overcome the non-uniformities common in MR images the filter based compensation method [42] was selected. I decided that using correction matrices [36] would be too restrictive because they limit the application of the system to only images produced on MRI machines which also provided phantom images. Filter based compensation is formed on the assumption that the inhomogeneity within an image is of a considerably lower frequency than the image detail and therefore by removing low frequency components the inhomogeneity can be corrected. Some literature states that this assumption is overly simplistic [23] and may not fully remove the inhomogeneity or degrade the image. Due to the lack of suitable alternatives and the prevalence of inhomogeneities within MR images it was decided that this method should be implemented and tested to see if the assumption was valid enough.

No standard toolkits could be found to implement the compensation and therefore this was done manually. The image volume was duplicated and a low-pass filter was applied to the duplicate to identify the low frequency component containing the non-uniformity. The low-pass filter selected to produce the smoothed function was a Gaussian filter as described previously. For this application the smoothing had to be much greater than was used previously and a standard deviation of 55 was found to give the best results.

Once the low frequency component had been extracted it was simply divided by the original image to produce the compensated image. From reading the literature the results were not expected to be particularly useful, but by setting the low-pass filter correctly the improvement was quite dramatic. This is illustrated in figure 3.4.

The draw back of this method is that the large Gaussian filter tends to cause spurious new noise patches outside of the main body of the patient. This is not considered to be too problematic as they can be eliminated during the segmentation and classification procedures.

# Chapter 4

# Analysis of Segmentation Techniques Specific to MR Image Properties

After reading several reviews and surveys on segmentation techniques it was clear to me that selecting a single technique for use would not be simple [23, 33, 55, 61, 91, 101]. The reason for this is that at each technique considers both the image and the segments produced in different ways. This diversity means that each technique has applications to which it is well suited and also those to which it is not. The interrelation between segmentation technique and application is very intricate and many factors must be considered. To ensure that the method selected was suited to the aims of this project, a thorough investigation of segmentation techniques was required with each one analysed based on the specific properties of MR images and the aims specified in Chapter 1. This chapter details the investigation carried out into possible segmentation techniques. For each candidate method, there is a description of its application and an analysis of its likely performance.

## 4.1 Introduction to Segmentation

Segmentation is as a process whereby an image is divided into meaningful regions [40]. The concept of 'meaningful' is somewhat ambiguous in this definition because there are many different applications for segmentation and each can have a very distinct concept of what might constitute a meaningful segment.

Image segmentation is defined formally by Zucker [137] in the following way:

Let $X$ represent the image to be segmented as a set of sample points.

Let $P$ be a predicate such that $P(Y)$ is *true* if and only if $Y$ is a set of sample points which satisfy some predetermined measure of uniformity, and is *false* in all other cases.

Then a segmentation of $X$ can be defined as its division into disjoint subsets $X_1, X_2, \ldots, X_N$ where:

$$\bigcup_{i=1}^{N} X_i = X \tag{4.1}$$

$$X_i, \quad i = 1, 2, \ldots, N \text{ is connected} \tag{4.2}$$

$$P(X_i) = \text{TRUE} \quad \text{for } i = 1, 2, \ldots, N \tag{4.3}$$

$$P(X_i \cup X_j) = \text{FALSE} \quad \text{for } i \neq j, \quad \text{where } X_i \text{ and } X_j \text{ are adjacent} \tag{4.4}$$

These four conditions state that for a segmentation to be valid:

(4.1) Every point in the image must be included in one of the regions;
(4.2) Every region must be connected;
(4.3) Every region must pass the uniformity test;
(4.4) All adjacent regions must fail the uniformity test if conjoined.

All of the technique reviews I investigated [23, 33, 55, 61, 91, 101] agreed that no single segmentation technique is guaranteed to perform well on all images and for all imaging modalities. Pham, Xu and Prince [101] further state that although no technique will perform well on all images, there are some techniques which are considered more generalised than others and which can achieve a 'good' level of performance on a wide range of data.

Wolpert and Macready's 'No Free Lunch (NFL) Theorems' [129, 130, 131] state and prove mathematically that all optimisation techniques perform equally as well as any other when their performance is averaged over all possible target functions.

Pham, Xu and Prince [101] state that image segmentation can be posed as an optimisation problem where some segmentation suitability function must be optimised given certain variables and parameters. Therefore I believe that NFL theorems can be applied to image segmentation and mathematically substantiate the empirical findings of the technique reviews investigated. The results of the NFL theorems for this application are that:

- Arbitrary application of a specific segmentation technique is statistically unlikely to produce 'good' performance; and
- For a given image the only way to ensure optimal segmentation performance (without testing all possible techniques) is to utilise *a priori* knowledge.

The lack of universal applicability in any single segmentation technique means that in order to select the optimal method for this application *a priori* knowledge must be employed. For this reason I think that a thorough investigation of a wide range of segmentation techniques is required so that each one can be evaluated relative to the specific features and intricacies of MR images and the aims of this project.

The literature divides segmentation techniques in many different ways. This research follows Lakare's convention [55] as the techniques in each category share similar advantages and disadvantages when used on MR images. The three categories he considers are: *Stochastic*; *Structural* and *Hybrid*.

## 4.2 Stochastic Techniques

Stochastic segmentation techniques are those which utilise some form of statistical analysis and are applied to individual IEs without taking into account the structure of the remaining image [55]. Stochastic techniques can utilise some information regarding surrounding IEs, but this generally is only carried out in a localised way.

### 4.2.1 Thresholding

Thresholding based segmentation aims to derive a function which allocates each IE to a single class such that each class consists of a different segment within the image. Sahoo et al. [113] defined this formally for a two dimensional image with two segments (object and background) as follows:

Let: $N$ be the set of natural numbers
$(x, y)$ be the spatial coordinates for a digital image
$G$ be the set of integers which represent pixel grey-scale values
$f$ be an image function mapping from an image point, $N \times N$, to a value, $G$
$t$ be a threshold value such that $t \in G$
$B$ be the binary pair of grey-scale classes, ($b_0$, $b_1$), such that $b_0$, $b_1 \in G$

Then the thresholding function, $f_t$, for image mapping $f$, is one which maps from an image point, $N \times N$, to a binary value, $B$, such that:

$$f_t(x, y) = \begin{cases} b_0 & \text{if } f(x, y) < t \\ b_1 & \text{if } f(x, y) \geq t \end{cases}$$

This definition states that the function takes the grey-scale value at the given point, and if it is less than the threshold value, $t$, it is assigned to the class $b_0$, otherwise it is assigned to the class $b_1$.

This definition refers to the simplest thresholding techniques, where only the element's grey-scale value is considered for the image function. Sahoo et al. [113] refer to these techniques as *point-dependent*. Alternatively, *region-dependent* techniques utilise some local property for each IE so that the segment allocation is partly determined by the IE's neighbourhood.

Thresholding techniques may also be defined by the scope (area of applicability) for the threshold value. Thresholds may be applied *globally* so that there is a single value for $t$ across the whole image, or applied *locally* where the value for $t$ varies across the image [113]. Weszka [127] adds another level of distinction for local techniques which utilise a continuously changing threshold value, they are called *dynamic*.

A final important distinction between thresholding techniques is the number of different segments into which the image can be divided. Sahoo et al.'s definition above divides the

image into two different segments, $b_0$ and $b_1$, using one threshold (*bilevel thresholding*). If more than two different segments are expected then multiple thresholds can be utilised (*multithresholding*) [33]. Bilevel thresholding would not be adequate for this application because MR images contain many different tissue types. Multithresholding will therefore be the main focus of investigation[1].

A key aspect for all thresholding techniques is the selection of the threshold values. There are many methods described in the literature and the most common and suited to MR images are described below.

### Global Point-Dependent Thresholding

The simplest threshold selection methods are for global point-dependent techniques. Here the threshold is selected according to analysis of the image's grey-scale histogram. For example Prewitt and Mendelsohn's *modal* thresholding method [105] attempts to mathematically identify valleys in the histogram which represent natural divisions between segments. The grey-scale value at the bottom of each valley is then set as a threshold so that the segments can be divided. Other methods such as Doyle's estimated image composition based *p-tile* method [30] and Otsu's variance based method [89] also rely on mathematical analysis of the image's histogram to select threshold values.

### Global Region-Dependent Thresholding

Region dependent thresholding methods utilise some function to incorporate neighbourhood information for each IE. Sahoo et al. [113] break the techniques down into three main categories. *Histogram transform methods* convert the image's histogram by weighting the contribution from each IE based on a local property such as gradient magnitude [74, 128]. The transformed histogram can then be used by any thresholding technique which relies on an images histogram. *Second-order grey-level statistic based methods* analyse the image's statistical construction using second-order functions so that more information about the context for each IE can be incorporated [51, 80]. *Relaxation methods* take an initial estimate of an IE's segment membership based purely on grey-scale value, and then iteratively improve this estimate by calculating the probability of the estimate being correct based on the similarity between the element and it's neighbour's segment allocation [110].

### Local Thresholding

Local thresholding (as considered by Sahoo et al. [113]) is a method where the image is divided into smaller sub-images, each of which is thresholded individually using any of the global techniques described above. After this sub-image thresholding a smoothing filter

[1]It should be noted that multithresholding can be considered as multiple applications of bilevel thresholds [113]. For this reason bilevel thresholds will not be completely discounted.

can be applied to remove any discontinuities which may occur between neighbouring sub-images. The aim of this technique is to compensate for variations in exposure across the image where using a universal threshold would mis-segment areas over or under exposed in comparison to the image mean. The method described as *dynamic* by Weszka [127] is very similar, but here each individual IE's segment allocation is calculated based on the thresholding of a sub-image centred on it rather than arbitrarily placed in a grid.

#### Thresholding Conclusions

Much of the literature demonstrates accurate and precise segmentation when thresholding is applied to images that depict well defined objects and backgrounds with low noise [30, 89, 105, 110, 113, 127]. However, I feel that these results do not show a particular relevance to this application.

A significant difficulty for the application of thresholding to MR images is the presence of image errors such as noise, artifacts and RF non-uniformity. While some of the more complex thresholding methods are able to account for these factors, these methods are considered to achieve limited success [55]. Moreover, without any spatial information being utilised in the segmentation process I believe that even small artifacts will be very difficult to identify since on a local scale they would appear as valid anatomical segments.

An additional issue for the use of thresholding is that MR images typically consist of multiple different tissue types many of which can have very similar values. I do not think that using purely point or regional information a segmentation technique would be able to effectively distinguish between tissue types as some spatial information is vital. Thresholding has been used with a good level of success for MR images of the brain [57, 62], but these methods required a great deal of anatomical knowledge to be incorporated and one required a user to set threshold values manually. While this was practical for a single structure (i.e. the brain), I believe that the level of information required for multiple organs would be prohibitive.

### 4.2.2 Clustering

Clustering is typically considered as an unsupervised technique for grouping objects based on one or more of their properties [112]. The detail of clustering's application for segmentation is discussed below, but the basic concept is that each IE is considered as a feature vector that describes some of its properties and 'similar' elements are grouped together in some way.

To allow clusters to develop so that they best represent the data, clustering is an iterative process. The clusters are repeatedly updated with the aim that each repetition will improve the allocations to clusters. Each iteration is a two stage process where the algorithm employed alternates between allocating objects to clusters, and characterising the new clusters in some way.

There are two common clustering techniques relevant to image segmentation: *k-means*

and *Fuzzy C-means*. *Expectation maximisation* clustering has also been used in the literature, but due to its high sensitivity to initial parameters [101] it will not be included here as training parameters would be time consuming and unnecessary as the other methods can achieve similar performance.

### $k$-means

$k$-means clustering iteratively develops clusters by considering each object to be clustered as a point in an $N$-dimensional space where each dimension in the space relates to a specific object property. In MacQueen's method [66] the only metric used for clustering is the Euclidean distance between objects in the cluster-space, and the aim of the algorithm is to minimise the sum of the square distances between each object and the centroid of the cluster to which it has been allocated. Formally (in 1-dimension) the algorithm attempts to minimise the function below [31]:

$$J = \sum_{i=1}^{k} \sum_{x \in X_i} \| x - v_i \|^2 \tag{4.5}$$

where $k$ is the number of clusters
$X_i$ is the set of objects allocated to cluster $i$
$v_i$ is the centroid of cluster $i$
$\| a - b \|$ is the Euclidean distance between $N$-dimensional points, $a$ and $b$.

The algorithm is initialised by randomly selecting $k$ points in the cluster space which are taken as the centroids for the first generation of clusters. The Euclidean distance between every object and each centroid is then calculated and each object is allocated to the cluster centroid to which it is closest. The centroids are then re-calculated for each cluster and the process of cluster allocation and centroid re-calculation is repeated until there is no further change in centroid position.

The challenges for using clustering for image segmentation are the selection of features for use as dimensions in the cluster-space, and the selection of the number of clusters to use. Coleman and Andrews [24] describe a method for for overcoming these challenges, which is outlined in figure 4.1. They state that for each IE in the image you should compute multiple features which relate to the values and textures for 'windows' of various sizes centred on that IE. This process builds a profile of each IE in terms of both it's own properties and those around it and is referred to as *Feature Computation*. Due to the number of features that this process creates, the following two steps are vital for the efficient operation of this algorithm, particularly when considered for the large medical image volumes in this application.

Coleman and Andrews first use the Karhunen-Loeve transform to *decorrelate redundant features* which provide no unique information. The number of features is then reduced further (*feature reduction*) by calculating a Bhattacharyya measure[2] for each remaining

[2]The Bhattacharyya measure looks at feature means and variances in order to select a set of features with differing means and variances so that they are expected to describe different aspects.

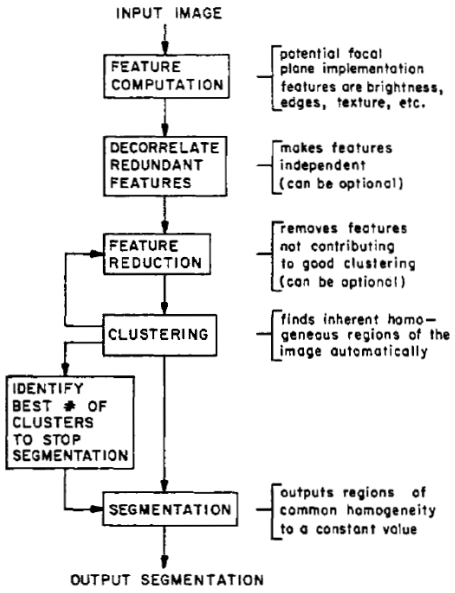


Figure 4.1: Block diagram for segmentation by $k$-means clustering using Coleman and Andrews' method [24]

feature and only retaining features which exhibit a good 'likelihood' of identifying clusters.

The remaining features are then used as dimensions in the cluster-space and $k$-means clustering is applied. The value of $k$ is calculated by a formula based on the cluster separability and cluster scatter matrices calculated as part of the feature reduction stage. In the final stage the IEs are allocated to a segment as defined by the cluster to which they belong.

### Fuzzy $C$-means

Dunn [31] identified that $k$-means clustering is non-deterministic given random starting conditions and that it does not necessarily produce a unique solution. To overcome this, Dunn formulated a modified version of $k$-means clustering which incorporates fuzzy set theory[3] so that the optimal clustering can "probably" be found.

In Dunn's method the function to be minimised is extended from 4.5 so that fuzzy set membership is considered for all of the possible clusters [31][4]:

$$J = \sum_{i=1}^{k} \sum_{x \in X} u_i(x) \parallel x - v_i \parallel^2 \tag{4.6}$$

[3]As defined by Zadeh [136].

[4]In Dunn's article he shows that the theoretical applicability of this approach is not limited to Euclidean metrics, but it is represented in this form here to show the comparison with MacQueen's method above.

where $u_i(x)$ is the fuzzy set membership of object $x$ to cluster $i$.

This function weights every object-to-cluster Euclidean distance by the membership value of the object to the cluster. For most object/cluster pairs the membership value is 0, but it does allow some objects to belong to more than one cluster with varying degrees of membership. The advantage of utilising fuzzy relationships is that clusters are developed with links to more objects than in $k$-means. This allows clusters to incorporate more information in their development process which should make evolution towards a 'globally' better solution more likely.

Bezdek [13, 14] shows that function 4.6 can be minimised iteratively in a method very similar to $k$-means except that when objects are allocated to cluster centroids it is not just the 'closest' centroid to which they are clustered, but they have fuzzy membership values calculated to every cluster centroid.

#### Clustering Conclusions

A key advantage of clustering as a method for segmentation is that it does not require any parameters, training or detailed anatomical information for efficient segmentation. This is advantageous for this application since there is such a variety of MR images. Being able to segment without having to input all of the image possibilities will simplify use and should negate difficulties caused by new or unexpected data.

The literature includes examples of clustering for medical image segmentation [9, 100], but these examples are limited to the brain, which I believe poses very different issues to body scans. In brain imaging there are comparatively few tissue types (white matter, grey matter, fluid and the skull) and these can be distinguished purely according to their MR values. The body contains many more tissue types, some of which have overlapping MR values and segmentation based purely on these values is therefore likely to be problematic.

Humans can identify different tissue types with similar values (grey-scale colours) by looking at spatial positioning and edge contours. Clustering does not offer any method for readily incorporating either of these pieces of information and even quite complex algorithms would classify tissue structures on opposite sides of an image as the same segment if they happened to have similar grey-scale values.

A further problem with clustering is that it operates at the individual IE level. For an image series of 512×512×100 voxels, if just 5 features per voxel are used in the clustering, then the algorithm has to operate on $26\times10^6$ points in a 5-dimensional cluster space. Moreover, establishing optimal clustering for all functional metrics (Euclidean etc.) is NP-hard and therefore no clustering methods are able to guarantee optimality in a reasonable operation time.

### 4.2.3 Markov Random Fields

A Markov Random Field (MRF) is defined by Li [60] as "a branch of probability theory which provides a tool for analysing spatial or contextual dependencies of physical phenomena". According to Kindermann and Snell [50] MRFs were initially developed to expand the Ising model[5] as a framework for simulating neighbour relations between random variables. MRFs take their Markovian component from the concept that each random variable's probability distribution is independent of all others given the probability distributions of it's neighbours, i.e.:

$$\mathbf{P}(\mathbf{X}_i|\mathbf{X}_j, j \in a, i \neq j) = \mathbf{P}(\mathbf{X}_i|\mathbf{X}_j, j \in n_i) \tag{4.7}$$

where $\mathbf{X}_i$ is the probability distribution of a random variable $i$
$\mathbf{X}_j$ is the probability distribution of a random variable $j$
$a$ is the set of all random variables in the Markov model
$n_i$ is the set of random variables which are neighbours of $i$.

Unlike a standard Markov process where *time* is the property on which variables change [112], in a MRF *position* takes this role. Neighbours are therefore not random variables which are temporally sequential, but are those which are physically 'close' to each other in some $N$-dimensional space. In reality the MRF does not actually enforce any restriction on what constitutes a neighbour and this can be set in any way required for the specific implementation.

The importance of MRFs for image segmentation is that spatial constraints exist for 'good' segments. These constraints can be used to specify a MRF which indicates the likelihood of a segmentation being correct by assessing it against the supplied constraints. The simplest example of a spatial constraint used in image segmentation is that a given IE will typically be a member of the same segment as its direct neighbours. The MRF for this constraint therefore constructs likelihood distributions such that segmentations where lots of neighbouring IEs are from different segments are considered less probable and vice-versa.

MRFs are only a spatial modelling tool for assessing the probability of a given segmentation and are not an independent segmentation technique. MRFs must therefore be used in combination with other techniques to achieve a segmentation. The most common method employed is the generation of a maximum *a posteriori* (MAP) image labelling within a Bayesian framework [101].

The following process explanation is based on the various work and descriptions shown in [50, 60, 92, 106].

[5]The Ising model models the two-dimensional direct neighbour interactions of two-state 'spin' in ferromagnetic materials where neighbours are more likely to spin in the same direction as each other than in opposite directions.

### Maximum *a Posteriori* Estimate Labelling

A labelling is an assignment such that every site in a predefined set is allocated a label from a set of candidate labels. For image segmentation the sites are the set of all possible IE positions and the candidate labels are all segment labels. Hence, a MAP labelling, $f^*$, is the single labelling from the set of all possible labellings, $\mathbf{S}$, which is believed to have the highest probability of being correct given the observed image, $\mathbf{r}$:

$$f^* = \arg\max_{f \in \mathbf{S}} \; p(f|\mathbf{r}) \tag{4.8}$$

According to this definition, segmentation is reduced to a problem of deducing a labelling for which its probability, given the observed image, is maximised. The total number of segmentations possible for an image with $N$ IEs is the same as the number of unique partitions for a set of size $N$. The Bell number, $B_N$, tells us the total number of segmentations which would require checking against equation 4.8 to ensure the most probable labelling was selected. As it is known that the sequence of Bell numbers is exponential [10], deriving the MAP by exhaustion is at best NP-complete and is likely to be too computationally complex. Before considering the method for maximising $p(f|\mathbf{r})$ it is first necessary to explain the method by which the probability distribution $\mathbf{P}(f|\mathbf{r})$ can be calculated using a Bayesian prior model.

### Bayesian Prior Model

The probability distribution for a labelling given the observed image is required, but this distribution cannot be specified directly. Therefore Bayes' theorem can be applied as shown below:

$$\mathbf{P}(f|\mathbf{r}) = \frac{p(\mathbf{r}|f)\mathbf{P}(f)}{\mathbf{P}(\mathbf{r})} \tag{4.9}$$

As the same image is used for all labellings, the probability distribution for the observed image, $\mathbf{P}(\mathbf{r})$, does not have any effect on the maximisation and can be safely disregarded leaving the equation:

$$\mathbf{P}(f|\mathbf{r}) = p(\mathbf{r}|f)\mathbf{P}(f) \tag{4.10}$$

The probability distribution for the observed image, given a labelling, $p(\mathbf{r}|f)$, (likelihood function) depends on the expected error sources in the imaging modality such as noise, inhomogeneities, and artifacts. The prior labelling distribution, $\mathbf{P}(f)$, is dependent on how well the labelling conforms to a predefined set of contextual rules which identify a 'good' segmentation.

### MRF Construction and the Gibbs Distribution

A MRF is a good method for modelling contextual and spatial constraints and is therefore ideal for deriving the prior distribution $\mathbf{P}(f)$. The Hammersley-Clifford theorem [41]

shows that a MRF must be expressible as a Gibbs distribution in the form shown below:

$$\mathbf{P}(f) = Z_p^{-1} \times e^{-\frac{U(f)}{T}} \tag{4.11}$$

where $Z_p$ is a normalising constant
$T$ is the temperature (i.e. the Gibbs global control constant)
$U(f)$ is the prior distribution energy.

The Gibbs distribution is explained fully by [50], but it is essentially an exponential distribution where the exponent is an 'energy' value distribution. For this application the aim is to create a prior energy value distribution such that when the labelling conforms to all contextual and spatial constraints it has a low energy and vice-versa.

The process by which the prior energy function is constructed for specific contextual constraints is non-trivial, but it can be considered simplistically as defining prior potential values for a range of cliques in $f$. For example, if the MRF was required to ensure that most IEs are in the same segments as their neighbours, then the cliques would be every pair of direct neighbouring IEs and their prior clique potential would be evaluated from a binary function such that matching segment allocations would have 0 energy, and differing allocations would have some positive value greater than 0. These prior clique potentials are summed across all cliques to give the total prior energy as shown below.

$$U(f) = \sum_{c \in C} V_c(f_c) \tag{4.12}$$

where $C$ is the set of all cliques (of any size or shape as required)
$V_c$ is the prior potential function for a clique $c$
$f_c$ is the elements of labelling $f$ in clique $c$.

The likelihood distribution, $p(\mathbf{r}|f)$, can also be expressed as an exponential function which is based on a likelihood energy function:

$$p(\mathbf{r}|f) = Z_l^{-1} \times e^{-U(\mathbf{r}|f)} \tag{4.13}$$

Here the likelihood energy function, $U(\mathbf{r}|f)$, is similar to the prior energy function above, but is based on the errors expected in the imaging modality such as noise. This energy function can be equally complicated to construct due to the variety of effects which can be compensated for. For example, Gaussian noise with a 0 mean and $\sigma$ standard deviation can be included with the function shown below:

$$U(\mathbf{r}|f) = \sum_{i \in d} \frac{(r_i - f_i)^2}{\sigma} \tag{4.14}$$

where $d$ is the set of all site positions
$r_i$ is the observed value at site $i$
$f_i$ is the labelling value at site $i$.

Both the prior Gibbs distribution equation (4.11) and the likelihood distribution equation (4.13) can be substituted into the Bayesian posterior distribution equation (4.10) to give the posterior distribution as a Gibbs distribution:

$$\mathbf{P}(f|\mathbf{r}) = Z^{-1} \times e^{-U(f|\mathbf{r})} \tag{4.15}$$

where the posterior energy function, $U(f|\mathbf{r})$, is given by:

$$U(f|\mathbf{r}) = \frac{U(f)}{T} + U(\mathbf{r}|f) \tag{4.16}$$

As the normalising constant, $Z^{-1}$, is identical for all possible labellings of an image, the MAP labelling can be found by deducing the minimum posterior energy function. Therefore equation 4.8 can be simplified to:

$$f^* = \arg\min_{f \in \mathbf{S}} U(f|\mathbf{r}) \tag{4.17}$$

### Posterior Energy Minimisation

For a MAP labelling to be calculated the posterior energy function 4.17 must be minimised. There is no single process for achieving minimisation for a MRF but some common approaches are: simulated annealing [34]; iterated conditional modes [106, 133] and $k$-means clustering [92], all of which are iterative techniques that seek to find an optimal labelling through trial and improvement.

Once MAP labelling has been established, each IE can be assigned a segment based on its labelling in $f^*$.

### Markov Random Field Conclusions

The primary advantage of segmentation using MRFs is the quantitative way in which it allows for complex constraints to be included in the segmentation process, something not possible under other approaches. Constraints can be based on well established contextual foundations which can be applicable across many different tissue regions and even imaging modalities.

Another useful aspect of MRFs for this application is that the final MAP value gives an identification of how 'good' the segmentation is likely to be. This information can be incorporated into the classifier at the next stage allowing decisions to be weighted according to how likely it is that the segment geometry is correct.

I think that a likely problem with this technique however is the complexity of constructing effective MRFs which incorporate all of the required information. The degree of quantitative control over constraints is a large advantage, but constructing them in a way which benefits the most from this control is highly complex [55]. Rajapakse, Giede

and Rapoport [106] stated that they were able to build a model to achieve good results with only three tissue types but expected it to be less successful with more tissue types (as is the case in this application). A further limitation of this approach is that the amount of detail required in the MRFs to allow a good segmentation with noisy, distorted and complex MR images means that operation is likely to be computationally intensive [55].

## 4.3 Structural Techniques

Structural segmentation techniques attempt to identify segments in an image by utilising structural information such the locations of edges and surfaces. These techniques use IE values (as with stochastic techniques) but apply functions to these values and their neighbours in order to perceive some property of the structure of the image at a point.

### 4.3.1 Edge Detection

The basis of segmentation using edge detection is that neighbouring segments should have significantly different MR values at the IEs either side of their shared boundary. Theoretically, the difference between values across the boundary should have a large enough differential value to indicate the occurrence of an edge. Identifying areas with high differential values is not in itself sufficient for segmentation because there is nothing in this process which combines multiple points identified as edges into a single contiguous edge. Therefore segmentation by edge detection is a two stage process. First the gradient magnitude of each IE is determined (*edge element extraction*), and then individual edge elements are combined to form complete edges (*edge element combination*) [33].

#### Edge Element Extraction

There are almost as many edge detection techniques in the literature as there are segmentation techniques, but due to the perceived limitations of this technique as a sole method of segmenting MR images [55] only the common and effective techniques will be discussed.

