

2008

A framework for tumor segmentation and interactive immersive visualization of medical image data for surgical planning

Jung Leng Foo
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Mechanical Engineering Commons](#)

Recommended Citation

Foo, Jung Leng, "A framework for tumor segmentation and interactive immersive visualization of medical image data for surgical planning" (2008). *Retrospective Theses and Dissertations*. 15785.
<https://lib.dr.iastate.edu/rtd/15785>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

**A framework for tumor segmentation and interactive immersive
visualization of medical image data for surgical planning**

by

Jung Leng Foo

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Co-majors: Mechanical Engineering; Human Computer Interaction

Program of Study Committee:

Eliot Winer, Major Professor

Adin Mann

James Oliver

Judy Vance

Julie Dickerson

Iowa State University

Ames, Iowa

2008

Copyright © Jung Leng Foo, 2008. All rights reserved.

UMI Number: 3383371

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

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

UMI[®]

UMI Microform 3383371
Copyright 2009 by ProQuest LLC
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

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

TABLE OF CONTENTS

List of Figures	v
List of Tables	ix
Abstract	x
1 Introduction	1
1.1 Medical Imaging	1
1.1.1 The Digital Imaging and Communications in Medicine (DICOM) Format	4
1.1.2 Windowing	6
1.2 Motivation	8
1.3 Dissertation Organization	12
2 Literature Review	13
2.1 Image Segmentation	13
2.1.1 Classical Approaches	13
2.1.2 Statistical and Probability Based Approaches	14
2.1.3 Deformable Models	16
2.1.4 Atlas Guided Approaches	17
2.1.5 Learning Algorithms in Image Segmentation	17
2.1.6 Fuzzy Segmentation Algorithms	18
2.1.7 Mixed/Hybrid Approaches	21
2.2 Medical Image Data Visualization	22
2.3 Virtual Reality in Medicine	25
2.3.1 Virtual Colonoscopy	25
2.3.2 Virtual Endoscopy	27
2.3.3 Laparoscopic Simulation/Training	29
2.4 User Interface and Interaction Design	33
2.4.1 Menu Interfaces	34
2.4.2 Designing a User Interface	35
2.5 Research Issues	39
3 Probabilistic Segmentation of Tumors	41
3.1 Introduction	41
3.2 Image Pre-processing	42
3.3 Automated Seed and Search Region Selection	42
3.4 Segmentation Procedure	45
4 Fuzzy Segmentation of Tumors	51

4.1 Fuzzy Logic	51
4.1.1 Fuzzy Inference System Example	52
4.2 Introduction	54
4.3 Pre-processing	55
4.4 Initialization	56
4.5 Inputs and Output of the Fuzzy System	57
4.6 Fuzzification	59
4.7 Fuzzy Rules	62
4.8 Defuzzification	63
4.9 Tumor Segmentation	64
4.10 Post-processing	65
4.11 Multi-Window Segmentation	65
5 ISIS: Interactive Immersive Visualization of Medical Patient Data	67
5.1 Introduction	67
5.2 Gamepad and Menu Interface	69
5.3 Default Mode	74
5.3.1 Information Menu	75
5.4 Windowing and Pseudo-coloring	77
5.4.1 Windowing Menu	79
5.4.2 Gamepad Controls and Shortcuts for Windowing Feature	81
5.5 Multiple Clipping Planes	81
5.5.1 Clipping Menu	83
5.5.2 Gamepad Controls and Shortcuts for Clipping Feature	84
5.6 Surgery Planning	85
5.6.1 Surgery Planning Menu	87
5.6.2 Tumor Inspection	90
6 Results and Discussion	92
6.1 Information and Properties of Test Datasets	92
6.2 Hardware and Software	94
6.3 Probabilistic Segmentation of Tumors	94
6.3.1 Segmentation Results	95
6.3.2 Segmentation Accuracy	100
6.3.3 Inter- and Intra-User Variability	103
6.4 Fuzzy Segmentation of Tumors	106
6.4.1 Multi-Window Fuzzy Segmentation Results	106
6.4.2 Segmentation Results	109
6.4.3 Segmentation Accuracy	113
6.4.4 Inter- and Intra-User Variability	116
6.5 Comparison of Both Segmentation Methods	117
6.6 Surgery Planning in Isis	119
6.6.1 Tumor Inspection	119
6.6.2 Simulated Endoscope Camera View	120
6.7 Performance Evaluation of Isis	122

7	Conclusions and Future Work	125
7.1	Summary and Conclusions	125
7.2	Future Work	128
7.3	Acknowledgements	129
	References	131

LIST OF FIGURES

Figure 1:	The first X-ray of the human body by William Rontgen in 1895 (http://inventors.about.com/library/inventors/blxray.htm)	1
Figure 2:	Illustration of a CT scanning process, and generation of a three-dimensional representation from two-dimensional medical image data.	2
Figure 3:	Example images of CT (left) and MR (right) scans.	3
Figure 4:	An illustration of a DICOM file structure, containing multiple data elements, each representing a particular value and type.	4
Figure 5:	Illustration of the windowing process, converting Hounsfield Units (HU) to 8-bit pixel intensities.	7
Figure 6:	Examples of windowing effects, using different window center and width values on the same image to emphasize tissue different types and densities.	8
Figure 7:	Examples of the three different levels of tumor complexity for automatic segmentation: a homogeneous tumor (left), a tumor with some inhomogeneity and fuzzy edges (center), and a highly heterogeneous tumor with calcifications and mixed tissue density (right). Tumors are highlighted with red dashed lines.	10
Figure 8:	Examples of multi-slice films (left), a conventional light box viewing station (center), and a digital “light box” on LCD monitors (right).	22
Figure 9:	Illustration of how multiple consecutive 2D slices are combined to generate a 3D volume representation of the patient data.	23
Figure 10:	Conventional colonoscopy and virtual colonoscopy. (http://www.rsna.org/rsna/media/pr2005/Summers-Colonoscopy/summers.cfm)	26
Figure 11:	Example of manual navigation for a virtual endoscopy.	28
Figure 12:	Illustration of a laparoscopic procedure. (Image from Laparoscopic Surgery by David Anaise, MD: http://www.danaise.com/laparoscopic_surgery_7-5.htm)	30
Figure 13:	A typical laparoscopic procedure using a two-dimensional video display. (Photo from VCU Medical Center, Minimally Invasive Surgery Center: http://www.vcu.edu/lesspainsurgery/fellowship_year_long.htm)	31
Figure 14:	Various minimally invasive surgery simulators/trainers.	32

Figure 15:	Examples of different frequently used menu interfaces. (Images from http://people.cs.vt.edu/~bowman/pubs.html)	34
Figure 16:	Screenshot of Kitware's Volview.	36
Figure 17:	Screenshot of open source Osirix radiological software application.	36
Figure 18:	Flowchart of segmentation process.	41
Figure 19:	Flowchart for automatic seed relocation and search region selection procedure.	43
Figure 20:	Values for Pr_l based on possible spatial and pixel intensity component values.	49
Figure 21:	Fuzzy inference system for tip calculations based on service and food.	53
Figure 22:	Flowchart of segmentation process using the fuzzy rule-based system.	55
Figure 23:	Illustration of the region of interest (ROI) selected by the user in the first slice of the image data set.	56
Figure 24:	Fuzzy system with three inputs and a single output.	57
Figure 25:	Example of a S-shaped membership function for a particular input.	60
Figure 26:	Software framework for the interactive immersive visualization system.	67
Figure 27:	Outline of the features of the Isis immersive visualization system.	68
Figure 28:	Buttons and joystick layout of wireless gamepad.	70
Figure 29:	Application in a four-walled VR projection system with wireless gamepad as interaction device (inset).	71
Figure 30:	Layout of menu interface.	73
Figure 31:	Displaying selected information about the patient and the image dataset.	76
Figure 32:	Windowing and pseudo-coloring features, settings of these features (circled in red) are viewable to the user within the system.	77
Figure 33:	Windowing for display of specific tissue densities and pseudo-coloring using Color Look-up Tables (CLUT)	78
Figure 34:	Windowing menu, editing current windowing parameters for manual input.	79
Figure 35:	Predefined bone window available as an option in menu.	80
Figure 36:	Interactive multiple clipping planes to slice into the 3D volume at any angle.	82
Figure 37:	Toggling a specific clipping plane on/off via the option menu.	83
Figure 38:	Resetting a specific clipping plane from within the menu.	84
Figure 39:	Trocar placement simulator for endo-surgery planning (trocars highlighted in circles).	86

Figure 40:	Toggling specific trocars on/off from within the option menu. T1 and T3 represent the first and third trocars that are turned on and currently visible. T1, which is in blue, is the trocar selected from the menu.	87
Figure 41:	Resetting a specific trocar from within the menu. T1, T2, and T3 represent the first, second, and third trocars that are currently turned on and visible. T2, which is in blue, is the selected trocar from the menu.	88
Figure 42:	Examples of a) an endoscope, b) the camera at the end tip of the endoscope, and c) a screenshot of the video output from an endoscope inserted into the digestive system.	89
Figure 43:	Viewing of automatically segmented tumors in context of patient data.	90
Figure 44:	Segmentation results for Test Case #1.	96
Figure 45:	Three-dimensional volume renders of the segmented tumor from Test Case #1, from manual segmentation and probabilistic segmentation.	97
Figure 46:	Probabilistic segmentation results for tumors with mixed tissue densities that are slightly heterogeneous. Actual tumors outlined by radiologist are shown in red dashed lines.	98
Figure 47:	Probabilistic segmentation results for tumors that are highly heterogeneous with calcifications. Actual tumors outlined by radiologist are shown in red dashed lines.	99
Figure 48:	Segmentation results from three different windowing settings of the same slice and the final merged segmented region.	108
Figure 49:	Selected results of the fuzzy segmentation for Test Case #1	109
Figure 50:	Three-dimensional view of the segmented tumor from Test Case #1 using manual segmentation (left) and from the fuzzy segmentation method (right).	110
Figure 51:	Fuzzy segmentation results for tumors with mixed tissue densities that are slightly heterogeneous. Actual tumors outlined by radiologist are shown in red dashed lines.	111
Figure 52:	Segmentation results for tumors that are highly heterogeneous with calcifications. Actual tumors outlined by radiologist are shown in red dashed lines.	112
Figure 53:	Performance comparison of both segmentation methods: MFN and MFP errors for probabilistic segmentation method, and MFN and MFP errors for fuzzy segmentation method.	117
Figure 54:	Comparison of both segmentation results with radiologist's manual segmentation for test cases #4 (category B) on the top row and #6 (category C) on the bottom row.	118

Figure 55:	Tumor volume and patient data viewed separately (left), and tumor model linked to a windowed patient model (right).	120
Figure 56:	Simulated endoscopic camera inside the patient model.	121
Figure 57:	Loading time of different volume sizes.	122
Figure 58:	Plot of the frame rate (in stereo) as the number of slices being loaded increases. Minimum requirement for user interaction shown as dashed red line at 14 frames per second.	123

LIST OF TABLES

Table 1:	Typical values of Hounsfield Units (HU) for common tissue types.	6
Table 2:	Fuzzy range limits of membership functions.	61
Table 3:	Fuzzy rules for the fuzzy inference system.	63
Table 4:	Mapping of application features to a specific menu tab.	72
Table 5:	Gamepad controls for the standard/default mode.	74
Table 6:	Gamepad controls for the windowing mode.	81
Table 7:	Gamepad controls for the clipping mode.	85
Table 8:	Gamepad controls for the endo-surgery planning mode.	88
Table 9:	Summary of information and parameters for the test cases presented.	93
Table 10:	Summary of the improvements made to the probabilistic segmentation method.	95
Table 11:	Accuracy evaluation of the probabilistic segmentation algorithm. Mean False Positive (MFP) when non-tumor pixels were selected and Mean False Negative (MFN) when tumor pixels were not selected for all 10 test cases. The results for “Version 1” are for the probabilistic method using the original seed relocation algorithm, while “Version 2” is with the new revised seed relocation and search region algorithm.	101
Table 12:	Statistical analysis of inter-user variability for probabilistic segmentation method.	105
Table 13:	Summary of the modifications made to the fuzzy segmentation method.	106
Table 14:	Summary of information and parameters for the test cases presented.	107
Table 15:	Accuracy evaluation of the fuzzy segmentation algorithm. Mean False Positive (MFP) when non-tumor pixels were selected and Mean False Negative (MFN) when tumor pixels were not selected for all 10 test cases. The results for “Version 1” are for the fuzzy method using a single window setting, while “Version 2” uses three window settings to generate a merged region.	114
Table 16:	Statistical analysis of inter-user variability for fuzzy segmentation method.	116

ABSTRACT

Medical imaging technology has changed the way doctors and surgeons diagnose and operate in the last 100 years. Since the discovery of the X-ray, doctors and surgeons have successfully utilized the ability to “see” inside a patient to further improve patient care. The advancement of computed tomography and magnetic resonance technology with faster and higher resolution scans has improved the analysis and visualization of patient data. As technology continues to accommodate the need for higher quality medical imaging, there is a continuing need for additional research in efficient ways of extracting crucial information from these vast amounts of data. This dissertation presents the framework for analyzing and visualizing digital medical images. Two new segmentation methods have been developed: a probability based segmentation algorithm, and a segmentation algorithm that uses a fuzzy rule based system to generate “similarity” values for segmentation. A visualization software application has also been developed to effectively view and manipulate digital medical images on a desktop computer as well as in an immersive environment.

For the probabilistic segmentation algorithm, image data are first enhanced by manually setting the appropriate window center and width, and if needed a sharpening or noise removal filter is applied. To initialize the segmentation process, a user places a seed point within the object of interest and defines a search region for segmentation. Based on the pixels’ spatial and intensity properties, a probabilistic selection criterion is used to extract pixels with a high probability of belonging to the object. To facilitate the

segmentation of multiple slices, an automatic seed selection algorithm was developed to keep the seeds in the object as its shape and/or location changes between consecutive slices.

The second segmentation method, a new segmentation method using a fuzzy rule based system to segment tumors in a three-dimensional CT data was also developed. To initialize the segmentation process, the user selects a region of interest (ROI) within the tumor in the first image of the CT study set. Using the ROI's spatial and intensity properties, fuzzy inputs are generated for use in the fuzzy rules inference system. Using a set of predefined fuzzy rules, the system generates a defuzzified output for every pixel in terms of similarity to the object. Pixels with the highest similarity values are selected as tumor. This process is automatically repeated for every subsequent slice in the CT set without further user input, as the segmented region from the previous slice is used as the ROI for the current slice. This creates a propagation of information from the previous slices, used to segment the current slice. The membership functions used during the fuzzification and defuzzification processes are adaptive to the changes in the size and pixel intensities of the current ROI. The proposed method is highly customizable to suit different needs of a user, requiring information from only a single two-dimensional image.

Segmentation results from both algorithms showed success in segmenting the tumor from seven of the ten CT datasets with less than 10% false positive errors and five test cases with less than 10% false negative errors. The consistency of the segmentation results statistics also showed a high repeatability factor, with low values of inter- and intra-user variability for both methods.

The visualization software developed is designed to load and display any DICOM/PACS compatible three-dimensional image data for visualization and interaction in an immersive virtual environment. The software uses the open-source libraries DCMTK: DICOM Toolkit for parsing of digital medical images, Coin3D and SimVoleon for scenegraph management and volume rendering, and VRJuggler for virtual reality display and interaction. A user can apply pseudo-coloring in real time with multiple interactive clipping planes to slice into the volume for an interior view. A windowing feature controls the tissue density ranges to display. A wireless gamepad controller as well as a simple and intuitive menu interface control user interactions. The software is highly scalable as it can be used on a single desktop computer to a cluster of computers for an immersive multi-projection virtual environment. By wearing a pair of stereo goggles, the surgeon is immersed within the model itself, thus providing a sense of realism as if the surgeon is “inside” the patient.

The tools developed in this framework are designed to improve patient care by fostering the widespread use of advanced visualization and computational intelligence in preoperative planning, surgical training, and diagnostic assistance. Future work includes further improvements to both segmentation methods with plans to incorporate the use of deformable models and level set techniques to include tumor shape features as part of the segmentation criteria. For the surgical planning components, additional controls and interactions with the simulated endoscopic camera and the ability to segment the colon or a selected region of the airway for a fixed-path navigation as a full virtual endoscopy tool will also be implemented.

1 INTRODUCTION

1.1 Medical Imaging

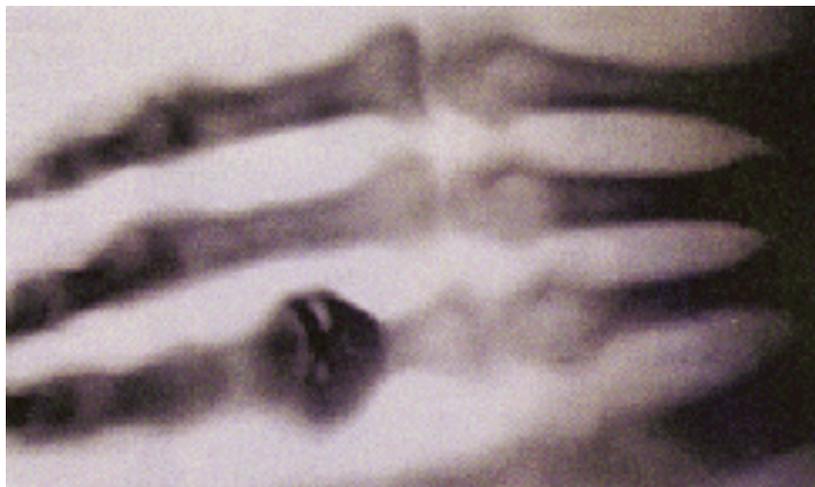


Figure 1: The first X-ray of the human body by William Rontgen in 1895
(<http://inventors.about.com/library/inventors/blxray.htm>)

In 1895, William Conrad Rontgen was experimenting with cathode ray tubes, discovering the X-ray, and created the first medical image of the human body. He took an x-ray of his wife's hand, shown in Figure 1. This event marked a significant milestone in medical examination, which paved the way for the research and development of modern medical imaging technology such as Computed Tomography (CT) scans, ultrasound imaging, and Magnetic Resonance Imaging (MRI). Despite being over a hundred years old, the X-ray remains a common and widely used form of medical imaging.