Marr and Hildreth [72] state that the main difficulty in identifying edges in images using value differentials is that the value changes across boundaries occur at multiple scales. For example, an edge between two similarly valued regions will have a much lower differential than one between two vastly differently valued regions, but both can be legitimate edges. In an attempt to solve this problem Marr and Hildreth proposed an edge detection method which identifies zero crossing points in a Laplacian of Gaussian (LoG) filtered image as edge locations. LoG filtered images are created by the application of two filters, a Gaussian filter (as explained in Appendix C) and a second directional derivative (Laplacian) filter. Utilising a Gaussian filter means that differentials occur across a limited range and are therefore comparable across the image. Taking the second

differential means that zero crossings will only be located at points in the image where there are sharp changes in element value.

Sobel [25], Roberts [108], Prewitt [104] and others have proposed edge detection operators based on the first order differential of the IE values surrounding a point of interest. These operators are typically implemented by convolving a kernel of some form with every IE to allocate it a new value based on some directed gradient property. The Sobel operator, for example, is a 3×3 kernel[6] as shown below:

$$\text{Vertical Edge Sobel Kernel} = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

When this operator is convolved with an image centred on a specific IE the value output is the gradient of a vertical (i.e. parallel to the *y-axis*) edge running through that element. If there is a high contrast vertical edge then there will be a large difference between the element values to the left and to the right of the IE and therefore the convolution will produce a high value. Conversely if the area is homogeneous (i.e. no vertical edge) then the elements to the left will be the same as those to the right and therefore the convolution will produce a zero value. This kernel is only able to detect the presence of vertical edges and therefore, as with Roberts' or Prewitt's operators, there is also a modified version for horizontal edges (as above but rotated clockwise by 90°).

It has been shown [63, 138] that much of the edge detection theory for two dimensions can be extended into three dimensions by replacing the two dimensional edge operator with a three dimensional kernel to be applied in the $x$, $y$ and $z$ directions.

### Edge Element Combination

The edge detection techniques described above are only able to provide locations of possible edges in the image and they do not guarantee that these edges join together such that they outline distinct segments or even form continuous contours. Davis [26] identifies two types of method for constructing continuous edges based on operators such as those explained previously. Davis calls the first set of methods "parallel", where edge operators are applied across the whole image in parallel to ascertain the gradient of every IE, and then the gradient values are used to create a contour through the image which passes along as many high probability edges as possible. An example of such a method is Martelli's heuristic search algorithm [73] which utilises a graph search to identify contours which minimise a cost function. In this case the cost function can be set so that it is dependent on contour properties such as length and smoothness.

The second set of methods identified by Davis are "sequential", where edge detection operators are not applied initially and are instead implemented as required. This method allows the edge operators to be modified so that they incorporate the edge values calculated previously in the sequential pass. The incorporation of previous values in

[6] When the image is two dimensional.

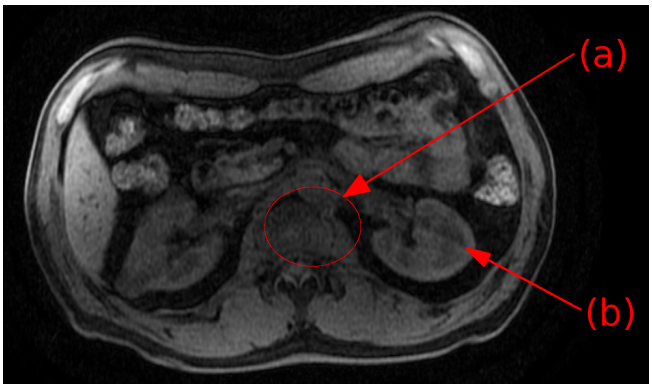


Figure 4.2: Human torso MR image showing low segment boundary gradient (a), and high internal texture gradient (b).

the same edge allows the algorithm to build a characteristic 'profile' of the edge so that new elements added to the edge can be better matched.

### Edge Detection Conclusions

Using edge detection for segmentation is shown in the literature to achieve good results, particularly with respect to the segment localisation. There are however several drawbacks to this technique aside from the common edge detection errors (offset edges, double edges etc.) when it is considered for MR images. Small image artifacts are very difficult to overcome with edge detection because on the local scale at which edge detectors consider the image, artifacts often have well defined and contoured edges. Therefore using edge detection for segmentation is very likely to identify image artifacts equally as well as 'real' tissue structures.

Another key limitation of edge detection for MR image segmentation is that only the image gradient is used to determine where segment boundaries lie. In MR images (as illustrated in figure 4.2) it is common for neighbouring tissue structures to have similar MR values and hence have a low boundary gradient, while other tissue structures can have an internal 'texture' or intensity homogeneities which display a large gradient. Tang et al. [123] implemented an edge detection algorithm for brain images, but had to use multi-spectral images (i.e. $T_1$ and $T_2$ simultaneously) to ensure that gradients were sufficient in all required areas. The results for this method showed good region definition, but even with the added complexity of multi-spectral images Tang et al. conceded that in practice the technique would probably have to be used as a preprocessing step to another technique such as deformable models in order to achieve the intended accuracy.

As with many of the segmentation techniques discussed, the performance of edge detection is dependent on the SNR of an image [61]. The reason for this is that edge operators only consider a small amount the image at any one time, and therefore a change in just one IE through noise can have a large effect on the calculated gradient and subsequent edge element combination.

### 4.3.2 Deformable Models

Deformable models (sometimes referred to a morphological models) can be used for segmentation by positioning them 'inside' the image such that they coincide with, and hence identify, segments of interest. In this process a model can be either a closed contour (in 2D) or surface (in 3D) which is constructed in such a way that it can be deformed to match its boundaries with those in the image [49, 75]. The two main types of deformable models are *energy based* and *probabilistic*, the processes for which are described below.

#### Energy Based Models

In energy based deformable model segmentation the model is considered as an elastic body which is deformed by an imbalance between internal and external forces so that it takes on the shape of a segment. The internal forces specify the 'relaxed' geometry of the segment and try to keep the segment in this shape. The external forces are produced by the model's relation the raw image and try to enforce typical segment properties such as having high gradient values at the boundaries. The model is implemented so that internal and external forces induce an energy in the model which represents the amount by which the model is away from its ideal structure and position [124, 75].

The aim of this technique is to create a model which has minimal total energy, $\mathcal{E}$, which is defined as:

$$\mathcal{E}(\mathbf{v}) = \mathcal{S}(\mathbf{v}) + \mathcal{P}(\mathbf{v}) \tag{4.18}$$

where $\mathbf{v}$ is the model representation
$\mathcal{S}(\mathbf{v})$ is the internal (deformation) energy in the model
$\mathcal{P}(\mathbf{v})$ is the external (segment validity) energy in the model.

The internal energy is typically (though not exclusively) calculated from two aspects of the model which define the general shape of a good segment. These are tension and rigidity [75]. The tension component tries to limit any extraneous contortion in the model's boundaries which are caused by the model being over-fitted to erroneous or noisy areas. The rigidity component controls the rate at which profile changes can occur to ensure that there is a degree of homogeneity across the segment surface. The typical equation for calculating the internal energy is shown below [75]:

$$\mathcal{S}(\mathbf{v}) = \int_0^1 \underbrace{w_1(s)\left|\frac{\partial \mathbf{v}}{\partial s}\right|^2}_{\text{tension component}} + \underbrace{w_2(s)\left|\frac{\partial^2 \mathbf{v}}{\partial s^2}\right|^2}_{\text{rigidity component}} ds \tag{4.19}$$

where $s$ is the parametric domain, $\{0,1\}$, on which the model is constructed
$w_1(s)$ is the tension weighting function
$w_2(s)$ is the rigidity weighting function.

The external forces act to align the model with the raw image using the following energy function:

$$\mathcal{P}(\mathbf{v}) = \int_0^1 P(\mathbf{v}(s))\, ds \tag{4.20}$$

where $\mathbf{v}(s)$ is the parametric position in the image domain
$P(i)$ is the scalar potential function of a point, $i$, in the image domain.

In this equation the scalar potential function is defined so that its value is low when the model segments the image well and high when the segmentation is poor. What defines a good segmentation is decided specific to the application, but two common examples are maximum edge gradients and maximum internal values.

### Minimising Energy Function

To minimise the energy function in energy based models mathematical optimisation techniques such as simulated annealing [68] can be used. The optimisation processes used require starting points. If the algorithm were initialised randomly (as is common practice for such techniques) then the model would most likely be fitted to the geometric structure closest to its random start point. This structure could be an organ, but is equally likely to be an area of noise at the periphery of the image. For this reason deformable models must be initialised manually by a user. The user can select a point on or close to a segment and the model will be constructed from there.

Once the energy function has been minimised the model can be converted back to the standard segment representation by labelling every IE within the model with a segment tag. Alternatively the model can be left in the geometric form as this is often advantageous for further processing such as classification.

### Probabilistic Models

Probabilistic methods are only superficially described here as they are very similar to MRFs described above and there is much less literature surrounding them than for energy based methods. The process for segmentation using these models follows an almost identical path to that for a MRF but rather than trying to identify a MAP labelling of the image, the MRF is used to identify a MAP model formation (geometry and position) [75, 122].

Instead of the MRF's prior distribution defining what constitutes a 'good' segment, probabilistic models define what constitutes a 'good' model to segment mapping using constraints similar to those employed in the energy based models described above. For example, some of the MRF terms penalise models which have convoluted boundaries to ensure that the model is smooth,. This is comparable to the tension term in the energy based methods.

#### Deformable Models Conclusions

It should be noted that deformable models are ultimately just a tool for implementing segmentation because they rely on definitions for the energy functions which are based around standard techniques such as image gradients and homogeneous valued regions.

Their advantage is that they construct naturally shaped segments with smooth boundaries. The internal force components are able to sustain such a shape in the presence of noise and intensity inhomogeneities. In addition, the initial energy functions can be constructed in a way that incorporates *a priori* knowledge about what the segment's geometry might be [75]. This means that the internal forces will attempt to structure a segment in a prototypical shape for the given tissue type and then the external forces can manipulate this structure by the small amount required to match the image.

The primary difficulty for deformable models is that each region model must be initialised, a task which is generally performed by a human user. Applications exist that have been designed to operate in a technique pipeline (as explained in §4.4) so that an initial rough segmentation is carried out by thresholding for example, and then the segment centres identified are used to initialise the deformable models. These approaches have been used with some success but as discussed in §4.4.5, a more promising implementation is the hybrid technique *atlas based segmentation*.

A further problem with deformable models results from their difficulties with certain geometric segments. Due to the internal forces, many deformable models struggle to converge when there are concave segment boundaries, long tube-like segments or segments with many irregularities or protrusions [75].

## 4.4 Hybrid Techniques

Hybrid techniques attempt to take the advantages of both stochastic and structural approaches to achieve better results by including more information in the segmentation process. These methods incorporate both statistical and structural analysis of the images though the ways in which these analyses are combined varies between applications.

Commonly the two types of data are considered simultaneously, but there are a set of techniques where a *pipeline* approach is employed to implement different techniques sequentially. The name "pipeline" comes from the way that an image is processed to some degree by one technique and then the result from this is passed to another technique to further improve it [8]. A pipeline can include as many different techniques as required. Numerous approaches to segmentation rely on the pipeline structure, but many of these are slight extensions on the main techniques covered and therefore are only explained where necessary at the implementation stage.

### 4.4.1 Region Growing

One category of segmentation techniques considers segments as a collection of IEs which satisfy a connection criterion. This collection of connected IEs is referred to as a *region* and there are multiple methods for developing regions in the literature. Region growing is one such technique where the region is 'grown' from seed points placed in the image. The process for using region growing in image segmentation is described below.

#### Region Initialisation

A seed point is typically a single IE and there can be one or many seed points placed in the same image depending on the specific application. Most versions of region growing defined in the literature utilise manually specified seed points where the user is required to place the point inside the region or regions which require segmentation [1, 33, 38, 55].

Alternatives to user generated seed points include placing a number of points randomly [58], selecting histogram extrema [38] and generating them intelligently as the algorithm progresses in locations with low region density [38].

#### Uniformity Criterion

The region growing process is based on a uniformity criterion which identifies when an IE is suitable for joining with another IE or region. There are a number of different criterion which can be constructed for assessing suitability, but the most common is *absolute difference* (sometimes called *average brightness*) [38, 97]. This criterion operates by calculating the difference between the candidate element's value and the values already included in the region (as shown in equation 4.21). If the difference is less than some threshold, $T$, then the region created by including the candidate would be considered uniform and would be acceptable. The threshold $T$ can be set manually based on some knowledge of the probable image construction or can be calculated automatically from the image's intensity histogram.

$$\Delta(R, C) = \max(maxval(R) - val(C), val(C) - minval(R)) \tag{4.21}$$

where $\Delta(R, C)$ is the absolute difference between region $R$ and candidate element $C$
$maxval(R)$ is the maximum IE value in the region $R$
$minval(R)$ is the minimum IE value in the region $R$
$val(C)$ is the value of IE $C$.

#### Region Connectivity

In addition to the uniformity criterion, some connectivity property must be specified for the IEs so that those which are spatial neighbours can be identified. In most applications

the image is two dimensional, and the connectivity property is taken as the Von Neumann neighbourhood (i.e. 4-connected) [96], but there is no reason to limit the property to this and much more complex connectivity schemes can be used. For this application the three dimensional version of the Von Neumann neighbourhood (i.e. 6-connected) could easily be implemented to account for the image's depth component.

### Region Growth

Once the uniformity criterion and element connectivity have been specified the process of region growing is a simple iterative process which starts with the seeds considered as single point regions. All of the neighbours for each region are then checked for whether they are suitable for joining with the region using the uniformity criterion. If a neighbour can be combined with the region to form a larger uniform region then the neighbour IE is incorporated. This process is then applied again with the new larger region so that second degree neighbours can also be incorporated if they fulfil the uniformity criterion. This is then repeated until every neighbour fails the uniformity criterion and the algorithm is complete [97]. If at any point in the process two regions become neighbours they can be joined together as long as every point in one region passes the uniformity criterion for the other region.

### Region Merging

Region merging is a slight variation on region growing which avoids the requirement for seed points and ensures that every point in the image is included in a region. It can be implemented in the same way as region growing with the exception that every IE is considered as a seed point for the first generation. This allows any two neighbours to join together at any point in the algorithm, allowing for a more even distribution of regions across the image [96].

### Region Growing Conclusions

There are many advantages of the region growing approach: it is guaranteed to create contiguous regions; it is very robust in terms of segment geometry and it lends itself well to the pipeline approach. However the applicability of region growing/merging depends largely on the uniformity criterion used. The most common method, absolute difference, suffers from many of the same problems as thresholding in terms of its inability to overcome region inhomogeneities and noise. While there are other criteria which can be used, these are very rare in the literature, and it appears that most applications which cannot operate with absolute difference utilise another method altogether rather than a more complicated uniformity criterion.

As region growing starts at the individual IE level, it is very susceptible to noise which can cause errors and prevent the correct growth of regions. At the initial stage, when each region is a single IE, if any one is influenced by noise then when it is combined

with another it will have a proportionately large effect on the region's properties and can prevent any further elements from being included. Methods such as *region splitting* and *split and merge* attempt to overcome this by considering the image at a larger scale so that there are more values per region, and any single noisy element will have less of an effect [97].

In its basic form, region growth does not actually guarantee that the segmentation produced will fulfil the first segmentation requirement identified in equation 4.1. It is possible that the growth process will terminate (i.e. when no more neighbours fulfil the uniformity criterion) before every IE has been included in a region. This limitation means that an additional stage of processing is required after termination to check for completeness and to correct un-segmented areas in some way.

Though not stated explicitly in the literature I believe that this process would be prohibitively slow to run on medical image volumes such as those being considered for this application. In an image with 26,000,000 IEs comparing each one with its neighbours is likely to take a substantial amount of time, particularly if the uniformity criterion is complicated.

### 4.4.2 Region Splitting

Region splitting is a region based segmentation technique which tries to segment an image by repeatedly splitting regions until they are considered uniform [88]. Much of the literature considers region splitting a weaker method than the other region based ones, and therefore only a basic overview is provided here [38, 96].

Region splitting requires a uniformity criterion almost identical to that used in region growing, with the only difference that just the single region is checked rather than a region and a candidate point. For example, if the absolute difference criterion were applied it would require that the maximum value in a region minus the minimum value in the region is less than some predefined threshold, $T$. As with region growing other criterion can be applied as required for a specific application.

The splitting of a region at a given step can be achieved in any way that is required for the application, but for a two dimensional image the split is generally into four quarters [97]. This method is widely used because it always creates rectangular regions which are straightforward to split at the next level. For a three dimensional application the quarter split can be expanded to an eighth split as shown in figure 4.3.

Region splitting is carried out by starting with a single region (the whole image) and then repeatedly checking all regions for uniformity and splitting those which are non-uniform down into smaller regions. The process will eventually terminate when every region in the image is uniform, even if the image is split down to the individual IE level by this point. This process is equivalent to creating an image split tree, where the root of the tree is the full image, the first set of children are the range of sub-regions created by splitting the whole image. Each sub-region then has a set of children of its own which correspond to a split of the sub-region and so on. This is shown in figure 4.4.

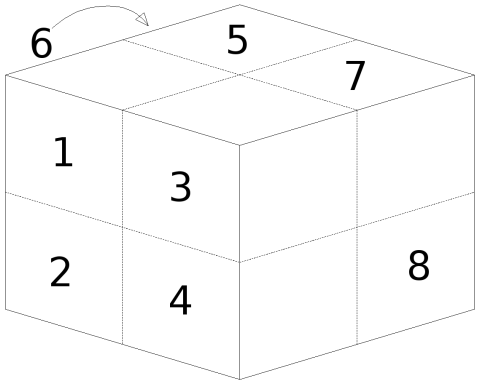


Figure 4.3: Three dimensional image volume split into eighths.

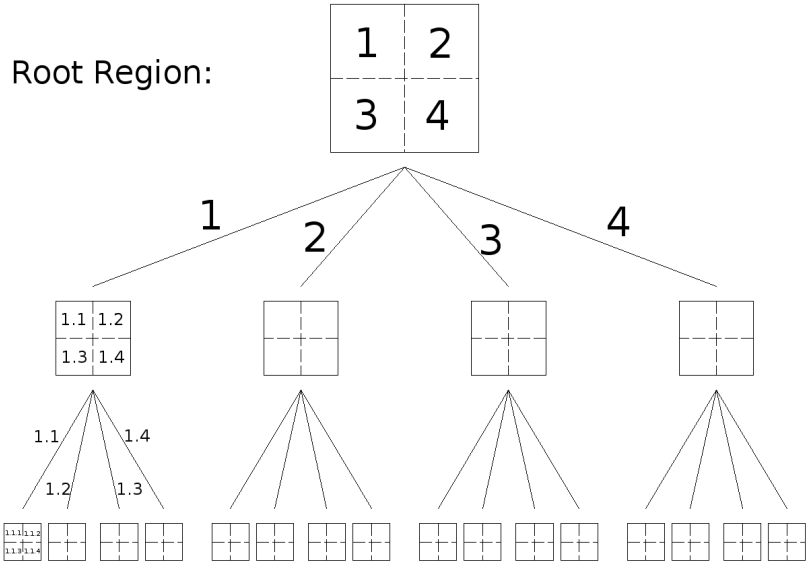


Figure 4.4: Image split tree for a two dimensional image which is split into four children at each node (i.e. a quad-tree)

#### Region Splitting Conclusions

The key advantage of region splitting over region growing is that when considering a region for uniformity, the region is typically much larger than a single point, and therefore noise has a smaller impact on the decisions made [96]. In addition, this process does not require seed points and can fully segment the whole image without any manual interaction.

The fundamental flaw with this technique in its most simplistic form is that it does not identify when neighbours are uniform together unless they are siblings in the image split tree. In fact, even if they are siblings in the image split tree they will only be interpreted as a single region if every sibling matches the same uniformity criterion. This means that the process commonly generates over segmented images unless the region borders happen to align with the straight lines generated by the region splits. *Split and merge* is the common method used to overcome this restriction and is described below.

A further problem with this method is that the fine detail in images does not often align with the split boundaries, and therefore many splits and uniformity checks have to be carried out before the main detail in the image can be identified. These 'wasted' steps lead many to state that region splitting is a computationally inefficient method [38, 96].

### 4.4.3 Split and Merge

Due to the limitations of both *region growing* and *image splitting* Horowitz and Pavlidis proposed a method which combines both in order to maximise each of their advantages [44]. Split and merge operates by taking an arbitrary set of regions, and then repeatedly either splitting or merging the regions until all of the regions in the image fulfil a predefined uniformity criterion. The following explanation of the split and merge process is reworked from references [44, 96, 97].

#### Initialisation

In order to limit the amount of memory required for an implementation of split and merge, Horowitz and Pavlidis' method does not generate the whole image-split tree (shown in figure 4.4), and only stores the equivalent of one layer[7] at any one time.

The start level is specified manually, and defines the layer in the image split tree at which the algorithm starts. The reason for starting in the middle of the tree rather than at either the root or the leaves is that it is expected that actual regions will be discovered at an intermediate range in the tree. Therefore starting at this level is likely to improve computational efficiency. Once the level has been specified the number of regions in the image at that level can be calculated and the image can be divided up as required.

[7]Note that "equivalent of one layer" means that all regions only ever add up to a single instance of the image, therefore the regions in memory can come from different layers in the image-split tree.

### Merging

The first stage in the split and merge algorithm is almost identical to that for region merging, with the only difference being that the process begins with the regions from the start level in the image split tree rather than at the individual pixel level.

As with region merging this process can use any uniformity criterion that is required and completes when there are no more regions in the tree which would pass the uniformity criterion if combined with all of their siblings.

### Splitting

Once the merging process has terminated the next stage of the algorithm splits every region down until it passes the uniformity criterion in a process similar to region splitting. The only difference from the standard region splitting algorithm is that it starts with the regions remaining after merging rather than a single region representing the whole image.

The uniformity criterion used for the splitting process is commonly the same one used for the merging stage though it can be modified if required. As with region splitting, the process continues until every region in the image passes the uniformity criterion.

### Grouping

The grouping stage is the key modification for split and merge which produces better results than pure region splitting. As stated, region splitting can identify neighbouring regions as different even if they would pass the uniformity criterion together just because they are from different parents in the image split tree. For region grouping the image split tree is discarded, and every region in the image is checked with every one of its neighbours for uniformity and they are combined if they pass the uniformity criterion. The process for selecting which neighbours to check first is non-deterministic in Horowitz and Pavlidis' method and they only state that every region must be checked against every one of its neighbours.

As with the splitting process, the grouping process typically uses the same uniformity criterion as region merging. The grouping process continues until all region-neighbour pairs have been checked for joint uniformity.

Once the grouping algorithm has terminated, the segmentation process is complete and points in the image can have their segment allocated based on which contiguous region they have been grouped into.

### Split and Merge Conclusions

Split and merge is generally considered to have all of the advantages of region merging and region splitting whilst minimising the disadvantages. I believe that the most important advantage is that split and merge is a robust technique, and in its basic form does not

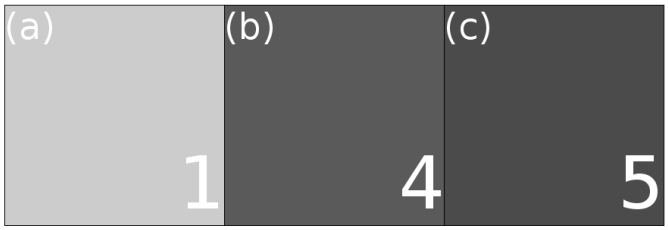


Figure 4.5: Image showing split and merge bias. It can be seen that regions (b) and (c) have close values and are likely to form a uniform region, but if regions (a) and (b) are siblings in the image split tree then they will be compared first. If the uniformity threshold is 3 then they will just pass and will be joined together. Due to the range between (a) and (c) being greater than the threshold, region (c) will never be able to join the region and will hence be segmented separately from (b).

place any restrictions on the formation of regions. It can therefore adapt very well to difficulties such as inter-patient variations.

As with region growing, the main factor controlling split and merge's performance is the suitability of the uniformity criterion used, but most of the literature focuses on absolute difference based measures and there is little available to suggest what would be the best uniformity criterion for use on MR images.

Split and merge improves on the computational efficiency of both region splitting and region merging, but due to the relatively low-level way in which it considers the image, a large amount of data is evaluated and performance can be affected.

One important limitation of other region based methods which split and merge does not solve is that it is, in its most basic form, unable to overcome intensity inhomogeneities due to its reliance on the inflexible absolute difference uniformity criterion.

As split and merge uses an image split tree for the first two stages, the process is inherently biased towards locating region edges along vertical and horizontal lines, and particularly lines which correspond to divides between high parent regions in the tree. This is because the algorithm will always check uniformity between siblings before it checks uniformity between non-sibling neighbours. This means that it may join regions which have 'reasonably' close values before it has a chance to check either region with another which may have an almost identical value. Figure 4.5 illustrates this bias.

### 4.4.4 Watershed

Watershed based segmentation is a method which is technically region based, but with similarities to edge detection and thresholding. It was initially proposed by Beucher and Lantuejoul [12] as a method for detecting non-parametric contours within grey-scale images. Because of the similarities between watershed segmentation and the region based techniques already described it will be discussed only briefly here.

Watershed based segmentation considers a two dimensional image as a three dimensional

topological relief where the height at each point is defined by its grey-scale value. A parallel can be drawn between this relief and a geographic landscape containing multiple water catchment basins such that the problem of segmentation becomes one of identifying the watershed lines which enclose each catchment basin. A catchment basin in this definition is an area of the image such that if one applied a hill descent from any point within it then it would finish at the same local minima. Basins are analogous to segments because they indicate areas of the image which are spatially contiguous and have values which are more similar to each other than other segments. Watershed lines surrounding segments are commonly identified by two approaches, *flooding* and *rainfalling* [37].

Flooding identifies watershed lines by simulating filling the landscape up with water fed in at each local minima. The concept is that, as each local minimum is filled, the water level will climb up the catchment basin until it comes into contact with the water which has been fed in from another minima. When two bodies of water meet the lines between them must be the watershed and are identified as such. The water levels are raised until the whole landscape is full and every catchment basin identified by a watershed.

Rainfalling simulates the path that a raindrop would take from each point in the image by following the path of greatest gradient until it reaches a local minima. Every point in the image which feeds into the same minima is then considered as a catchment basin and its edges are the watersheds.

Typically this process is applied once an image has been passed through a gradient magnitude filter[8]. This means that homogeneous sections of the image correspond to low-lying areas in the landscape which are commonly surrounded by high peaks representing the high gradient values of region edges.

#### Watershed Conclusions

The use of watersheds on gradient images is a novel way of overcoming the difficulties of joining edges in standard edge detection. Moreover, the watershed algorithm itself does not require any parameter settings which makes the operation very robust and reliable.

Watershed thresholding also has many of the same advantages as regions based techniques, but in this form it is very prone to over-segmenting images. As this method constructs watersheds for each local minima in the image it means that there will be as many regions segmented as there are local minima. For a noisy image there can be lots of these minima which makes this process unsuitable. This noise sensitivity means that images must be filtered thoroughly before segmentation to avoid producing too many regions. There are methods described in the literature for overcoming the over segmentation such as waterfall [11] and marker based [78] methods, but these increase complexity and reduce the algorithms generality.

[8]This filter is equivalent to an application of edge detection operators across the whole image and replacement of each image value with its gradient magnitude (see §4.3.1).

### 4.4.5 Atlas-Guided

Atlas-Guided segmentation is not a single technique, but a group of methods based around the concept of using an anatomical template to inform the segmentation process. In the literature atlas-guided segmentation techniques are almost exclusively in used medical applications due the way that the constituents of a medical image are a reasonably stable between patients. They are the most commonly used in neural images [2, 3, 93, 103] which exhibit the minimum amount of inter-patient variation.

There are various approaches to implementing atlases for segmentation, some of which are designed to improve some of the standard methods discussed in this chapter such as deformable models [132] and region growing [95], but the most common method for implementing an atlas guided segmentation is to treat it as a registration problem.

#### Registration Based Approach

In registration based atlas-guided segmentation a pre-segmented image is mapped (through registration) onto the image which requires segmentation in a way such that the segments from the pre-segmented image overlay and define the corresponding segment of the same type in the un-segmented image. The key process in this approach is the method for registering the pre-segmented image onto the un-segmented one.