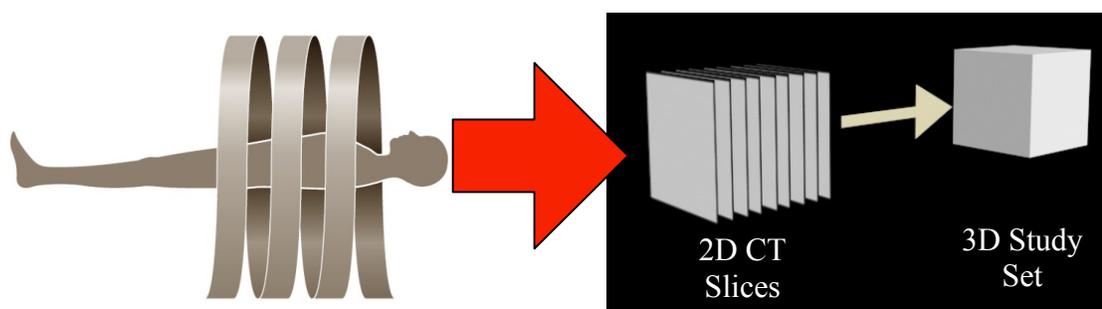
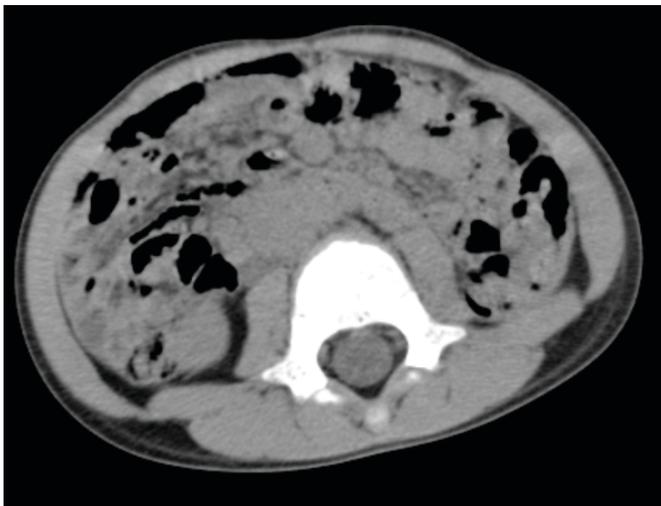


Figure 2: Illustration of a CT scanning process, and generation of a three-dimensional representation from two-dimensional medical image data.

CT scans are cross-sectional images obtained from different angles of the patient's body, using an x-ray tube [1]. As the patient slides into the rotating x-ray tube, consecutive 2D slices of image data are generated. These slices can then be joined together to form a 3D representation of the tissues and organs studied as shown in Figure 2. Sir Geoffrey Hounsfield introduced the first CT scanner in 1972, which was capable of scanning images at 80 x 80 pixels resolution and each image slice took approximately 5 minutes to complete. Comparable improvements have been made in the last few decades with the latest generation of CT scanners scanning at 1024 x 1024 pixels and completing a single slice scan at approximately 30 milliseconds. CT may be enhanced by use of contrast agents containing elements of a higher atomic number than the surrounding flesh (iodine, barium).

The first MRI scanner was introduced in 1977 by Paul Lauterbur and Peter Mansfield and unlike the CT scanner, which uses ionizing radiation (x-ray) to obtain the image data, MRI imaging is based on the emission and absorption of electro-magnetic energy in the radio frequency (RF) range of the electrostatic spectrum [2]. The scanned object will absorb and emit different variations in the phase and frequency of the RF

range. These spatial variations form the basis of producing MRI images. A bone will absorb and emit a unique set of RF phase and frequency, while tissues will have a different set. Contrast agents for MRI are those that have paramagnetic properties and an example is gadolinium. Figure 3 shows examples of the output from both CT and MR imaging scans.



CT image of abdomen



MR image of head

Figure 3: Example images of CT (left) and MR (right) scans.

CT and MRI scans both enable the physician to “see” inside a patient, non-invasively. These medical images help physicians discuss and examine a patient’s condition, and aid them in making critical medical decisions and conclusions. CT and MRI scans are widely used in medical facilities to produce digital images of a cross-section of a patient’s body tissues and internal organs. Typically, multiple 2D cross-sections are taken during an MRI or CT session, and collectively are called a study.

1.1.1 The Digital Imaging and Communications in Medicine (DICOM) Format

Most digital medical imaging data is stored in the Digital Imaging and Communications in Medicine (DICOM) format [3]. DICOM is a standard created and approved by the National Electrical Manufacturers Association (NEMA). The DICOM standard format facilitates the distribution and viewing of medical images created from MRI, CT, angiographs, and ultrasound scans. The DICOM standard format enables the storage of both medical image information and pertinent patient's information into a single file for easy exchange of medical information.

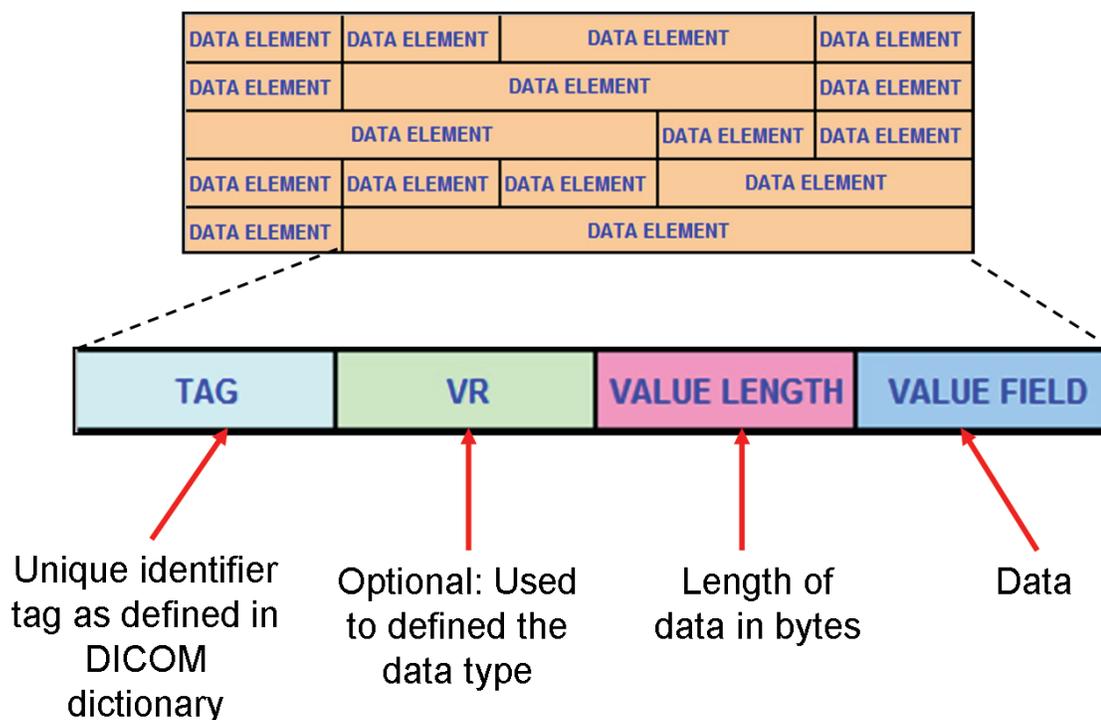


Figure 4: An illustration of a DICOM file structure, containing multiple data elements, each representing a particular value and type.

A DICOM formatted file composes of multiple DICOM data elements that can be of any type, length, and value, as shown in Figure 4. Common elements available in all DICOM files are information such as: patient's names, sex, age, scan type (CT/MRI), image resolution, and the image data. The information structure of a DICOM data element block consists of a unique identifier tag code, an optional value representation, the length of the data stored (in bytes), and finally the data itself.

When reading in each data element, the identifier tag is referenced with the DICOM 'dictionary', a list of over 1000 tags with its corresponding meaning. The identifier tags are descriptive of what the information in the data element contains. Examples are 'DCM_PatientsName' for the patient's name, 'DCM_PatientsAge' for the patient's age, 'DCM_Manufacturer' for the manufacturer of the scanner, and 'DCM_AcquisitionDate' for the date the image data was acquired. The second field is the optional value representation, which represents the data type of the data store and indicates whether the data is floating point (float or double), integer (unsigned or signed), or character. This field is optional because the data type can be inferred from the identifier tag. For example, if the data element is for a patient's name it will be a string data type and if it is storing the image resolution it will be an integer data type. The next field stores the length of the image in bytes, useful for allocating memory before reading the actual data stored in the fourth and final field. Depending on what the data element stores, the data field can be as short as just a few character bytes to as large as a 512 x 512 data array for image data.

1.1.2 Windowing

Unlike conventional computer images where the image data are stored as pixel intensities, CT image data are stored as Hounsfield Units (HU), which is a measure of tissue density relative to the density of distilled water. Standard CT scans have a range between -1000 to +1000 HU [1], although recent CT scanners now have the capability of scanning at ranges between -1000 to +3000, for a higher resolution. Each number in this range represents a shade of grey, with +1000 HU being white and -1000 HU being black. Table 1 shows some typical ranges for some common tissue types of the human body. Since the HU is a comparison of density, water is assigned a HU of zero while heavier tissues like bone will have HU in the positive range, and tissues less dense than water such as fat will have HU in the negative range.

Table 1: Typical values of Hounsfield Units (HU) for common tissue types.

Type	HU
Bone	+400 to +1000
Soft Tissue	+40 to +80
Water	0
Fat	-60 to -120
Air	-1000

In order to format the HU into regular intensity values for proper viewing and display, a process called 'windowing' is performed. This is because the human eye cannot accurately distinguish between 2000 or more different shades of grey. Therefore, only a limited number of HU are displayed for easy viewing and interpretation. A

grayscale image is achieved by setting the window center (C) and the window width (W) to a suitable range of Hounsfield units, depending on the tissue being studied.

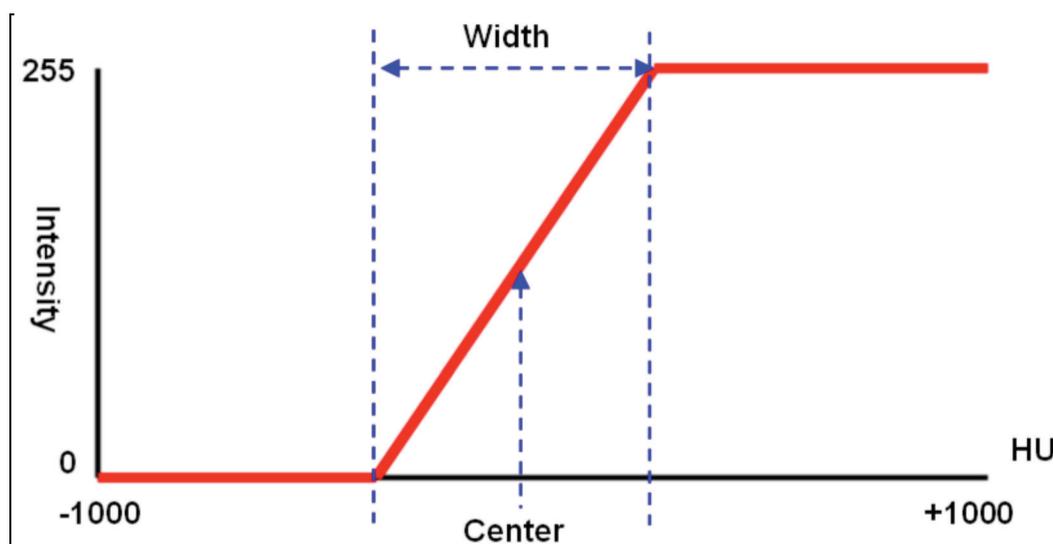


Figure 5: Illustration of the windowing process, converting Hounsfield Units (HU) to 8-bit pixel intensities.

A simple illustration of how the HU in a CT image can be converted to regular 8-bit grayscale pixel intensities to target a specific tissue type is shown in Figure 5. The window center represents the central Hounsfield unit of all the numbers within the window width, and the window width covers the HU of all the tissues of interest and these are displayed as various shades of gray. Tissues with CT numbers outside this range are displayed as either black or white. Both the window center and width can be set independently and their respective settings affect the final displayed image.

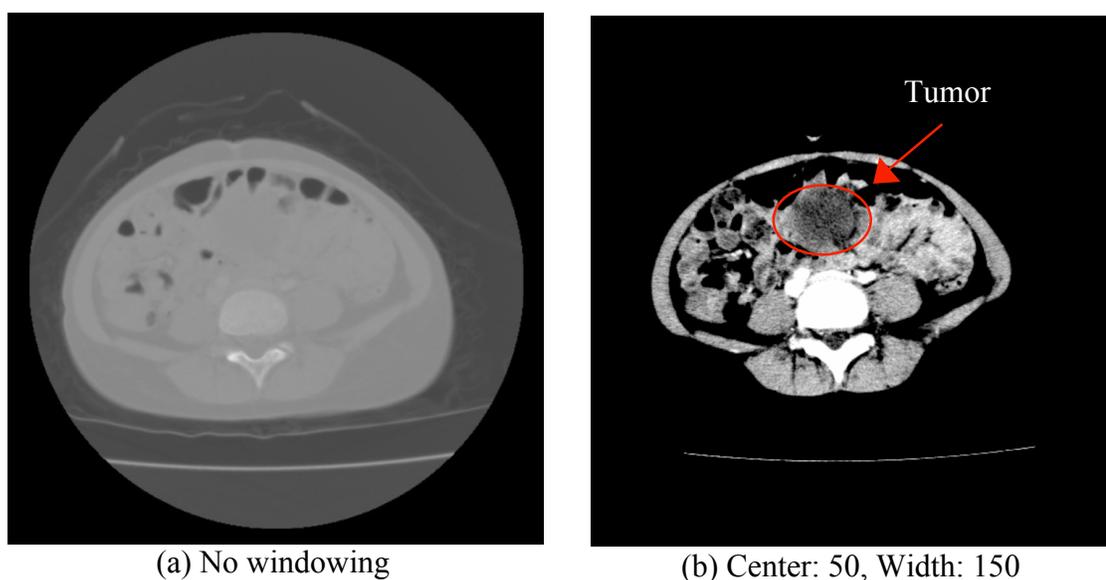


Figure 6: Examples of windowing effects, using different window center and width values on the same image to emphasize tissue different types and densities.

Figure 6 illustrates an example of manipulating the window center and width to achieve different display effects of the same CT image data. Figure 6a displays the original CT image data with no windowing, leaving many structural details (i.e. organ edges, etc.) unclear. However, when windowing is applied in Figure 6b, with the center and width set for viewing soft tissues and muscle, the details of the internal organs appear to be clearer, with the tumor highlighted with a circle. In Figure 6a, the tumor is barely, if at all, discernable.

1.2 Motivation

With the recent introduction of high-resolution multi-slice medical imaging technology, methods of acquiring and visualizing high quality medical image data in a non-invasive manner are now commonplace. With this detailed information of the patient

body, doctors and surgeons can now effectively use this visual information to substantially improve patient diagnosis, treatment monitoring, and pre-operative planning. However, there is still a significant amount of analysis and examination on the surgeon's part to properly identify unhealthy tissues such as tumors and individual organs. The ideal scenario would be when an automated process can intelligently identify if a tumor or lesion is present in the image data and simultaneously delineate that malignant region as well as the other healthy tissues and organs. Surgeons could then view any organ on its own or as a collection within the anatomy. An important factor is to properly segregate the tumor from the rest of the healthy organs in a patient. The ability to automatically segment a tumor could substantially improve the diagnosis as well as surgical planning for patients. Surgeons could analyze the shape and size of a tumor prior to surgery, or monitor the progress of a patient by comparing the segmented tumor through various stages of treatment.

However, due to the complexity of the medical images, and the irregular shapes and sizes of tumors, this segregation process can be a difficult task. The edges of a tumor can be difficult to delineate from the other organs due to noise during the image data acquisition process, or part of the tumor having a similar tissue density as other surrounding organs. Manual segmentation of a tumor from individual images can be a daunting task, since partial body scans range from tens to hundreds to even thousands of slices. Even if a person could take between 2-4 minutes per image, it would take hours or even days to manually segment a tumor from a 500-slice dataset.

There is a need for an automated process where a tumor can be properly identified with substantial accuracy using computerized segmentation algorithms. Automated

computer tumor segmentation works by having the user provide some form of *a priori* information about the tumor such as estimated tissue density and shape. The algorithm will then process the rest of the medical image dataset to extract the tumor based on the information provided. Chapter 2 provides a discussion on the various forms of medical image segmentation techniques currently available. The difficulty and complexity of automatic tumor segmentation can be loosely categorized at three different levels: 1) tumors with homogeneous tissue densities, 2) tumors with some inhomogeneity and fuzzy edges, and 3) tumors with heterogeneous tissue densities. Depending on the type of tumor, some tumors will have consistent tissue densities, while some will be highly calcified (calcium build-up in tissues) and intermixed with tissues of different densities, examples of all three categories are shown in Figure 7.

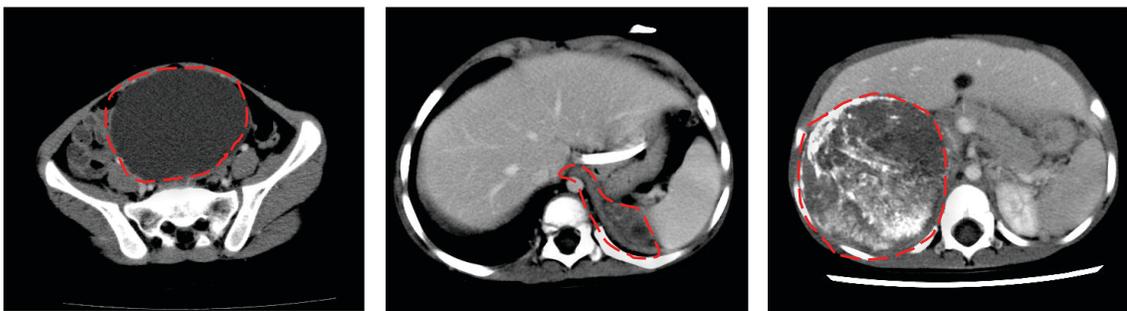


Figure 7: Examples of the three different levels of tumor complexity for automatic segmentation: a homogeneous tumor (left), a tumor with some inhomogeneity and fuzzy edges (center), and a highly heterogeneous tumor with calcifications and mixed tissue density (right). Tumors are highlighted with red dashed lines.

Due to the large scale of these image data, visualizing them in an effective and intuitive manner can be a challenging task as well. Medical image data are commonly stored as 512 x 512 pixels per image, while newer scanners are capable of producing

1024 x 1024 pixel slices. For a dataset of 400 slices, this means the amount of graphics data being processed is approximately 105 million voxels. Graphic processing technology has come to the point where it can be applied to efficiently visualize these large-scale medical image data. If this is combined with the current state of virtual reality technology, medical image data can be loaded into an immersive environment where doctors and surgeons will be able to interact and manipulate the 3D data in real time. Currently, there are several software packages available to radiologists for viewing and analyzing medical image data in three-dimensional visualization, as will be discussed in the next chapter. However, these applications are specifically designed for radiology and are very complicated for any non-radiology personnel such as a surgeon or doctor to use. Should a surgeon need to re-examine an image dataset, it is most likely he or she would have to make a trip to the radiology department and even then, manipulation control would be in the hands of the radiologist and not the surgeon. This tedious process wastes crucial time and efforts. Surgeons must have better tools to access and manipulate medical image data. Any such application will be beneficial as an assistive tool for medical diagnosis or surgical planning, which will ultimately lead to improved patient care.

These requirements form the basis of the research presented. This dissertation will discuss the development of a framework for tumor segmentation and immersive interactive visualization of medical image data.

1.3 Dissertation Organization

The remainder of this dissertation is organized as follows: Chapter 2 presents an in-depth literature review of research in medical image segmentation, current advances and uses of virtual reality in medicine, and the research issues identified. Chapter 3 describes the probabilistic segmentation method in detail, while Chapter 4 describes a second segmentation method developed based on fuzzy logic. A framework that was developed to visualize medical image data in an interactive immersive virtual environment is discussed in Chapter 5. The results of the segmentation algorithms will be presented in Chapter 6, including statistical analyses of the segmentation errors incurred from both methods and performance statistics, such as frame rates and initialization time, of the developed visualization framework. Finally, the research presented in this dissertation is discussed and summarized, with conclusions formed and some future work defined in Chapter 7.