Park [93] defines registration as the process of finding "the best transform $T$ such that both data sets are best aligned by a particular similarity measure", with the formalised definition:

$$T^* = \arg\max_{T \in F} \mathrm{SIM}(A(\bullet), B(T(\bullet))) \tag{4.22}$$

where $T^*$ is the best (estimated) transform
$A(x)$ is the reference data set (i.e. the pre-segmented image)
$B(x)$ is the floating data set (i.e. the image to be segmented)
$F$ is a family of all possible image transforms
SIM is a binary degree of similarity function.

Park goes on to state that multiple definitions of the similarity function exist but *mutual information* is the primary method for registering inter-modality images, while the *correlation coefficient* is used when the comparison is intra-modality. The two main methods for transforming images are *linear* and *non-linear* (sometimes referred to as warping) transformations.

Linear transformations are only able to translate and rotate objects and are therefore unable to be applied to 'soft' or complex structures such as are found in medical images. Non-linear transformations extend the properties of linear transformations and include the ability to both scale and deform the images to achieve the optimum registration. The methods for achieving this are numerous but the majority are based on the principle of defining a mathematical function which can represent the intra-image free-form-deformation quantitatively [111]. The deformation is constructed so that structurally

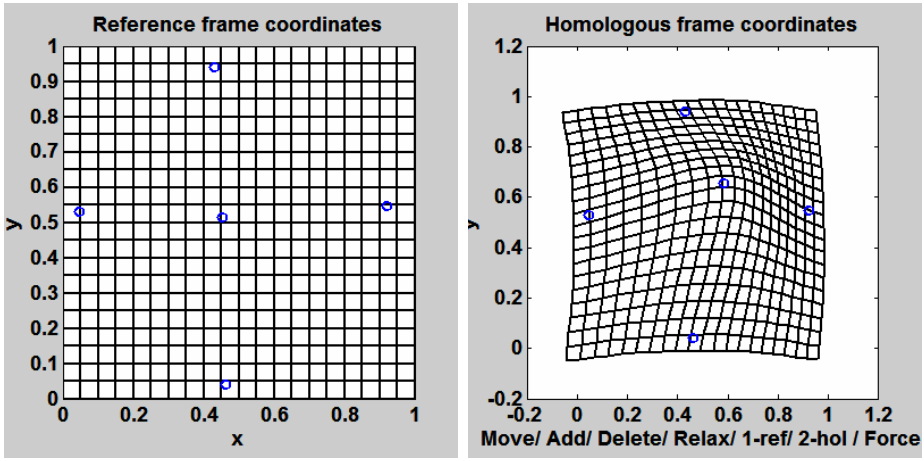


Figure 4.6: Images showing the deformation represented by a non-linear transformation from the pre-segmented image (reference frame) to the un-segmented image (homologous frame) [111].

identifiable points (control points) in each image match-up. An example deformation function is shown in figure 4.6, where it is essentially a representation of the uniform Cartesian grid from the pre-segmented image as a set of curves in the un-segmented image.

The way that the transformations are optimised so that the similarity function is maximised is dependent on the technique and application, but typically the process is based on modifying free-form-deformation parameters until the control points are correctly aligned. This is a standard optimisation problem and can be processed using one of the iterative or gradient descent algorithms [111].

Once the optimal transform has been found the process of extracting the segments from the un-segmented image is straightforward and the corresponding segment identifiers from the pre-segmented image can be transferred across.

### Probabilistic Approach

Another common approach to deformable model segmentation is to incorporate a probabilistic component so that greater patient variation can be overcome. This method is based around constructing a probabilistic atlas which represents the expected probability distribution of tissue structure geometries and is defined from statistical analysis of a large set of pre-segmented images. This atlas is then registered onto the image which requires segmentation and rather than a standard linear or non-linear transform, the transform is carried out to maximise the registration probability. The maximisation can be carried out via various methods but MAP is often used [39, 94, 125].

### Atlas-Guided Conclusions

An important property of atlas-based segmentation is that each pre-segment in the atlas can very easily be labelled with the name of the tissue structure it represents and these labels can be transferred across to the un-segmented image during registration [101]. For this application the ability to do this is very useful as it would negate the requirement for a final classification stage and the model would be ready for use immediately.

Atlas-based approaches are a useful framework for incorporating a large amount of information into the segmentation process. As explained previously, the only way to guarantee a good segmentation is to employ prior knowledge and this is an area where atlas based methods are likely to perform well. The problem is that the knowledge is imparted through the construction of each atlas which can be complicated. The usefulness of an atlas is based on its representation of the population of all possible tissue structures, and therefore the only way to make sure the atlas is representative is to process a large amount of data.

According to Pham, Xu and Prince [101] atlas guided segmentation is most suitable for structures which are geometrically stable across most patients. This is due to the inflexibility of the registration process which is unable to overcome significant differences between the images, particularly when the structures of interest are elongated or complicated. As this application is intended to be robust in the presence of patient variation it is thought that the limitation of atlas guided segmentation to just geometrically stable structures could make it incompatible.

As part of Park's investigation of the limitations of 'classic' non-linear registration he states that for an abdominal image one would typically need thousands of degrees of freedom in the free-form-deformation and this degree of complexity is likely to be too computationally expensive [93]. Park and others suggest methods for overcoming this problem with non-linear segmentation, but their approaches do not achieve equivalent results, particularly when the amount of deformation is variable across the image.

There are many other segmentation techniques described in the literature some of which are based on the standard techniques above and some which take a completely different approach. No more will be detailed here, but useful additions and modifications will be explained if they become required during implementation.

## 4.5 Evaluation and Selection of Techniques

### 4.5.1 Important Considerations

It is clear from the literature that there are no methods available which have been shown to perform well across all tissue types, patients, imaging errors and modality settings for MRI. Instead there are a range of factors which must be considered and balanced in order to select the technique best suited for this application. An explanation of the key considerations follows.

#### Utilisation of Knowledge

From the literature surrounding various segmentation techniques it is apparent that the degree to which they utilise specialised *a priori* knowledge is a very important factor, with a large bearing on the segmentation that is produced. It is known from Wolpert and Macready's no free lunch theorems that the only way to guarantee optimal segmentation performance is to utilise large amounts of knowledge.

Almost all segmentation techniques require some modification for use, whether it is setting a threshold or constructing a complete probabilistic atlas, but it is clear that some incorporate more specific knowledge than others. For example, the level of knowledge incorporated into a threshold value is likely to be very general, such as knowing what the image modality is and therefore knowing what the likely range of values will be. Creating a probabilistic atlas on the other hand requires a large amount of information about the statistical distribution of tissue structure geometries. From this it would be logical to deduce that the best segmentation technique for this application would be the one which can be created so that it includes as much information about the MRI modality and patient anatomy as possible. However, I believe that the conclusion would not be so assured.

All of the medical implementations described in the literature which utilised knowledge intensive techniques (such as MRFs and atlas-based methods) were limited to small, well studied and patient-invariant regions, which typically were constructed from a low number of tissue types. A good example of this is Lee et al.'s thresholding process for midsagittal brain images [57]. They identify anatomical datum points such as the tip of the nose and then use these datum points to pre-process the image to remove extraneous sections which would confuse the thresholding process. This pre-processing technique requires a large amount of anatomical knowledge. Compared to the torso, the brain and skull are simple structures which do not vary greatly between patients. I therefore feel that the amount of knowledge required to achieve similar results in the torso would be very large. Implementing such a complicated knowledge structure would be inherently difficult and beyond the scope of this project. Moreover, as there is no literature describing such a system I also think that it is beyond the practical scope of current knowledge representation and utilisation techniques.

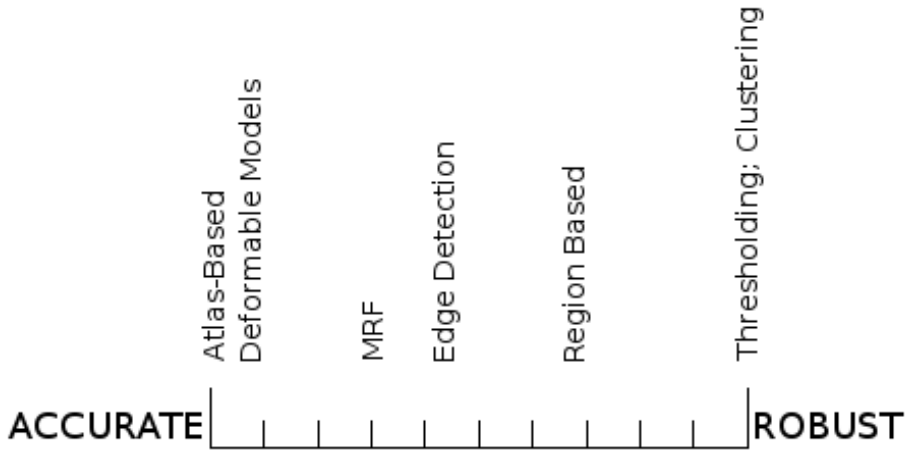


Figure 4.7: Robustness-Accuracy scale for image segmentation techniques

A secondary problem with the utilisation of knowledge is that it makes the process very inflexible towards unexpected variation such as from noise and image artifacts. In addition, tumours for example can take up a large amount of space and force other organs out of the way. There is no way that a knowledge base could account for every possible tumour and therefore such occurrences would cause the process to fail without even partial or semi-useful results.

Since the amount of knowledge which can be included is limited, the problem of selecting a segmentation technique becomes a balance between robustness and accuracy. One can either implement lots of knowledge in a specific area which is likely to make the method very accurate when it segments exactly what it was built for, but inflexible on anything else, or one can implement general knowledge which allows for good robustness at the cost of accuracy in certain areas, or any point in between. Figure 4.7 shows the perceived positioning of the different methods on this scale from my analysis of the literature.

As the aim is to construct an application which can operate on a wide range of data, methods on the robust side of this scale will be preferred for use as they will be more likely to operate effectively in this way.

### Applicability in Three Dimensions

Each segmentation technique considered can be created to work in three dimensions, but some of the methods are more suited than others. A common problem with the extension to three dimensions is the large increase in computational demand as is found with atlas-based based methods. To perform three dimensional segmentation with an atlas based technique you must define a three dimensional deformation mapping which requires an enormous number of parameters, each of which must be developed by a mathematical optimisation algorithm. Segmentation techniques which can operate in three dimensions without a large increase in computational demand will therefore be preferred.

### Modifiability

Almost all of the literature reviewed which attempted to implement segmentation techniques used either slightly modified versions of those described previously or a combination of more than one method in a pipeline approach. I therefore feel that the same approach is likely to be required for this project.

All of the methods can be adapted, but it is more natural for some. For example, the region based methods can be very easily set up to over-segment the image allowing a secondary technique to improve on the process. Deformable models conversely are quite fixed in the way that they are applied because of the construction of each elastic body and their outputs cannot easily be used in another process.

### Computation Time

There is a noticeable correlation in segmentation techniques between the accuracy of segmentation and the time taken to create it. The fastest methods such as thresholding can be computed in real-time, but these methods are inherently simplistic. The alternative is more complicated methods which analyse specific properties of the image at varying scales and in varying positions. These algorithms are typically much more computationally expensive, but make decisions based on a deeper knowledge of the image construction and are therefore considered to be more likely to produce accurate results.

Due to the amount of data in a series of MR images and the complexity that is innate in each image, I do not believe that any algorithm will be able to make well informed decisions in a short amount of time. However, processing time has been given a low priority in the construction of this program because most potential applications will pre-segment images well before use.

### User Interaction

One of my main aims for this project has been for the program produced to operate without any human supervision. During the investigation it was discovered that many segmentation techniques rely on a user interacting in some way with operation. Many of these processes could be adapted to operate without this interaction though this is often at the cost of segment accuracy.

Even though it limits the number of candidate techniques, the criterion for unsupervised operation has been retained as I believe that the advantages of autonomous operation in terms of its wider range of applications outweighs the advantages that supervised methods would bring.

| Segmentation Technique | Robustness [W=10] | Accuracy [W=3] | Applicability in 3D [W=5] | Modifiability [W=5] | Computation Time [W=1] | User Interaction [W=10] | Image Error Resilience [W=6] | Parameter Complexity [W=2] | WEIGHTED AVERAGE |
|---|---|---|---|---|---|---|---|---|---|
| Thresholding | 10 | 1 | 10 | 8 | 10 | 5 | 1 | 9 | 6.60 |
| Clustering | 10 | 1 | 10 | 8 | 4 | 10 | 1 | 10 | 7.69 |
| MRF (with clustering) | 3 | 8 | 8 | 5 | 5 | 10 | 10 | 3 | 6.90 |
| Edge Detection | 5 | 8 | 7 | 3 | 6 | 10 | 5 | 7 | 6.52 |
| Deformable Models | 3 | 8 | 7 | 2 | 5 | 1 | 5 | 3 | 3.57 |
| Region Grouping / Merging | 8 | 5 | 10 | 10 | 4 | 10 | 3 | 9 | 7.98 |
| Region Splitting | 8 | 1 | 10 | 10 | 3 | 10 | 6 | 9 | 8.10 |
| **Split and Merge** | **8** | **5** | **10** | **10** | **5** | **10** | **6** | **8** | **8.38** |
| Watershed | 8 | 5 | 10 | 10 | 5 | 10 | 5 | 8 | 8.24 |
| Atlas-Based | 1 | 10 | 1 | 1 | 3 | 10 | 3 | 1 | 4.12 |

Figure 4.8: Quantitative comparison of segmentation techniques (weighting for each criterion based on the aims in chapter 1 is shown in brackets).

### 4.5.2 Technique selection

A full quantitative comparison of the main segmentation techniques is shown in figure 4.8. I have scored each technique for its suitability across a range of criteria, including the considerations described above, in order to asses which is the most suitable for this application. The scores have been derived from my analysis of each technique in the literature.

From this comparison the highest weighted average is scored by split and merge with the other regions based techniques scoring very similarly. Their key attributes are that they are all highly robust in terms of inter- and intra-patient variability because they do not incorporate large amounts of anatomically specific information. They also satisfy the requirements for extension into three dimensions, modifiability and unsupervised operation.

Out of the top four segmentation methods I decided to implement split and merge for this this application. The primary reason was that much of the literature considers split and merge to be an improvement on both region grouping and region splitting in almost every way. The way in which it combines aspects from both is very effective at taking their advantages whilst minimising their disadvantages. From my review it appears that

the only real price of improved performance is the increase is the number of parameters which must be set. This increased complexity is due to one parameter being required for each of the three operation stages. Split and merge and watershed both exhibit very similar properties with respect to the aims of this project and selecting between them was difficult. Ultimately the decision to use split and merge was due to the fact that it is less dependent on complex preprocessing techniques as it is more resilient to errors such noise which can cause over-segmentation within watershed methods.

# Chapter 5

# Implementation and Critical Analysis of Split and Merge

## 5.1 Implementation

The basic concept of the split and merge algorithm was explained in §4.4.3, but that only covered a very general approach. This chapter describes its implementation with any modifications, improvements or parameters that were required specifically for MR images.

### 5.1.1 Image Split Tree

Clearly using a quad-tree (as described by Horowitz and Pavlidis [44]) would not be suitable for three dimensional segmentation because it would only allow division in two of the three axes at any one step. Manousakas et al. [70] propose the concept of using using an oct-tree[1], but rejected it due to the perceived computational cost of splitting and merging the regions in spite of its perceived advantages in terms of region boundary uniformity. This analysis was made in 1998 and therefore I feel that with the increase in both processing and memory it should now be implementable. In addition, processing time is considered a low priority for this project and therefore the risk of slow operation is worth the improvement for boundary uniformity which comes from using information in all thee dimensions.

### 5.1.2 Region Connectivity

To fulfil criterion 4.2 of the segmentation definition (page 19), each region produced must be connected. As 'connected' does not have a fixed meaning, the concept must be

[1] An oct-tree is where there are eight children per parent so that a region is halved in each axis at each spiting step, see figure 4.3.

specified for each implementation. To do this, a neighbourhood scheme must be defined which identifies which voxels can be considered as adjacent. For this implementation the three dimensional Von Neumann neighbourhood with a range of 1 was selected because it guarantees that any touching regions will only be considered as neighbours if they have an interfacing surface with an area of at least $1 \times 1$ units.

Using this neighbourhood definition a region can be considered 'connected' if there is a path between any two voxels within the region which goes from neighbour to neighbour without passing through any voxels that are not contained within the region.

### 5.1.3 Region Representation

A fundamental component of the split and merge process is a data structure which is able to store the details of each region. Moreover, there must be a method available which is able store the split tree structure used in the merging and splitting stages and a neighbourhood structure which is able to identify region neighbours during the grouping stage.

As they are guaranteed to be rectangular cuboids, the obvious method for storing region detail at the merging and splitting stages is with their position and size. Each region's position can be stored with three integer values identifying the Cartesian coordinates of its lowest corner[2], and its size can be stored with three integer values identifying its height, width and depth[3]. The problem with this representation is that during the grouping stage regions are no longer guaranteed to be rectangular cuboids and therefore cannot be specified in this way.

It is possible to create two region representations and switch between them after the splitting stage, but my approach was to use a single, adaptive representation for a region throughout. This was implemented by creating the Java object `Region`, which can be considered as an array of rectangular cuboids, each of which is encoded as the Java object `CubicVolume`. `CubicVolume`'s store the six integer values explained above to define a single rectangular cuboid region. For the merging and splitting stages only a single `CubicVolume` is associated to each `Region`, but when regions are combined during grouping this is done by merging their arrays of `CubicVolume`s ensuring that the representation defines the entire region accurately. I contemplated using a more complex region object using boundary representations but I believe that the computational cost of precessing each boundary would be prohibitive.

In [97], Pavlidis uses a linked list of regions to store the image split tree, such that there are identifiable links between siblings. This method ensures the tree structure is maintained by only merging when there are four siblings of the same size. Once the linked list had been used for merging and splitting it is discarded for the grouping stage where a new method must be used because all neighbours require identification rather than just siblings.

[2]In this definition the lowest corner is the one with minimum $i$, $j$ and $k$ values.

[3]This is a simple extension of how Pavlidis defines a two dimensional region in [96].

There are various data structures proposed in the literature which aim provide a framework for representing neighbour relations at the grouping stage. *State-Space* and *Low-Level Boundary* approaches are not suitable for this application because implementing them results is a large increase in model size [7], but using a region adjacency graph (RAG) has been shown to operate well [96, 97].

A RAG is a graph where the vertices represent regions and edges represent adjacency relationships between them. To avoid having to switch data structures between splitting and grouping a modified RAG was used. The modified RAG was implemented using the Java object `RegionAdjacencyGraph` which is based around a region list (`RegionList`) and an adjacency list (`AdjacencyList`).

`RegionList` is an extension of a hash map which stores all of the `Region`s in the image with unique integer values as keys. These unique keys are fundamental for the identification of inter-region adjacencies and will be referred to as region identifiers in the following explanation. One modification required for using a RAG as a tree structure is that each region has an additional integer value included which identifies its level in the image split tree[4].

`AdjacencyList` is also an extension of a hash map where the key values correspond to the region identifiers for every region in the region list. Stored against each identifier is another hash map which lists all of the regions adjacent to the region and a level for the adjacency. The adjacency level allocated to each adjacency relationship represents the level in the tree at which the two regions would be considered siblings. This implementation is slightly wasteful on memory because it duplicates every adjacency, but if only one record were stored per adjacency then the only method to find out what neighbours a region has would be to search through every one and check.

Using the region and adjacency levels stored in these data structures the image split tree can be represented in the following way. When the program requires the identity of a regions siblings it can retrieve this information by searching through its neighbours, and selecting those where the adjacency level between them is the same as the region's level in the split tree[5]. Using this RAG structure in combination with the original image, the following implementation is able to identify the grey-scale values contained within any region in the image.

### 5.1.4 Initialisation

My split and merge algorithm is initialised by creating an intermediate level in the image split tree. The actual start level selected has no affect on the results produced by the algorithm, and is only to improve the computation time. For this application various values were tested and level 6 (i.e. $\approx$ 262,000 regions) gave the best results.

[4]For example, a region with a level 0 would be the whole image, a region with a level 1 would be one eighth, a region with a level 2 would be one sixty-fourth etc.

[5]As siblings do not have to be adjacent to each other, the actual implementation of this search function performs a graph search along same-level adjacencies so that it can identify them all.

Algorithm 5.1 shows the initialisation process which is used to generate a RAG, $G$, for the voxel array, $V$, with the start level, $q$.

**Algorithm 5.1** Split and Merge - Initialisation

**Input:** $q$, the start level in the image-split tree
**Input:** $V$, the voxel array representation of the image series
1: $n \leftarrow$ CALCULATE-NO-OF-START-REGIONS($q$)
2: $G \leftarrow$ a new empty RAG
3: **for** $index = 0$ to $n - 1$ **do**
4: $key \leftarrow$ a new unique region identifier key
5: $coordinates \leftarrow$ CALCULATE-COORDINATES-OF-REGION($index$, $V$)
6: $dimensions \leftarrow$ CALCULATE-DIMENSIONS-OF-REGION($index$, $V$)
7: $region \leftarrow$ CREATE-NEW-REGION($key$, $coordinates$, $dimensions$)
8: $adjacencies \leftarrow$ CALCULATE-ADJACENCIES($coordinates$, $dimensions$, $G$)
9: $G \leftarrow$ ADD-REGION($G$, $region$, $adjacencies$)
10: **end for**
11: **return** $G$

This algorithm and its implementation is reasonably simple, but some aspects require a slightly more detailed explanation.

The function "CALCULATE-NO-OF-START-REGIONS" simply uses equation $n = 8^q$ to calculate the expected number of regions that there will be in the start level in the image split tree.

Figure 5.1 shows a representation of the region configuration after initialisation with uniform regions representing level 4 in the image split tree[6].

### 5.1.5 Uniformity Criterion

The absolute difference uniformity criterion (page 37) was used for this implementation because there is a lack of detailed information on any other criterion and the only example of split and merge being used on MRI also uses it [70]. The mathematical function for the absolute difference uniformity criterion for a single region is shown below. The values used for the uniformity threshold are explained in the individual splitting, merging and grouping sections.

$$U(R) = \begin{cases} \text{TRUE} & \text{if } maxval(R) - minval(R) < \varepsilon \\ \text{FALSE} & \text{else} \end{cases} \tag{5.1}$$

[6]Level 4 is used here to make the regions large enough that they are clearly visible. In practice level 6 is used where the regions are considerably smaller.

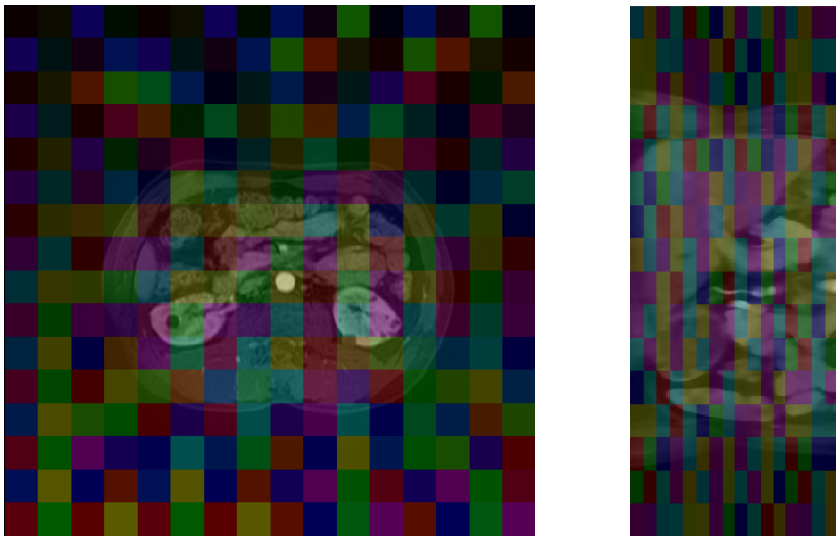

Figure 5.1: Screen capture from the Java implementation of the split and merge initialisation process where coloured sections represent discrete regions - abdominal MRI scan slice view on X-Y axis (left) and top view on X-Z axis (right).

### 5.1.6 Merging

The merging process is implemented exactly as stated in the previous description where each region in the image is checked for uniformity with its siblings in the image split tree. This method is shown in algorithm 5.2.

The checking is carried out with a process similar to a graph-search, where a list of regions is maintained which identifies all of the regions which have not yet been checked (i.e. the list *to_check*). This list is initially a duplicate of every region in the RAG, but as each one is checked it is removed from this list so that it is not checked any more than once.

There are two types of region joining which are required in the split and merge algorithm, one is explained in §5.1.8, and the other is called in line 5 of this algorithm. This is the simpler join operation, where eight sibling regions are joined together to form their parent region. This process is simple because the parent region is known to be a rectangular cuboid and can therefore be defined with a three point coordinate and a three value dimension, as with all regions up to this point. The 'new' region is constructed in the implementation by calculating the minimum and maximum edges for each the the eight siblings in all three axes and then using these extrema to define the parent region.

The literature does not state what values should be used for the uniformity threshold, $\varepsilon$. For reasons explained in §5.1.8 it was decided that the image should be over segmented at this point and therefore testing discovered that the value of 10 should be used.

The remaining functions in algorithm 5.2 are self-explanatory and will not be covered in any greater detail. Figure 5.2 shows a representation of the region configuration after merging with the uniformity threshold set to 10.

**Algorithm 5.2** Split and Merge - Region Merging

**Input:** $G$, the RAG
**Input:** $V$, the voxel array representation of the image series

```
1: to_check ← ALL-REGIONS-IN(G)
2: while to_check ≠ empty list do
3:    region ← GET-ANY-REGION(to_check)
4:    siblings ← SIBLINGS-OF(region, G)
5:    new_region ← JOIN-8-REGIONS(siblings)
6:    if UNIFORM(new_region, V) then
7:       new_adjacencies ← COMBINE-ADJACENCIES(siblings, G)
8:       G ← REMOVE-REGIONS(G, siblings)
9:       G ← ADD-REGION(G, new_region, new_adjacencies)
10:      to_check ← ADD(to_check, new_region)
11:   end if
12:   to_check ← REMOVE(to_check, siblings)
13: end while
14: return G
```

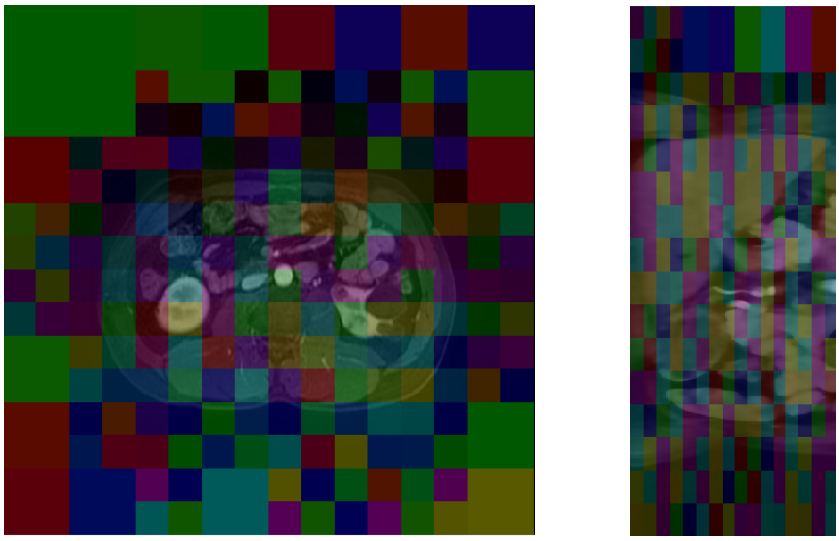

Figure 5.2: Screen capture from the Java implementation of the split and merge merging process where coloured sections represent discrete regions - abdominal MRI scan slice view on X-Y axis (left) and top view on X-Z axis (right).

### 5.1.7 Splitting

As with the merging process, the splitting process was implemented exactly as described in the literature for a standard split and merge application. This is shown in algorithm 5.3.

**Algorithm 5.3** Split and Merge - Region Splitting

**Input:** $G$, the RAG
**Input:** $V$, the voxel array representation of the image series

```
1: to_check ← ALL-REGIONS-IN(G)
2: while to_check ≠ empty list do
3:    region ← GET-ANY-REGION(to_check)
4:    if ¬UNIFORM(region, V) then
5:       new_regions ← GENERATE-CHILDREN(region)
6:       for each new_reg in new_regions do
7:          new_adjacencies ← GENERATE-ADJACENCIES(new_reg, G)
8:          G ← ADD-REGION(G, new_reg, new_adjacencies)
9:          to_check ← ADD(to_check, new_reg)
10:      end for
11:      G ← REMOVE-REGION(G, region)
12:   end if
13:   to_check ← REMOVE(to_check, region)
14: end while
15: return G
```

The algorithm is similar to that used in region merging in that a list of all regions which require uniformity checking is maintained. This list initially contains every region the the RAG output from the merging process and then uniform regions are removed, and non-uniform regions are replaced with their children in the image split tree.

When a regions is non-uniform it is split and the algorithm then progresses through all eight of the new child regions and uses the function "GENERATE-ADJACENCIES" to identify which regions in the RAG are adjacent to each. The actual Java implementation for this method also utilises the adjacency list for the parent region to limit the number of regions which must be checked and improve the computational efficiency.

The remaining functions in algorithm 5.3 are self-explanatory and will not be covered in any greater detail. Figure 5.3 shows a representation of the region configuration after splitting with the uniformity threshold kept at 10. This shows large regions in homogeneous areas and very small regions close to areas of high gradient such as edges.

### 5.1.8 Grouping

The first implementation of the grouping process was created to operate exactly as stated in Horowitz and Pavlidis' original method. This process is shown in algorithm 5.4.

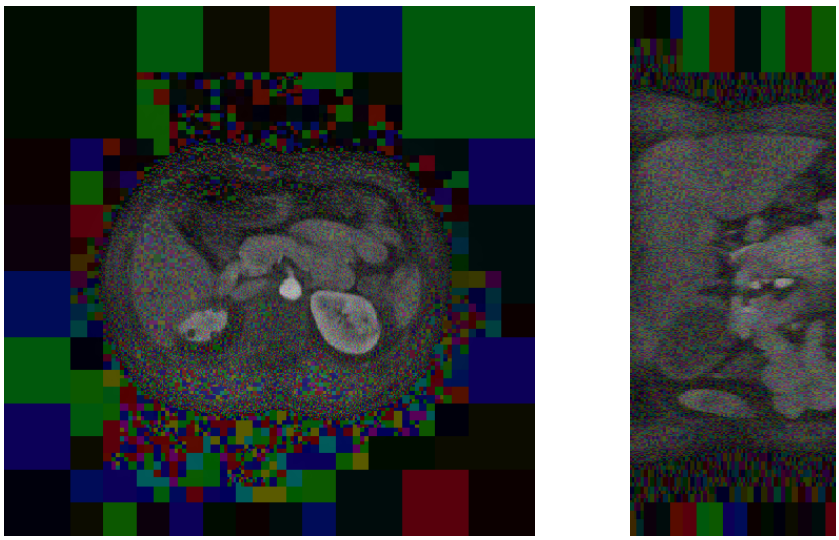

Figure 5.3: Screen capture from the Java implementation of the split and merge splitting process where coloured sections represent discrete regions - abdominal MRI scan slice view on X-Y axis (left) and top view on X-Z axis (right).

**Algorithm 5.4** Split and Merge - Region Grouping

**Input:** $G$, the RAG
**Input:** $V$, the voxel array representation of the image series

```
1: to_check ← ALL-REGIONS-IN(G)
2: while to_check ≠ empty list do
3:    region ← GET-ANY-REGION(to_check)
4:    adj_to_check ← ALL-ADJACENT-TO(region, G)
5:    while adj_to_check ≠ empty list do
6:       adj_region ← GET-ANY-REGION(adj_to_check)
7:       new_region ← JOIN-2-REGIONS(region, adj_region)
8:       if UNIFORM(new_region, V) then
9:          new_adjacencies ← GENERATE-ADJACENCIES(new_region, G)
10:         adj_to_check ← ADD(adj_to_check, new_adjacencies)
11:         G ← ADD-REGION(G, new_region, new_adjacencies)
12:         G ← REMOVE-REGION(G, region)
13:         G ← REMOVE-REGION(G, adj_region)
14:         to_check ← REMOVE(to_check, region)
15:         to_check ← REMOVE(to_check, adj_region)
16:         region ← new_region
17:      end if
18:      adj_to_check ← REMOVE(adj_to_check, adj_region)
19:   end while
20:   to_check ← REMOVE(to_check, region)
21: end while
22: return G
```

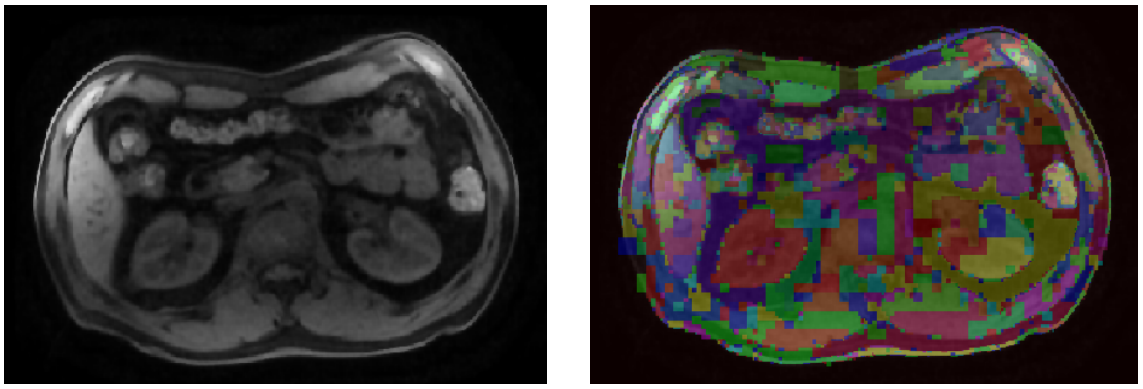

Figure 5.4: Screen capture from the Java implementation of the complete split and merge process in its basic form alongside the raw image for comparison. The only valid segment is the (red) kidney on the left hand side of the image.

This algorithm operates by checking every pair of neighbour regions to see if they can be combined to form a uniform region. If they can then their combined region is placed in the RAG and they are both removed from it.

The function "JOIN-2-REGIONS" is used in this algorithm to combine any two neighbour regions which are uniform. Unlike in the previous joining function, it cannot be guaranteed that the two regions will form a rectangular cuboid when joined, and therefore the new region cannot be described in terms of three point coordinate and a three value dimension. This joining function operates by combining the lists of `CubicVolume`s from both neighbour regions.

Region grouping is the final stage in the split and merge algorithm and therefore its results would ideally be suitable for classification. After testing this implementation it became clear that it was of limited applicability for MR images. The best results obtained are shown in figure 5.4 which was achieved using a uniformity threshold of 85 for merging, splitting and growing.

It should be noted that the absolute difference uniformity criterion is not completely robust with regards to different images. Each MR image can have a different value range and therefore the split and merge parameters are likely to require different settings for image series produced on MRI apparatus which scales the images differently.

It can be seen in figure 5.4 that the basic split and merge implementation is able to identify some structures. A problem with this segmentation is that the segment edges are very course and do not represent the smooth edges which are seen in real structures. This segmentation also over-segments any regions which are slightly non-uniform due to its strict adherence the the uniformity criterion.

A fundamental problem with basic split and merge is that it is completely non-deterministic in the way that it checks for uniformity between adjacent regions as any two neighbours can be checked at almost any point in the process. The non-determinism can be seen in this algorithm with the function "GET-ANY-REGION" in lines 3 and 6. The

problem with the non-determinism is similar to that shown in figure 4.5 but rather than the tree structure determining which regions to merge, it is the random order in which they are checked. This means that if the algorithm happens to check two regions which are 'reasonably' similar they will merge and potentially prevent either of the regions merging with another which is much more similar. Clearly the non-deterministic operation does not guarantee an optimal grouping, but checking every possible combination of regions would require exponential time and is not implementable.

There are various method suggested in the literature for improving the errors identified in the basic split and merge implementation. They are discussed briefly below.

### Boundary Elimination

Boundary elimination is a technique to improve a segmentation by removing any region boundaries which do not appear to outline real image features [98]. False boundaries are identified in this method using two quantifiable boundary attributes: edge contrast and straightness. The process operates by assessing a merit value for every inter-segment boundary and then removing those which fall below a certain quality threshold. The merit function suggested by Pavlidis and Liow [98] is shown below.

$$f(e) = \frac{|\text{sum of contrast along boundary } e|}{\text{length of boundary } e} + \beta \frac{\text{number of direction changes along boundary } e}{\text{length of boundary } e} \tag{5.2}$$

where $\beta$ is the weight associated with the boundary straightness.

In this function the boundary contrast is calculated by summing the difference between the IEs on either side of the boundary across every point in the boundary. The second component simply penalises edges which follow perfectly straight lines as these are unlikely in natural images and are most likely the result of errors. For the application after split and merge this is particularly useful because the split tree bias explained previously has a tendency to create false straight boundaries between non-sibling regions.

### Locally Adaptive Thresholding

There are improvements proposed in the literature which are based around the concept of using an adaptive uniformity threshold which changes to suit each local region in the image [18, 20]. The implementations of these methods were shown to be robust and successful, but the research carried out shows that they are of limited use when regions become small. The reason for this limitation is that the local area used for setting threshold values does not incorporate enough data and suffers from poor SNR. Moreover, these methods are still highly non-deterministic in their treatment of the region growing and cannot guarantee an optimal solution.

#### Simulated Annealing

Simulated annealing is a meta-heuristic which has been referenced at various other points in this document. Is is a method for locating the global optima in an optimisation problem by considering it as the metallurgical process of annealing. The concept is that a global control constant called 'temperature' is gradually reduced, controlling the formation of partially random solutions and forcing them towards a global optima [52].

Manousakas et al. [70] describe the process for using simulated annealing for region grouping to reduce non-determinism in the process and encourage the formation of an optimal solution. Simulated annealing operates by gradually increasing the threshold value, $\varepsilon$, used for a number of grouping stages. This method appears at first to be a simple threshold relaxation, but due to the grouping's non-determinism the groups at each stage are partially random and therefore the implementation does follow the definition of simulated annealing.

#### Space-Graph Representation

Cheevasuvit, Maitre and Vidal-Madjar [19] identified that split and merge does not necessarily produce a unique solution and proposed a method to improve the results. In their method they perform split and merge with multiple different threshold values and construct a space-graph from the segments produced by each. This space graph is then used to identify which regions are stable across a majority of the threshold ranges and these regions are taken forward. Though Cheevasuvit, Maitre and Vidal-Madjar report some good results with this approach, I believe that it is not suitable for this application due to the size of the image volume and the time it would take to process each tolerance range.

### 5.1.9 Simulated Annealing

Simulated annealing appeared to be the most suitable method for improving the optimality of split and merge as it is a well understood optimisation technique and can be implemented using only small modifications to the standard algorithm. It is also be able to minimise the straight edge bias seen in figure 5.4 because the splitting stage can be much finer allowing it to match segment outlines clearly. Simulated annealing was implemented using algorithm 5.5. This also shows the sequence used to combine the previous split and merge algorithms detailed above.

This algorithm operates by increasing the uniformity threshold after each application of the the grouping process. The function "FINALISE-REGIONS" decodes the RAG and allocates each voxel in the array a single value to identify which region it is a part of. Testing this algorithm showed that the best and most robust settings were an initial threshold of 10, a step threshold increase of 3 and 60 steps. The result of applying these settings is shown in figure 5.5.

**Algorithm 5.5** Split and Merge - Simulated Annealing

**Input:** $V$, the voxel array representation of the image series
**Input:** $q$, the start level in the image-split tree
**Input:** $\varepsilon$, the initial uniformity threshold (global constant)
**Input:** $\delta$, the step threshold change
**Input:** $s$, the number of annealing steps

```
1: G ← INITIALISE-RAG(q, V)
2: G ← MERGE-RAG(G, V)
3: G ← SPLIT-RAG(G, V)
4: for loop_count = 0 to s do
5:     G ← GROUP-RAG(G, V)
6:     ε ← ε + δ
7: end for
8: V ← FINALISE-REGIONS(V, G)
9: return V
```

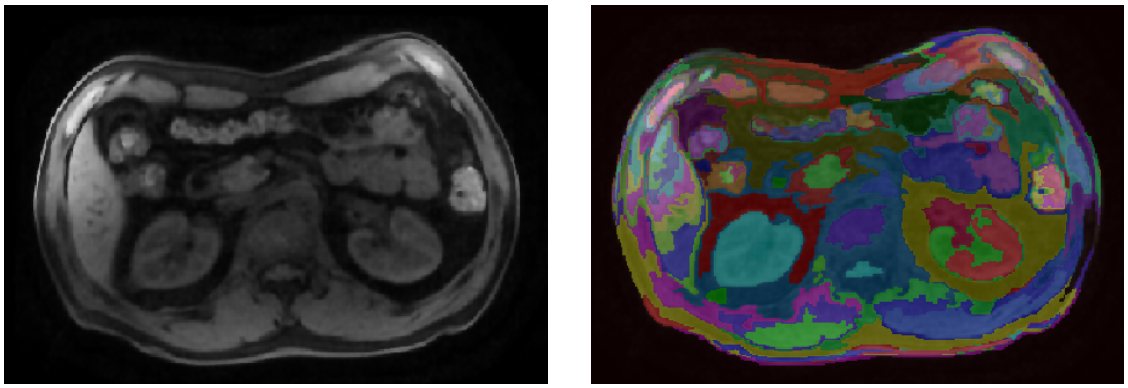

Figure 5.5: Screen capture from the Java implementation of the complete split and merge process with simulated annealing alongside the raw image for comparison. The segment boundaries are considerably smoother than without simulated annealing and the right hand kidney is improved, though still not complete.

It can be seen that simulated annealing vastly improves on the standard split and merge algorithm, but the segmentation is not perfect. A clear problem with this implementation is that it still fails on the inhomogeneous region towards to top of the liver and at the perimeter of the right hand kidney. Even after the inclusion of simulated annealing, split and merge still relies on point values and neighbour relationships to asses region membership and therefore cannot be expected to overcome the gradual spatial variation that is observed in inhomogeneous regions such as these.

### 5.1.10 Boundary Elimination

To overcome the difficulties with inhomogeneities the boundary elimination method was employed. By removing poor boundaries irrespective of absolute difference uniformities regions which are over segmented by purely a statistical analysis can be joined to form a complete region. Boundary elimination is implemented by analysing every boundary within the image volume using some merit function, and then removing all boundaries which have a merit function below a predetermined threshold, $\sigma$. The algorithm used to implement boundary elimination is shown in 5.6.

**Algorithm 5.6** Boundary Elimination

**Input:** $V$, the segmented voxel array (as produced by split and merge)

```
1: boundary_list ← ANALYSE-BOUNDARIES(V)
2: for each boundary in boundary_list do
3:    merit ← GET-MERIT-VALUE(boundary)
4:    if merit < σ then
5:       V ← JOIN-REGIONS-ACROSS-BOUNDARY(V, boundary)
6:    end if
7: end for
8: return V
```

The key aspect of boundary elimination is the function "ANALYSE-BOUNDARIES" which processes the whole image volume and assesses the properties of each region boundary. This function constructs a list of boundaries and associates a merit value with each.

Once the required boundary information has been acquired the merit function can be calculated for each boundary using equation 5.2. During the implementation of this merit function some variations were tested which included: penalising small regions; encouraging smooth regions, second order edge detection and penalising long boundaries, but it was found that these did not greatly improve segmentation accuracy but increased the computational demand so they were removed.

It was found empirically that a value of 800 yielded the best results when used as the value for the constant, $\beta$, in the boundary elimination merit function (seen in formula 5.2)[7]. Similar testing was used to deduce that the optimum boundary merit threshold, $\sigma$,

[7]This value sounds very high, but as the typical values for contrast are much larger than those for

**Algorithm 5.7** Boundary Elimination - Boundary Analysis

**Input:** $V$, the segmented voxel array

1: *boundary_list* ← an empty list
2: **for** each *voxel* in $V$ **do**
3: *voxel_region* ← GET-REGION-ALLOCATION(*voxel*, $V$)
4: *neighbours* ← GET-NEIGHBOURS(*voxel*, $V$)
5: **for** each *neighbour* in *neighbours* **do**
6: *neighbour_region* ← GET-REGION-ALLOCATION[6](*neighbour*, $V$)
7: **if** *voxel_region* ≠ *neighbour_region* **then**
8: *contrast* ← GET-CCONTRAST-BETWEEN(*voxel*, *neighbour*, $V$)
9: *changes* ← GET-BOUNDARY-CHANGES(*voxel*, *neighbour*, $V$)
10: *key* ← GET-KEY-FOR(*voxel_region*, *neighbour_region*)
11: *boundary_list* ← INCREMENT-BOUNDARY(*boundary_list*, *key*)
12: *boundary_list* ← UPDATE-CONTRAST(*boundary_list*, *key*, *contrast*)
13: *boundary_list* ← UPDATE-CHANGES(*boundary_list*, *key*, *changes*)
14: **end if**
15: **end for**
16: **end for**
17: **for** each *boundary* in *boundary_list* **do**
18: *merit* ← CALCULATE-MERIT-VALUE(*boundary*)
19: *boundary_list* ← SET-MERIT-VALUE(*boundary_list*, *boundary*, *merit*)
20: **end for**
21: **return** *boundary_list*

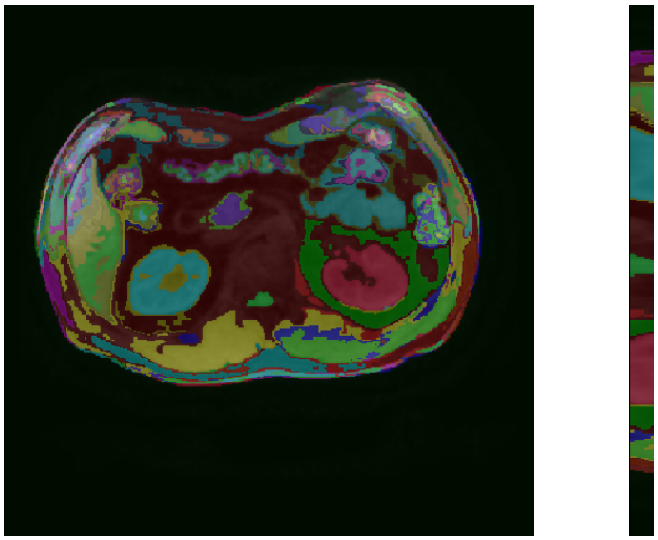

Figure 5.6: Screen capture from the Java implementation of split and merge with simulated annealing and boundary elimination showing abdominal MRI scan slice view on X-Y axis (left) and top view on X-Z axis (right). The side view is thin due to the presence of large tumours at the top and bottom of the kidneys which affected the segmentation.

for this application is 1550. The results of using boundary elimination with these values can be seen in figures 5.6 and 5.7.

Boundary elimination works by discarding poor boundaries, therefore if the previous segmentation has already merged regions which it should not have, then boundary elimination will not improve the results. For this reason the split and merge algorithms must be implemented with low uniformity thresholds so that the image is over segmented and will be likely to have an edge at each point that there is an anatomical boundary. These results were achieved with an initial uniformity threshold of 10, a step threshold increase of 3 and 30 steps in the simulated annealing process.

## 5.2 Critical Analysis

The segmentations in figures 5.6 and 5.7 show the best results achieved using split and merge with simulated annealing and boundary elimination. Though many variations on parameters and settings were tested I was not able to create a segmentation which accurately defined multiple organs in a single image series. From researching the MRI modality and various segmentation techniques it was discovered that such a segmentation could not be expected due to some of the reasons described below.

It can be seen that both of the kidneys can be identified in one series, but even with the use of filter based compensation and boundary elimination the inhomogeneity is too large

direction changes this is actually equivalent to a ratio of approximately 9:1 for contrast and changes respectively.

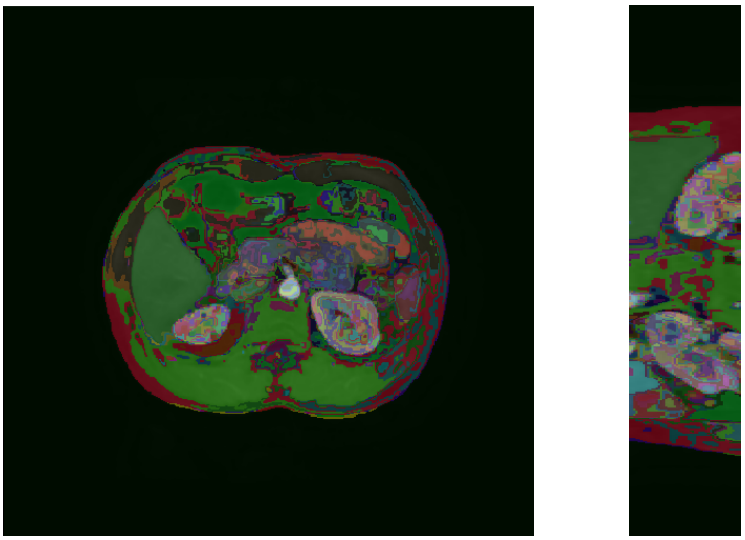

Figure 5.7: Screen capture from the Java implementation of split and merge with simulated annealing and boundary elimination showing abdominal MRI scan slice view on X-Y axis (left) and top view on X-Z axis (right). In this segmentation the liver has been well segmented and can be seen in the dark green segment on the left hand side.

and the top of the liver is over-segmented. The liver is well segmented in the other image series but the kidneys in this image are heavily affected by tumours and the segmentation struggled to overcome this. In both of these series the remaining structures such as: the spine; the intestines and the rib cage are poorly represented if at all.

I think that this inability to detect all tissue structures does not point to a flaw with the method that I have implemented. The initial aim to robustly segment multiple different tissue structures was defined before I had a full understanding of both the MRI modality and the methods used in segmentation. I think that it was naive to expect segmentation to work across all structures when in the vast majority of cases the MR image is only created to identify one. For this reason I believe that the method which I have implemented is robust, as it can segment multiple tissue structures, but it must be used on images of suitable quality in the area of interest. A method to overcome this image dependence would be to include multi-spectral images as is discussed in §8.2.

It should be noted that the majority of the images used in testing were from renal (kidney) tumour studies and as a result tumours were prominent in the images and some of the areas were un-segmentable using the implemented algorithm. There are two possible effects caused by tumours in this segmentation process. The first is when the tumour has similar MR properties to the kidneys. In this case the tumour and kidney are commonly merged together into one segment. This is not necessarily a problem as the outline would be correct, but large tumours often press against other tissue structures within the patient. When this happens the boundary elimination algorithm in ineffective and consistently eliminates the boundaries between both organs and the tumour merging all three structures into one, invalid, segment. I do not think that this can easily be

overcome within this segmentation framework because the values are similar and the tumour boundaries are naturally smooth and difficult to identify. The other set of tumours are those which have highly different MR values to the kidneys in which they are located. When this occurs the split and merge algorithm tends to over-segment the kidneys because there are so many high contrast boundaries within them. Even if one could accurately extract all of the tumour segments from a kidney, the remaining structure would be very porous and I think that interpolating the full geometry would be very complicated. I believe that my algorithm is highly robust, but large and densely packed tumours are too complicated a problem to be resolved here. I therefore selected portions of kidney within the images which were reasonably tumour free and focused on accurately segmenting them. This is the reason for the limited thickness in some of the segmentations shown above.

During the process of investigating and implementing these algorithms there were several areas identified which could be improved or modified to achieve better results.

**Initialisation Improvements**

I believe that split and merge would be more computationally efficient if the process were not initialised at an intermediate level in the image split tree. It was explained previously that the intermediate initialisation was used by Horowitz and Pavlidis to speed-up the algorithm's operation as actual region boundaries were not expected to be found at the top or bottom of the image split tree. During its implementation I noticed that the merging stage which follows this initialisation is the only one which requires details of the tree structure. Splitting is only carried out on single regions making it independent of the tree details and grouping was designed to be performed once the tree has be discarded. This means that by starting with the splitting stage from a full image volume there is no requirement that each region is positioned in a tree and this can simplify the data structure used to store regions. I think the reduction in memory requirement would be worth the slight increase in operation time as the splitting stage only forms a small fraction of the total computation.

I believe that this implementation of split and merge could also be improved by incorporating a more complex tree structure that can accommodate for the difference in spatial resolution between the image axes. In all of the test images used the $z$-axis had a lower resolution than $x$ and $y$ and therefore the oct-tree structure meant small regions were elongated in the $z$-axis. Having tree regions uniformly elongated in the same direction causes a bias in the segmentation procedure and it is more likely to identify elongated segments as a result. This could be overcome by considering each elongated voxel as multiple sub-voxels so that division could be carried out equally and accurately in all dimensions.

### Splitting Improvements

The main problem with the splitting process is that the uniformity threshold has to be set to a very low value to get accurate region representations. If low uniformity thresholds are not used then boundaries identified at this stage are very irregular and follow a 'zig-zag' like path which corresponds to the perimeters of high level regions within the image split tree (as is seen in the implementation of basic split and merge in figure 5.4). Having a low uniformity threshold means that the splitting process does take longer, but this time is typically less than 20 seconds for a full medical volume. The real problem is the number of regions which are generated and the resulting memory requirement. In order to process test images using this method up to 12GB of memory was required. This would be acceptable if such a process was used in a commercial system, but it is impractical in terms of using the software on regular desktop PCs. Effective preprocessing improved the problem by increasing region uniformity, but I do not believe that there is a clear method for overcoming this problem and it is inherent in the split and merge process.

### Region Grouping Improvements

Region grouping was by far the biggest contributor to the long processing times, and in most of the tests it made up over 95% of the total operating time. An idea which I had to improve the computation speed was to maintain a list of regions which had been checked together for uniformity. In the current implementation, if region A is checked and found to be non-uniform with region B then this fact is remembered and the two are not checked again, but if region A is later grouped with another region C which is also a neighbour of B, then the grouped region A-C is again checked for uniformity with B even though it will clearly fail. By using a list of all checked region combinations many unnecessary uniformity calculations could be removed, reducing the operation time.

Another possible method for reducing the operating time is to process the volume as multiple thin sections and then recombine them at the end. The time taken for grouping is exponential in the number of voxels, and therefore applying the process multiple times to sections 3 or 4 slices thick would improve the speed. Using this many slices would still have advantages over two dimensional application because there is some depth information incorporated. The thin slices could be combined using any of the standard methods currently used in two dimensional segmentation such as Kalman filtering [112]. I believe that this method might provide a good compromise between the efficiency of two dimensions and the improved information and accuracy of three dimensions.

The grouping process could be further improved to minimise the non-determinism which leads to the requirement for simulated annealing. Non-determinism could be reduced by employing a more 'intelligent' approach to selecting the order of neighbour pair uniformity checks. One possible method for improving this process would be to create a directed gradient map of the image and then give grouping priority to regions which are tangential to the direction of the identified gradients at each point. A map could be created reasonably simply which identifies the general gradient 'flow' across the whole image

and then by first checking regions tangential to this gradient they are more likely to have similar values and therefore be more homogeneous. This method would simulate checking all neighbours and selecting the closest but would do this in a more efficient way as checking all neighbours would be very computationally demanding.

# Chapter 6

# Tissue Structure Classification

The output from the split and merge procedure described above is a three dimensional voxel array where each voxel within it is allocated to a single region by a unique integer labelling. That is to say that a region in the array is a collection of all voxels which have been labelled with a the same integer.

Constructing an object classifier is defined by Russell and Norvig [112] as the process of defining discrete-valued function which outputs a decision based on a set of input attributes. For this application, a classifier is required to allocate a label to each segment in the voxel array [23], but the meaning behind 'label' is specific to the type of classifier. There are generally considered to be two types of classifier: supervised and unsupervised.