2 LITERATURE REVIEW

2.1 Image Segmentation

The complexity of medical image segmentation has encouraged many fields of research into improving automatic and adaptive segmentation algorithms. Current segmentation methods can be broadly categorized as: 1) classical approaches such as thresholding and region growing, 2) statistical or probability based classifiers, 3) learning algorithm approaches such as artificial neural networks and fuzzy clustering, 4) deformable models approaches, and 5) database or atlas guided approaches. Each of these methods will be discussed in detail.

2.1.1 *Classical Approaches*

Classical approaches like thresholding and region growing are simple and common segmentation algorithms that perform by inspecting the intensity properties of each pixel of an image slice. In thresholding segmentation, a threshold value is selected by the user and every pixel in the image is classified as either having intensities less than the threshold or having intensities greater than or equal to the threshold [4]. The region growing algorithm performs in the same manner but only adjacent pixels with similar (within a small tolerance) pixel intensities are grouped together as a region, creating multiple regions within the image [5]. Unfortunately, while these methods are easy to implement, they are not effective enough to segment complex medical image data as

pixel intensity alone is not sufficient to accurately distinguish organs and/or tumors. This limitation of classical approaches spurred research and development for more robust and adaptive segmentation algorithms.

2.1.2 Statistical and Probability Based Approaches

These methods perform by analyzing the pixels in an image and generating a statistical histogram or a probability map. Statistical histograms are generated based on the pixel intensities values and will incorporate statistical properties such as mean intensity, the distribution of intensity values, and/or the homogeneity of the pixel intensities. Similarly, probability maps are generated based on predefined criteria set by the user. In most cases, probabilities of a pixel being the object of interest are generated for each pixel based on its intensity and location [6-8].

For example, Vincken *et al.* [9] and Kupinski *et al.* [10] developed methods of using probability as a means to merge regions for segmentation. Vincken's method divides a three-dimensional image data representation created from slices into a voxel data tree structure, and links adjacent voxels based on a parent-child voxel probability linking scheme. Kupinski's method performs a statistical analysis on the pixel intensity histogram using a small-variance Gaussian function on previously defined partitions of the image data, to generate a probability distribution for these partitions. Based on these probability distributions, the partitions are evaluated and the partition with the maximum probability is selected as the region of interest.

Even though the current probabilistic segmentation methods available for use have shown success in segmenting the required object of interest, these current

implementations usually require large amounts of computer memory and processing resources. This is because these methods perform a first-pass inspection of the entire medical study in order to generate statistical histograms or probability maps. Loading the entire image study in 3D for this processing requires immense computational resources. Thus, very few methods are able to generate and calculate probabilities for segmentation in real-time.

In addition to the methods discussed, several early methods that used the Monte Carlo probability simulation for image segmentation have also been investigated [11-17]. Examples include early work done Zhang *et al.* [18,19] developed a segmentation algorithm that implements the Monte Carlo algorithm to create a stochastic approximation model. This model is later used in conjunction with a Markov Random Field (MRF) to perform image segmentation. Chou *et al.* [20] proposed an image segmentation procedure that uses MRF to generate likelihood values within an image for region labeling using a Bayesian inference system.

More recently, many methods have implemented Markov Random Fields (MRF) as a prior model building for image segmentation [21-31]. Melas *et al.* [32] used MRF as a model estimation tool to generate likelihood values based on texture information within the image, which was later segmented using a Bayesian approach. Zhang *et al.* [33] used hidden MRF models with model fitting performed using the Expectation-Maximization (EM) algorithm, to extract features from brain MR images.

Based on the detailed review of the methods discussed, probability based segmentation algorithms have been a prevalent concept in image segmentation. It is based on the success of these methods that a probabilistic algorithm is implemented in the

proposed segmentation method for generating pixel selection probabilities. The formulation of the probability calculations using a pixel's spatial and intensity relations in this proposed method is a unique approach based on the literature review performed.

2.1.3 Deformable Models

Deformable models use closed parametric curves or surfaces to delineate a region of interest. These methods work by detecting region boundaries or edges and then deforming curves [34,35,36] or surfaces [37,38,39] to match them. In doing so, a smooth representation of the object of interest is obtained at the end of the segmentation process. Another development to deformable models is a technique called "live-wire" [40,41], which is a semi-automatic algorithm. This algorithm performs interactively with the user, locating optimal boundaries or edges as the user moves the mouse. When the mouse is moved close to an optimal edge the live-wire algorithm snaps the curve to that located edge. In addition, *a priori* information can be used to create a deformable model template. Montagnat and Delingette [42] built and used deformable surface templates of the liver and ventricles to segment the liver from CT scans and ventricles of the brain from MR images. A drawback of deformable surface methods is that they are usually laborious since they require additional post processing calculations to connect the curves from each slice into a surface. Also, the shape of the object segmented is limited to the order of the curve defining the shape.

2.1.4 Atlas Guided Approaches

These methods create a compiled atlas of correctly segmented images from previous experiments as a reference to segment new but similar images [43-46]. For example, Lorenzo-Valdes *et al.* [47] proposed a method that built a digital probability distribution map or atlas based on manual segmentations of the left and right ventricle, myocardium, lungs, liver, stomach and skin on 3D cardiac MRI image sets of 14 subjects. Segmentation was performed by comparing the generated maps to other similar datasets, and then individually segment the left and right ventricles, and the myocardium with 85% - 91% accuracy.

Atlas-guided approaches are better suited for brain imaging where the object of interest to be segmented (i.e. cerebral cortex of the brain) does not differ greatly in size, shape or location from slice to slice within the same dataset. Due to the method's dependence on familiar and similar data, atlas-based methods are at a disadvantage when segmenting a rare or unique data set. Furthermore, they require an extensive database of accurately segmented results either from previous experiments or from manual segmentations, which can be very time consuming to create and query.

2.1.5 Learning Algorithms in Image Segmentation

Researchers have also investigated the use of learning algorithms derived from the field of artificial intelligence for medical image segmentation. Examples are methods using data mining [48,49], artificial neural networks [50] and developing a digital atlas from previous experiments for use as training data [51,52,53]. These algorithms simulate

a biological process to “learn” the properties of the object to segment. For example, an artificial neural network (ANN) [54] consists of a parallel network of nodes to simulate a network of neurons such as the brain, where each node has a corresponding weight assigned to it. Through a training process, these weights are computed. Then, the ANN can then be used to segment a new data set. In order to properly train and implement a segmentation algorithm using an ANN, a significant amount of *a priori* knowledge of the data set is required. This can take significant time and resources to complete and can be a disadvantage when trying to segment a unique or rare data set.

Some methods will use *a priori* information combined with a learning algorithm such as an ANN to perform segmentation on medical image data. For example, Ma *et al.* [55] developed a classifier that incorporates the probabilities computed based on set features of the voxel data. Self Organizing Maps (SOM) and Neural Networks are then used to analyze these probabilities and further simplify them to classify the voxel dataset into multiple groups. Another method that combines multiple forms of learning algorithms was proposed by Reddick *et al.* [56] where the ANN is first trained in a two-stage approach using Self Organizing Maps (SOM) to classify white matter and grey matter from multi-spectral MR images.

2.1.6 Fuzzy Segmentation Algorithms

In addition to the discussed learning algorithms, fuzzy logic has also been used as a primary tool for image segmentation algorithms. Fuzzy logic [57,58] is applied to determine the similarities and/or correlation between pixels, voxels, or regions in the

image data and to group or classify these voxels/regions appropriately. A brief discussion on fuzzy logic and fuzzy rule based systems is presented in Chapter 4.

One approach is to measure the similarity of adjacent voxels/regions (fuzzy connectedness) based on pixel intensity. Developed by Udupa and Samarasekera in 1996 [59,60], the original implementation of this approach performs a statistical analysis on the tissue density information prior to using region growing methods to join adjacent pixels. Fuzzy logic is used to evaluate the degree of “connectedness” between pixels during the region growing process. Further improvements have been made to the method to work on connected regions and incorporate information such as size, shape, and texture. Fuzzy connectedness or fuzzy region growing as it is sometimes known, has been successfully implemented in methods such as the separation of arteries and veins in magnetic resonance angiograms via an iterative process of connectedness checking [61], as well as in the estimation of brain tumor volume from MR images [62]. Other implementations of fuzzy connectedness include [63,64,65]. However, methods that implement some form of fuzzy connectedness checking incur excessive computational requirements. To improve computational speeds, Nyul *et al.* [66] utilized tree-graph searching algorithms as a pre-processing step prior to fuzzy connectedness checking, reducing computational time by almost 34%.

Another approach to using fuzzy logic for segmentation is to group similar voxels/regions based on a set of criteria, known as fuzzy clustering. However, unlike fuzzy connected methods, clustering does not take into account the location of the voxel (unless location is set as part of the criteria) or require that the voxels/regions are adjacent to each other. Thus, two disconnected voxels far apart from each other could be in the

same cluster or group, if the criteria were non-spatial. A popular implementation of fuzzy clustering is the fuzzy C-means clustering, developed by Dunn in 1973 [67]. In the original implementation, a single piece of data can belong to two or more clusters, creating a cross membership, and is based on the minimization of difference between any new data point to the overall mean of the cluster. If a minimum difference is met, then this new data point is accepted into the cluster. Since then, there have been many variations and improvements to the original implementation and thus remains a widely used classification method. Methods such as [68-72] are where pixels/regions were grouped using fuzzy clustering.

Yao *et al.* [73] developed a method combining *a priori* knowledge, fuzzy clustering and deformable models to segment colonic polyps from computed tomography images. The pixels were first clustered based on the algorithms estimation if they were air, polyp tissue, or non-polyp tissue. Then an initial deformable shape is placed within the centroid of a polyp cluster and the model is driven until it forms a closed shape that resembles the polyp cluster. Pham and Prince [74] developed an adaptive fuzzy C-means (AFCM) segmentation algorithm that adapts to inhomogeneities within an image using a bias gain field with AFCM. The gain field smoothes out any sudden intensities variations within MR images and has proven effective in simulated cases of corrupted images. Various other methods such as [75,76,77] also implemented the use of fuzzy sets to determine pixel intensity and/or spatial relations within an image.

Some of the strengths in these methods are that users can now devise fuzzy rules based on observations as segmentation criteria, and the fuzziness nature allows the algorithms to segment pixels/voxels that would otherwise have been discarded, especially

in cases when there is high inhomogeneity in the region of interest. However, this could lead to over segmentation or region leaks when the algorithm accepts more pixels than needed, resulting in a segmented region that is far larger than the actual object. For fuzzy clustering methods, there is also a high dependency on the number of classes or clusters defined during initialization.

2.1.7 Mixed/Hybrid Approaches

There are methods that will incorporate approaches from more than one of the categories discussed. Tolia and Panas implemented an AFCM method by building a probability vector based on inter-pixel correlations [78]. Colliot *et al.* [79] integrated fuzzy subsets to represent spatial relations within the image when performing segmentation using deformable models and has been successful in segmenting brain sub-cortical structures in MR images. Toledo *et al.* [80] introduced the idea of a new deformable model called eigen-snakes to segment vessels in angiographs based on a probabilistic framework that performed with encouraging results. These methods try to combine the best of various methods, such as the flexibility of a fuzzy segmentation method with the shape information available in deformable curves, but usually at the expense of additional computation resources and longer computation time. Hybrid methods are also usually more complex to implement.

2.2 Medical Image Data Visualization

The constant advancement in medical imaging technologies has allowed newer multi-slice CT or high-gradient MR scanners to scan at higher resolutions and faster times than ever before. Soon these scanners will be delivering thousands of image slices for a single patient to medical personnel. While this level of detail will usher in new advances it will also present new challenges such as storing, processing, and visualizing these large amounts of data. It has become evident that the light boxes (examples shown in Figure 8) used to view the conventional medical image on film will become insufficient and obsolete, they can typically only display between 40-60 images. This is where the use of three-dimensional (3D) visualization techniques can be used as a more intuitive and efficient way of representing large scale medical image data. 3D visualization combines multiple sequential slices of two-dimensional medical image data to generate a three-dimensional volume render of the dataset.



Figure 8: Examples of multi-slice films (left), a conventional light box viewing station (center), and a digital “light box” on LCD monitors (right).

In a 3D representation, data can be represented as either a surface or a solid. When drawn as a solid representation, the data points are converted to voxels. Four pixels from a single slice create the “front” face of the voxel and another group of four pixels from the next slice create the “back” face of the voxel. Linear interpolation is performed to fill in the gaps between these two faces, generating a three-dimensional representation as illustrated in Figure 9.

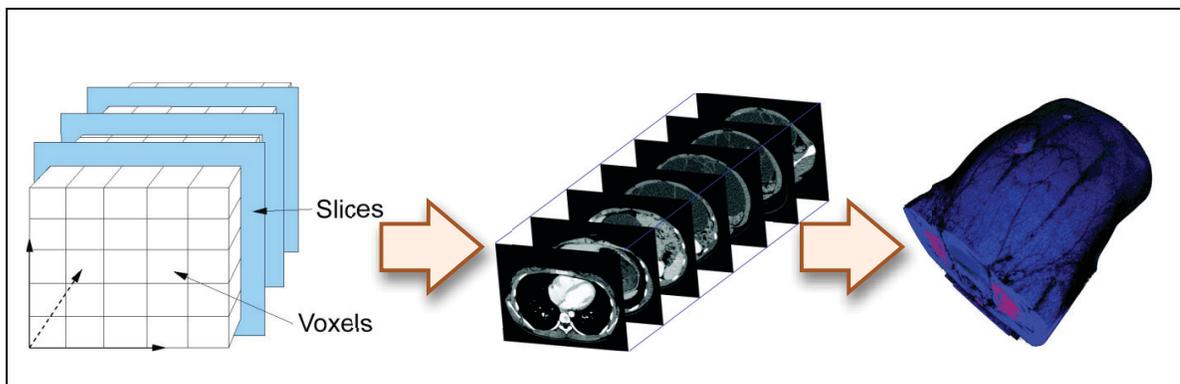


Figure 9: Illustration of how multiple consecutive 2D slices are combined to generate a 3D volume representation of the patient data.

The second form of 3D visualization is through surface representations known as isosurfacing. To generate an isosurface, a voxel set is searched for a particular intensity value related to a specific tissue type (e.g., bone, skin, etc). The result of this search is a collection of 3D points that represent where the desired tissue type exists in the volume. The isosurface is drawn using this set of points, creating a 3D graphical representation of CT/MRI data. The advantage of this type of representation is that it uses much less computing and graphical processing resources compared to a solid volume rendering

representation. However, the tradeoff is that when the model is simply a surface representation, it contains no interior information.

The Visible Human Project [81] is one of the early attempts at addressing the need for research into visualizing and registration of medical image data. Initiated in 1989 by the National Library of Medicine (NLM); cadavers of a female and a male were scanned in using CT, MRI, and X-ray, with the resulting data stored on a clustered database. After almost two decades, this project is still in progress with its objective to build an atlas of the human body that will completely and correctly identify every organ in the human anatomy.

Since then, various implementations have been developed to visualize complex medical data. Amira [82], Vitrea [83], OsiriX [84] and KitWare's VolView [85] are examples of desktop applications catered for medical image data visualization. These software packages all have built-in features such as windowing, surface and volume rendering, and pseudo-coloring via Color Look-Up Tables (CLUT), all which enhance the visualization process. Recent research has already shown that the use of 3D representations of medical image data can be beneficial and increase efficiency in providing additional information to medical professionals [86]. OsiriX is available as an open-source freeware on Apple Mac OS computers, while VolView is available on the Microsoft Windows OS platform as shareware with a limited number of features available for free and additional advanced features available with purchase. With these freely available tools, it is now possible for non-radiological specialists such as surgeons to load in a medical image dataset and view it on a personal computer. However, the

aforementioned software packages still require a significant learning curve since they are designed with the radiologist as the targeted end user.

Most currently available visualization software applications prefer isosurface representations compared to a full volume rendered representation to save on graphics and processing resources. Surface or contour representations assumes that the voxel data in CT/MRI to be discrete when in fact the opacity and voxel intensity transitions between air and other tissue densities are continuous and usually span between three to five voxels, depending on the resolution of the original image data. Thus, in order to accurately represent large-scale medical image data, a volume rendered representation is preferred [87], which is the method used in this research.

2.3 Virtual Reality in Medicine

Virtual reality (VR) has been successfully utilized in many fields such as engineering design, drug design, military simulations, and now medicine. When used in medicine, applications such as virtual colonoscopy, virtual endoscopy, and laparoscopic simulation and training have all proven to be great assets to medical professionals. Current research on these applications will be reviewed and discussed in this section.

2.3.1 Virtual Colonoscopy

Colorectal cancer is the third most diagnosed type of cancer and the second leading cause of death in the United States [88]. Despite this fact, most people choose not to participate in the screening process because of the associated discomfort, high risk, and

high cost. In order to encourage a higher screening rate, much research has gone into improving the accuracy and validating the use of virtual colonoscopy (VC) as an alternative to the more common conventional optical colonoscopy (CC) procedure. Virtual colonoscopy is performed by taking spiral CT scans of the patient's abdomen after the colon has been thoroughly cleansed and inflated with either room air or CO₂. During 30-40 seconds of a single breath-hold, several hundred high resolution CT scans are taken to form a 3D volumetric representation of the abdomen, after which a model of the inner colonic surface is generated and inspected for polyps (abnormal tissue growth), as shown in Figure 10.

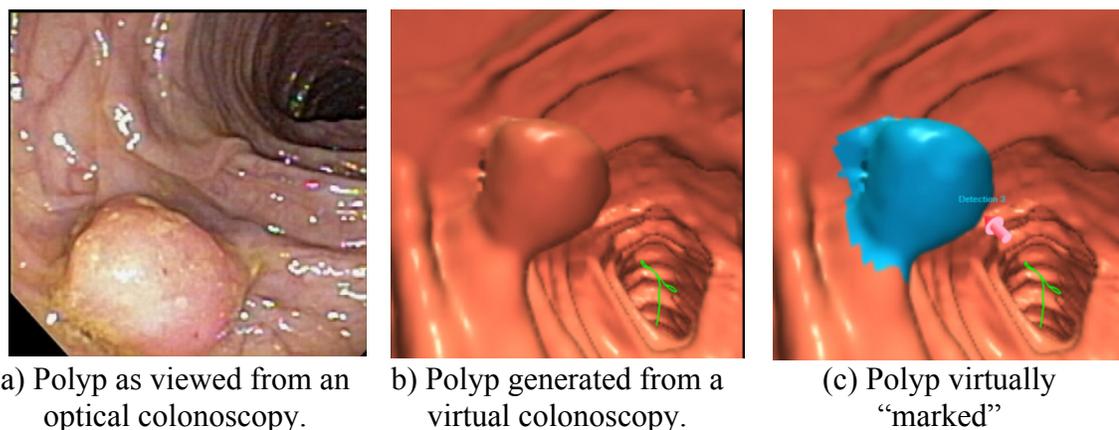


Figure 10: Conventional colonoscopy and virtual colonoscopy.
(<http://www.rsna.org/rsna/media/pr2005/Summers-Colonoscopy/summers.cfm>)

The accuracy of VC compared to CC has been the focus of many recent case studies. The majority has stated that despite being less invasive, the accuracy of VC is still relatively low to that of CC but shows great promise. Studies conducted by Pescatore *et al.* in 2000 [89] and by Cotton *et al.* in 2004 [90] concluded that VC is not yet ready

for widespread use to replace CC. This is due to the low sensitivity rates when using VC, which is a measure of the detection of polyps or lesions on the colon. The results of both studies showed low sensitivity rates of 36%-38% for small polyps (smaller than 5mm) and 63%-69% for large polyps (larger than 10mm) when using VC, while CC showed sensitivity rates as high as 99% for small polyps and 100% for large polyps. The low reported sensitivity rates are due to inefficient polyp identification algorithms in VC. However, both studies have also stated that the sensitivity rates for VC did increase when radiologists and gastroenterologists performed a manual inspection of the generated model of the inner colonic surface. As CT scanning technology continues to evolve and become faster and more accurate, virtual colonoscopy will someday replace conventional colonoscopy as a noninvasive alternative.