Supervised classifiers are the most commonly used methods and rely on knowledge input from human operators to make informed decisions about the class allocation for a given input. The knowledge can be imparted by methods such as: annotated examples and logical rules. The common component with all of these methods is that external knowledge is used to link the attributes of an input to some 'real world' label. For this application the label would be tissue or organ type.

Conversely, unsupervised classifiers are methods which operate completely independently from any user input. As they have no external knowledge they are not able to assign 'real world' labels, but their use comes from being able to assign general non-specific labels that are consistent between all inputs that they process. That is to say that an unsupervised technique would be able to identify a segment in one image being of the same class as a segment in another image, but could not state what the class is. There are many applications where identifying similarities would be sufficient without knowing a 'real world' label, but for this project the aim was to label tissue or organ types. They can however be applied with a single annotated example allowing them to extrapolate labels across identified classes, but this use of an annotated example technically makes them supervised. The inability of fully unsupervised classifiers to attach real medical labels to the segments means that they will not be considered any further for use in this application.

An important consideration with respect to supervised classification techniques is the

way in which they acquire the rules used in the classification function. I will refer to the two different approaches as: *learnt knowledge-based* and *explicit knowledge-based*.

Learnt knowledge-based classifiers are not told the actual rules used in their classification function, and instead they are given many examples of the mappings which the function should make so that they can derive their own rules from them. These techniques do not require that the person building the classifier understands the class attributes, but do rely on the training data being representative of the full population of possible inputs. An example of such a technique is an artificial neural network [112], where a back propagation algorithm can be used to automatically set weights and biases based on a training set.

Explicit knowledge-based classifiers are those which are built with all rules needed for the classification process explicitly included. These techniques require that the person constructing them fully understands what properties the constituents of each class should take and that they are able to represent these rules fully. An example of such a technique is a Bayesian classifier, where the person creating the network must understand and construct dependency links between variables.

There is a certain amount of crossover between these two techniques and some classification methods can legitimately be considered to operate in both ways with minimal modifications. Most of the explicit knowledge-based techniques found can be adapted for learnt knowledge-based operation but the converse is much less common.

As radiographers are able to accurately identify the names of different tissue structures within MR images there must be a set of 'human-understandable' rules which can be used to classify the segments for this application. This means that explicit knowledge-based methods could be implemented as long as these rules are describable in a machine understandable way.

Learnt knowledge-based methods pose many advantages for classification but they have limitations which mean that they will not be considered for this application. The key problem with them is that they require a large range of annotated training images which are difficult to acquire. Moreover, the performance of these techniques on real data is dependent on how well the test images represent the total population and assuring this property is highly complex.

The segmentation algorithm was not able to accurately segment every tissue structure within the test images, therefore the classifier will only be constructed to label those which are. The typical definition for a classifier states that it should allocate a label to every input, and therefore an 'unknown segment' label was included for all of the poorly segmented areas. A full list of the labels included in the classifier is shown below:

- Kidney
- Liver
- Background (i.e. the air inside the MRI machine which is not a part of the patient)
- Unknown (i.e. a segment which cannot be allocated to one of the other categories)

By restricting the classification methods considered to only those which are supervised

and operate on explicit knowledge there are not a great deal of techniques to decide between. Expert systems [35, 64] are by far the most common methods which operate with these requirements. Appendix D contains a review of expert systems for this application and includes some detail on two sub-fields which allow for the incorporation of uncertainty within the knowledge base. This review is included for background information can be disregarded for the purposes of this application, and any key information within it is incorporated in this chapter.

## 6.1 Selection of a Classification Technique for MRI Segments

From my investigation of potential classification techniques I believe that the most suitable method for the aims of this project is fuzzy inference [28, 67, 118]. 'Hard' logical expert systems [35, 64] are far too restrictive for this application due to the large amount of variation expected in patient anatomy. As 'hard' logical systems do not allow for any uncertainty within rules the rule base would have to be very large and each rule would have to be so specific to an organ size/shape/etc. that they would be difficult to specify. It was stated above that the knowledge for these systems is explicitly included and therefore it must be produced by a human expert. It is not just the large number of rules that make the specification difficult, it is also the fact that each rule would have to be so specific to an organ configuration that an expert, such as a radiographer, is unlikely to express their knowledge in such a way.

Bayesian classifiers [112] overcome this difficulty to a degree be considering rules using probability distributions for cause and effect. The problem with this is that the expression of rules as probability distributions is likely to be too unnatural for a radiographer and therefore they would be less effective rules and would lead to a less accurate solution. Fuzzy inference includes this ability to work with inexact and general rules using fuzzy set membership, but the rules used are of of the form "IF... THEN..." and could be implemented with confidence that the system is operating with the conditions that were intended by the expert who specified them.

## 6.2 Implementation of a Fuzzy Inference Classifier

### 6.2.1 Classification Preparation

The split and merge segmentation procedure is not able to segment every organ within the image volume and in some areas the organs are very fragmented. This means that there are many more segments than organs and the classifier is required to identify all of those which do not relate to true organs.

Each segment must be processed by the classifier independently and therefore operation

could be slow for all of the very small regions in over segmented areas of the image. To improve this I decided that all regions with a volume of less than a certain threshold should be immediately identified as *unknown* without being processed by fuzzy inference. All of the organs which are identifiable in the segmented image have a volume well above the segmentations mean and therefore this stage is able to reduce the number of inference evaluations from 2,500 to 600 for a full image volume of 512×512×100 voxels with a threshold of 1,000 $mm^3$.

### 6.2.2 Attribute and Rule Selection

All fuzzy inference systems operate by evaluating "IF...THEN..." rules which are specified relating to certain attributes of the objects which require classification. The main differentiation between different implementations of fuzzy inference is the way in which the rules are computationally represented and evaluated. Mamdani style rules [67] were selected for use in this inference system because they are the most similar to a how a human would express such details. They strictly follow the "IF...THEN..." framework where the antecedent is a possible attribute of the object requiring classification and the consequent is a classification. For example, "IF volume is large THEN classification is liver". An alternative to this inference system would be Sugno rules [118] which allow the consequent to be a function of the antecedent. The problem with these is that a human would not naturally state a rule in such a way and therefore would overcomplicate its specification. Mamdani inference also suits the non-standard evaluation process selected which is explained in §6.2.5.

Radiologists commonly use a large amount rules based on spatial information to classify an organ within a MR image. This is reasonably simple for a human to achieve because they can process each organ in the image in parallel and identify the spatial relationships between them. It is important to note that according to Saladin [115] the spatial information available in anatomy teaching materials such as text books is only correct for approximately 70% of the population due to the potential for patient variation. This means that radiographers have a very general knowledge of possible variations in organ position which is built up from an understanding of the function of different structures and their previous experience.

The aim of using a fuzzy inference based classifier is to employ rules which are as human like as possible, but for two reasons my method focuses more on geometric constraints rather than spatial ones. The first of these reasons is that the segmentation method is not able to accurately segment all of the organs within a medical volume, this means that there are not enough points within the volume which can be used as datums from which to generate spatial data. Moreover, a DICOM MR image does not necessarily contain any information about the position of the patient within the imaging apparatus. This means that any spatial information would have to be relative to other structures shown within the image rather than any absolute points.

Secondly, the only way for an automated system to make human-like positional variation

judgements would be to incorporate an enormous knowledge base. In addition to being impractical, I think that similar results can be achieved without this degree of complication by using more general geometric rules rather than specific positional ones.

The rules used in classifying each segment were selected for the following reasons:

### Kidney Rules

The kidneys are a good candidate for geometric based classification because they are highly patient invariant. Unlike the heart and stomach which vary greatly between patients based on their use, the kidney performs the same tasks for all people and is therefore a similar shape and size for all. Kidneys typically take a rough shape 12 cm long, 5 cm wide and 2.5 cm thick [115]. These dimensions are equivalent to a volume of approximately 150,000 $mm^3$. By comparing this volume to the other organ descriptions in [115] it is clear that the kidneys are one of the smaller organs and therefore this can be used as a rule in the fuzzy inference process.

The medial margin[1] is known to be concave in all kidneys as this is the area where the main arteries connect to the interior [81]. This is a useful identification property because there are few other organs with such a shape in the human body and will make the kidneys stand out.

A common description of kidneys which uses loose spatial information, is that they are found to the sides of the body and not in the centre. This can be used in the classification by including a rule which says the kidneys must be to the left or to the right.

One non-medical identification rule which I included related to the surface of the kidneys. One difficulty in the classification process was mislabelling of organ cavities[2] as organs such as kidneys. The problem is that there are many such cavities within an image and therefore it is likely that one will be approximately the same size as a kidney, be in a similar position and even display attributes such as concavity. To overcome this I included a smoothness assessment to eliminate such erroneous classifications. The cavity segments typically have a very complex shape and have very irregular surfaces. A rule was therefore included which stated that the segments surface should be smooth.

The rules used in the classifier to identify kidneys are shown below:

**IF** size is small
  **AND** medial margin is concave
  **AND** (horizontal position is left
    **OR** horizontal position is right)
  **AND** surface is smooth
**THEN** segment is kidney

[1]The medial margin is the thin inward facing edge.
[2]Organ cavities are the spaces between tissue structures.

### Liver Rules

According to Moore and Agur [81] the liver is the single largest internal organ within a humans body. Size is therefore a good pointer for identifying the liver and was used in the fuzzy inference with large values required for the liver classification.

Another well defined anatomical property of the liver is that it is found on the right had side of the body [115]. This property can greatly narrow the number of potential candidates for the liver and therefore the horizontal position of each segment will be used to identify the liver.

As with kidneys, smoothness was incorporated to ensure that organ cavities were not classified as livers.

The rules used in the classifier to identify kidneys are shown below:

**IF** size is large
**AND** horizontal position is right
**AND** surface is smooth
**THEN** segment is liver

### Background Rules

The background in a MR image is the area within the imaging volume which it not part of the patient. The signal recorded is based on the MR properties of air. No literature could be found to describe the properties of background segments within a MR image, but from my analysis of images several rules were identified.

The most obvious method for identifying the background segment within a MR image is that it is the one which touches the edge of the imaging volume in the most places.

I intended to avoid using MR values in the classification process because the algorithm is aimed at being independent of contrast weighting, but for identifying the background I decided to use this information. Air has negligible MR and therefore the value recorded is very low for all of the common contrast weightings. This means that MR value can be used robustly and all background segments will have a low MR value.

Segment centrality was selected as a method for identifying the background as the patient is always central within the imaging volume and therefore, assuming a full slice is used, the background must also be.

Using this information about the properties of a background segment within and MR image, the classification rule shown below was generated:

**IF** edge contact is high
**AND** MR value is low
**AND** centrality is high
**THEN** segment is background

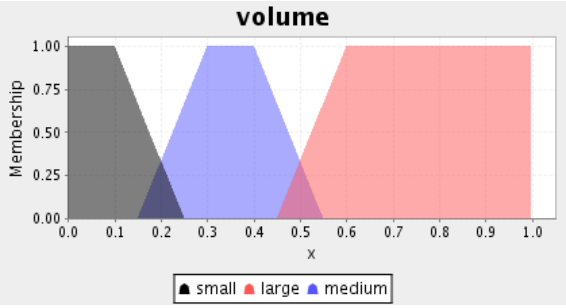


Figure 6.1: Fuzzification function for the variable 'Size' with the linguistic values 'large' and 'small' ('medium' is not necessary but is included for completeness).

This system was constructed without any explicit rules for what defines an 'unknown' segment, instead these are just regions which cannot be classified into one of those above.

### 6.2.3 Fuzzification Functions

As stated previously, fuzzy inference rules operate on a set of attributes for each object requiring classification. The key component in fuzzy inference is that it does not enforce strict attribute values and instead incorporates fuzzy set membership to allow for a degree of uncertainty within the rules. This means that each potential attribute for input objects is a fuzzy rather than hard set membership value and a set of functions must be specified to extract these set memberships from the input object. Electronic systems can asses quantitative properties of objects very well, but they cannot be expected to understand qualitative attributes such as is required for fuzzy inference. Therefore a set of mapping functions (referred to a *fuzzification functions*) must be specified to turn quantitative object properties into fuzzy set membership values for each potential object attribute used in the rules involved in the classification. Below is the reasoning and description of each of the fuzzification functions required for the attributes used in the rules specified above.

#### Size

As it cannot be guaranteed that the image volume will contain entire organs the size cannot be implemented in an absolute way. Instead the size of each segment was considered relative the the others within the same image volume so that larger and smaller organs could be identified. The size of each segment is normalised to a value between 0 and 1, where 0 is the smallest segment in the image and 1 is the largest.

Linguistic values for both small and large were required and the percentage of segments in each can be seen in figure 6.1. The 'large' set is extended because the distribution of organ sizes is non-uniform and it overcomes the high variance seen in larger organs.

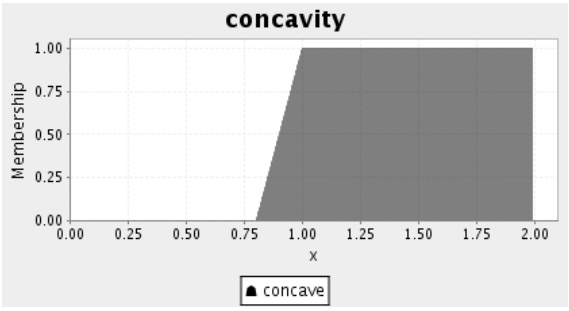


Figure 6.2: Fuzzification function for the variable 'Concavity' with the linguistic value 'concave'.

### Medial Margin Concavity

Some research was carried out into possible methods for assessing directional concavities within segments. There are a set of mathematical tools for analysing segments in this way [109], but the methods require complex surface mappings and I believe that they would be too computationally demanding to apply to multiple three dimensional segments.

Instead of assessing concavities in this way I decided to apply an isotropic method which calculates a segments direction-invariant concavity. This method calculates concavity as the ratio of concave to convex voxels on the segments boundary.

Using this definition of concavity, a concave segment is one which has more convex than concave boundary voxels and hence has a ratio greater than 1. The fuzzification function for segment concavity is shown in figure 6.2.

### Horizontal Position

The horizontal position variable was created to operate on values between 0 and 1 such that 0 is the far right-hand side of the imaging volume and 1 is the far left-hand side. This value was calculated for each segment by comparing its centre of mass to the width of the image.

Linguistic values were required for both left and right and their membership functions can be seen in figure 6.3.

### Surface

The smoothness of each segment was assessed by dividing its volume by its surface area, so that smooth segments would have high values and uneven segments would have low values. As this is not a well defined medical parameter several tests were carried out to deduce the optimal fuzzification function with the linguistic value 'smooth'. The selected function is shown in figure 6.4.

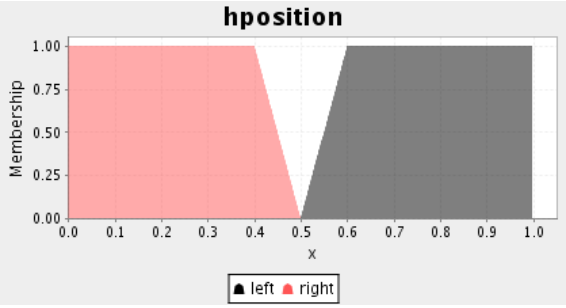


Figure 6.3: Fuzzification function for the variable 'Horizontal Position' with the linguistic values 'left' and 'right'.

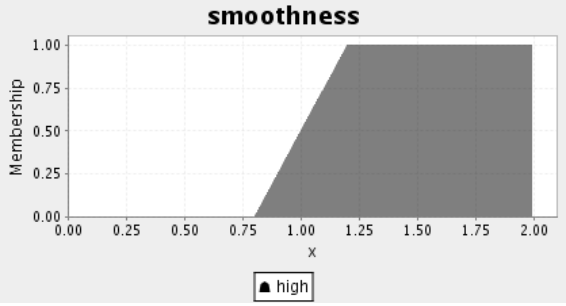


Figure 6.4: Fuzzification function for the variable 'Surface' with the linguistic value 'smooth'.

### Edge Contact

I decided to use relative values for the edge contact variable because different image series are of different sizes. The edge contact variable was calculated for a region by summing the total number of voxels within the region which are at an edge and then dividing this value by the total number of edge voxels in the whole image. This produces edge contact values in the range 0 to 1, where 0 is a region which has no contact with the perimeter of the imaging volume, and 1 is a region in contact with every point on the image volume.

The only linguistic value required for the edge contact variable was "high". I used the fuzzification function shown in figure 6.5 so that regions with high values must contact at least 75% of the total imaging volume.

### Centrality

The variable for centrality was created to operate based on millimetre displacements from the centre of the image volume. The displacements were calculated using the Euclidean distance from each segments centre of mass to the middle of the image.

From assessing the position of organs within a body, it was decided that the centrality fuzzification function for the linguistic value 'high' should take the form shown in figure 6.6.

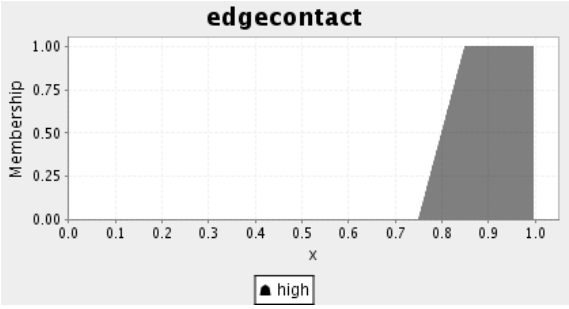


Figure 6.5: Fuzzification function for the variable 'Edge Contact' with the linguistic value 'high'.

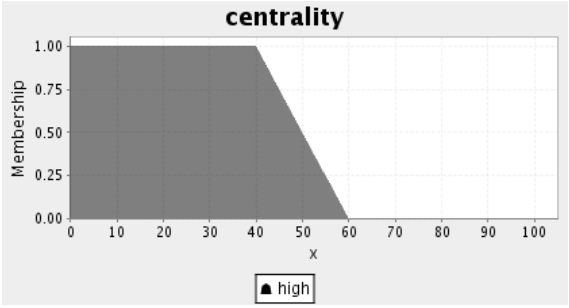


Figure 6.6: Fuzzification function for the variable 'Centrality' with the linguistic value 'high'.

#### Magnetic Resonance Value

MR values on their own do not tell the user anything about the property of a tissue structure and require a knowledge of the imaging parameters to identify anything absolutely. In this application the MR values are only being used to identify the background where it is known that the values will be low for all images. It would not make sense to specify the fuzzification function using absolute values and therefore relative values are used. The value at a point is divided by the maximum value across the whole image volume to give a value in the range 0 to 1, where a region with a value of 0 has the lowest values in the image and a region with a value of 1 has the highest values in the image.

The only linguistic value required for the MR value variable was "low". I used the fuzzification function shown in figure 6.7 so that regions with low values must be in the bottom 25% of the entire population.

### 6.2.4 Rule Evaluation

Fuzzy inference rules are evaluated by combining the set membership values of the antecedents to produce a set membership value for the consequent. The detail of this process is included in Appendix D but the process typically sets the consequent's

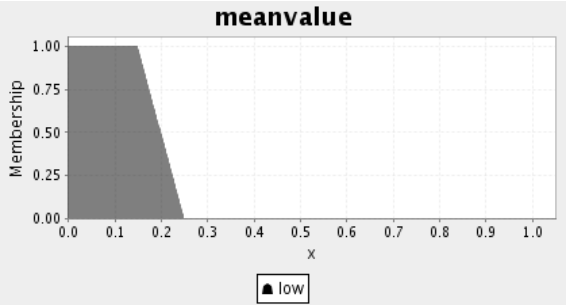


Figure 6.7: Fuzzification function for the variable 'MR Value' with the linguistic value 'low'.

membership value to be the same as the antecedent's. When antecedents are paired using AND or OR their membership values are combined using minimum and maximum functions respectively.

The jFuzzyLogic Java toolkit [22] was used to implement and evaluate each of the rules above. This toolkit operates by parsing a file which specifies inference rules, fuzzification functions and a defuzzification function (explained below) and creates a *fuzzy inference engine* Java object. Once the inference engine has been created each of the input variables can be set to numerical values and then it automatically fuzzifies them, evaluates them and produces a result.

The rules used by the jFuzzyLogic toolkit are specified in a separate text file using the fuzzy control logic (FCL) syntax as defined in IEC 1131-7 [46].

### 6.2.5 Defuzzification Function

Once each of the rules in a fuzzy system has been evaluated they allocate a membership value to each potential object class. To select a single best solution these classifications must be combined in some way in a process called *defuzzification*. There are many possible methods for this, some of which are described in Appendix D. In this implementation the required output is a segment classification with a discrete label such as *kidney*. Many of the methods in the literature merge the outputs from each rule into one single numerical value which represents them all. This would not be suitable here because it is not possible for a segment which has properties of both *kidney* and *background* to belong to a class in between them both and instead must be one or the other.

The defuzzification method I selected was very close the Mamdani's initial proposal and is essentially a parallel application of several different fuzzy inference classifiers, each of which was constructed for a separate organ class. The single output class was selected by taking the set with the highest confidence value.

I implemented this defuzzification using the jFuzzyLogic toolkit by specifying a singleton 'spike' in the output domain for each possible segment classification. This function is shown in figure 6.8. The single class is then selected from this output function by selecting

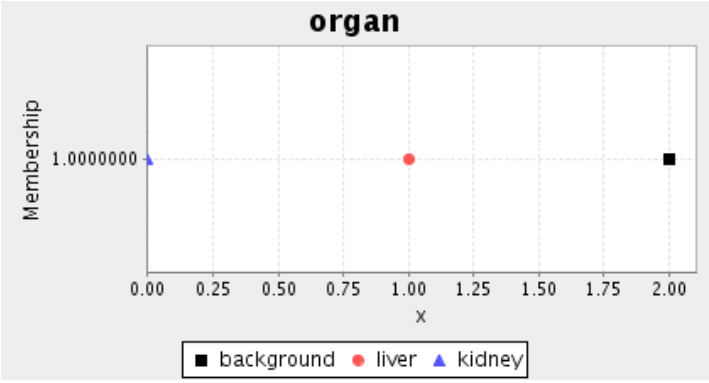


Figure 6.8: Defuzzification function showing singleton points relating to each possible discrete segment classification. The values 0, 1 and 2 are arbitrary and have no effect on the classification process.

the class which has the highest membership value for its representative singleton.

The membership values for each discrete classification can be thought of as confidence values for the classification. My application uses these confidence values to ensure that the best classification from the inference is good enough to be provided as an output. This check is carried out by comparing the confidence value against a threshold value, $T_c$, and if the confidence is high enough then the class inferred is output, otherwise the segment is classified as 'unknown'.

A value of 0.5 was used for $T_c$ so that only segments which are considered 'more likely than not' in a class are used. I believe that this value could be reduced in practice because the inference rules specified above are 'strong' enough that very few segments have a value greater than 0 and they are typically very low for incorrect classifications.

The full classification process is shown in algorithm 6.1.

Applying this classification algorithm to the output from the segmentation stage produces the classification shown in figures 6.9 and 6.10. It can be seen in these images that both kidneys, liver and the background can all been identified and labelled if the segmentation is of good enough quality. There is a more thorough discussion of the accuracy of the classification process in §7.3.

**Algorithm 6.1** Segment Classification

**Input:** $V$, the segmented voxel array
**Input:** *rules*, the FCL file with inference rules and functions
**Input:** $T_s$, the segment size threshold
**Input:** $T_c$, the classification confidence threshold

```
1: inference_engine ← GENERATE-INFERENCE-ENGINE(rules)
2: for each segment in V do
3:    segment_attributes ← DERIVE-SEGMENT-ATTRIBUTES(segment, V)
4:    size ← GET-FROM(segment_attributes)
5:    if size < T_s then
6:       V ← CLASSIFY-SEGMENT-AS("unknown", segment, V)
7:    else
8:       segment_class ← INFER-CLASS-FROM(segment_attributes, inference_engine)
9:       class_confidence ← RETRIEVE-CONFIDENCE(segment_attributes,
         inference_engine)
10:      if class_confidence < T_c then
11:         V ← CLASSIFY-SEGMENT-AS("unknown", segment, V)
12:      else
13:         V ← CLASSIFY-SEGMENT-AS(class, segment, V)
14:      end if
15:   end if
16: end for
17: return boundary_list
```

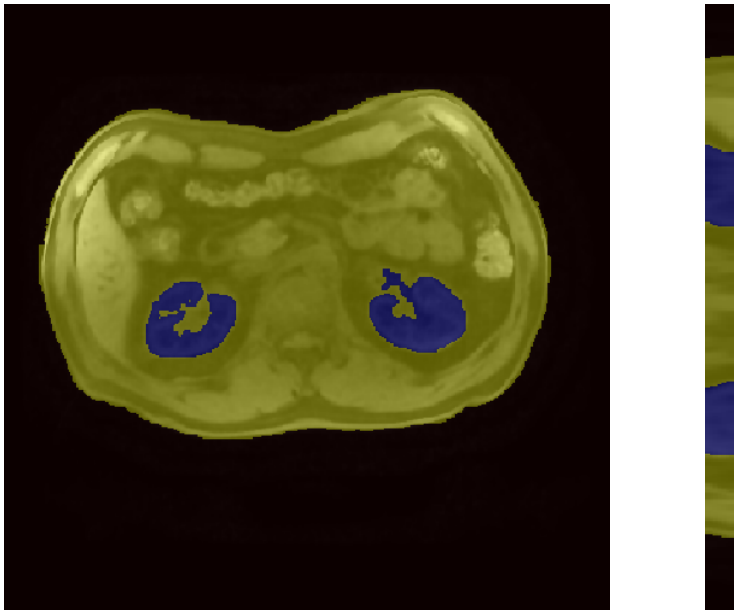

Figure 6.9: Screen capture from the Java implementation of fuzzy inference segment classification showing slice view on X-Y axis (left) and top view on X-Z axis (right). The black region is classified as background, the blue region is classified as kidney and the yellow region is classified as unknown.

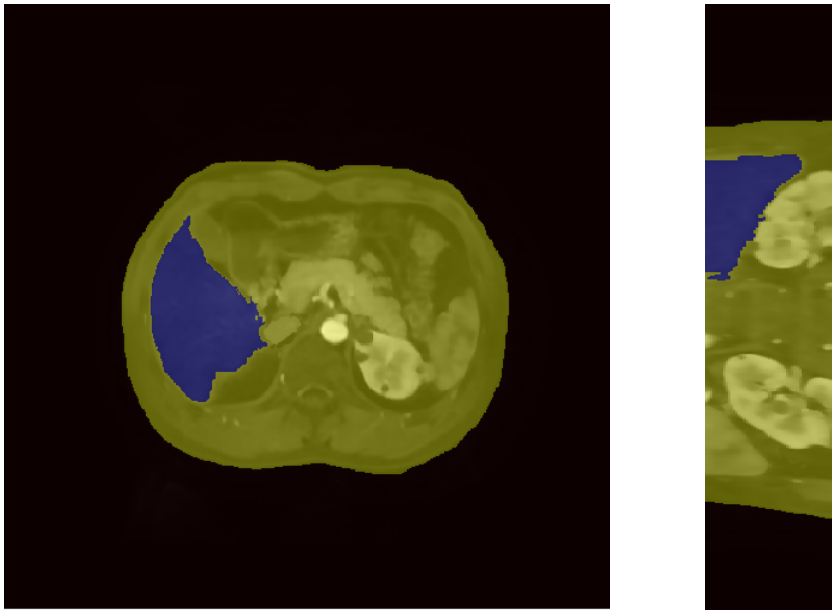

Figure 6.10: Screen capture from the Java implementation of fuzzy inference segment classification showing slice view on X-Y axis (left) and top view on X-Z axis (right). The black region is classified as background, the blue region is classified as liver and the yellow region is classified as unknown.