2.3.2 *Virtual Endoscopy*

Virtual endoscopy expands this virtual exploration to view other anatomical structures such as the colon, airways, and joints. For example, air can be segmented from CT or MRI images to obtain outlines of inner airway walls. These outlines are then interpolated and joined to generate a three-dimensional representation with a fly-through path that a virtual camera can navigate. However, in the cases when a fly-through path is not available, a path can be created manually using two-dimensional slices from three separate views.

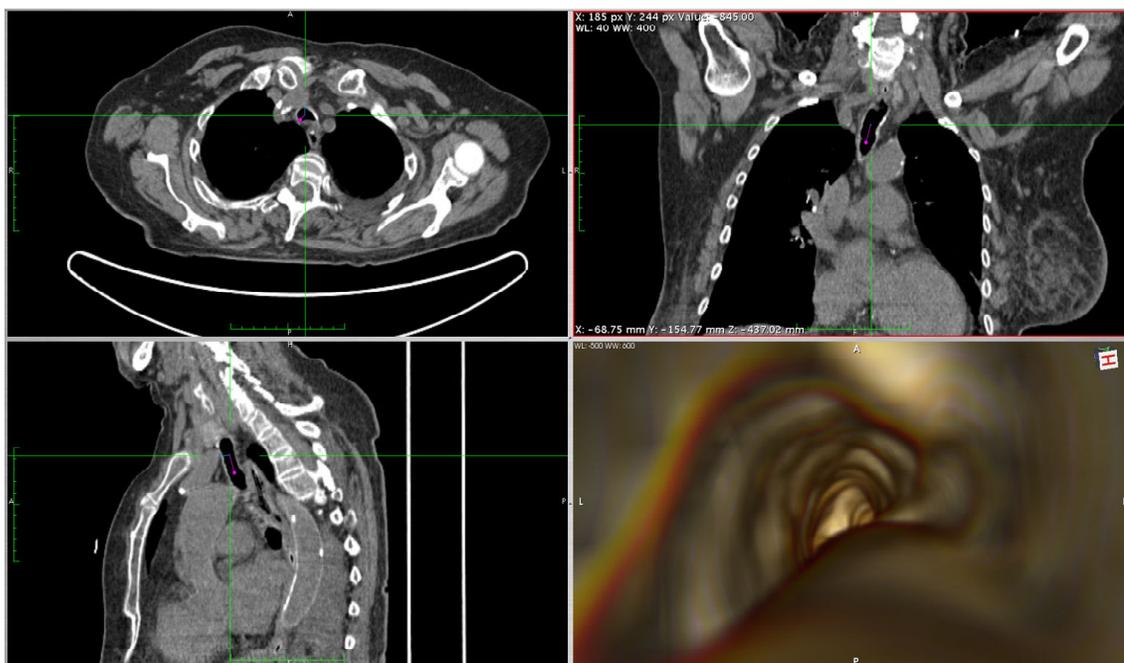


Figure 11: Example of manual navigation for a virtual endoscopy.

Figure 11 shows a screen display for a manual navigation of a virtual endoscopy. There are three two-dimensional views and a three-dimensional camera view. The top left of the screen shows the axial (top-down) view, top right of the screen shows the front-back view, and the bottom left of the screen shows the left-right view. The camera view is shown in the bottom right of the screen. By clicking the mouse pointer at a specific location in any of the three two-dimensional views, the camera shifts to the corresponding location in the three-dimensional camera view. Although this implementation provides sufficient information to the user, it can be a tedious process to use the mouse as a navigational input device as well as keeping track of each of the views simultaneously, with much ongoing research on improving the navigation interface [91].

2.3.3 Laparoscopic Simulation/Training

In the last decade, laparoscopic surgery has replaced open surgery for several procedures because of the benefits of minimally invasive procedures. Laparoscopic surgery is a modern technique of minimally invasive surgery in which procedures are performed through small incisions, usually between 0.5–1.0 inches. Through these incisions, trocars can be inserted into the patient's body. Trocars are hollow cylinders and act as instrument ports during a laparoscopic surgery to introduce instruments such as scissors, graspers, or a laparoscope. The laparoscope is the key element in a laparoscopic surgery as it allows the surgeons to view the operative region inside the patient's body. The laparoscope is a telescopic rod with a lens and light source connected to a video camera through a fiber optic cable. If possible, the area of the body is inflated with carbon dioxide to create a larger working space. Carbon dioxide is used because it is a gas that can be easily absorbed by tissues and it is a non-flammable, which is important since electrosurgical instruments are commonly used in laparoscopic surgery. Figure 12 illustrates a laparoscopic procedure.

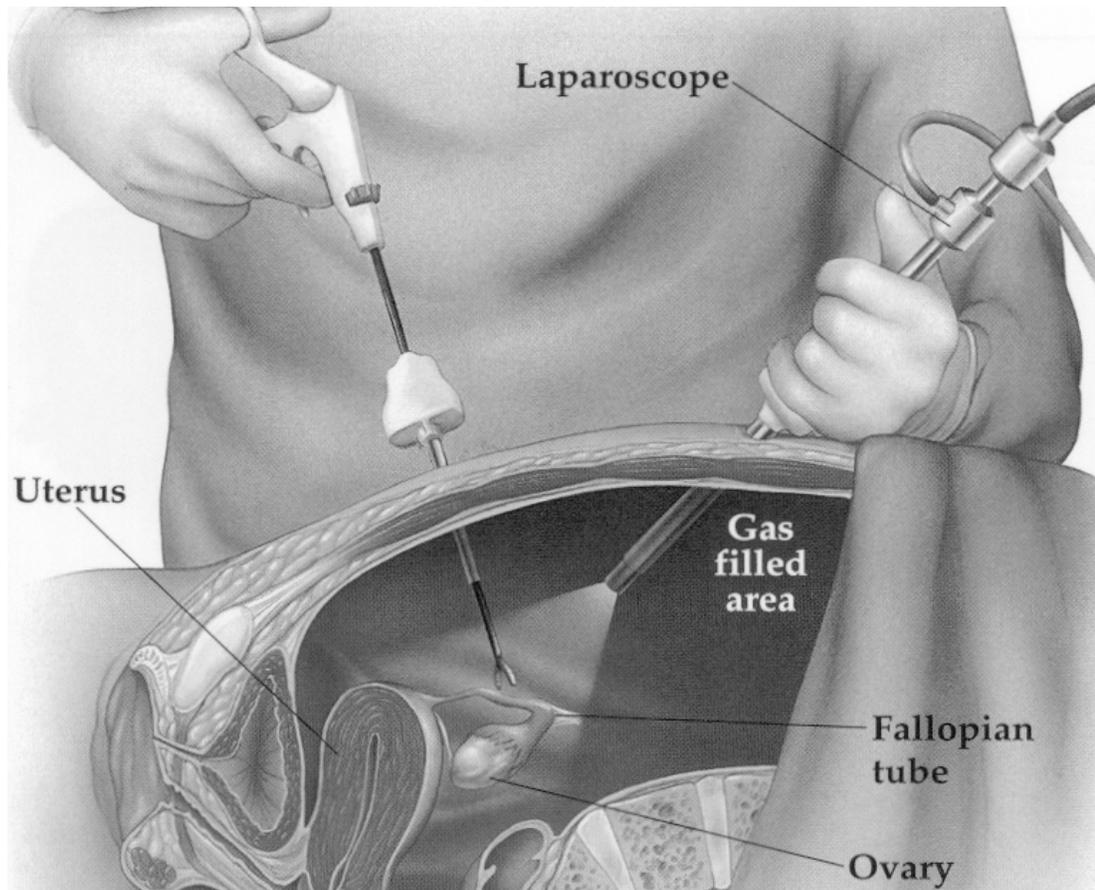


Figure 12: Illustration of a laparoscopic procedure.
 (Image from Laparoscopic Surgery by David Anaise, MD:
http://www.danaise.com/laparoscopic_surgery_7-5.htm)

A laparoscopic surgery procedure imposes a strain onto the surgeon since it requires a high degree of precision and psychomotor skills due to the two-dimensional video feedback interface. The manipulation of laparoscopic instruments in a three-dimensional environment (inside the patient) based on a two-dimensional visual can be disorienting for the surgeon. A typical laparoscopic procedure is shown in Figure 13 below.



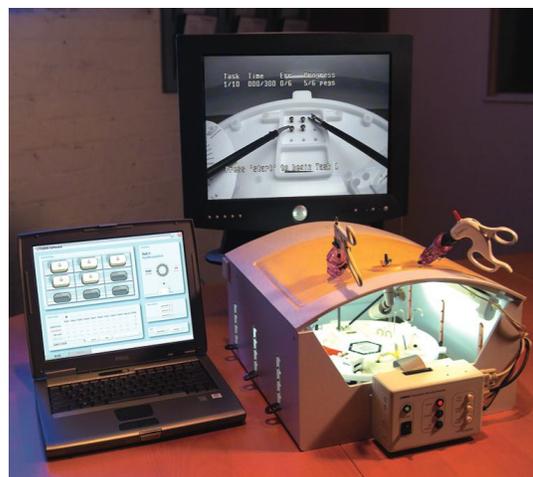
Figure 13: A typical laparoscopic procedure using a two-dimensional video display. (Photo from VCU Medical Center, Minimally Invasive Surgery Center: http://www.vcu.edu/lesspainsurgery/fellowship_year_long.htm)

Several applications have emerged for use as training platforms for laparoscopic surgery to improve the psychomotor skills of surgeons and recent research have placed a focus on improving and validating such training applications. The Virtual Laparoscopic Interface (VLI) by Immersion Corp. [92], LTS by RealSim Systems [93], Computer Enhanced Laparoscopic Training System (CELTS) by Simulation Group [94], LapMentor by Simbionix [95], and Minimally Invasive Surgical Trainer – Virtual Reality (MIST-VR) system by Mentice Corp. [96] are examples of laparoscopic simulators/trainers available in the market today, shown in Figure 14. Most of the

mentioned systems do provide the option of force feedback but have yet to be validated by experts and peers in the field. In a recent study published by Schijven and Jakimowicz [97] in 2003, the force feedback feature is an indispensable component to a realistic simulation environment.



LapMentor by SimBionix



LTS by RealSim Systems



VLI by Immersion Corp.



MIST-VR by Mentice

Figure 14: Various minimally invasive surgery simulators/trainers.

The MIST-VR system is a system that assesses the psychomotor skills of a surgeon while performing standardized laparoscopic procedures. Trainees or surgeons are

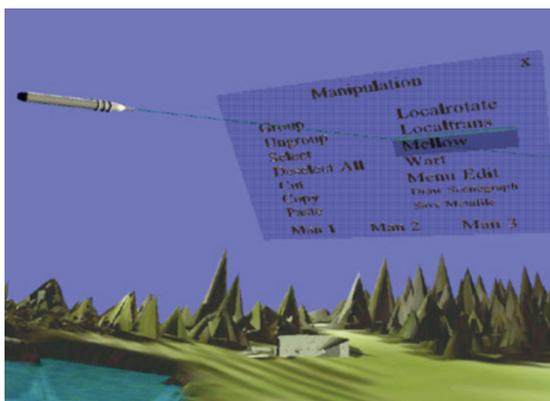
required to perform several basic laparoscopic procedures in the simulator, and criteria such as dexterity, length and number of movements, and total completion time are recorded and assessed. Despite lacking in haptic force feedback, this system is one of the foremost acceptable alternatives to box-trainers [98] and has been validated by many experts in this field. Its popularity could stem from its low cost of US\$25,000 (as of October 2007) compared to other laparoscopic simulation systems, which could be as high as US\$100,000. In 2004, Gallagher *et al.* [99] performed a study to evaluate the MIST-VR system and showed promising results as to the effectiveness and validity of the application. Surgical residents that were trained using MIST-VR were able to perform laparoscopic gallbladder dissection faster with fewer errors when compared to those who received standard surgical training. Additional studies by Ahlberg *et al.* [100], and Grantcharov *et al.* [101] also showed similar positive conclusions.

2.4 User Interface and Interaction Design

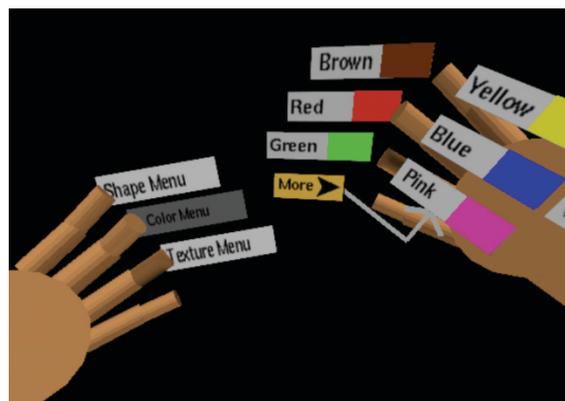
In their articles [102,103], Bowman *et al.* suggested that user interactions with a virtual environment can be generalized as four universal tasks: navigation, selection, manipulation, and system control. Navigation refers to the users changing their viewpoints in the environment, selection is the task of choosing and picking objects within the scene, and is often paired with the manipulation, which is to modify the properties of the selected object's properties such as position or orientation. The final universal task, system control is defined as changing the state of the environment and is most commonly tied to input interfaces such as menus or device shortcuts.

2.4.1 Menu Interfaces

Early attempts of menus in a virtual environment were very similar to the ones available in 2D desktop applications, floating menus within the virtual environment with pull-down options and “click-able” via a ray-casted selection by a 3D input device such as a wand [104].



a) Floating 2D menu with ray-casting for selection and manipulation



b) Menus attached to a pinch-glove



c) Tablet-pen menu interface

Figure 15: Examples of different frequently used menu interfaces.
(Images from <http://people.cs.vt.edu/~bowman/pubs.html>)

Examples of these are shown in Figures 15a and 15b. However, these types of menus suffered from the inaccuracy of the ray-casting, using tracked movements for

selection (i.e. jitter), and occupying a large portion of the virtual scene if many sub-menus were needed. Other early implementations include a tablet type interface [105,106] where buttons and menus were displayed on a tablet computer that could be physically carried by the user. A stylus allows the user to tap and select specific buttons, sliders, or icons. An advantage is that this is a 2D interface that users are already familiar with, and users have a physical surface and constraint to interact with. This interface is shown in Figure 15c.

Recent research into virtual environment interface designs such as the development of the “Spin Menu” by Gerber and Bechmann [107], a contextual hierarchical 2D menu with sub-menus that are arranged either in a concentric manner or stacked on top each other in the virtual environment. Other implementations such as [108,109,110], also focused on modifying how 2D menus are displayed in a 3D virtual environment, trying to balance the number of menu features with usability.

2.4.2 Designing a User Interface

Interface design has always been a challenging task, especially when trying to balance functionality and advance features with a usable and intuitive control interface. When reviewing applications such as Amira, Vitrea, OsiriX, and VolView that incorporated advance radiology imaging utilities and features, their user interfaces can be very unintuitive and at times tricky to the unfamiliar user, as shown in Figures 16 and 17. This is because these applications are developed with the radiologist as the target user groups. Although these applications are packed with advance features, most of these features are those that a surgeon or non-radiology person might find unnecessary.

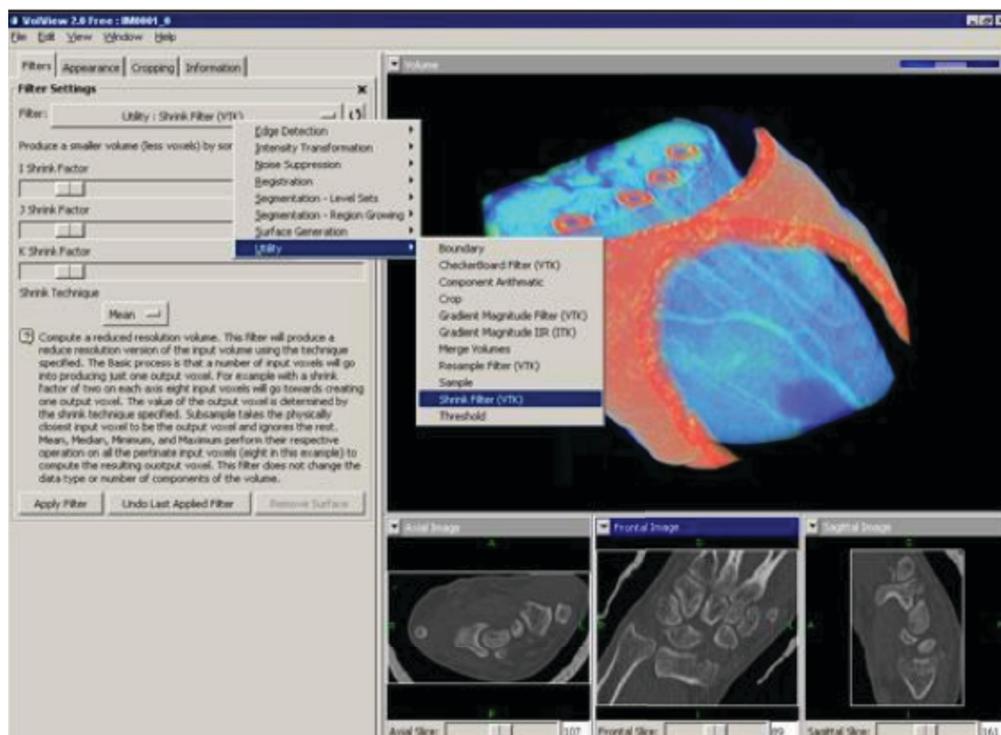


Figure 16: Screenshot of Kitware's Volview.

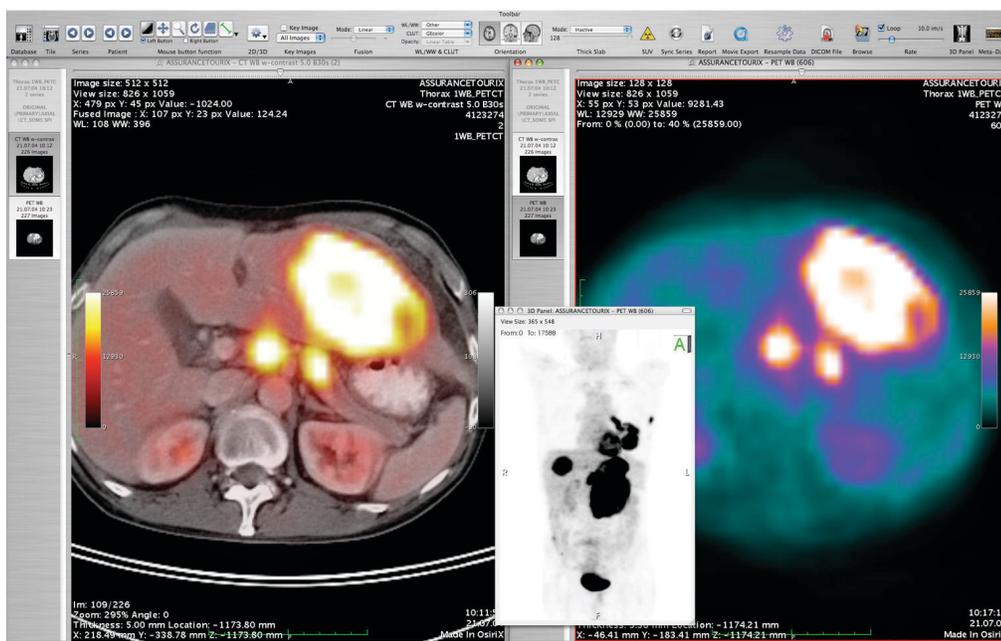


Figure 17: Screenshot of open source Osirix radiological software application.

According to Don Norman in his book “*The Design of Everyday Things*” [111], in order for users to efficiently execute difficult and complex tasks as simple as possible, the interface has to satisfy seven basic principles:

1. Use both knowledge in the world and knowledge in the head – The system needs to be able to provide the necessary knowledge and cues within the environment itself, but at the same time the operations and tasks performed should also be clear and appropriately match the mental model that the user already has.
2. Simplify the structure of tasks – Mental aids, visual cues and feedback, and automating part of or all of the tasks (as long as it does not distract the user) are ways to simplify a task. However, it is also important that the simplification does not result in taking control away from the user.
3. Make things visible – The interface should be clear of what the features are and what they do, and the effects of those features should be visible in the system.
4. Get the mappings right – Users usually have a mental model of how certain things are implemented, thus it is important that these models and the user’s actions are clearly mapped to the system events. A small movement should have a small effect event, and a large movement should have a large effect event.
5. Exploit the power of constraints – Constraints make it impossible to do anything else but the correct action in a correct way at a specific situation.
6. Design for error – Designers should always expect the user to make a mistake and anticipate them, and should a mistake happen, the user should be able to recover from that.

7. When all else fails, standardize – If natural mappings are unable, it is important to standardize the controls so that users can learn them quickly and perform similar tasks within the system easily.

Ben Shneiderman also provided his eight rules when it comes to designing a user interface [112], most of which closely parallels Norman's:

1. Strive for consistency in action sequences, layouts, terminology, and so on.
2. Enable frequent users to use shortcuts or special key sequences and macros, to make frequently performed actions more accessible.
3. Offer appropriate informative feedback for specific user actions, at appropriate times and levels that will not be distracting to the user.
4. Design dialogs to yield closure, such as alerts or prompts for the user when the designated task or action has completed.
5. Offer error prevention and simple error handling, so that ideally the users will not make any mistakes and when they do, they are offered clear and informative options to recover from those mistakes.
6. Permit easy reversal of actions, similar to error prevention, but more towards encouraging the user to explore the system. A simple reset or state reversal option will help to relieve the anxiety that they might make a mistake.
7. Support internal locus of control, so that the user is in control of the system and the system will respond appropriately to the user's actions.
8. Reduce short-term memory load by maintaining a simple and organized display to the user and providing sufficient time to learn the interface.

Both Norman and Shneiderman have very similar views on user interface design and both authors stress highly on consistency of interface controls, minimizing user's ambiguity and errors, and to provide an appropriate amount of information feedback to the user. These will be the guidelines followed during the course of developing the interface discussed in this dissertation.

2.5 Research Issues

Based on the various literature review of current research in medical image segmentation and the use of virtual reality in medicine, three research issues have been identified. They are:

1. To efficiently and accurately segment tumors within CT/MR image data.

Most segmentation methods require significant amounts of computing and memory resources to implement or substantial *a priori* information. Segmentation algorithms that are efficient, accurate, and can perform with minimal *a priori* information to allow for processing of new or unfamiliar datasets would be very useful for medical personnel. These algorithms would allow a wider range of computational resources and software to exist without affecting the algorithm's performance.

2. To utilize virtual reality and visualization technology to assist in and improve tumor diagnosis and surgical planning for tumor removal.

The ability to examine and manipulate patient data in an immersive environment will provide surgeons with additional visual information to aid in the diagnosis procedure or as an invaluable aid in surgical planning.

3. To develop an intuitive interaction scheme for manipulation of large medical studies within a virtual environment.

When interacting with a vast amount of image data in an immersive environment, the interaction between the user and the system has to be intuitive and easy to comprehend. The volumetric representations generated from medical studies can be problematic to navigate and ultimately hinder the necessary exploration a surgeon may require. It is crucial that the interaction method be simple yet powerful to enhance surgical planning.

3 PROBABILISTIC SEGMENTATION OF TUMORS

3.1 Introduction

This chapter describes the first of two segmentation methods developed, which is a probability based segmentation algorithm. The flowchart in Figure 18 illustrates the steps of the segmentation process.

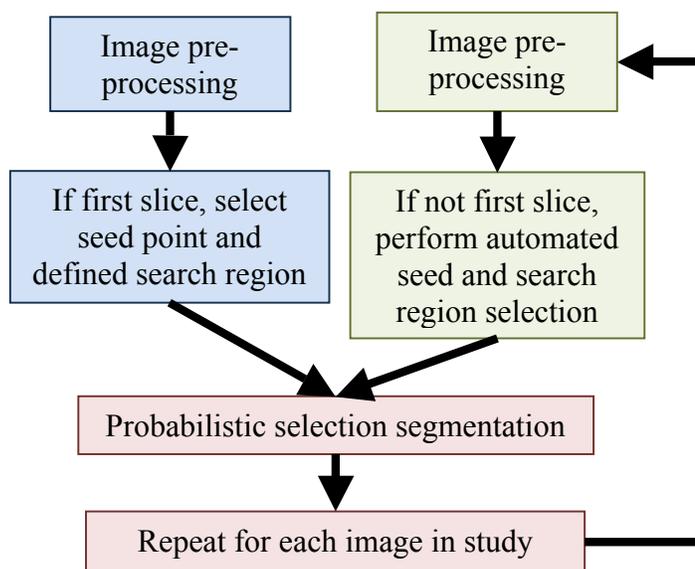


Figure 18: Flowchart of segmentation process.

The method begins with a pre-processing step that includes noise removal and performing a window center/width operation to increase the contrast between the object of interest and the background. To initialize the segmentation process, a user selects seed point(s) within the first slice in the study and defines a search region. For each remaining slice in the set, an automated algorithm is performed to automatically place the seed and redefine the search region for segmentation. Lastly, probabilities are generated for every

pixel in the search region, and pixels with high probabilities are segmented as the object of interest. The process is then repeated for every image slice in the data set.

3.2 Image Pre-processing

Before any segmentation is performed, the image data is pre-processed for improved visual quality. The first step in this pre-processing is to obtain the image data in terms of pixel intensities via windowing. The window center and width of the Hounsfield Units are set to improve the contrast between the object of interest and the background, converting the raw CT data to actual pixel representations for display. If needed, additional processing to remove image noise using a standard Gaussian smoothing filter is performed. The Gaussian filter was selected because of its simple implementation and effectiveness in reducing noise at a low computation cost. The kernel of the Gaussian filter is set to a default of 5x5 pixels, but is customizable to any desired setting. A larger kernel filter may be required for noisier images at the user's discretion. These evaluations are all performed on the first image of the dataset, under the assumption that 1) the quality of the first image represents the quality of the dataset, and 2) any enhancements made to the first image will yield similar improvements for the remaining image slices.

3.3 Automated Seed and Search Region Selection

Most of the objects in a study do not remain consistent in terms of position and size from slice to slice, thus the location of the seed must change accordingly. A process to automatically re-position the seed and search region was developed to abstract this

process from a user. As each subsequent slice is processed for segmentation, the process ensures that the seed remains in the object of interest. This is accomplished by using information from the previous slice as well as the first slice of the study (where the user selects the original seed and search region).

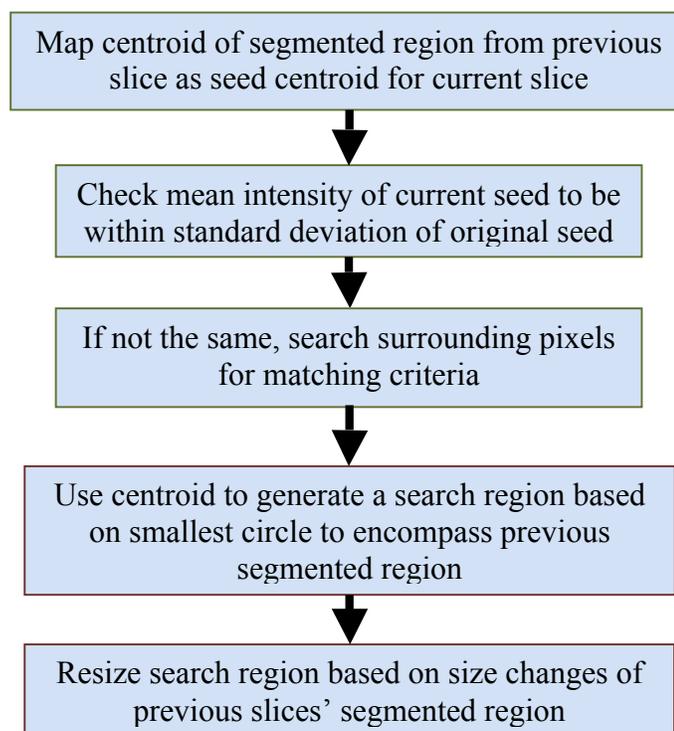


Figure 19: Flowchart for automatic seed relocation and search region selection procedure.

To initialize the method, the user selects a seed, which is defined as a square region of pixels, within the target object on the first slice of the study as the seed region. The default area is a 5x5 grid square, but can be set to any size region. By using a seed region rather a single point, instances when pixels within the tumor or object of interest are not homogeneous are accounted for. The mean intensity of the group is used as the seed intensity referenced during the segmentation procedure. For the first slice of the study, the user will also be required to define a search region, which is a circle of a fixed

radius using the seed as the center. If this is not the first slice, the following method is applied.

The centroid of the segmented region from the previous slice is used as the new seed region centroid in the current slice. If there are multiple regions (due to segmentation noise creating multiple smaller clusters of stray regions), only the biggest region will be considered. The centroid is calculated by averaging the x and y coordinates of all the pixels inside the region, shown in Equation (1).

$$centroid = \left(\frac{x_1 + x_2 + x_3 + \dots + x_n}{n}, \frac{y_1 + y_2 + y_3 + \dots + y_n}{n} \right) \quad (1)$$

where n is the number of pixels inside the region.

The mean intensity at this new seed region is then compared to the mean intensity of the original seed region from the first slice as well as the mean intensity of the segmented region from the previous slice. If they are not within a tolerance of one standard deviation of the mean intensity, the algorithm searches the neighboring pixels that make up a seed for a mean intensity within one standard deviation. If a group of pixels satisfies these criteria, it becomes the new seed. Otherwise, the algorithm reverts to the seed mapped from the previous slice to be used for segmentation.

The new search region is defined based on the object segmented in the previous slice. In the previous slice, the smallest circle required to completely encompass the object segmented is calculated and used as the search region for the current slice. Then the running average of the change in size of the circle from the last three images is calculated and used to shrink or enlarge the new search region accordingly. The average size change is defined as:

$$\bar{C} = \frac{3C_1 + 2C_2 + C_3}{6} \quad (2)$$

where C_1 , C_2 , and C_3 are the change in segmented region size for the previous slice, 2nd previous slice, and 3rd previous slice respectively. This is done to accommodate for a change in size and location of the object in between slices, this also helps to reduce processing time since the algorithm will not have to evaluate the entire image. However, if desired, a user may select to search the entire image on each slice. This would be useful for the case where multiple disconnected tumors are located in the image slice. The drawback is the additional processing time needed to search every pixel in the image.

3.4 Segmentation Procedure

The segmentation process consists of steps that combine aspects of thresholding and region growing. The segmentation method processes the volume data's two-dimensional slices one at a time as opposed to processing the entire three-dimensional data to minimize memory usage and computing resources. Incorporating the use of the automated seed selection algorithm developed, these two-dimensional slices can be processed individually without compromising information from previous slices.

Pixel selection is based on the probabilistic selection criterion used in the Simulated Annealing (SA) heuristic optimization method [113,114]. An optimization problem is a set of coupled equations where one is designated as the objective function. It is the aim to either minimize or maximize the objective function while not going over preset values on the other equations (i.e. the constraints of the problem). All the equations are composed of the same set of independent design variables. An optimization method

intelligently chooses design variable values to test as it moves to the optimal set of these values. The goal is to find the set of values that either minimizes or maximizes the objective function, satisfies all the constraints, in the fewest possible design variable combinations (i.e. iterations).

In the SA method, a probability is responsible for deciding if a new design point (i.e. set of independent design variable values) is accepted to be the starting point for searches in subsequent iterations. This is done to facilitate the possibility that a temporary move to a less desirable design point might eventually lead to a better overall solution (i.e. moving out of a local minimum or maximum solution to eventually find the global one). The concept of accepting a less desirable point to eventually achieve the best overall solution can be thought of as analogous to accepting pixels that do not satisfy the similarity selection criterion but are needed to complete the object region in a segmentation process. In this manner, a formal optimization method can be mapped onto a medical image segmentation process. Most objects within a medical image do not have clearly delineated edges, and segmentation based purely on a specific range of intensity values does not produce a region that isolates the object completely. A method based on probabilities can address this issue very well. Like any heuristic method, the resulting region grown will not be 100% accurate, but will be significantly closer to completing the region when compared to deterministic methods such as conventional thresholding and region growing.

The probability of a pixel being accepted into the object region depends on two factors: 1) the distance of a pixel from a seed (spatial properties) and 2) the difference between the intensities of a pixel to a seed (pixel intensity properties).

The distance of a pixel from a seed is computed by:

$$r_{PIXEL}(x_{PIXEL}, y_{PIXEL}) = \sqrt{(x_{SEED} - x_{PIXEL})^2 + (y_{SEED} - y_{PIXEL})^2} \quad (3)$$

where x_{SEED} , and y_{SEED} , are the coordinates of the seed, and x_{PIXEL} and y_{PIXEL} are the coordinates of the current pixel being processed.

The maximum allowable value for $r_{PIXEL}(x,y)$ is r_{MAX} , which is the radius of the search region and defines the area of the image that will be inspected during the segmentation process.

A ratio of the radiuses is defined as δ_R :

$$\delta_R(x, y) = \frac{r_{PIXEL}(x, y)}{r_{MAX}}, \quad r_{PIXEL} \leq r_{MAX} \quad (4)$$

This ratio is calculated as the radius of the search region, $r_{PIXEL}(x,y)$ divided by the maximum allowable search region, r_{MAX} . This ratio describes the properties of a pixel relative to the defined search space based on its location. A pixel close to a seed will have a value close to zero, while a pixel that is far away from a seed will have a value approaching one.

Now that the spatial component of the probabilistic selection process is defined, the pixel intensity component is computed. The difference between the intensity of the seed, I_{seed} , and the intensity of the current pixel, $I(x,y)$ is defined as, $\delta_I(x,y)$:

$$\delta_I(x, y) = 1.0 - |I_{SEED} - I(x, y)| \quad (5)$$

From Equation (5), pixels with very similar intensities to the seed pixel will have a pixel intensity component value of 1.0, and this value will approach zero as the intensities continue to differ.

Using the relations for spatial and pixel intensity properties, the selection probabilities are calculated. Combining Equations (4) and (5), the probability Pr_1 is defined as:

$$Pr_1(x,y) = e^{-a} \quad (6)$$

$$a = \frac{\delta_I(x,y)}{\delta_R(x,y)} \quad (7)$$

$Pr_1(x,y)$ is the selection probability and is modeled after the Monte Carlo acceptance probability used in SA [45]. The closer the value of Pr_1 is to one, the higher the probability of the current pixel being selected into the region. When a equals zero, Pr_1 has a value of one; and as a grows larger, Pr_1 approaches a value of zero. For Pr_1 to have a value of one (high probability of selection), either δ_I is close to or equal to 1.0 (minimal difference in intensities between seed and pixel), or δ_R is close to or equal to 0.0 (pixel is close to seed).

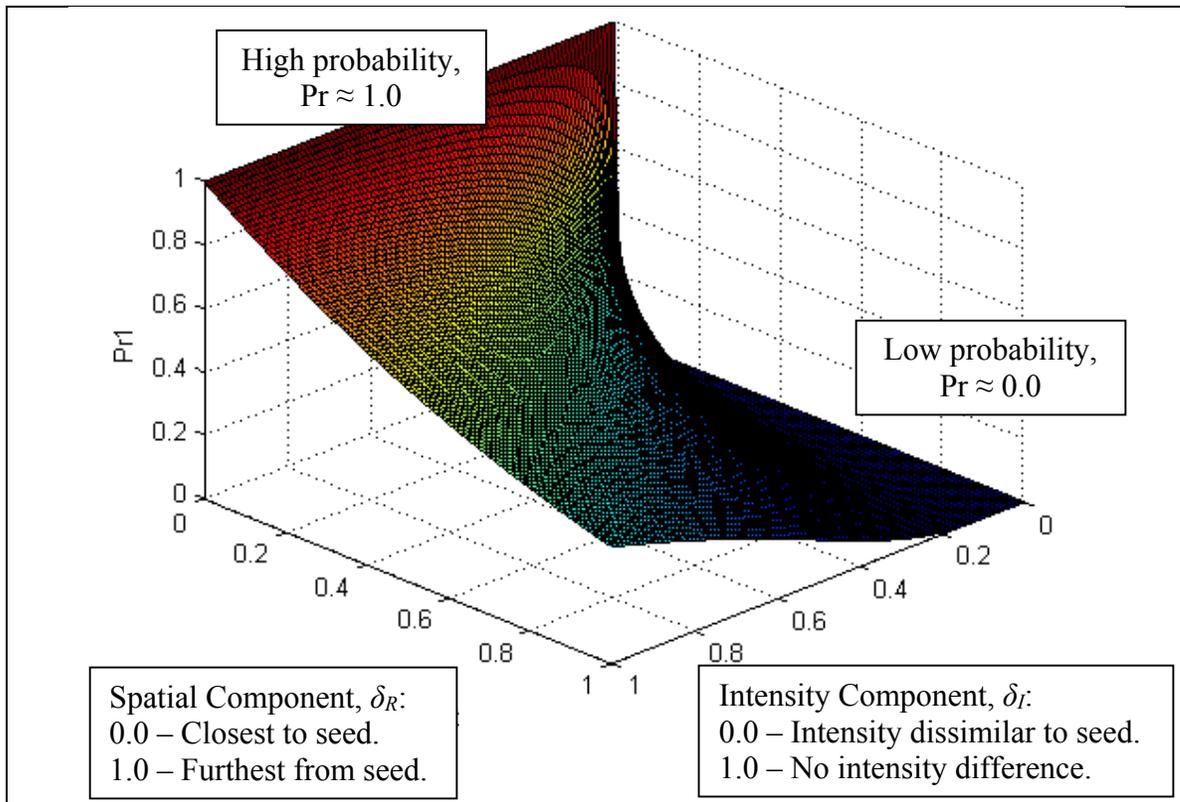


Figure 20: Values for Pr_I based on possible spatial and pixel intensity component values.