# Chapter 7

# Validation of Results

The preprocessing, segmentation and classification stages described previously are able to produce an annotated voxel array where each voxel is classified to one of the possible tissue classes. This model would be useful in a full medical application, but at this development stage the results must be checked to ensure that they are accurate and conform to the original aims of the project. This chapter gives a description of the tools and methods used to ensure this conformity.

The implementation was checked against the project aims specified in chapter 1 by a qualitative visual analysis of the models produced from test images. This visual analysis was performed by a general practitioner who has an understanding of both human anatomy and the MRI modality. Two tools were used to allow the doctor to investigate the models created and so that he could make informed judgements on the implementation's performance. The two tools were: a two dimensional slice viewer and three dimensional volume visualisations.

## 7.1 Two Dimensional Slice Viewer

The main tool used to ensure that the processes implemented were performing as required was a slice-by-slice viewer which displays the image volume in two dimensional sections. All of the screen captures shown in previous sections were created using this tool as it allows for the results at each stage to be quickly and accurately displayed.

I constructed this viewer using the classes included in the Java Swing graphical user interface (GUI) toolkit [120]. The viewer is based around a class I created called `ImageDisplay` which is an extension of the Swing object `JPanel`. The panel created by `ImageDisplay` contains three radio buttons to switch between viewing directions, and a scroll bar which allows the user to move through multiple slices in the same image volume. This section of the GUI is shown in figure 7.1.

`ImageDisplay` takes the voxel array from any of the previous stages and converts it into and RGB array where each value within the array represents an RGB colour represented

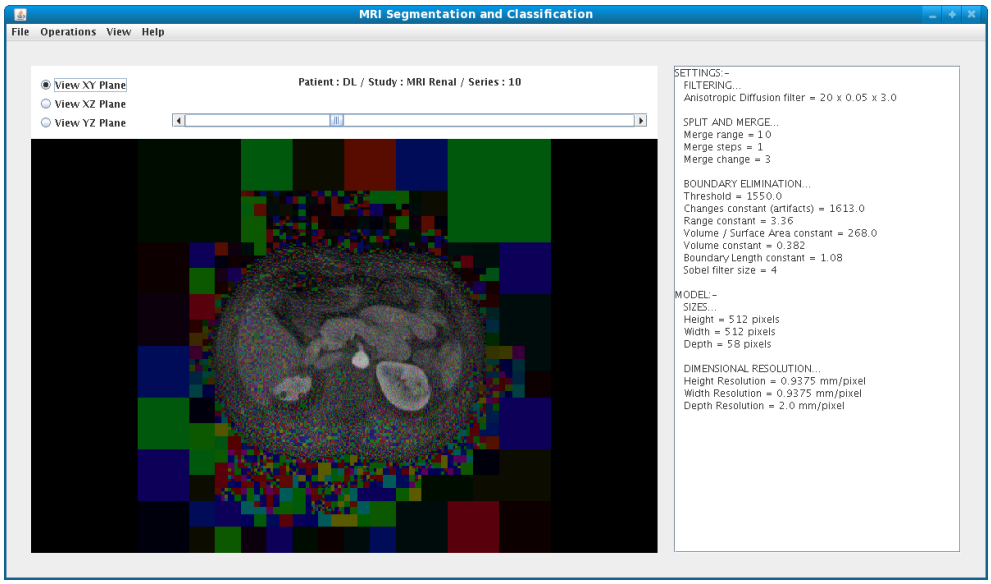


Figure 7.1: Screen capture of the user interface showing the panel produced by `ImageDisplay` on the left hand side.

as an integer. For raw and preprocessed MRI images the values within the voxel array are scaled according to their extrema and converted into grey-scale values. Regionalised and classified images are converted into colour images where each regions is represented by a unique colour, with a key identifying labels for those regions which have been classified.

Using the plane selection radio buttons and the position of the slider, `ImageDisplay` retrieves a two dimensional set of values from the RGB array and displays them on the screen using `JPanel`'s `paintComponent` method.

The important attribute of this viewer is that it overlays the tissue structure profiles on top of the raw MRI images. Overlaying them in this way means that their relation can clearly be seen and analysed. A limitation with validating images using this viewer is that it can only view one slice at a time and only in one of three directions. This means that it cannot be used to asses the continuity and smoothness of surfaces which are out of plane.

## 7.2 Three Dimensional Volume Visualisation

To ensure that all of the structures identified were naturally shaped and had smooth representative boundaries they were checked in three dimensions. This was done by converting the labelled voxel array into a set of three dimensional meshes each of which related to a separate tissue structure. These meshes were then rendered so that they could be orientated and viewed as required. To perform this conversion and visualisation I used Golodetz's [37] visualisation program, the results of which can be seen in figures 7.2 to 7.4.

Figure 7.2: Screen capture from the three dimensional visualisation of the segments classified as part of the kidneys.

Figure 7.3: Screen capture from the three dimensional visualisation of the segments classified as part of the kidneys.

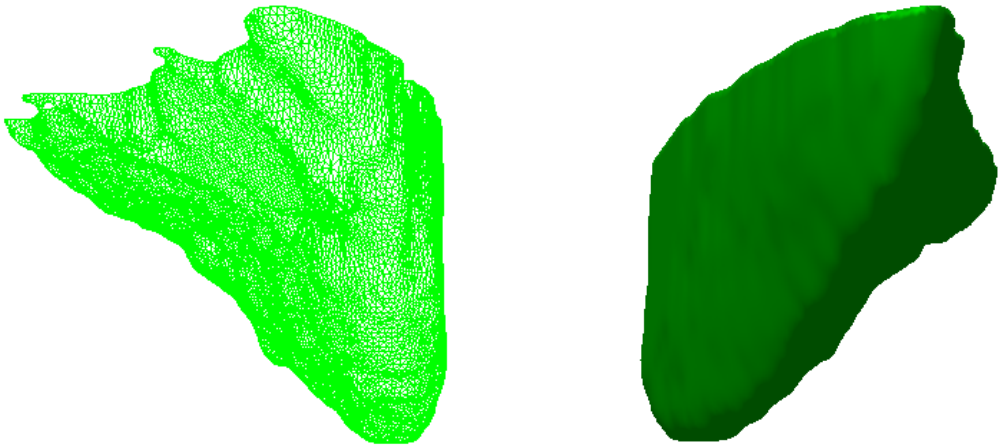

Figure 7.4: Screen captures from the three dimensional visualisation of the segment classified as part of the liver. Image mesh (left) and rendered image (right).

## 7.3 Qualitative Medical Assessment

The models produced from a set of test images were analysed by Dr. James Crossingham of Birmingham NHS using the visualisation tools described above. The two key areas which required validation were the quality of the structure profiles and the correctness of the structure labellings.

### Position and Geometry of Identified Structures

Dr. Crossingham's assessment of the model's segments is that they represented the real tissue structures within the MR images with a good degree of accuracy and would be suitable for a range of medical applications. He stated that the kidney segments depicted both the smooth external surface and the hollow internal cavity correctly and the liver segmentation corresponded to the raw MRI images.

By looking at the rendered three dimensional models for each segment Dr. Crossingham stated that the segment surfaces for both the kidneys and liver were as he would expect and that rendered images such as those could be used to represent structures in a practical application.

One inaccuracy noted in the segments produced was that the process had merged parts of smaller conjoined structures into the main organ segments. A good example of this is shown in the right-hand kidney segment in figure 7.2. It can be seen in this image that there is a small protrusion from the side of the internal cavity. From investigating the MR images Dr. Crossingham believes that this is the renal artery which is the main blood vessel which feeds into the kidney. As the renal artery is physically connected to the kidney this is invariably a difficult structure to separate using segmentation techniques without a detailed anatomical knowledge. Dr. Crossingham stated that incorrect segmentation such as this would be harmful for fully robotic surgery, but for many of the potential applications which he envisioned it would not be a problem.

### Validity of Structure Labels

Dr. Crossingham stated that the labels in the test images were the same as he would specify for the segments to which they were allocated. He also said that the rules utilised were medically sound, and their human-understandable form was helpful.

After receiving an explanation of the classification process, Dr. Crossingham stated that this system could be used for diagnostic purposes. There are various illnesses which affect the physical attributes of patient tissue structures in such a way that they could easily be recognised by "IF... THEN..." rules similar to those already in use. Information such as the number, size and vertical position of organs can be enough to identify certain illnesses. For example, a rule could be included for ectopic kidneys with slightly modified inference criteria from the standard kidneys and the system would be able to produce a warning for such illnesses with almost no program modification.

### Additional Medical Comments

While assessing the models produced, Dr. Crossingham commented on some other areas which I believe to be relevant. One comment was that the the ability of this technique to produce accurate surface rendering was particularly useful. There are certain diseases such as polycystic kidneys and liver cirrhosis, where the surface of the organ is of particular diagnostic importance. Surface texture is difficult to detect in two dimensional slice images and therefore the three dimensional visualisations produced from a segmented organ are very useful.

Dr. Crossingham identified that this system would be useful for teaching medical students as it can give a clear identification of the position, size and geometry of organs to those who are not familiar with MR images. In medical training a consultant teaching a group of students would select an image series displaying a certain attribute and point to the rough position of key structures. Using this program the image series could be processed prior to the lesson and the consultant could cycle through the various components identified. Teaching is this way will show their boundaries and components as clearly highlighted areas rather than rough points and Dr. Crossingham believes this will make the information more understandable.

# Chapter 8

# Conclusions

## 8.1 Research Assessment and Contributions

The system which I have constructed is able to convert raw MR imaging data into patient models which define the position and geometry of a patient's kidneys and liver.

The method implemented follows three stages to detect these positions and geometries:

- It parses the raw MRI data into a processable format;
- It preprocesses this data to limit the effect of various imaging errors;
- It then segments the data into distinct segments which potentially represent tissue structures.

For each of these stages there is a very wide range of available methods described in the literature and thorough research was carried out to ensure that the methods selected were the best suited to this application.

The majority of the techniques utilised were well documented in the literature and have each individually been applied to similar applications in the past. However, none of these techniques could be applied blindly as standard and each had to be modified in some way to ensure they would perform as required. The split and merge segmentation was a particular example of this adaptation. It required that a range of parameters and functions were specified and had to be combined with two other secondary segmentation techniques to achieve the best results.

Though they were each well documented, I could find no mention of all of the techniques being used in this combination, hence I believe the composite process is unique. Its performance against the main aims is discussed below, but I believe that the results produced, particularly in terms of robustness, are comparable to state of the art methods described in the literature.

In addition to the common techniques, I included two original elements. The first of these was using an oct-tree for split and merge. This has been mentioned in the literature, but I do not believe that it has been implemented successfully before now. Additionally, I

created a modified version of a region adjacency graph so that it could be used to simulate an image split tree making the segmentation process more efficient.

Once the system has defined the position and geometry of several structures it can correctly identify and label kidney, liver and background segments. The fuzzy inference system used was very successful at classifying segments and its primary advantage is the way that it operates based on human understandable rules as would be expressed by a radiographer. In all of the tests carried out the fuzzy inference classifier identified the exact segment that a human would for liver, kidney and background, even when the segmentation was less than perfect and contained flawed surfaces.

I concede that the classification test is somewhat limited as it only operates on four classes, but I believe that the rules incorporated are robust enough that they would be suitable even if many more classes were included for similar tissue structures. Importantly, the flexibility of fuzzy inference as a classification method means that new classes could very easily be included if required.

In addition to these main goals, the program developed also meets many of the secondary aims specified in Chapter 1.

### Accuracy and Precision

It was shown in §7.3 that the models produced would be considered accurate and precise enough for use in a medical application. No quantitative assessment was made of these properties, but a visual examination of the models produced show that their boundaries correspond closely to real tissue structures and I approximate that at least 75% of the tissue boundaries are within 2-3mm of the ideal positioning.

This level of accuracy is good enough for the models produced to be of use for some applications, but there are others, such as surgical robotics, where the errors make the model unsuitable. Accuracy was a particular problem in areas where two or more tissue structures with similar MR properties were in contact with each other. Boundaries such as these are difficult to identify for humans, but most radiologists would be able to approximate a boundary down the middle. The problem with these boundaries for my implementation is that it merges the two separate structures into one as is seen in the rendered kidney images in Chapter 7.

I explained in §4.8 that segmentation methods must balance the type and amount of knowledge which they incorporate. I believe that my segment inaccuracies are a result of my focus on robustness and due to my selection of techniques which apply general rather than specific knowledge. There is lots of literature relating to segmentation, but nowhere is there a description of a method which is able to segment every tissue structure accurately. I think that this lack of 'all-round' techniques is due to the amount of specific knowledge which would have to be incorporated. I noticed during my investigation that there was a plateau in new segmentation research approximately five to ten years ago and very few ideas have shown particular improvements since then. I feel that this is due to the limitations of applying specialised knowledge in a general way. I think that it

would require a very novel approach to representing and incorporating information before a technique combining both robustness and accuracy could be created.

### Robustness

It became clear from researching each component in the process that the initial aim for robustness as "accurate identification of most organs in one image volume" was unrealistic. The primary reason for this is that MR images are simply not able to give the required level of definition, clarity and focus across a large range of organs in one image. In addition, my analysis of the segmentation methods available showed that a certain amount of specific knowledge is required in order to achieve a reasonable level of accuracy and there is no all-round method which can achieve an acceptable accuracy for all possible segments.

In spite of my initial overly ambitious definition of robustness, I believe that the method I have employed is actually very robust in terms of variation between structures and patients as well as in terms of imaging errors. Robustness has been used as a key requirement in selecting each of the component techniques used in the complete program. I therefore know that the fundamentals of each section are based on general information and can be applied to a wide range of tissue structures and imaging parameters. My validation of the program shows that it can segment and classify multiple tissue structures which have very different physical appearances. The only aspect that required modification between tissue structures is the number of steps in the simulated annealing process and I believe that if suitably clear images of other tissue structures were used they could be segmented equally successfully.

### Fully Unsupervised Operation

Throughout the process of developing this program, all methods have been selected so that they can operate without any user interaction. Once the MRI data is input the program automatically carries out all of the required actions to produce the patient model. This meets the requirement set in Chapter 1, but the result is that there are many parameters which I have had to manually tune. The reason for this compromise relates back to the incorporation of prior knowledge into the segmentation process. Without a user imparting information at runtime the only way that a system can accurately segment complicated images is to have enough general knowledge built in. In my segmentation method this knowledge is included through the parameters for the different component processes and they are fundamental to unsupervised operation. The detail of this is explained in Chapter 4.

### Processing Time

The main limitation of the implemented techniques is their computational demand. It was stated throughout that this would be a low priority for the project and therefore I do

not think that this is a significant problem but it requires some consideration. In testing, processing a full medical volume of 50 images has taken up to 5 hours to complete. I believe that this is reasonable and therefore meets the initial requirement but it is not ideal. The reason for this slowness is the split and merge grouping process, where each of the regions is checked for uniformity with each of its neighbours. This is an inherently slow process for an image with many regions due to the high number of relations and would be difficult to completely circumvent. As the operation is fully unsupervised and as the majority of applications will not require models urgently the model generation time is of very low significance.

A related limitation of this system is that it requires a large amount of memory to run. To process the testing data required 10-12GB of memory. This memory requirement is due to the regional operation of the split and merge algorithm. In order that the boundaries are represented accurately, the regions must be split so small that there can be up to 8,000,000 regions in a full volume. I feel that the memory requirement is also an innate component of the split and merge algorithm, but as processing was a low priority it was an acceptable compromise.

## 8.2 Further Work

In addition to the proposed improvements for segmentation identified in §5.2, there are some general areas which could be modified to improve the processes discussed previously.

In much of the literature relating to preprocessing, segmentation and classification there are methods described which operate on multi-spectral images[1] [23, 90, 101, 107]. Throughout the course of this research it was clear that the amount of information utilised in the model construction was reflected in the accuracy of the results. Using multi-spectral images can increase the amount of information greatly and I believe that the accuracy improvement, particularly in segmentation, would be worthwhile. In addition, using multi-spectral images would make a significant improvement with respect to identifying multiple organs within one model as each image could focus on creating good contrast for a specific set of structures.

Another area where my process could be further developed is by integrating the segmentation and classification stages. I stated previously that the performance of classification is highly dependent on how accurately the segments correspond to the true patient anatomy. It should be possible to adapt the boundary elimination stage to interact with the classifier so that segments could be constructed in way which suits the requirements of the classifier. The classifier could be used as a heuristic in the boundary elimination stage to encourage segments with a high classification confidence.

A more detailed investigation of preprocessing methods would lead to improvements in segmentation. It is evident from the research that the quality of the data is very

[1] Multi-spectral images are those which have more than one value per spatial position. Each value is acquired using a different MRI contrast or possibly using a different modality all together.

important to segmentation and the methods that I employed were not able to remove all of the errors present. Non-uniformities caused by RF inhomogeneities were the largest problem during the development process. Creating a segmentation technique which could accurately recognise whole regions affected by non-uniformities was very difficult as the difference in value across such regions can be very large. If a more complex method could be employed to eradicate non-uniformities then the segmentation process could be simplified and processing times as well as the number of parameters would reduce dramatically.

Ideally I would have quantitatively assessed the models produced using either manual labelling, phantom images or imaging simulations [23]. By implementing one of these methods the accuracy and precision of the implemented techniques could be assessed as exact values and would allow for the performance to be validated more consistently.

## 8.3 Concluding Remarks

Throughout the course of this project I have developed many skills and built an understanding of many varied new areas. I was completely unaware of segmentation previously but as a result of the research I have carried out, I now have a deep understanding of this field. Moreover, I feel that my research and analysis of segmentation has developed an awareness of the complexities of applying *a priori* knowledge to computational problems. Researching new topics such as segmentation and image processing has been a large component of this project and this has helped me to develop strong research skills. I am now able to synthesise large amounts of new information from many difference sources and can asses this new information by comparing it to what I have acquired from elsewhere. To implement the techniques I researched I was required to learn object oriented programming using Java as well as user interface construction for the validation stage.

In addition to these new skills, this project has given me a good opportunity to apply much of what I have been taught as part of the masters course. The *machine learning*, *intelligent systems* and to a lesser extent *computational linguistics* courses were the most relevant to this project and components of these topics were involved in many of the processing stages. Their relevance to classification is obvious, but fields such as Bayesian networks and Markov chains were pertinent for both preprocessing and segmentation as well. Elements from the *foundations of computer science* course were useful throughout the research stages and an understanding of complexity was invaluable in assessing some of the proposed segmentation techniques when considered with respect to the large size of medical data. The *functional programming* course provided me with some useful tools for designing implementing the various components of this project. Even though I used the Java programming language, techniques such as recursion, function typing and data parsing were all relevant.

# Appendix A

# Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI, also referred to in the literature as Nuclear Magnetic Resonance Imaging) is an imaging modality commonly used in medical applications which utilises the resonance of atomic nuclei to measure and visualise internal tissue structures [16, 56, 76, 79].

### Proton Spin

Hydrogen is the most common element in the human body and is contained in water, fat, protein and other more complex biological tissues. Hydrogen's atomic structure is a single proton and a single electron, but the electron has a negligible effect on the the nuclear magnetic resonance and therefore takes no part in the following explanation. All protons (hydrogen nuclei) have a magnetic field and hence also have a magnetic axis; according to current particle and quantum theory protons rotate about this magnetic axis with a property termed 'spin'. In nature (not under the influence of magnetic fields) proton magnetic axes are random in their orientation and hence all spin in different directions.

### Magnetic Fields

When an external magnetic field is applied to a proton the proton rotates so that its magnetic axis is aligned with the external field. If a strong enough magnetic field is applied to a tissue which contains multiple protons then all of the protons align themselves with this field and their uniform spin is termed a 'precession'.

When a group of protons are precessing their individual magnetic fields form a larger group field (referred to as 'net magnetisation') which is aligned parallel to the external field ('longitudinally'), but this longitudinal net magnetisation is much weaker than the external field required to form the precession. Due to the comparative weakness of the net magnetisation and the fact that it is aligned parallel to the external magnetic field it cannot be detected by any external instruments, and therefore has limited potential for determining any tissue properties.

### Nuclear Resonance

When protons are precessing their frequency of rotation (termed 'resonant frequency') is a direct property of the strength of the magnetic field which induced the precession and the compound structure containing the proton[1]. This relationship means that for a known magnetic field strength the resonant frequency of protons within a tissue of interest can be calculated.

### Radio Frequency Pulses

If protons in precession interact with a radio frequency pulse which matches their resonant frequency they resonate causing them to move out of alignment with the main magnetic field. If the amplitude and duration of the pulse are of a high enough value then the protons can be rotated by a full 90° into the transverse plane. The resonant frequency of hydrogen nuclei in water based tissues, lipid (fat) based tissues and fluids are all in a similar range and therefore the pulse can be tuned so that it will cause all of them to resonate and 'flip' in to the transverse plane at the same time.

Once the pulse has been turned off and the protons are unsupported in the transverse plane they eventually succumb to the external magnetic field and flip back into the longitudinal plane. When they do this they release the energy that they acquired from the pulse in the form of an electro-magnetic pulse of their own which can be detected externally.

### Image Extraction

MRI units operate on the principles stated above so that they can detect specific hydrogen nuclei within a patient. As the electro-magnetic pulses generated by the protons are non-directional (as with all electro-magnetic waves) a small magnetic field gradient is applied to the patient so that the protons will respond the the excitation pulse in a slightly different manner in different locations allowing for spatial information to be extracted. This final value extraction is actually carried out with with a Fourier transform as the apparatus detects the signal from protons as frequencies.

The method above explains how proton density (PD) can be extracted from a patient allowing for the construction of a PD-weighted image where areas of high proton density have a higher intensity and areas of low proton density have a lower intensity. This method of building image contrast is considered inadequate in modern radiography as proton density is of limited diagnostic importance.

Instead, MRI units detect and record the time taken for each region of protons to return to longitudinal precession once the pulse has been applied (termed the 'relaxation time'). Protons in different tissue structures have different relaxation times, and therefore

[1]This relationship is defined formally in the Larmor equation, for a full explanation the reader is referred to [16] and [79].

monitoring some aspect of this property allows for the extrapolation of more detailed tissue information.

There are three aspects of relaxation time which are commonly monitored and these aspects allow for the creation of $T_1$, $T_2$ or $T_2^*$ - weighted images. Each of these contrasting schemes highlight different tissue aspects and boundaries, for example, $T_1$ is typically more suited to lipids and fat based structures and $T_2$ is more suited to water based fluids. Due to the complexity of selecting between them it will not be discussed further in this document.

It should be noted that for all of the techniques researched and implemented in this document it will be assumed that the radiographer who acquired the images selected the optimal contrast and settings to provide good general organ identification.

An important feature of magnetic resonance (MR) images is that the raw intensity values do not actually represent any physical property of the tissue, and are a product of multiple settings. For this reason the values should only ever be considered relative to each other rather than as absolute.

# Appendix B

# DICOM File Construction

## B.1 File Format

The specific details of the DICOM file format are listed formally in part 10 of the DICOM standard (PS. 3.10-2009) [82] but a brief explanation of the format is included here for completeness.

DICOM image files are essentially two lists of image attributes which are stored in a standard binary representation. The first list consists of file meta information and the second contains the actual image data and any associated information.

#### Data Elements

An image attribute is an individual piece of information which defines some specific property of the image, patient or imaging modality. Some examples of image attributes are image dimensions, patient name, modality settings and pixel values. Each image attribute is encoded in the DICOM file as a *data element*, which has the construction specified in part 5 of the DICOM standard (PS. 3.5-2009) [84]. The standard states that each data element should contain either three or four distinct pieces of information depending on whether *explicit* or *implicit* value encoding is used. These pieces of information are shown for a single element in a sequence of data elements (data set) in figure B.1.

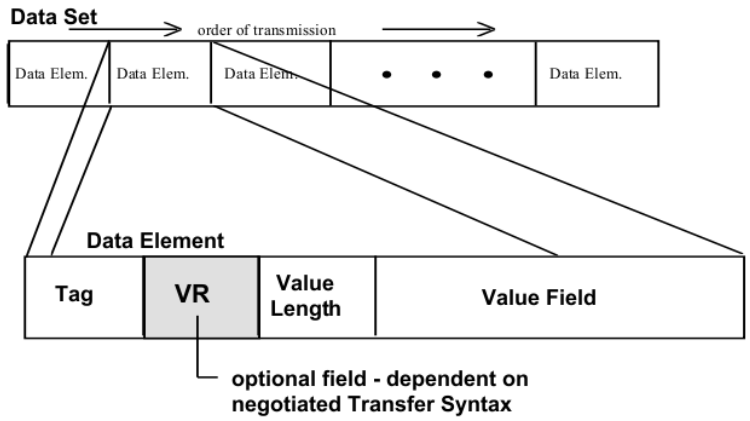


Figure B.1: DICOM Data Element Construction [84]

The first piece of information included in a data element is a *tag* which identifies what image attribute the information is pertaining to. Each tag is actually a pair of 2-byte values, where the first value is the group number, $g$, and the second value is the element number within the group, $e$. The numbers are stored in the binary file in little-endian format, and are represented visually as 4-digit hexadecimals in the form "($gggg$,$eeee$)". Part 6 of the DICOM standard (PS. 3.6-2009) [85] is a data dictionary which includes a list of the tags used to identify each standard image attribute.

The second (optional) section of a data element is the value representation (VR) which defines the method in which the attribute value has been encoded into the binary file. There are multiple different value representations used in a single DICOM file, and the meanings and decoding methods are listed in PS 3.5-2009 [84]. This field is optional, because in some files the file meta information (explained below) states that all data set elements shall have implicit value representations, that is to say that for all data elements the value representation shall be assumed to be the one 'usually' employed. Each value representation is identified by a two character string, which is encoded in the DICOM file as 2-bytes, where each byte refers to a single ISO/IEC 8859 character.

The third piece of information included in a data element is the value length. The value length refers to the number of bytes in the binary file which have been allocated to store the given attribute (excluding tag, value representation and value length). The value length is an integer, encoded in either 2 or 4-byte binary depending on whether an explicit value representation is included. With some exceptions (stated below) the total number of bytes allocated for value representation and value length is 4-bytes, therefore if an implicit value representation is used then all 4-bytes are available for the value length. However, if an explicit value representation is included then only 2-bytes are available.

The exception to this data element structuring is for explicit value representations (OB, OW, OF, SQ, UT and UN) where the value length is potentially quite large and 4-bytes are required to store it. In these cases 8-bytes are used to store the value representation and length. 4-bytes are allocated for the value representation (the standard 2-byte string

and 2-bytes of 0s to fill space), and an additional 4-bytes are included to accommodate the large value length. There is a further complication to this system when some of these 'special' value representations (OB, OW, SQ, and UN) have undefined value lengths. In these cases the value length field takes the value FFFFFFFFH, and the value termination is set by some predetermined delimiter.

The final data element field is the actual attribute value itself. It has a length of the number of bytes defined in 'value length', and can be decoded by a method dependent on the value representation.

### File Meta Information

As stated previously, a DICOM file consists of two lists of image attributes (data elements). The first list is referred to as the "DICOM file meta information" and identifies that the file is indeed in the DICOM format and defines some key attributes used in decoding the remainder of the data elements.

The first two pieces of information in the file meta information are not standard data elements. The first is the 128-byte file preamble which is allocated for use by specific DICOM implementations or application profiles. There is no defined structure for this section of a file, and each byte can be set to 00H if it is not used. The second non-standard part of the DICOM file is the 4-byte DICOM prefix. This is simply a representation of a four character string, which should be "DICM" to confirm that the file is in the DICOM format.

After these two pieces of information the file meta information consists of between 6 and 10 data elements which identify key aspects about the way in which the data has been encoded as a DICOM file. To avoid any potential confusion this section of the file is always encoded using explicit value representations and little endian ordering. The full list of required data elements is contained in table 7.1-1 of PS 3.10-2009, but the two fundamental ones are "Transfer syntax" and "File meta information group length". The transfer syntax data element is a unique identifier which prescribes whether the main data set is encoded explicitly or implicitly and in big or little endian. The group length defines the length of the meta information and hence identifies the point at which the main data set begins and where the given transfer syntax should be used.