Figure 20 shows the probability distribution for different combinations of spatial component and intensity component values. When the intensity difference is small or negligible ($\delta_I(x,y) \approx 1.0$) and the pixel being evaluated is further away from the seed pixel ($\delta_R(x,y) \rightarrow 1.0$), the probability Pr_I decreases to approximately 0.368. With this behavior, as long as the intensity of the pixel being evaluated is similar to the seed pixel, the value of Pr_I will never go to zero regardless of its location. This allows the algorithm to identify disconnected regions that are actually part of the object of interest. When a pixel's intensity differs greatly from the seed pixel's ($\delta_I(x,y) \rightarrow 0.0$), the value for Pr_I reduces to zero, regardless of its current pixel location within the search region. This ensures that pixels that are close to the seed are not selected as the object simply because

of their close proximity to the seed region. Conversely, if the pixel is at the seed location, Pr_1 will equal to 1.0 regardless of the intensity difference, since any pixel at the seed's location is essentially the seed itself. However, the value of Pr_1 reduces rapidly as soon as the current pixel being evaluated is away from the seed location. This reduction in Pr_1 becomes more apparent when there is a significant difference in intensities ($\delta_I(x,y) \rightarrow 0.0$). Thus, using Equations (4) and (5) ensure that both properties are represented equally. In addition, the quantities are only dependent on the pixel being evaluated and the seed region, thus allowing discontinuous regions to be selected (e.g., multiple, disconnected tumors).

Once Pr_1 is computed, a second probability, Pr_2 , a random number between zero and one, is generated. The two probabilities, Pr_1 , and Pr_2 , are then compared as follows:

$$R(x,y) = \begin{cases} 1.0 & Pr_1 \geq Pr_2 \\ 0.0 & Pr_1 < Pr_2 \end{cases} \quad (8)$$

Pixels that are selected from the probabilistic selection criteria are allocated a value of one, indicating that they now belong to the segmented region. Pixels whose probabilities, Pr_1 , do not exceed the generated probability, Pr_2 , are allocated a value of zero. Since this step of the segmentation process is highly probabilistic, pixels that do not belong to the object will have a possibility to be selected as well. This is a tradeoff when applying probabilistic methods. However, as the test cases will demonstrate, this tradeoff does not affect the method's capability to efficiently and accurately segment an object. The amount of pixels incorrectly chosen will be negligible compared to the correct ones. When viewed they will not distort or create an incorrect representation.

4 FUZZY SEGMENTATION OF TUMORS

4.1 Fuzzy Logic

In 1965, Zadeh [57] introduced a mathematical set theory to represent vagueness or fuzzy-ness in everyday life called fuzzy logic (also called fuzzy sets). Fuzzy sets were designed to be easy and natural to implement and setup. A good example would be when making a right turn in a car. Would you tell the driver to “turn the steering wheel at five degrees increment until 92 degrees to the right, starting at three meters from the intersection”? Or would the instructions be more like, “turn the steering wheel *a little* to the right when you get *close* to the intersection”? The former (a crisp instruction) would be too precise to be carried out by any person while the latter (a fuzzy instruction) is of a more natural and intuitive to be performed based on the driver’s interpretation.

Fuzzy sets represent a degree of membership using membership functions. Inputs to the fuzzy inference system must first be fuzzified, a process of converting crisp inputs to fuzzy inputs. The output(s) from the fuzzy inference system must then be defuzzified, converting the fuzzy output into a crisp value. In a fuzzy set, a variable/object can have memberships to more than one fuzzy subset, with different degrees of membership [58]. For example, a car (object) at three meters from the intersection can have a high degree of membership (0.95) of the fuzzy subset *veryCloseToIntersection* and at the same time have a mid degree of membership (0.5) of the fuzzy subset *quiteCloseToIntersection*, and possibly even a third membership of the fuzzy subset *notCloseToIntersection* with a low degree of membership (0.1).

In order to convert these crisp inputs (distance in meters) to a fuzzy membership value, a process called fuzzification is used. After this, the fuzzy inputs are used in a fuzzy inference system consisting of a set of predefined fuzzy rules. Fuzzy rules operate based on an IF-THEN relation. Using the example of turning a car at an intersection, an appropriate fuzzy rule would be “*IF carIsVeryCloseToIntersection THEN turnSteeringSlowlyToRight*”. Using these fuzzy inputs and fuzzy rules, the fuzzy output can be calculated and then converted to a crisp output value using a defuzzification process, which converts the fuzzy output degree of membership to the fuzzy subset *turnSteeringSlowlyToRight* into an actual value in degrees to turn the steering wheel. This entire process from input fuzzification to using the fuzzy rules, and then defuzzification for output(s) can be collectively called a fuzzy inference system.

4.1.1 Fuzzy Inference System Example

The following is an example of the fuzzy inference system, where the situation is to calculate the tip for a meal. Figure 21 illustrates the fuzzification of crisp input values, the application of the fuzzy rules to generate a fuzzy output, and finally the defuzzification process to obtain a resulting crisp output value. The tip as a percentage of the total bill depends on two criteria, the service at the restaurant and the taste of the food. If the service is poor or the food is bad, then the tip will be cheap. If the service is great or if the food is delicious, then the tip will be great. Scores between 0 and 10 are assigned to each criterion. However, in order to quantify the definition of poor service or delicious food, a fuzzy inference system is used.

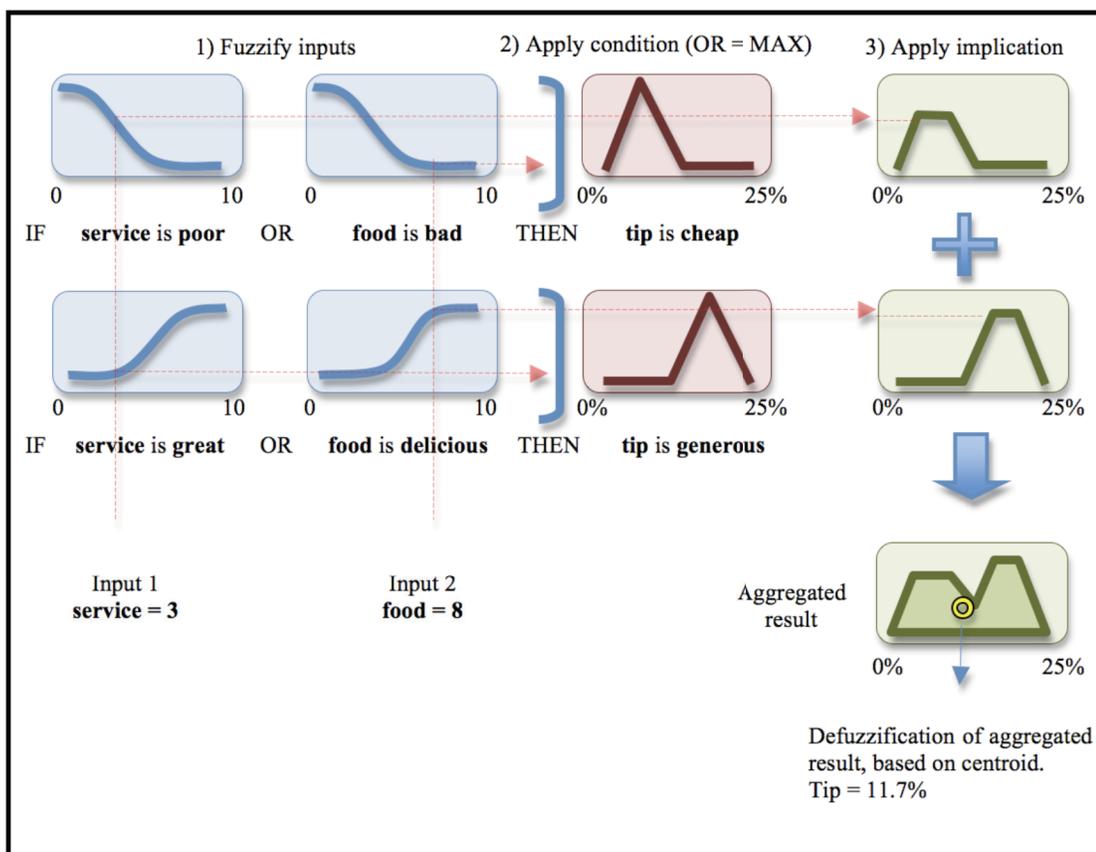


Figure 21: Fuzzy inference system for tip calculations based on service and food.

Suppose the “score” for food is eight, to what extent is it delicious? The membership functions for great service and poor service, delicious food and bad food are in blue, at the first two columns respectively. In all membership functions, the x-axis represents the crisp input/output value and the y-axis represents the degree of membership. These membership functions will fuzzify the crisp input values. Using the membership functions, a score of eight for food is fuzzified to a 0.85 membership of being delicious. The same goes for the service, with a score of three can be fuzzified to a 0.05 membership for great and also at the same time a 0.6 membership for poor.

The two fuzzy rules are defined as 1) IF service is poor OR food is bad THEN tip is cheap, and 2) IF service is great OR food is delicious THEN tip is generous. A logical operator (AND/OR) is applied to the multiple inputs, and in the case of this example, the logical operator is OR. Both inputs are evaluated and the OR operator applies the condition by taking the maximum degree of membership values for all inputs. The “implication” stage takes into account only the area of the output membership function below the maximum degree of membership. This is done for all the rules in the inference system and combined to generate an aggregated result output. The centroid of the output is calculated and the x-axis value of the centroid represents the crisp output value for the fuzz inference system.

4.2 Introduction

Figure 22 summarizes the second segmentation process which uses a fuzzy rule based system. The user initializes the process by selecting a region of interest (ROI), and inputs to the fuzzy system are generated based on properties of the ROI. The segmentation of the image is performed based on the output of the fuzzy system, and this process is repeated for every slice in the data set.

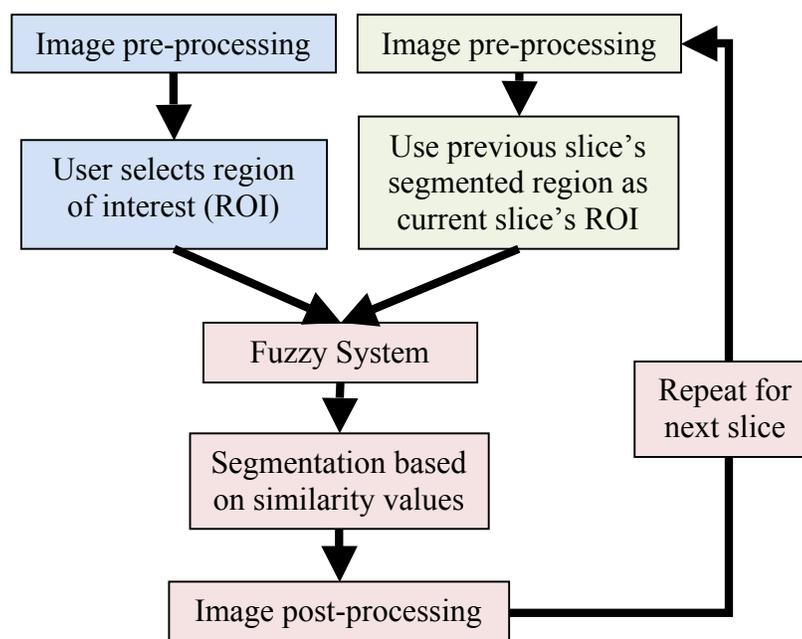


Figure 22: Flowchart of segmentation process using the fuzzy rule-based system.

4.3 Pre-processing

In order to convert the image data into actual pixel intensities, a windowing process is applied. Apart from formatting the image data into standard image pixels, this process also allowed for contrast and brightness adjustments to the image data to improve identification of the tumor. The windowing parameters used are as provided by the radiologist, but can be manually set if needed by the user. Once windowing of the image data is performed, pre-processing of the data is required if it is too noisy from acquisition imperfections. To remove noise from an image, a Gaussian filter with a 3×3 pixels mask was used as a smoothing filter.

4.4 Initialization

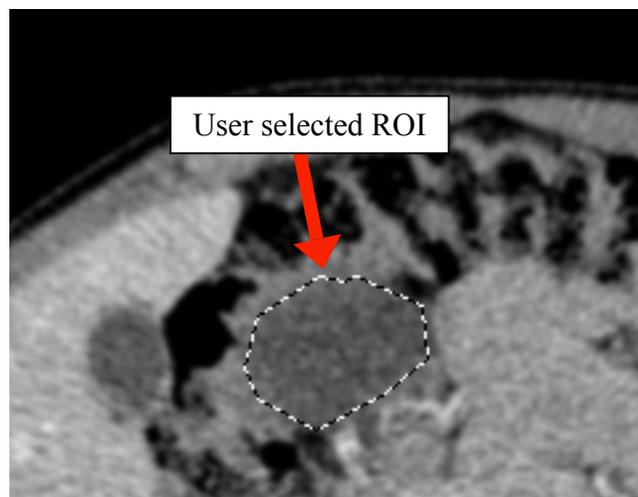


Figure 23: Illustration of the region of interest (ROI) selected by the user in the first slice of the image data set.

The segmentation process is initialized by having the user select the ROI inside the tumor (object) on the first slice in the dataset, as shown in Figure 23. For optimal performance of the algorithm, the ROI selected should be as similar to the tumor as possible since this is the only *a priori* information provided to the algorithm. If the ROI selected by the user is too small, the resulting segmented region for the first slice will suffer from under segmentation where the edges of the tumor will not be segmented completely. If the ROI selected is too large, then over segmentation will occur where the segmented region will also include the non-tumor pixels surrounding the actual tumor.

For subsequent slices in the dataset, the resulting segmented region from the previous slice is mapped as the current slice's new ROI. By doing so, the method will be able to automatically update the shape and size of the ROI as the object changes its shape

and size in each consecutive slice. This allows segmentation of a three-dimensional dataset to be performed from user information based on only the first slice.

4.5 Inputs and Output of the Fuzzy System

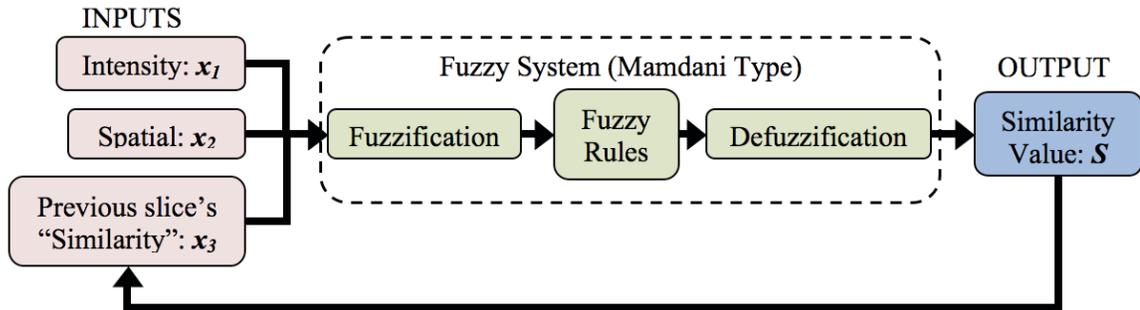


Figure 24: Fuzzy system with three inputs and a single output.

A Mamdani type fuzzy system [115] was constructed with three inputs, shown in Figure 24. A single output calculated is based on eight fuzzy rules. The three inputs are calculated from properties of the pixels inside the ROI. All inputs will be between 0.0 and 1.0, where the extremes represent the least desirable value (0.0) and the most desirable value (1.0) and the degree of memberships are calculated using a set of membership functions that are adaptive to each different slice in the image dataset.

The first input x_1 , is based on the intensity properties of the region, and is a measure of the difference in the current pixel's intensity to the mean intensity of the pixels inside the ROI. Pixels with minimal intensity difference to the mean intensity will have x_1 values close to 1.0, whereas pixels showing a significant intensity difference will have x_1 values close to 0.0. This can be expressed as:

$$x_1 = 1.0 - |I_{CURRENT} - I_{MEAN}| \quad (9)$$

where $I_{CURRENT}$ and I_{MEAN} are pixel intensities of the current pixel and the mean of the ROI respectively, with values between 0.0 (black) and 1.0 (white).

The second component x_2 is based on the spatial properties of the pixel, where the distance from the current pixel to the centroid of the ROI is computed. The further a pixel is from the centroid of the ROI, the smaller the value of x_2 , and the closer the value to 1.0. This relationship is expressed as:

$$x_2 = 1.0 - \left(\frac{(x_i - x_c)^2 + (y_i - y_c)^2}{R_{MAX}^2} \right) \quad (10)$$

where (x_i, y_i) : coordinates of the current pixel
 (x_c, y_c) : coordinates of the ROI's centroid
 R_{MAX} : maximum search radius

R_{MAX} is the radius of the search region and only pixels within the search region will be evaluated. Any pixels outside of this radius will have a distance component of zero. The new search region is defined based on the object segmented in the previous slice. In the previous slice, the smallest circle required to completely encompass the object segmented is calculated and used as the search region for the current slice. Again, the running average of the change in size of the circle from the last three images is calculated and used to shrink/grow the new search region. The average size change is defined as:

$$\bar{C} = \frac{3C_1 + 2C_2 + C_3}{6} \quad (11)$$

where C_1 , C_2 , and C_3 represent the change in the segmented region size for the previous slice, 2nd previous slice, and 3rd previous slice, respectively.

The third and last input x_3 is the similarity output, S , from the previous slice. Since there is no reference to a previous slice in the first slice of the dataset, a default value is assigned. So for the first slice only, based on the selection of the ROI by the user, pixels inside the selected ROI will have a similarity of 0.75, while pixels not selected (outside the ROI) will have a similarity of 0.25. This is based on the assumption that if a pixel was selected as the object in the previous slice or at least has a high similarity value, that pixel should be given some degree of advantage to be selected as part of the object in the current slice. This propagation system allows information from the previous slice to be passed on and used in the current slice. It should be noted that even if the pixel has a high similarity value in the previous slice, the resulting similarity output for that pixel in the current slice could be low if the other two inputs do not satisfy the requirements of the fuzzy rules.

4.6 Fuzzification

Based on these three inputs, the fuzzification process generates membership values from a set of adaptive membership functions. The three inputs are fuzzified in the fuzzy system using specific S-shaped spline membership functions. An example is illustrated in Figure 25. Fuzzification converts these crisp inputs to membership values within a fuzzy set. All three inputs' fuzzy sets have the same subsets and are defined as $\{LOW, HIGH\}$, where the fuzzy subset LOW describes the particular input as having a low or insignificant value, while the fuzzy subset $HIGH$ describes the input as having a

high or significant value. The only difference between these fuzzy sets is the way the subsets' membership functions are defined for each input.

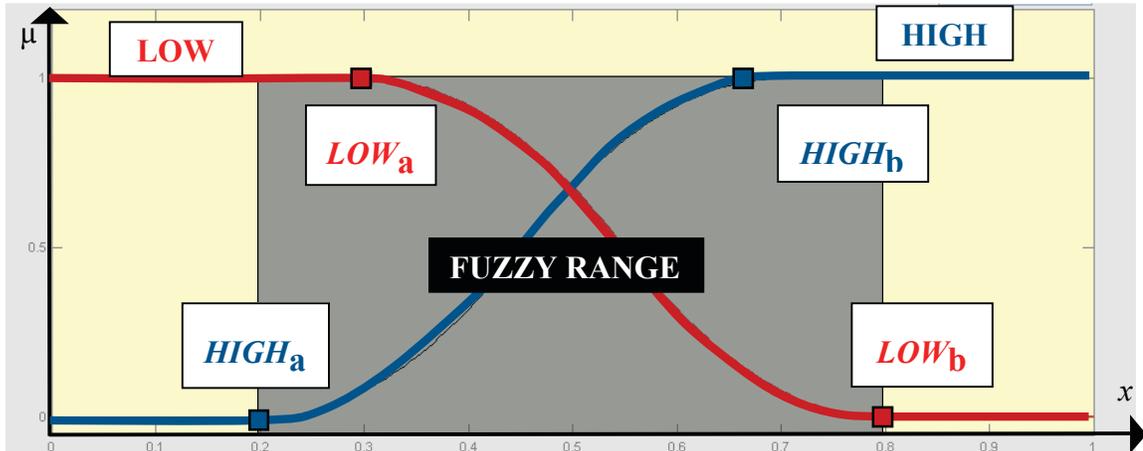


Figure 25: Example of a S-shaped membership function for a particular input.

Referring to the membership function for the subset *LOW* (in red) in Figure 25, if input x is less than a , the membership value is constant at 1.0. The membership value starts to decrease when x is larger than a until x becomes greater than b . At this point onwards the membership value remains constant at 0.0. The case is the opposite for the membership function of the subset *HIGH* (in blue), where it starts out at 0.0 and increases to 1.0 between a and b .

The fuzzy range highlighted is where the value of b for *LOW* is greater than the value a for *HIGH*. This forms an overlap between the two subsets and it is because of this overlap that the fuzzy inputs can have both *LOW* and *HIGH* membership with different degrees of membership (non-zero). For example, based on the membership function illustrated in Figure 25, if input x has a value of 0.65, it is *LOW* with 0.125 degree of membership, but at the same time, it is also *HIGH* with 0.875 degree of membership. The membership functions for each fuzzy subset can be expressed as follows:

$$\mu_{LOW} = \begin{cases} 1 & x \leq a \\ 1 - 2 \left[\frac{x-a}{b-a} \right]^2 & a < x < \frac{a+b}{2} \\ 2 \left[\frac{b-x}{b-a} \right]^2 & \frac{a+b}{2} \leq x < b \\ 0 & b \leq x \end{cases} \quad (12)$$

$$\mu_{HIGH} = \begin{cases} 0 & x \leq a \\ 2 \left[\frac{x-a}{b-a} \right]^2 & a < x < \frac{a+b}{2} \\ 1 - 2 \left[\frac{b-x}{b-a} \right]^2 & \frac{a+b}{2} \leq x < b \\ 1 & b \leq x \end{cases} \quad (13)$$

By making the values of a and b variable at every slice, the membership functions are then adaptive to the changes in the properties of the object being segmented (such as pixel intensity, size, shape, and location) within the current image. The values for a and b are defined in Table 2.

Table 2: Fuzzy range limits of membership functions.

Membership Function	a	b
$\mu_{LOW}(x_1)$	$1.1 \times \min(x_1)$	$1.1 \times \text{mean}(x_1)$
$\mu_{HIGH}(x_1)$	$0.9 \times \text{mean}(x_1)$	$0.9 \times \max(x_1)$
$\mu_{LOW}(x_2)$	0.1	0.5
$\mu_{HIGH}(x_2)$	0.25	0.9
$\mu_{LOW}(x_3)$	$1.1 \times \min(x_3)$	$1.1 \times \text{mean}(x_3)$
$\mu_{HIGH}(x_3)$	$0.9 \times \text{mean}(x_3)$	$0.9 \times \max(x_3)$

Two membership functions control the fuzzification of the first input, $\mu_{LOW}(x_1)$ for the fuzzy subset *LOW* and $\mu_{HIGH}(x_1)$ for the fuzzy subset *HIGH*. From Table 2, $\mu_{LOW}(x_1)$ has the limit a defined as the minimum of the previous slice's x_1 values with an additional 10% buffer, and the limit b is defined as the mean of previous slice's x_1 values, also with an additional 10% buffer. The value of a 10% buffer is an ad-hoc default value determined through the test cases run, but can be set to any value. The other membership function, $\mu_{HIGH}(x_1)$ has the limit a defined as the mean of the previous slice's x_1 values with a reduction of a 10% buffer, and the limit b is defined as the maximum of the previous slice's x_1 values, also with a reduction of a 10% buffer. This ensures a fuzzy range with a width of 20% of the mean, centered at the mean. The fuzzy range for x_2 is defined as between 0.75 of R_{ROI} and 1.25 of R_{ROI} . The fuzzy subset *LOW* membership function, $\mu_{LOW}(x_2)$ has limits a and b at 0.1 and 0.5 respectively and the fuzzy subset *HIGH* membership function $\mu_{HIGH}(x_2)$ has limits a and b at 0.25 and 0.9 respectively. The limits for x_2 are constant because x_2 is already defined in terms of R_{MAX} , which changes based on the size of the current ROI. The fuzzy range of the third input x_3 is defined in a similar fashion to the fuzzy range of the first input x_1 , using a 10% buffer.

4.7 Fuzzy Rules

With three inputs (x_1 , x_2 , and x_3) and each input having two possible descriptions (*LOW* or *HIGH*), this gives us eight possible input combinations and hence eight rules to address each combination. These rules are listed in Table 3.

Table 3: Fuzzy rules for the fuzzy inference system.

Rule	IF x_1	AND x_2	AND x_3	THEN y
1	High	High	High	Object
2	High	High	Low	Object
3	High	Low	High	Object
4	High	Low	Low	Background
5	Low	High	High	Object
6	Low	High	Low	Background
7	Low	Low	High	Background
8	Low	Low	Low	Background

The output fuzzy set, y , is defined as either *BACKGROUND* or *OBJECT*. From the rules defined in Table 3, if at least two of the inputs are *HIGH*, then the output will be the *OBJECT*, otherwise the output will be the *BACKGROUND*. These rules also ensure that even if a pixel has a high similarity value in the previous slice, it will be assigned a low similarity value in the current slice if both its intensity and distance components are low.

Based on the inputs to the fuzzy inference system, an output fuzzy set $\mu(y_i)$ is generated, where i represents the rule number defined. Using the general approach of *MIN-MAX* aggregation, all of these output fuzzy sets are aggregated into a single output fuzzy set $\mu(y)$.

4.8 Defuzzification

A defuzzification process is performed to convert the output fuzzy set into a crisp output of similarity values, S . This is done by using the general center-of-area method

shown in Equation (14), to calculate the centroid of the area defined by the single output fuzzy set $\mu(y)$.

$$S = \frac{\int \mu(y) \cdot y \cdot dy}{\int \mu(y) \cdot dy} \quad (14)$$

4.9 Tumor Segmentation

The segmented region, R , is obtained by evaluating the entire image, determining the maximum similarity value, S_{MAX} , and selecting pixels with similarity values equal to S_{MAX} .

$$R(x,y) = \begin{cases} 1, & S(x,y) = S_{MAX} \\ 0, & S(x,y) \neq S_{MAX} \end{cases} \quad (15)$$

Where x and y are the coordinates of the current pixel being evaluated.

As shown in Equation (15), only pixels with similarity values equivalent to the maximum similarity value detected in the current image are segmented to be the object. A specific value is not defined since the similarity value obtained will differ in every image of the study due to the adaptive nature of the method. By only accepting pixels with the maximum similarity value, the method assures that only pixels that best describe the object of interest are segmented.

4.10 Post-processing

Additional post-processing steps using basic morphological operations are performed to eliminate segmentation noise such as disconnected pixels from the object that create stray pixels and unselected pixels that create ‘gaps’ within the segmented object. Stray pixels are first removed by the morphological process of pixel erosion and then gaps are filled to be part of the object using the morphological process of pixel dilation.

By default, these post-processing morphological routines are performed using a 3×3 pixel mask. The process of erosion operates by evaluating each pixel (the center) and the surrounding eight other pixels. If all of the surrounding eight pixels are not selected, then the center pixel will be set as the background. The process of dilation is the opposite of erosion, where if all of the surrounding eight pixels are selected as the tumor, then the center pixel is selected and set as a tumor pixel as well. Both these processes are performed for every pixel in the 2D image. Thus, any stray pixels and gaps in the segmented object smaller than the mask will be eliminated. In addition to making the final segmented result more visually presentable, this process also helps to reduce the classification error of the results.

4.11 Multi-Window Segmentation

In addition, the method described can also be run on multiple window settings. Different tissue densities may appear more pronounced under different window settings. Thus, in order to capture a wider range of tumor tissues, the same segmentation

procedure is performed multiple times under different window settings. The user, selecting windowing settings where the tumor is visually identifiable at varying intensities, determines the variation in the window settings manually. The segmented regions based on three different window settings are combined to generate a final segmented region, R .

$$R_{Total} = R_{Window1} + R_{Window2} + R_{Window3} \quad (16)$$

$$R_{Final} = \begin{cases} 1, & R_{Total} \geq 2 \\ 0, & R_{Total} < 2 \end{cases} \quad (17)$$

where $R_{Window1}$, $R_{Window2}$, and $R_{Window3}$ are the segmentation results of using three different windowing settings, and R_{Final} is the resulting segmented region. Additional morphological routines to remove any noise artifacts such as stray pixels are then performed.

5 ISIS: INTERACTIVE IMMERSIVE VISUALIZATION OF MEDICAL PATIENT DATA

5.1 Introduction

Isis, named after the Egyptian goddess of healing, is an interactive application developed to visualize patient medical data in a virtual environment. The principle during the development of Isis is that it will be a medical image data viewer designed for non-radiologists with a low learning curve.

The outline of the software framework is illustrated in Figure 26. The input of medical image data can be in two different formats: 1) any DICOM standard formatted medical image data or 2) any three-dimensional volume image data stored as 8-bit pixel intensities (between 0 and 255). The latter format is to accommodate the visualization of non-DICOM formatted image data, such as the segmented tumor results from the segmentation methods discussed.

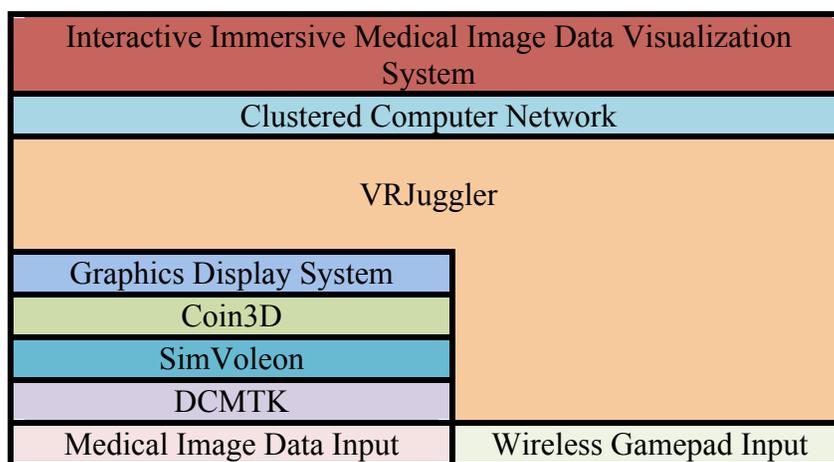


Figure 26: Software framework for the interactive immersive visualization system.

The input DICOM formatted medical image data is processed using the open source library DCMTK: DICOM Toolkit [116]. Based on the information in the DICOM headers, the slice location of each slice and the pixel spacing values are obtained. This will ensure a correct representation of the image data in accurately scaled proportions. In order to display the image data on a scalable platform, from a single desktop display to a multi-tiled immersive display, VRJuggler [117] was utilized for immersive visualization. Coin3D [118] was used for scenegraph handling of objects/models in the virtual environment, while SimVoleon [119] performs the volume rendering of the medical image data. User interaction with the system is handled using a wireless gamepad, where the joysticks are used for rotation and translation of the objects, while the button presses perform different functions depending on the current mode of the system.

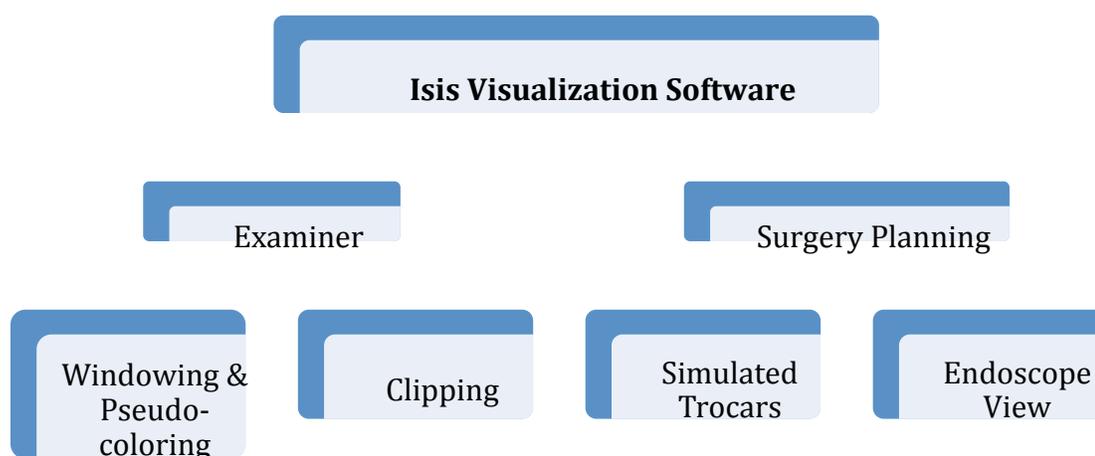


Figure 27: Outline of the features of the Isis immersive visualization system.

The software may be used in either examiner or surgical planning mode as shown in Figure 27. The examiner category contains features that will allow the user to explore the dataset using the gamepad as an interaction input device. Here the user can rotate and translate the 3D model of the image data. Pseudo-coloring allows the user to change the color scheme of the volume model to highlight different details within the dataset. Users can perform windowing on a DICOM formatted image dataset to focus on tissues of a desired density range and the multiple clipping planes allow the user to slice into the 3D model at any angle for an unobstructed view of a specific region of interest. The surgical planning mode allows inclusion of multiple simulated trocars that the user can manipulate and a simulated endoscope view for use in minimally invasive procedure planning.

5.2 Gamepad and Menu Interface

The main input device to interact with the application is a wireless gamepad. The gamepad controls are showed in detail in Figure 28, where the gamepad interface consists of two joysticks, a directional pad (with up/down, left/right arrow buttons), and four buttons on the right side of the gamepad. There are four additional buttons on the front of the gamepad, shown in Figure 28b. Buttons #5 and #7 are located at the left front side of the gamepad, while buttons #6 and #8 are located at the right front side of the gamepad.

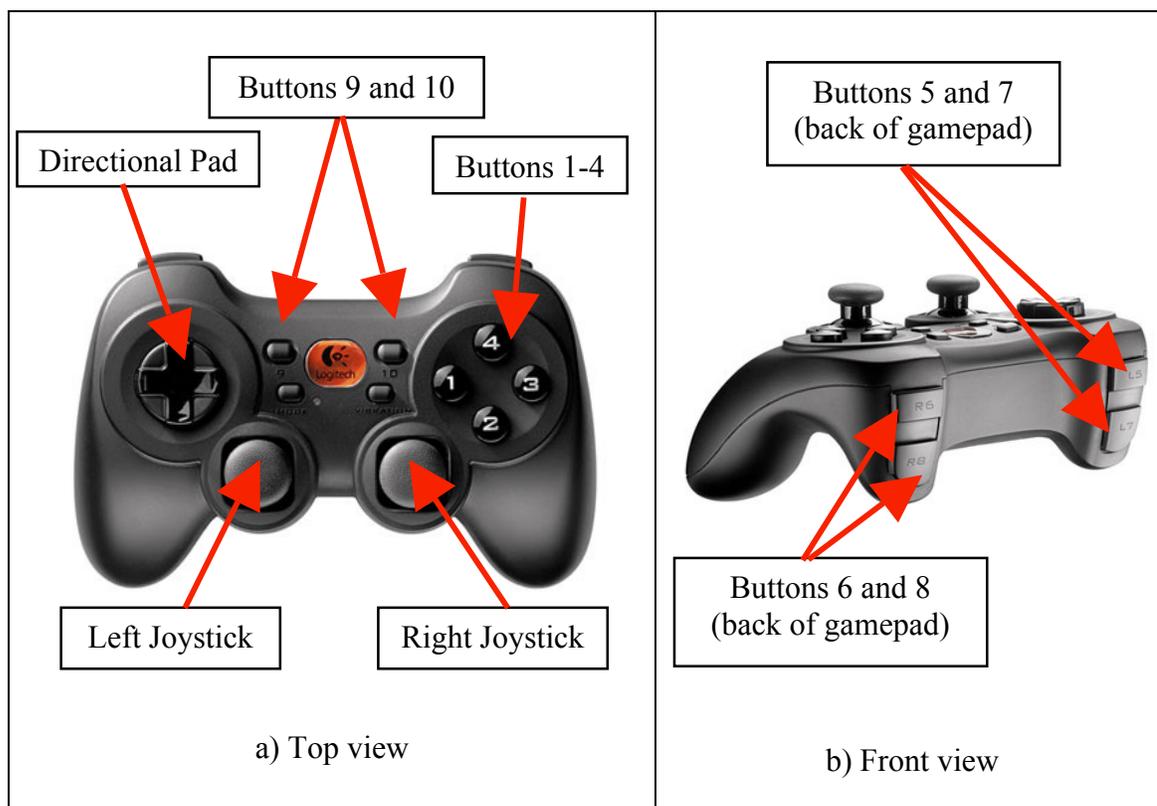


Figure 28: Buttons and joystick layout of wireless gamepad.

The joysticks handle the rotation and translation of objects, while the buttons perform specific functions based on the current mode that the application is in. For example, button #2 will turn on clipping planes when the application is in clipping mode, but will recalculate the pixel values based on the set windowing parameters in windowing mode. Figure 29 shows a user interacting with the application in a four-wall VR projection system with multiple patient data sets loaded.

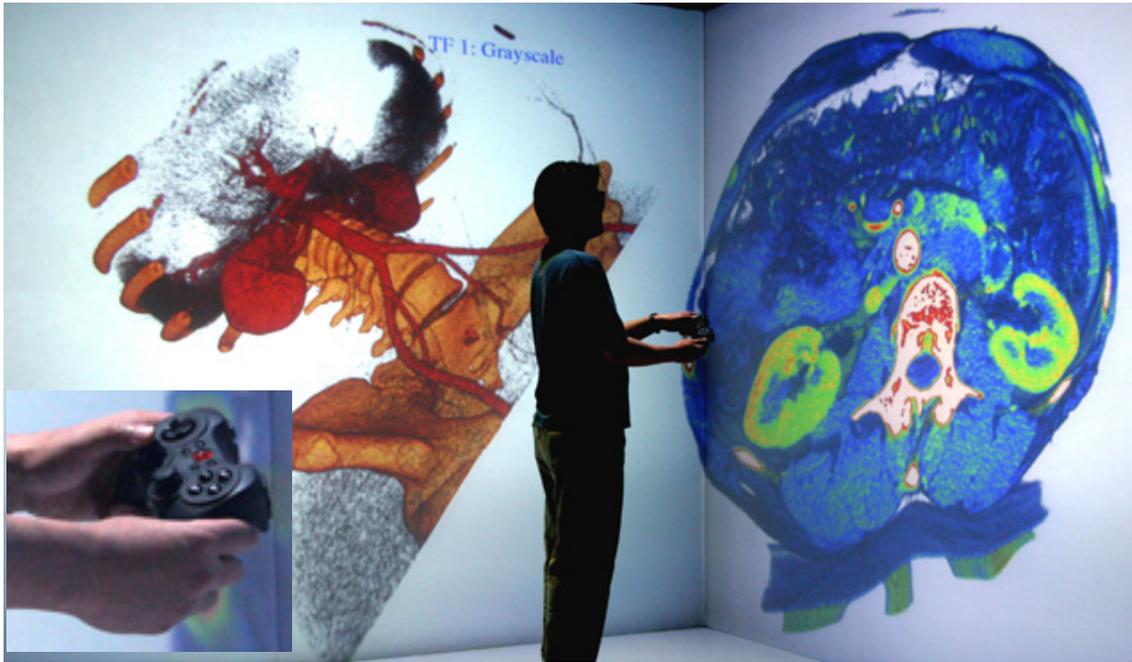


Figure 29: Application in a four-walled VR projection system with wireless gamepad as interaction device (inset).