### Data Set

The second list of data elements is referred to as the "Data set" and contains information about the image and associated parameters. Each DICOM file must contain a single data set which represents an instance of a single SOP class. It should be noted that limiting the DICOM file to a single data sat does not mean that only one image can be stored per file, as multiple images can be included in one SOP class and hence in one data set.

The data set is simply a list of data elements which are placed in order of ascending attribute tag. Each data element in the set can be decoded using the transfer syntax set

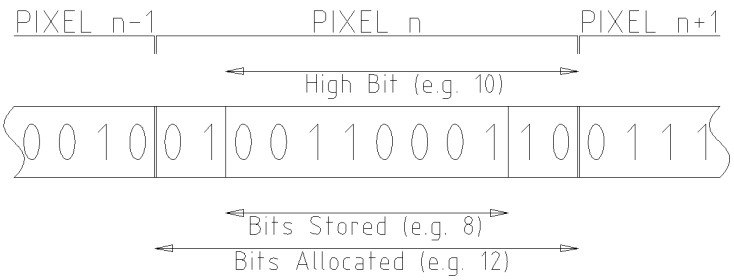


Figure B.2: DICOM Pixel Construction Example

in the file meta information allowing a particular attribute to be set for the given image. There are numerous different pieces of information which can be included in a DICOM data set as data elements but not all of them have to be included in a file if the encoding software/imaging equipment does not deem them necessary.

## B.2 Image Encoding

The DICOM file format utilises multiple data elements to encode a single image. For example one data element stores the number of pixel rows, another stores the number of pixel columns, and others store information relating to the pixel value range. The actual pixel values are always encoded as a pixel data element (7FE0,0010) with an overlay data element (60xx,3000) included when transparency is required in sections of the image.

Pixel values can be encoded in the pixel data element in various ways, some of which employ compression (encapsulated) and some of which do not (native).

### Uncompressed Pixel Data Elements

Uncompressed pixel data is stored in a data element's value field as a concatenation of pixel values represented in binary such that all rows and columns of an image are encoded as a single string. Each pixel has a set number of bits allocated (listed in the data element (0028,0100)), but not all of the allocated bits have to be used to store the pixel value. In some cases some of the allocated bits are used for application specific pixel information such as planar overlay values, and therefore the number of bits used to store the actual pixel value is stated explicitly in the DICOM file (listed in the data element (0028,0101)). In order to understand where the range of bits used to store the value sit within the range of bits allocated, a third property is included in the file which is the location (in reference to the allocated bits) of the high stored bit. The relation between these three values is illustrated in figure B.2.

As the pixels are stored as a single one dimensional string, in order to be converted into a two dimensional image they require dividing and reconstruction. The pixels are ordered row by row left to right and top to bottom; therefore knowing the image width (detailed

in a separate data element) allows for decoding and encoding.

#### Compressed Pixel Data Elements

The DICOM file format supports many lossless and lossy compression techniques such as various JPEG methods, run length encoding and MPEG2 MP@ML (for multi-frame images/video). Due to the layered construction of the DICOM file format compressed images (as binary) are simply included in the pixel data element as with an uncompressed image, and a special transfer syntax (explained previously) is used to identify the type of decompression that the binary data will require.

Due to the fact that the sample images for this project do not contain any compressed data the exact processes for compression will not be covered in any greater detail.

#### Multi-Frame Images

As stated previously, the DICOM restriction that a single file can only contain one data set does not restrict the file from containing multiple images. This is because the pixel data element can act as a sequence of sub-elements, each of which contains the pixel information for a single frame. This sequence is constructed in the same way as a sequence is for the value representation SQ, where each sub-element (or item) is identified with an item tag (FFFE,E000), and the sequence is ended with a delimiter tag (FFFE,E0DD).

## B.3 Directory Files

The DICOM standard allows for groups of DICOM files to be identified as a set with the use of directory files, referred to as DICOMDIR. The construction of these directory files is defined in part 3 of the DICOM standard (PS. 3.3-2009) [83]. The standard states that DICOMDIR files are constructed in a similar fashion to standard DICOM files in that they consist of a meta information header and a main body which consists of multiple elements. The meta information sets identify basic information such as the transfer syntax, while the data set contains the directory record sequence (0004,1220), which shows the relation between each individual DICOM file. The directory record sequence contains four types of record: Patients; Studies; Series and Data (Images).

#### Patients

Patient records are considered as the root directory in the sequence and they define which patient the data has been taken from. There can be one or more patients in the directory, each of which has a disparate set of studies.

### Studies

The second level directory in the record sequence is the studies. These are sets of patient data which have been collected in the same area of interest. This can be an anatomical area of interest or a set related to a specific type of treatment. There can be one or more study per patient, each of which has a disparate set of series.

### Series

Series are the lowest directory level and group images together into sets which have been taken in a single imaging pass. That is to say that a series will consist of multiple images of different areas which were taken at the same session and can be conjoined to form an overall picture of the patient at that instant. There can be one or more series per study, each of which has a disparate set of images.

### Data (Images)

These records are the main body of the directory sequence and correspond to the individual DICOM files. The directory record contains information about the file and image name. Therefore taking the DICOMDIR file (where the folder containing the images is known), the exact images for a given patient, study and series can be retrieved from their respective DICOM files.

# Appendix C

# Review of MR Image Preprocessing Techniques

## C.1 Noise Suppression

### Low-Pass Filtering

McVeigh, Henkelman, and Bronskill [77] propose the use of a low-pass spatial filter to remove random noise in MR images. They found that the random noise common in MR images is of a higher frequency than the general tissue structure, and therefore by attenuating the high frequencies the noise can be reduced.

Methods such as mean, median and adaptive filtering can all be used as low-pass filters [38], but a Gaussian averaging operator (*Gaussian filter*) is considered to be optimal for smoothing high frequency noise [86]. A Gaussian filter calculates a filtered value for a point in an image by taking the weighted average of all points in a window centred on it. The weight for each point within the window is defined by its distance from the centre element and the Gaussian distribution. The standard form for the two dimensional Gaussian distribution with a 0,0 mean is shown below.

$$g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \tag{C.1}$$

where $g(x,y)$ is the Gaussian distribution value at point $x$,$y$
$\sigma^2$ is variance of the Gaussian distribution
$\frac{1}{2\pi\sigma^2}$ is a normalising value so that the distribution sums to 1.

The common method for applying a Gaussian filter is through the application of a discretised convolution mask to every IE. An example mask for a 5×5 operator with

standard deviation 1.0 is shown below.

| | | | | |
|---|---|---|---|---|
| 0.002 | 0.013 | 0.022 | 0.013 | 0.002 |
| 0.013 | 0.060 | 0.098 | 0.060 | 0.013 |
| 0.022 | 0.098 | 0.162 | 0.098 | 0.022 |
| 0.013 | 0.060 | 0.098 | 0.060 | 0.013 |
| 0.002 | 0.013 | 0.022 | 0.013 | 0.002 |

(Adapted from [86])

One advantage of this method is that it can easily be extended into three dimensions by the creation of a three dimensional convolution matrix. As the processing time is linear on the number of IEs, it is also computationally efficient.

McVeigh, Henkelman, and Bronskill [77] state that low-pass filtering reduces the spatial variance of an image, which is an important property for detecting tissue segment homogeneity. However, this is at the cost of blurring edges and decreasing spatial resolution. The reason for the loss of spatial resolution is that the filter can combine values from either side of an edge causing the middle elements to be allocated an intermediate value which does not correctly belong to either. Another limitation of this technique is that noise in MRI is often considered as Rician [65] and low-pass filtering does not provide a way for correcting this.

### Anisotropic Diffusion Filtering

As stated above, the problem with simple smoothing filters such as the Gaussian filter, is that they smooth the whole image including the edges of any pictured objects. As edges take an important role in many segmentation techniques this loss is not ideal. Perona and Malik [99] proposed a process called *Anisotropic Diffusion Filtering* (ADF) to overcome this difficulty by only applying smoothing within pictured objects and avoiding their edges.

A parallel is drawn between the process of recursively smoothing an image using a Gaussian filter[1] and the process of temporal thermal diffusion in a two dimensional plate. Considering the smoothing process in this way means that a Gaussian filter can be applied anisotropically so that it is dependent on certain properties of the image location[2] [86]. The full detail of the derivation will not be shown here[3], but the thermal diffusion equations can be used to show that a Gaussian filter can be applied anisotropically in an iterative way such that:

$$I_{i,j}^{t+1} = I_{i,j}^{t} + \lambda \left[ c_N \cdot \nabla_N I + c_S \cdot \nabla_S I + c_E \cdot \nabla_E I + c_W \cdot \nabla_W I \right]_{i,j}^{I} \tag{C.2}$$

[1] This recursive smoothing process is referred to in the literature as the construction of a multi-scale image representation where the image resolution is gradually reduced.

[2] The parallel here is that different materials have different thermal conductivities in different directions and therefore the process of modelling thermal diffusion over time in such materials is anisotropic and has well established methods.

[3] See references [86] and [99].

where $I^s_{i,j}$ is the filtered image at iteration step $s$
$\lambda$ is the step smoothing control constant such that $0 \leq \lambda \leq \frac{1}{4}$
$N$, $S$, $E$, $W$ are the four directions in the image: North, South, East and West
$c_D$ is the 'conduction coefficients' in direction $D$
$\nabla_D I$ is the nearest neighbour differences in direction $D$.

The important component of this equation is the *conduction coefficients* which define how much each direction influences the smoothing at each step as a function of the image gradient in that direction. Gradient identification is fundamental to ensuring that filtering is applied orthogonally to the gradient rather than parallel. The starting point (boundary condition) for this process, $I^0_{i,j}$, is the image to be filtered, and this equation can be used to iteratively create smoother and smoother images which respect edges.

A modified version of Perona and Malik's ADF has been applied to MR images [54], where they claim the technique is robust and its quantitative performance is better than other "state of the art" methods. However, due to the iterative and complex nature it is slower to compute than more basic convolution mask based methods. Moreover, according to Manjon et al. [69] anisotropic filtering is prone to suppressing small features and the edge enhancement process can modify the statistical properties of the whole image.

### Frequency Domain Filters

There are a class of filtering methods discussed in the literature which analyse and modify images based on their representation in the frequency domain. To use a frequency based filter the image (or part of it) must first be converted into the frequency domain. This transform is often carried out with a Fourier based transform such as the *discrete Fourier transform* or the *discrete cosine transform* [135].

Filtering an image based on its frequency representation can be done in many different ways, but the general concept is that certain frequencies can be amplified, or more commonly attenuated, based on an assessment of all or part of the frequency domain. For example, low-pass filtering can be applied in the frequency domain by attenuating all frequencies above a certain threshold [38].

The key drawback of pure frequency based filters is that no spatial information is available and this restricts their ability to fully compensate for Rician noise which is believed to be common in MRI[4] [65]. A solution to this is to consider the image as wavelets by implementing a *discrete wavelet transform*. As wavelets incorporate both frequency and spatial information they can be used to implement more complex filtering methods such as *wavelet thresholding* [29].

Wavelet based filtering has been applied to MR images [87] and is shown to perform well at improving SNR, but the effect of the process on spatial resolution has not been discussed. It should also be noted that due to the computation involved in converting

[4]Rician noise is considered to have a non-0 mean and instead its mean typically depends on local intensity values, hence why spatial information is required for its compensation [87].

between the frequency and spatial domain, frequency domain filtering is not commonly used for 'simple' filters such as low-pass because they can be applied more efficiently in the spatial domain as described previously.

### Non-Local Means Filtering

Originally suggested by Buades, Coll and Morel [17], non-local (NL) means filtering is an extension to standard means filtering. Rather than using some arbitrary spatial neighbourhood to decide on the noise-free value of a pixel, the method uses pixels anywhere in the image if they have some similarity to the target pixel. The NL-means filtered value for a pixel $i$ in an image $I$ is calculated from the weighted sum of every pixel in the image, as shown in equation C.3.

$$NL(v(i)) = \sum_{j \in I} w(i,j)v(j) \tag{C.3}$$

where $w(i,j)$ is the similarity weight between $i$ and $j$ such that $1 \geq w(i,j) \geq 0$ and $\sum_{j \in I} w(i,j) = 1$
$v(j)$ is the value of pixel $j$.

In this equation the weight between two pixels is determined by their similarity to each other as assessed by comparing the neighbourhoods in which they occur. A neighbourhood function is defined, which identifies all of the neighbours for a given pixel[5]. This allows for a grey-level vector to be constructed containing all neighbours' values. The similarity of two pixels can be assessed by applying the Gaussian weighted Euclidean distance between the neighbour vectors for each.

NL-means filtering has been applied to MR images by Manjon et al. [69], and appears to produce some very good results when compared to ADF and a wavelet based filter. Their visualisation of the noise extracted from MR images shows a clear improvement in areas of high image gradient where most methods over-identify noise and damage the image as a result (see figure C.1). They state that they would expect a three dimensional application to be even more successful due to the number of similar voxels which could be found.

The fundamental flaw with this method is the number of neighbourhood vector comparisons which are required to calculate the weightings for each pixel pair. Manjon et al.'s method only searches an 11×11 pixel window instead of the full image, but processing time is 7 minutes for ninety 256×256 pixel slices and they state that they would expect it to be "several hours" for a full three dimensional implementation[6].

[5]The neighbourhood function is typically a square, rectangular or circular window centred on the pixel of interest. Manjon et al. [69] found that a circle with a radius of 2 was optimal for their implementation.

[6]Manjon et al. do not state what platform they are basing their time estimate on, but the paper was published in 2008 and therefore I believe that their estimate is still valid.

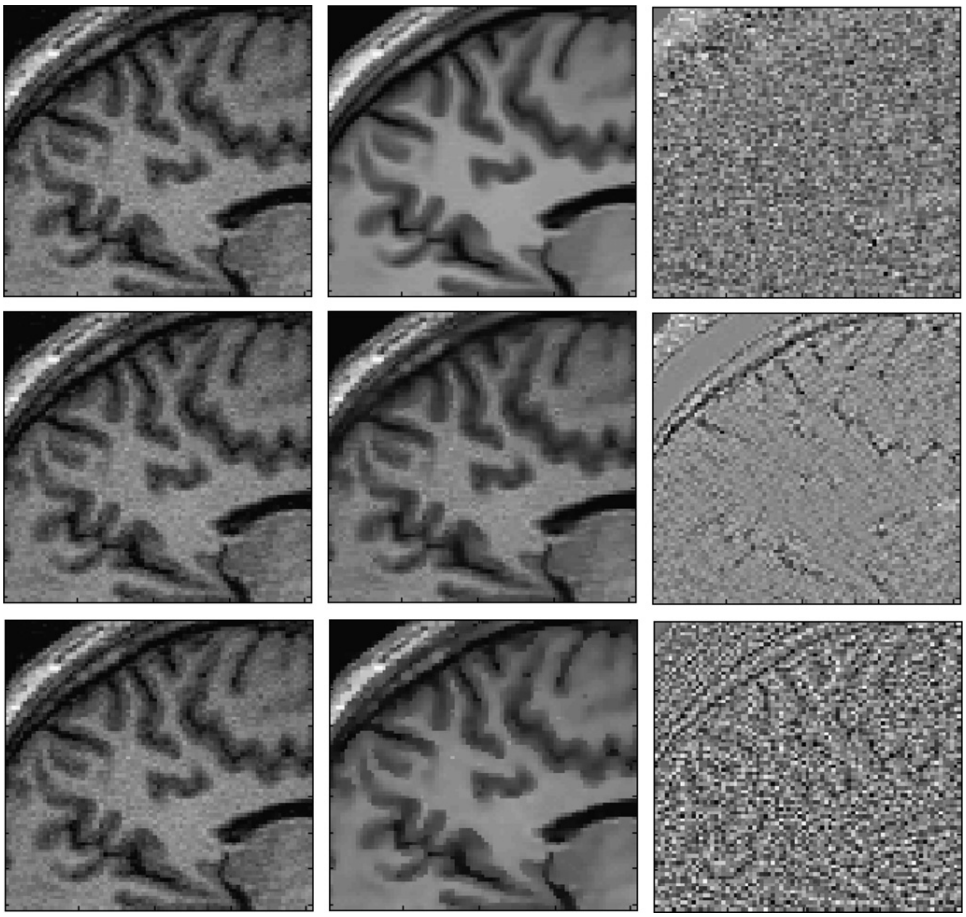

Figure C.1: Images showing noisy raw image (left), filtered image (centre) and residuals (right) for non-local means filtering (top), wavelet based filtering (middle) and anisotropic diffusion filtering (bottom) [69]. This comparison shows that the NL-means method extracts a much smoother noise profile across the whole image and does not produce over filtered edges as is shown for the other two techniques.

There are many more filtering techniques in the literature but only those with a well documented use with MRI have been considered. The reader is referred to [17] for a more thorough review.

## C.2 Image Artifact Removal

Image artifact removal during image preprocessing is a very difficult task due to the way that artifacts can appear as legitimate tissue structures when viewed at most scales. This review was unable to find many descriptions of methods for achieving this with the exception of Bayesian/MRF based methods such as those described below. Many of the segmentation reviews and implementations which considered noise and RF non-uniformities completely disregard artifacts at the preprocessing stage [5, 43, 110] and it appears that the common view is that artifacts should be removed at the image acquisition stage by the radiographer rather than with complicated digital methods.

#### Bayesian Restoration of Images

Geman and Geman [34] developed a technique for both noise and artifact removal which involves the creation of a hierarchical stochastic model that is used to derive the maximum '*a posteriori*' estimate of an un-degraded image[7]. In this method a Gibbs distribution is used to generate a prior model representing prototypical un-degraded image construction and then simulated annealing is used to identify the maximum likelihood estimate of this un-degraded image given the detected signal [48].

This method has previously been applied to MR images and Hu et al. [45] state that "reduction of motion artifacts can be expected if it appears pseudo-random". The method can also reduce noise and enhance edges [45]. This ability to compensate for multiple factors is the largest advantage of Bayesian image restoration, but constructing a prior model which is able to do this is complicated and requires the setting of multiple parameters.

I believe that it would be more logical to incorporate the prior noise model into the segmentation stage if this approach was deemed necessary. The restoration process involves an implementation of a Markov random field in a very similar fashion to image segmentation discussed in §4.2.3. All of the constraints included in a model used for image restoration could be used in a segmentation Markov random field and would therefore make a separate restoration process redundant.

My investigation has identified that image artifacts pose a very complex problem because the recorded signal is fundamentally incorrect. Unlike noise and non-uniformity, where the signal is the product of truth and some error, image artifacts are almost entirely erroneous values and attempting to reconstruct true data can be impossible. For this reason, as well as the very limited range of artifacts which Bayesian restoration can

[7] In this application the un-degraded image is the actual human tissue structure.

overcome, and the lack of alternatives, I have decided that no attempt to remove image artifacts will be implemented for this application. Instead, my implementation will operate under the assumption that the radiographer has limited any large artifacts but will also attempt to be flexible towards any small artifacts which remain. Judging from the lack of methods reported in the literature, this seems to be in-line with the general approach.

## C.3 RF Non-Uniformity Compensation

**Correction Matrices**

A common method for considering an image signal from a MR imaging machine is as the product of three components [36];

$$f(x,y) = a(x,y)b(x,y)e(x,y) \tag{C.4}$$

where $f(x,y)$ is the image recorded at point $(x,y)$
$a(x,y)$ is the true signal at point $(x,y)$
$b(x,y)$ is a spatially dependent illumination function
$e(x,y)$ is a spatially dependent sensor sensitivity.

This relation illustrates that if the two components of RF non-uniformity (illumination function and sensor sensitivity) can be removed from the recorded image then the true signal can be extracted. Both components of the RF non-uniformity in this equation are spatially dependent but more importantly are consistent between images taken from the same apparatus. This means that if the value of $b(x,y)e(x,y)$ can be established from one image it can be used to correct all others produced on the same apparatus. This is the concept behind the application of correction matrices.

A correction matrix can be constructed to represent $b(x,y)e(x,y)$ by taking an image of a homogeneous contrast material. As it is known that the contrast material is homogeneous, the value for $a(x,y)$ becomes constant and therefore the distribution for $b(x,y)e(x,y)$ can easily be extracted. Once this has been done the non-uniform recorded image can simply be divided by the correction matrix to extract the true signal.

A problem with this method is that the matrix can be affected by noise and become convoluted and non-representative of the true non-uniformity. A solution to this is to approximate the matrix as a smoothed mathematical function in a process called surface fitting [27].

These are well established methods of compensating for RF non-uniformities, but are not suitable for this application due to their reliance on 'phantom' images. Using this method for compensation would limit the potential applications to only images produced on MRI machines for which phantom images are available and this goes against the robustness aim of this project.

### Filter Based Compensation

Haselgrove and Prammer [42] identified that the requirement of correction matrices for phantom images was restrictive and developed an algorithm to overcome this. Their method is similar to the correction matrix method, but they derive the correction matrix (which they call a *smoothed function*) from the image itself.

To create a smoothed function Haselgrove and Prammer make two important assumptions:

- The image consists of two components; a uniform background and a group of high frequency objects.
- The RF non-uniformity varies slowly compared to the detail objects.

This first assumption means that the true signal component in equation C.4 is the sum of a constant background, $B$, and a varying foreground, $F(x, y)$, giving:

$$\begin{aligned} f(x,y) &= [B + F(x,y)]b(x,y)e(x,y) \\ &= \underbrace{F(x,y)b(x,y)e(x,y)}_{\text{high frequency}} + \underbrace{B \cdot b(x,y)e(x,y)}_{\text{low frequency}} \end{aligned} \tag{C.5}$$

Using the second assumption, separate high and low frequency components can be identified. This means that if a low-pass filter is applied then only the low frequency component, $B \cdot b(x, y)e(x, y)$, will remain which can be used as the smoothed function. By dividing the recorded image by this function the true signal can be extracted[8].

This method is a useful way to overcome the problem with correction matrices, but the two assumptions used are not always valid therefore the process can fail to remove all of the non-uniformity, or more likely, can degrade the image. The common method for correcting this is to incorporate some form a phantom image which poses the same problems as in correction matrix based methods [23].

[8]The actual result is the true signal divided by a uniform background, but background uniformity means that relative values are correct.

# Appendix D

# Review of Supervised Explicit Knowledge-Based Classifiers

## D.1 Expert Systems

Expert systems are defined by Lucas and Van Der Gaag [64] as knowledge based problem solving agents which employ heuristics that have been taken from human expert knowledge.

The two primary components of expert systems are heuristic representation and heuristic evaluation. A human expert's knowledge is unlikely to be expressed in a machine understandable form and therefore the knowledge must be converted into a representation which is. Once these heuristics have been defined a framework is required which is able to apply them to a query and deduce an output.

#### 'Hard' Logical Deduction

The most commonly used method for representing knowledge is with formal logics such as propositional and first order predicate [35]. The detail of these logics will not be included here, but it is clear that they can be used to represent a range of rules using the standard logical connectives. For example the rule "if the input is large and dark then it is a liver" could be represented with the predicate logic rule:

$$\forall x \; L(x) \wedge D(x) \Rightarrow V(x) \tag{D.1}$$

where $L$ is a unary predicate for largeness
$D$ is a unary predicate for darkness
$V$ is a unary predicate for liver.

If an expert system has a single rule then it can simply be applied to any input to directly deduce an output. It is much more likely that an expert system will include multiple

rules and therefore some deductive system is required which can combine multiple rules. Different logics have different methods available for achieving this deduction, resolution and chaining are two common examples [35].

Expert systems can be used to represent uncertain or inexact rules with Bayesian classifiers and fuzzy inference, but they are considered separately below.

#### Application of Expert Systems to Tissue Classification

Li et al. [59] use an expert system based on on multiple "IF... THEN... ELSE..." rules to classify tissue types in brain MR images. Their method uses a range of features to perform the classification, but key decision attribute is the mean MR value for a segment. They use settings from the MR apparatus to convert grey-scale values back into absolute MR values so that if a segment is within the known range for a tissue type then it can be identified. Their approach can be thought of as a rule based decision tree where if the output from one rule is indecisive then it is fed into another to further refine a label.

#### Expert System Conclusions

Expert knowledge systems lend themselves very well to the problem of tissue classification due to the fact that rules for identification are already well understood by human radiographers. The process can also be computationally efficient if forward chaining is employed as complex spatial constraints do not have to be evaluated for a segment unless absolutely required for the classification.

Li et al.'s classification method achieved good results on white matter within the brain, but was less successful in other, less value-identifiable, areas [59]. Another problem with their approach is that they incorporated rigid structural constraints within some of the rules and therefore the model struggles in regions which had a large degree of variation. I believe that these difficulties highlight the key limitation of this approach which is that the human tissue structure does not lend itself to 'hard' logic as it requires a certain degree of flexibility.

I think that the reliance on absolute MR values in this method is a large disadvantage for two reasons. If there are intensity inhomogeneities within the image then they will lead to incorrect value calculations and therefore incorrect classifications, and the technique required a lot of detailed information about the imaging parameters which may not always be readily available.

## D.2 Bayesian Classifiers

Uncertainty within an expert system requires a different approach to standard logics which cannot incorporate rules which are not perceived to be perfectly representative. A Bayesian classifier (or naïve Bayesian classifier) is a probabilistic framework which uses

Bayes' rule and assumes that all of the *effects* are conditionally independent given a single *cause* allowing a full joint probability distribution to be constructed. This is represented as follows [112]:

$$\mathbf{P}(Cause, Effect_1, Effect_2, \ldots, Effect_n) = \mathbf{P}(Cause) \prod_i \mathbf{P}(Effect_i|Cause) \qquad \text{(D.2)}$$

From this definition it can be seen that if a prior probability is known for a *cause*, and conditional probabilities are known for that *cause* given a set of *effects*, then the *cause's* probability given the recorded *effects* can be calculated. This distribution can be used as a classifier by constructing a joint distribution for each candidate class and then allocating each input to the class for which it has the highest joint probability.

Russell and Norvig [112] describe a method which can be used to derive the values for $\mathbf{P}(Cause)$ and $\mathbf{P}(Effect_i|Cause)$ from training data, but if these values are already known, or can be estimated, by a human expert then they can be implemented directly.

### Application of Bayesian Classifiers to Tissue Classification

I could not find an example in the literature of a Bayesian classifier being applied to medical images, but in similar image based applications [71, 134] the Bayesian classifier operates on a feature vector for each segment with values such as shape, size, colour (grey-scale value) and texture.

Relating this application to equation D.2, the organ name would be the *cause* and the set of recorded segment attributes would be the *effects*. The prior probability for each organ can simply be calculated from number of that organ type expected per patient divided by the number of number of segments detected. The difficult component would be the establishment of conditional probabilities for each organ-attribute pair. This would have to be derived from discussion with radiographers to ascertain the causality that they perceive between the organ and attribute.

### Bayesian Classifier Conclusions

Though the assumption that each segment attribute is conditionally independent is not factually correct, the literature [112] states that in practice the method can still perform well. Also this method overcomes the uncertainty limitation of 'hard' logic and allows radiographers to express rules which they believe to only be true part of the time.

I believe that the major draw back with this method is that the implementation of probability distributions based on qualitative radiographer statements is very unnatural. Unlike in the formal logic rules in the previous section, people do not often express a belief in a way that can be tied to an exact probability. A radiographer may be able to identify certain degrees of strictness for a rule, but this accuracy is not as great as the system would achieve if it were trained on statistical data. Additionally the radiographer would have to be explicitly asked how likely they think a given combination of attribute and

organ type are and I think that their ability to respond to this would be less representative then if they were allowed to simply list rules of their own, as can be done with both expert systems and fuzzy inference.

## D.3 Fuzzy Inference Systems

A fuzzy inference system (or fuzzy expert system) is a computing framework which is able to use a set of entailment rules to make a judgement from a set of inputs. The important aspect of these systems is that the entailment rules can be based on fuzzy concepts without a full deterministic definition. The notion of fuzziness in these systems follows from Zadeh's fuzzy set theory [136], where rather than considering set membership as either true or false, it is considered in varying degrees. That is to say that an object can belong to one or more set in different 'amounts'.

### Mamdani Fuzzy Inference Systems

Mamdani was one of the first to propose a control system based on fuzzy set theory [67] and his method is the most common in use. He aimed to create a control system for a steam engine which was based on human defined heuristic rules rather than a mathematical understanding of the full system. Mamdani states that human defined rules can be considered as "loose linguistic expression[s]" rather than as a fixed control strategy, and therefore he employed the concepts of fuzzy theory into rules so that they could be implemented electronically while incorporating the generality with which they were created. Mamdani's process is as follows:

### Quantisation

Quantisation (now commonly called *fuzzification* [47]) is the method used to convert from continuous input values into membership values for discrete fuzzy sets. For each input variable a quantisation function must be defined that converts a continuous value into a membership value for each fuzzy set which the variable can take. For example, if a random variable was *size*, then the fuzzy sets could be: *large*; *medium* and *small*. The quantisation function for this example would take a size, asses it against the expected range, and then allocate it a membership value to *large*, *medium* and *small*. A possible mapping for this example is shown in figure D.1.

### Rule Implementation

The fuzzy rules should be defined by consultation with a human expert before operation. The rules specified can be of the form "IF $X$ THEN $Y$", where the antecedent, $X$, is one, or a combination of, fuzzy-set allocations for input variables and the consequent, $Y$, is a fuzzy-set allocation for an output variable. Input variable set allocations can be

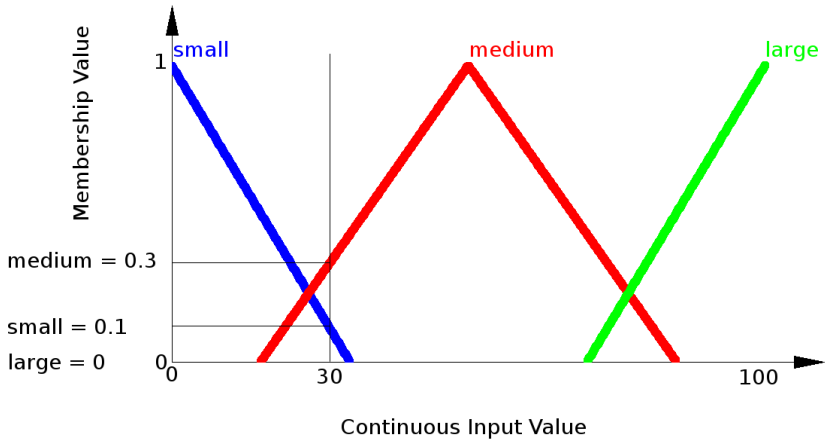


Figure D.1: An example quantisation mapping for the variable "*size*", which takes an input in the range 0 to 100 and allocates it to three possible fuzzy sets. Shown is the conversion for a size of 30 which is *large* with a membership value 0, *medium* with a membership value 0.3 and *small* with a membership value 0.1.

combined or modified in three ways: *union* (i.e. size is large OR surface area is small); *intersection* (i.e. size is large AND surface area is small) and *negation* (i.e. size is NOT large).

The rules are then evaluated by setting the consequent's membership value to the same value as the antecedent's. If a combined antecedent is used then membership values are combined using the following criterion:

- $A$ OR $B = \max(\mu(A), \mu(B))$
- $A$ AND $B = \min(\mu(A), \mu(B))$
- NOT $A = 1 - \mu(A)$

where $A$, $B$ are fuzzy-set allocations for random variables (i.e. size is large)
$\mu(C)$ is fuzzy membership value associated to $C$.

Takigi, Sugeno and Kang [118] proposed a variation on fuzzy inference which was aimed to allow the generation of rules in a more systematic way. In their method the consequent in each fuzzy rule is some function of the input variables which are involved in the antecedent. This method means that the consequents are non-fuzzy and the process of computing the output value is simplified. Another variation on fuzzy inference is the Tsukamoto fuzzy model [47] for which the consequent has a monotonical membership function allowing for a non-fuzzy evaluation of each.

### Output Computation

If a single rule is used then the output is immediately available, but the uses of such a system would be limited. It is more common that multiple rules are specified and a

stage is therefore required to consolidate their results and extract a single output. It is possible that different rules would allocate positive membership values to more than one fuzzy-set for the output variable and therefore some method is required to select which single output to give. To compute the output (now commonly called *defuzzification* [47]) Mamdani initially proposed selecting the one with the largest membership value.

This area has been widely developed since its first proposal and there are many more methods available based around calculating the centre of gravity of all output set memberships. The reader is referred to [47] for more detail.

### Application of Fuzzy Inference to Tissue Classification

Dellepiane, Venturi and Varnazza [28] present and prove a methodology for tissue classification in MR images using fuzzy inference. They use rules relating to many segment properties such as: *size*; *mean grey-value*; *elongation*; and *centrality*. Their method for rule generation actually learns from training data rather than the typical human expert approach, but their method proves rules can be applied, and therefore I believe that their work is relevant to this application. Scheipers et al. [116] implement a similar process for Sugeno inference on ultrasound images where they are able to accurately identify and label tumours.

### Fuzzy Inference Conclusions

The key advantage of fuzzy inference systems is their ability to utilise human expert rules in a way which adapts to the inherent impreciseness. Unlike many of the other classification methods, the rules which a human radiographer would state could be input almost directly into a fuzzy inference system in a recognisable form. As the rules are applied in a fuzzy way they are very general and robust towards variation and this makes them very suited to the classification of tissue structures with a degree of inter-patient variability.

Another advantage is that these methods can be used to output a 'belief' value with a proposed classification. That is to say that the system always selects a class, but if the class is only be best of a bad set then the system identifies this with a low membership value allowing the program to take this into account.

A limitation of this approach to classification is that results produced are highly dependent on the selection of input fuzzification functions. There are many different ways in which these functions can be defined [47] and each one can be suited to different applications. They can be based on human expert knowledge, but they are not as human-understandable as "IF... THEN..." rules and I consider them as the weak link within the process.

Additionally, the defuzzification process used in the modern Mamdani model, where the centre of gravity must be calculated for a potentially complex two dimensional shape, is computationally complex and can slow the evaluation process down.

# Bibliography


[1] R. Adams and L. Bischof. Seeded Region Growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–648, 1994.

[2] A. Akselrod-Ballin et al. Atlas Guided Identification of Brain Structures by Combining 3D Segmentation and SVM Classification. *Medical Image Computing and Computer-Assisted Intervention*, 9(2):209–216, 2006.

[3] N. C. Andreasen et al. Automatic Atlas-Based Volume Estimation of Human Brain Regions from MR Images. *Journal of Computer Assisted Tomography*, 20(1):98–106, 1996.

[4] U. Anliker et al. AMON: A Wearable Multiparameter Medical Monitoring and Alert System. *IEEE Transactions on Information Technology in Biomedicine*, 8(4), 2004.

[5] R. Archibald et al. Improving Tissue Segmentation of Human Brain MRI Through Preprocessing by the Gegenbauer Reconstruction Method. *NeuroImage*, 20(1):489–502, 2003.

[6] M. Awad and N. Bourbakis. Automated MRI Driven Image Guided Robotic Brain Surgery. In *International Conference on Information Intelligence and Systems*, pages 126–132, 1999.

[7] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall Inc., first edition, 1982.

[8] S. Battiato, G. Di Blasi, G. M. Farinella, G. Gallo, and G. C. Guarnera. Adhoc Segmentation Pipeline for Microarray Image Analysis. In *In IS and T-SPIE Electronic Imaging*, 2006.

[9] S. Z. Beevi and M. M. Sathik. A Robust Segmentation Approach for Noisy Medical Images Using Fuzzy Clustering with Spatial Probability. *European Journal of Scientific Research*, 41(3):437–451, 2010.

[10] E. T. Bell. Exponential Numbers. *The American Mathematical Monthly*, 41(7):411–419, 1934.

[11] S. Beucher. Watershed, Heirachical Segmentation and Waterfall Algorithm. In *Proceedings of ISMM 1994*, pages 69–76, 1994.

[12] S. Beucher and C. Lanteujoul. Use of Watersheds in Contour Detection. International Workshop on Image Processing: Real-Time Edge and Motion Detection / Estimation, 1979.

[13] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press New York, first edition, 1981.

[14] J. C. Bezdek, R. Ehrlich, and W. Full. FCM: The Fuzzy c-Means Clustering Algorithm. *Computers and Geosciences*, 10(2-3):191–203, 1984.

[15] R. Bourne. *Fundamentals of Digital Imaging in Medicine*. Springer-Verlag, first edition, 2010.

[16] M. A. Brown and R. C. Semelka. *MRI Basic Principles and Applications*. Wiley-Liss, third edition, 2003.

[17] A. Buades, B. Coll, and J. M. Morel. A Review of Image Denoising Algorithms, With a New One. *Multiscale Modeling and Simulation*, 4(2):490–530, 2005.

[18] Y. L. Chang and X. Li. Adaptive Image Region-Growing. *IEEE Transactions on Image Processing*, 3(6):868–872, 1994.

[19] F. Cheevasuvit, H. Maitre, and D. Vidal-Madjar. A Robust Method for Picture Segmentation Based on a Split and Merge Procedure. *Computer Vision, Graphics and Image Processing*, 34:268–281, 1986.

[20] S. Y. Chen, W. C. Lin, and C. T. Chen. Split and Merge Image Segmentation Based on Localized Feature Analysis and Statistical Tests. *CVGIP: Graphical Models and Image Processing*, 53(5):457–475, 1991.

[21] Yi-Yu Chou et al. Automated 3D Mapping Shape Analysis of the Lateral Ventricles via Fluid Registration of Multiple Surface-Based Atlases. In *4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1288–1291, 2007.

[22] P. Cingolani et al. jfuzzylogic. Online Resource, 2010. http://jfuzzylogic.sourceforge.net.

[23] L. P. Clarke et al. MRI Segementation: Methods and Applications. *Magnetic Resonance Imaging*, 13(3):343–368, 1995.

[24] G. B. Coleman and H. C. Andrews. Image Segmentation by Clustering. *Proceedings of the IEEE*, 67(5):773–785, 1979.

[25] P. Danielsson and O. Seger. *Generalized and Separable Sobel Operators in Machine Vision for Three Dimensional Scenes*. Academic Press Inc., first edition, 1990.

[26] L. S. Davis. A Survey of Edge Detection Techniques. *Computer Graphics and Image Processing*, 4(3):248–270, 1975.

[27] B. M. Dawant, A. P. Zidjdenbos, and R. A. Margolin. Correction of Intensity Variations in MR Images for Computer-Aided Tissue Classification. *IEEE Transactions on Medical Imaging*, 12(4):770–781, 1993.

[28] S Dellepiane, G Venturi, and G Vernazza. A Fuzzy Model for the Processing and Recognition of MR Pathological Images. In *Information Processing in Medical Imaging*, volume 511 of *Lecture Notes in Computer Science*, pages 444–457. Springer Berlin / Heidelberg, 1991.

[29] D. L. Donoholo and I. M. Johnstone. Ideal Spatial Adaptation by Wavelet Shrinkage. *Biometrika*, 81(3):425–455, 1994.

[30] W. Doyle. Operations Useful for Similarity-Invariant Pattern Recognition. *Journal of the ACM (JACM)*, 9(2):259–267, 1962.

[31] J. C. Dunn. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Cybernetics and Systems*, 3(3):32–57, 1973.

[32] D. Evans. DCM4CHE: Open Source Clinical Image and Object Management Toolkit. Online Resource, 2010. http://www.dcm4che.org/.

[33] K. S. Fu and J. K. Mui. A Survey on Image Segmentation. *Pattern Recognition*, 13:3–16, 1981.

[34] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis Maching Intelligence*, 6:721–741, 1984.

[35] J. Giarratano and G. Riley. *Expert Systems: Principles and Programming*. PWS-Kent Publishing, first edition, 1989.

[36] D. T. Glennon et al. MR Image Non-Uniformity Correction Techniques. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 3, pages 89–90, 1991.

[37] S. M. Golodetz. *Zipping and Unzipping: The Use of Image Partition Forests in the Analysis of Abdiminal CT Scans*. PhD thesis, Oxford University Computing Laboratory, DRAFT.

[38] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Pearson Education Inc., international edition, 2008.

[39] G. L. Goualher et al. Automatic Identification of Cortical Sulci Using a 3D Probabilistic Atlas. *Medical Image Computing and Computer-Assisted Interventation*, 1496:509–518, 1998.

[40] E. L. Hall. *Computer Image Processing and Recognition*. Academic Press, Inc, 1979.

[41] J. M. Hammersley and P. Clifford. Markov Fields on Finite Graphs and Lattices. Unpublished, 1971.

[42] J. Haselgrove and M. Prammer. An Algorithm for Compensation of Surface-Coil Images for Sensitivity of the Surface Coil. *Magnetic Resonance Imaging*, 4(6):469–472, 1986.

[43] C. L. Hoad and A. L. Martel. Segmentation of MR Images for Computer-Assisted Surgery of the Lumbar Spine. *Physics in Medicine and Biology*, 47:3503–3517, 2002.

[44] S. L. Horowitz and T. Pavlidis. Picture Segmentation by a Tree Traversal Algorithm. *Journal of the ACM (JACM)*, 23(2):368–388, 1976.

[45] X. Hu et al. Bayesian Image Processing in Magnetic Resonance Imaging. *Magnetic Resonance Imaging*, 9:611–620, 1991.

[46] International Electrotechnical Commision. *Programmable Controllers Part 7: Fuzzy Control Programming (IEC 1131-7)*, 1997.

[47] J. S. R. Jang, C. T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice-Hall, 1997.

[48] V. E. Johnson et al. Image Restoration Using Gibbs Priors: Boundary Modeling, Treatment of Blurring, and Selection of Hyperparameter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):413–425, 1991.

[49] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, pages 321–331, 1988.

[50] R. Kindermann and J. L. Snell. *Markov Random Fields and their Applications*. American Mathematical Society, first edition, 1980.

[51] R. L. Kirby and A. Rosenfeld. A Note on the use of (Gray Level, Local Average Gray Level) Space as an Aid in Threshold Selection. *IEEE Transactions on Systems, Man and Cybernetics*, 9(12):860–864, 1979.

[52] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science Magazine*, 220(4598):671–680, 1983.

[53] Kitware Inc. Insight Toolkit (ITK). Online Resource, 2010. http://www.itk.org/.

[54] K. Krissian and S. Aja-Fernanez. Noise-Driven Anisotropic Diffusion Filtering of MRI. *IEEE Transactions on Image Processing*, 18(10):2265–2274, 2009.

[55] S. Lakare. 3D Segmentation Techniques for Medical Volumes. *State University of New York at Stony Brook: Research Proficiency Exam*, 2000.

[56] L. Landini, V. Positano, and M. F. Santarelli. *Advanced Image Processing in Magnetic Resonance Imaging*. Taylor and Francis Group, 2005.

[57] C. Lee et al. Unsupervised Connectivity-Based Thresholding Segmentation of Midsagittal Brain MR Images. *Computers in Biology and Medicine*, 28:309–338, 1998.

[58] U. M. Leloglu and U. Halici. Multi-Image Region Growing for Integrating Disparity Maps. In *Computer Analysus of Images and Patterns*, pages 403–410. Springer Berlin / Heidelberg, 1999.

[59] C. Li et al. Knowledge-Based Classification and Tissue Labeling of MR Images of Human Brain. *IEEE Transactions on Medical Imaging*, 12(4):740–750, 1993.

[60] S. Z. Li. *Markov Random Field Models in Computer Vision*, pages 361–370. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, first edition, 1994.

[61] Z. Liang. Tissue Classification and Segmentation of MRI Images. *IEEE Engineering in Medicine and Biology*, pages 81–85, March 1993.

[62] K. O. Lim and A. Pfefferbaum. Segmentation of MR Brain Images into Cerebrospinal Fluid Spaces, White and Gray Matter. *Journal of Computer Assisted Tomography*, 13(4):588–593, 1989.

[63] H. K. Liu. Two- and Three-Dimensional Boundary Detection. *Computer Graphics and Image Processing*, 6:123–134, 1977.

[64] P. Lucas and L. Van Der Gaag. *Principles of Expert Systems*. Addison-Wesley Publishers Ltd., first edition, 1991.

[65] A. Macovski. Noise in MRI. *Magnetic Resonance in Medicine*, 36(3):494–497, 1996.

[66] J. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. *Proceeding of the Fifth Berkeley Symposium on Mathematics, Statistics and Probability*, 1:281–297, 1967.

[67] E. H. Mamdani and S. Assilian. An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *International Journal of Man-Machine Studies*, 7:1–13, 1975.

[68] J. F. Mangin et al. *Information Processing in Medical Imaging: 14th International Conference*, chapter Deformable Topological Models for Segmentation of 3D Medical Images, pages 153–164. Kluwer Academic Publishers, 1995.

[69] J. V. Manjon et al. MRI Denoising Using Non-Local Means. *Medical Image Analysis*, 12:514–523, 2008.

[70] I. N. Manousakas et al. Split and Merge Segmentation of Magnetic Resonance Medical Images: Performance Evaluation and Extension to Three Dimensions. *Computers and Biomedical Research*, 31:393–412, 1998.

[71] H. Marin-Castro, E. Sucar, and E. Morales. Automatic Image Annotation using a Semi-Supervised Ensemble of Classifiers. In *Progress in Pattern Recognition, Image Analysis and Applications*, volume 4756 of *Lecture Notes in Computer Science*, pages 487–495. Springer Berlin / Heidelberg, 2007.

[72] D. Marr and E. Hildreth. Theory of Edge Detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 207(1167):187–217, 1980.

[73] A. Martelli. An Application of Heuristic Search Methods to Edge and Contour Detection. *Comunication of the ACM*, 19(2):73–83, 1976.

[74] D. Mason et al. Measurement of c-bands in Human Chromosomes. *Computers in Biology and Medicine*, 5(3):179–180, IN9–IN10, 181–201, 1975.

[75] T. McInerney and D. Terzopoulos. Deformable Models in Medical Image Analysis: A Survey. *Medical Image Analysis*, 1(2):91–108, 1996.

[76] D. W. McRobbie et al. *MRI from Picture to Proton*. Cambridge University Press, second edition, 2007.

[77] E. R. McVeigh, R. M. Henkelman, and M. J. Bronskill. Noise and Filtration in Magnetic Resonance Imaging. *Med Phys*, 12(5):586–591, 1985.

[78] F. Meyer and S. Beucher. Morphological Segmentation. *Journal of Visual Communication and Image Segmentation*, 1(1):21–46, 1990.

[79] D. G. Mitchell. *MRI Principles*. Saunders, first edition, 1999.

[80] M. M. Mokji and S. A. R. Abu-Bakar. Adaptive Thresholding Based on Co-occurence Matrix Edge Information. *Journal of Computers*, 2(8):44–52, 2007.

[81] K. L. Moore and A. M. R. Agur. *Essential Clinical Anatomy*. Williams and Wilkins, third edition, 2007.

[82] National Electrical Manufacturers Association. *Digital Imaging and Communications in Medicine (DICOM) Part 10: Media Storage and File Format for Media Interchange*, 2009 edition, 2009.

[83] National Electrical Manufacturers Association. *Digital Imaging and Communications in Medicine (DICOM) Part 3: Information Object Definitions*, 2009 edition, 2009.

[84] National Electrical Manufacturers Association. *Digital Imaging and Communications in Medicine (DICOM) Part 5: Data Structures and Encoding*, 2009 edition, 2009.

[85] National Electrical Manufacturers Association. *Digital Imaging and Communications in Medicine (DICOM) Part 6: Data Dictionary*, 2009 edition, 2009.

[86] M. S. Nixon and A. S. Aguado. *Feature Extraction and Image Processing*. Academic Press Inc., second edition, 2008.

[87] R. D. Nowak. Wavelet-Based Rician Noise Removal for Magnetic Resonance Imaging. *IEEE Transactions on Image Processing*, 8(10):1408–1417, 1999.

[88] R. Ohlander, K. Price, and R. Reddy. Picture Segmentation using a Recursive Region Splitting Method. *Computer Graphics and Image Processing*, 8:313–333, 1978.

[89] N Otsu. A Threshold Selection Method from Grey-Level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-9(1):62–66, 1979.

[90] M. Ozkan, H. G. Sprenkels, and B. M. Dawant. Multispectral Magnetic Resonance Image Segmentation Using Neural Networks. In *IJCNN International Joint Conference on Neural Networks*, volume 1, pages 429–434, 1990.

[91] N. R. Pal and S. K. Pal. A Review on Image Segmentation Techniques. *Pattern Recognition*, 26(9):1277–1294, 1993.

[92] T. N. Pappas. An Adaptive Clustering Algorithm for Image Segmentation. *IEEE Transactions on Signal Processing*, 40(4):901–914, 1992.

[93] H. Park. *Adaptive Registration and Atlas Based Segmentation*. PhD thesis, The University of Michigan, 2003.

[94] H. Park, P. H. Bland, and C. R. Meyer. Construction of an Abdominal Probabilistic Atlas and its Application in Segmentation. *IEEE Transaction on Medical Imaging*, 22(4):483–492, 2003.

[95] N. Passat et al. Region-Growing Segmentation of Brain Vessels: An Atlas-Based Automatic Approach. *Journal of Magnetic Resonance Imaging*, 21:715–725, 2005.

[96] T. Pavlidis. *Structural Pattern Recognition*. Electrophysics 1. Springer-Verlag, first edition, 1977.

[97] T. Pavlidis. *Algorithms for Graphics and Image Processing*. Springer-Verlag, first edition, 1982.

[98] T. Pavlidis and Y. T. Liow. Integrating Region Growing and Edge Detection. *IEEE Transactions on Pattern Analysis Maching Intelligence*, 12(3):225–233, 1990.

[99] P. Perona and J. Malik. Scale-Space and Edge Detection Using Anisotropic Diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[100] D. L. Pham and J. L. Prince. An Adaptive Fuzzy C-Means Algorithm for Image Segmentation in the Presence Intensity Homogeneities. *Pattern Recognition Letters*, pages 57–68, 1999.

[101] D. L. Pham, C. Xu, and J. L. Prince. A Survey of Current Methods in Medical Image Segmentation. *Annual Review of Biomedical Engineering*, January 1998.

[102] O. S. Pianykh. *Digital Imaging and Communications in Medicine (DICOM) A Practical Introduction and Survival Guide*. Springer-Verlag, first edition, 2008.

[103] K. M. Pohl et al. Anatomical Guided Segmentation with Non-Stationary Tissue Class Distributions in an Expectation-Maximization Framework. *IEEE International Symposium on Biomedical Imaging*, pages 81–84, 2004.

[104] J. M. S. Prewitt. *Object Enhancement and Extraction in Picture Processing and Psychopictorics*. Academic Press Inc., first edition, 1970.

[105] J. M. S. Prewitt and M. L. Mendelsohn. The Analysis of Cell Images. *Annals of the New York Academy of Sciences*, 128(3):1035–1053, 1966.

[106] J. C. Rajapakse, J. N. Giedd, and J. L. Rapoport. Statistical Approach to Segmentation of Single-Channel Cerebral MR Images. *IEEE Transactions on Medical Imaging*, 16(2):176–186, 1997.

[107] W. E. Reddick et al. Automated Segmentation and Classification of Multispectral Magnetic Resonance Images of Brain Using Artificial Neural Networks. *IEEE Transactions on Medical Imaging*, 16(6):911–918, 1997.

[108] L. G. Roberts. *Machine Perception of Three-Dimensional Solids*. PhD thesis, Massachisetts Institute of Technology, 1963.

[109] A. Rosenfeld. Measuring the Sizes of Concavities. *Pattern Recognition Letters*, 3:71–75, 1985.

[110] A. Rosenfeld and R. C. Smith. Thresholding Using Relaxation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 3(5):598–606, 1981.

[111] D. Rueckert et al. Nonrigid Registration Using Free-Form Deformation: Application to Breast MR Images. *IEEE Transactions on Medical Imaging*, 18(8):712–721, 1999.

[112] S. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, international edition, 2003.

[113] P. K. Sahoo et al. A Survey of Thresholding Techniques. *Computer Vision, Graphics and Image Processing*, 41:233–260, 1988.

[114] K. Saito and R. Nakano. Medical Diagnostic Expert System Based on PDP Model. In *EEE International Conference on Neural Networks*, volume 1, pages 255–262, 1988.

[115] K. S. Saladin. *Anatomy and Physiology: The Unity of Form and Function*. McGraw-Hill, third edition, 2004.

[116] U. Scheipers et al. Neuro-Fuzzy Inference System for Ultrasonic Multifeature Tissue Characterization for Prostate Diagnostics. In *IEEE Ultrasonics Symposium*, volume 2, pages 1379–1382, 2002.

[117] D. W. Shattuck et al. Magnetic Resonance Image Tissue Classification Using a Partial Volume Model. *NeuroImage*, 13:856–876, 2001.

[118] M. Sugeno and G. T. Kang. Structure Identification of Fuzzy Model. *Fuzzy Sets and Systems*, 28(1):13–33, 1988.

[119] Sun Microsystems. Java Advanced Imaging: JAI. Online Resource, 2010. http://java.sun.com/javase/technologies/desktop/media/jai/.

[120] Sun Microsystems. Java Swing. Electronic Software, 2010. http://java.sun.com/products/jfc/tsc/articles/architecture/.

[121] S. Suzuki et al. Tele-Surgery Simulation with a Patient Organ Model for Robotic Surgery Training. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 1(4):80–88, 2006.

[122] R. Szeliski. Bayesian Modeling of Uncertainty in Low-Level Vision. *International Journal of Computer Vision*, 5(3):271–301, 1990.

[123] H. Tang, E. X. Wu, Q. Y. Ma, D. Gallagher, G. M. Perera, and T. Zhuang. MRI Brain Image Segmentation by Multi-Resolution Edge Detection and Region Selection. *Computerized Medical Imaging and Graphics*, 24(6):349–357, 2000.

[124] D. Terzopoulos and K. Fleischer. Deformable Models. *The Visual Computer*, 4(6):306–331, 1988.

[125] P. M. Thompson et al. Detection and Mapping of Abnormal Brain Structure with a Probabilistic Atlas of Cortical Surfaces. *Journal of Computer Assisted Tomography*, 21(4):567–581, 1997.

[126] N. Wadley et al. Pre-Operative Planning and Intra-Operative Guidance in Modern Neurosurgery: A Review of 300 Cases. *Annals of The Royal College of Surgeons of England*, 81(4):217–225, 1999.

[127] J. S. Weszka. A Survey of Threshold Selection Techniquesss. *Computer Graphics and Image Processing*, 7:259–265, 1978.

[128] J. S. Weszka and A. Rosenfeld. Histogram Modification for Threshold Selection. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):38–52, 1979.

[129] D. H. Wolpert and W. G. Macready. No Free Lunch Theorems for Search. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[130] D.H. Wolpert. The Lack of a priori Distinctions Between Learning Algorithms. *Neural Computation*, 8(7):1341–1390, 1996.

[131] D.H. Wolpert and W.G. Macready. No Free Lunch Theorems for Optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

[132] M. Wu et al. Quantitative Comparison of Air, SPM, and the Fully Deformable Model for Atlas-Based Segmentation of Functional and Structural MR Images. *Human Brain Mapping*, 27:747–754, 2006.

[133] P. P. Wyatt and J. A. Noble. MAP MRF Joint Segmentation and Registration. In *MICCAI*, pages 580–587. Springer-Verlag, 2002.

[134] Y. Xin et al. Texture Classification of Aerial Image Based on Bayesian Network Augmented Naive Bayes. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 37, 2008.

[135] L. P. Yaroslavsky and M. Eden. *Fundamentals of Digital Optics*. Springer-Verlag, first edition, 1996.

[136] L. A. Zadeh. Fuzzy Sets. *Information and Control*, 8(3):338–353, 1965.

[137] S. W. Zucker. Region Growing: Childhood and Adolescence. *Computer Graphics and Image Processing*, 5:382–399, 1976.

[138] S. W. Zucker and R. A. Hummel. A Three-Dimensional Edge Operator. *IEEE Transactions on Pattern Analysis Maching Intelligence*, 3(3):324–331, 1981.