The features have been categorized into four separate modes: 1) default/standard mode, 2) windowing mode, 3) clipping mode, and 4) surgery planning mode. The different modes of the system are outlined in this chapter, discussing the system controls where some button presses will perform different actions depending on the current mode that the system is in. The left joystick, directional pad, and Buttons #2, #7, #8, and #9 are the only controls that change depending on the current mode that the application is in and unless stated otherwise, the buttons/joysticks will perform as defined for the standard/default mode when the application is in any of the four defined modes.

With so many different features and functionalities, there are several ways that a user can interact with Isis. A menu interface has been designed to ensure that the user has a smooth experience when navigating and controlling the application. In addition,

shortcut buttons have also been programmed into the application for the more experienced users who are familiar with the controls. During the entire interface design process, much effort has gone to incorporate Norman's and Shneiderman's rules of interface design such as standardized controls with natural mappings, informative feedback, and several automated sequences to minimize user errors.

There are currently four menu tabs available: 1) Windowing, 2) Clipping, 3) Surgery planning, and 4) Information, and each menu corresponds to a specific mode within the application. Table 4 lists the corresponding menu tab that will appear depending on what mode the application is in when the menu is activated.

Table 4: Mapping of application features to a specific menu tab.

Feature	Corresponding menu tab
Default/standard	Information
Windowing	Windowing
Clipping	Clipping
Surgery planning	Surgery planning

The general layout of a menu is explained in Figure 30. On the top of the menu is the title of the current menu, and directly below it are the titles of the menu before and after the current menu. The red arrow icon for the previous menu corresponds to how that menu can be accessed by pressing the red button on the gamepad, and the green arrow icon represents that the next menu can be accessed by pressing the green button. This gives the user a sense of order for the various menus for fast access. For each specific menu, an information box at the bottom of the menu provides the user with a brief description of the current menu or instructions on how to activate specific actions. The

majority of the real estate of the menu interface consists of option lines with the current limit being seven. A maximum of nine lines could be fitted on the menu if needed. Each option line will have a description and an action button, and the options will change according to the current menu displayed.

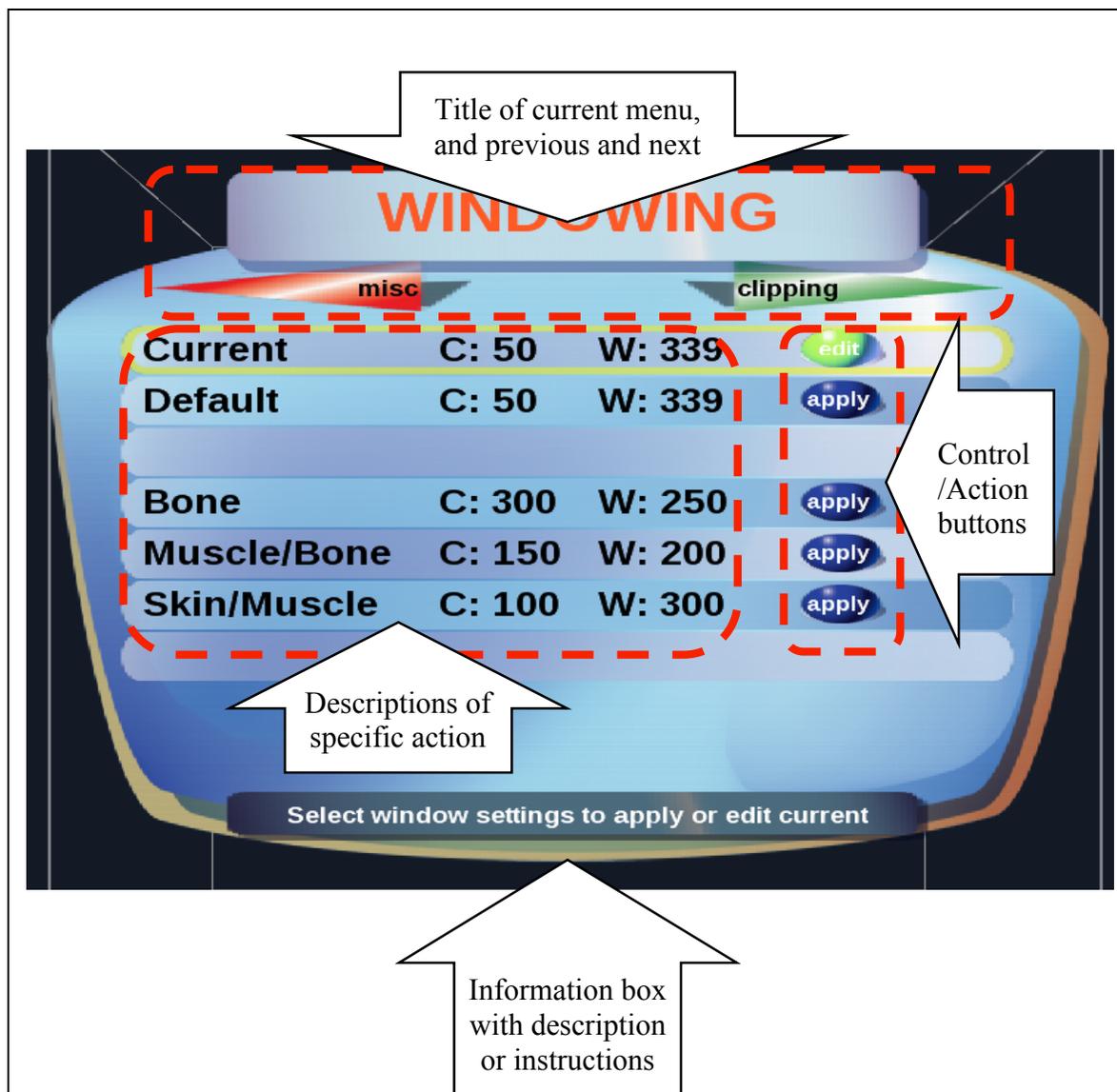


Figure 30: Layout of menu interface.

5.3 Default Mode

Table 5: Gamepad controls for the standard/default mode.

Gamepad controls	Function
Left Joystick	Translate model (forward/backward, left/right)
Right Joystick	Rotate model
Directional Pad	Translate model (up/down, left/right)
Button #1	Go to previous mode
Button #2	N/A
Button #3	Go to next mode
Button #4	Hide/unhide current selected model
Button #5	Go to previous transfer function
Button #6	Go to next transfer function
Button #7	Select previous model
Button #8	Select next model
Button #9	Reset position of current selected model
Button #10	Exit

When in the standard/default mode, the gamepad controls are as shown in Table 5. The joysticks and the directional pad will allow the user to rotate and translate the current selected model (if more than one dataset is loaded). Buttons #1 and #3 will cycle between the four modes mode of the application, while Button #4 hides or unhides the current selected model. Button #2 is reserved for activating specific functionalities such as recalculating the voxel values based on new windowing parameters when in windowing mode, or turning the clipping planes on and off when in clipping mode. Button #2 in default mode is just a dummy button which does nothing since there are no special functionalities when in default mode, but will serve as the “activation button” in other modes. Buttons #5 and #6 cycles between the different transfer functions loaded into the application to change the color scheme used for pseudo-coloring. Buttons #1 - #6

are the only buttons that will perform the same functionality regardless of the current mode that the application is in.

Buttons #7 and #8 have been designated as the “selection buttons” to cycle between the objects (dataset models, clipping planes, or trocars). In default mode, these buttons will simply cycle between the dataset models loaded into the system. Button #9 serves as the “reset button” and will reset specific parameters to its initial state, and in default mode this button resets the position and rotation of the current selected model to the origin. And finally, Button #10 quits the application.

5.3.1 Information Menu

When the menu is activated in “Default” mode, it presents the user with selected information about the patient and the dataset read in directly from the DICOM file as shown in Figure 31. Since there is a variety of information embedded in the DICOM file, only selected pertinent information is displayed, such as:

- Patient’s information: Age and sex (Name is not displayed)
- Modality of dataset (CT or MRI)
- Dataset’s resolution (number of rows × number of columns × number of slices)
- Dataset’s acquisition date
- Manufacturer of scanning machine
- Attending institution (hospital or clinic name)
- If segmented tumor model (from segmentation methods) is available or not

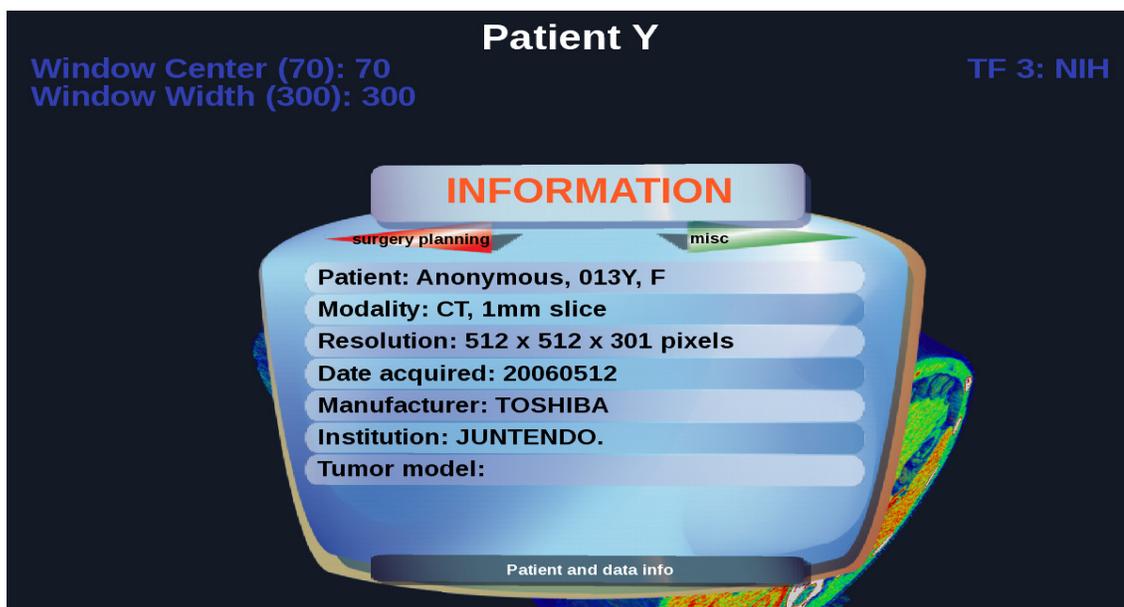


Figure 31: Displaying selected information about the patient and the image dataset.

The number of slices in the dataset and whether a tumor model is available or not are input parameters to the application, while all other information is obtained at runtime and read in directly from the DICOM dataset.

5.4 Windowing and Pseudo-coloring

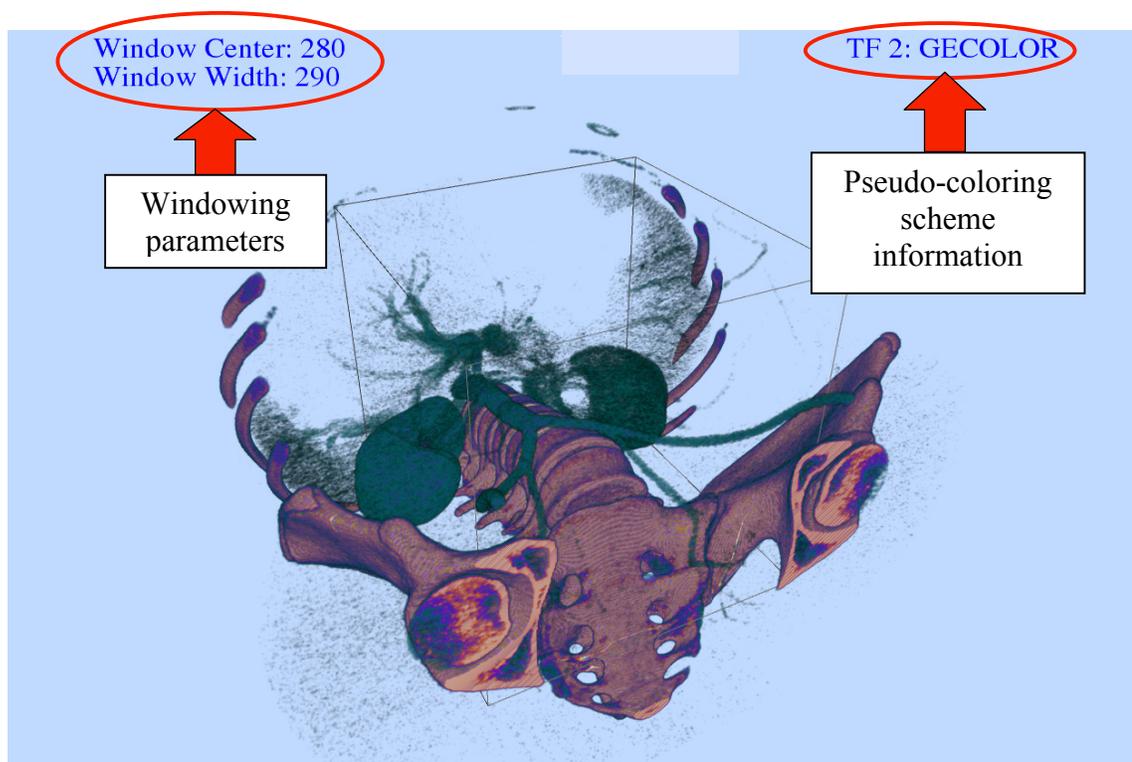


Figure 32: Windowing and pseudo-coloring features, settings of these features (circled in red) are viewable to the user within the system.

Figure 32 shows the values of the window center and width displayed to the user (left) and the current color scheme in use (right). As discussed in Chapter 1, CT/MRI data are stored in terms of tissue densities and windowing is a process that converts a specific range of these densities into 8-bit pixel data. This process has been implemented into the visualization system with direct user interaction via the wireless gamepad. Windowing parameters can be set using directional (arrow) buttons on the gamepad. The up arrow increases the window center value and the down arrow decreases the window center value, while the left arrow decreases the window width value and the right arrow

increases the window width value. Once the windowing parameters are set, the 3D representation can be regenerated based on the new windowing parameters, and depending on the size of the image dataset processed, regeneration of the displayed data can take between 1-5 seconds.

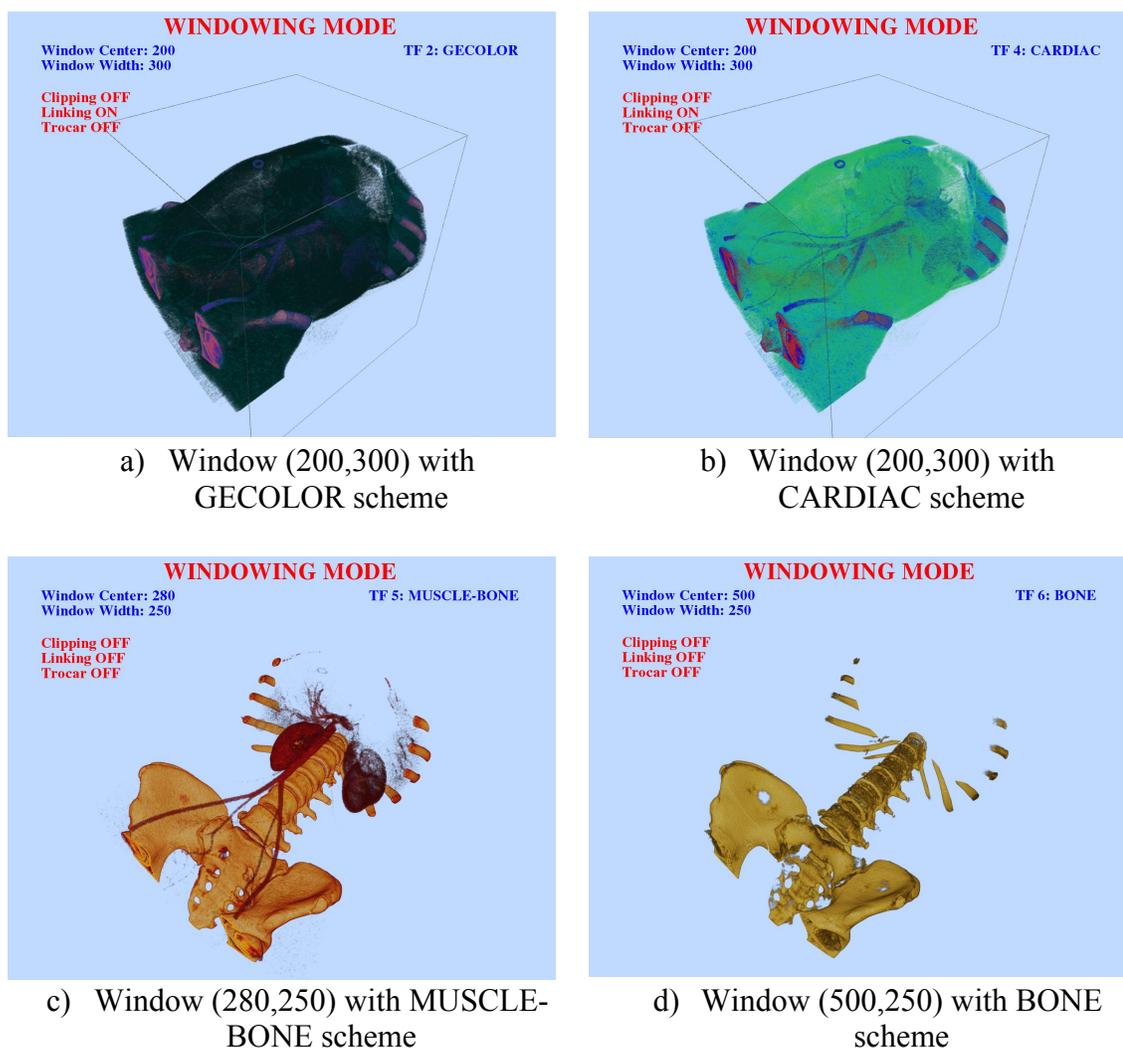


Figure 33: Windowing for display of specific tissue densities and pseudo-coloring using Color Look-up Tables (CLUT)

A set of Color Look-up Tables (CLUT) is created to enable pseudo-coloring of the image data. The application will be able to assign different coloring schemes based on the original pixel intensity after windowing. These features help enhance details within the image data. Different colors and opacities can highlight organs or tissues that are of interest to the user. Examples of the windowing pseudo-coloring process are shown in Figure 33. When assigning pseudo-colors to the model, the system reads in a text file containing the values for the CLUT, which is a 256 x 4 array containing the red, green, blue, and transparency (RGBA) values for every possible 8-bit intensity value. Based on these values, a voxel of a specific intensity (between 0 and 255) will be assigned its corresponding color and transparency value.

5.4.1 Windowing Menu

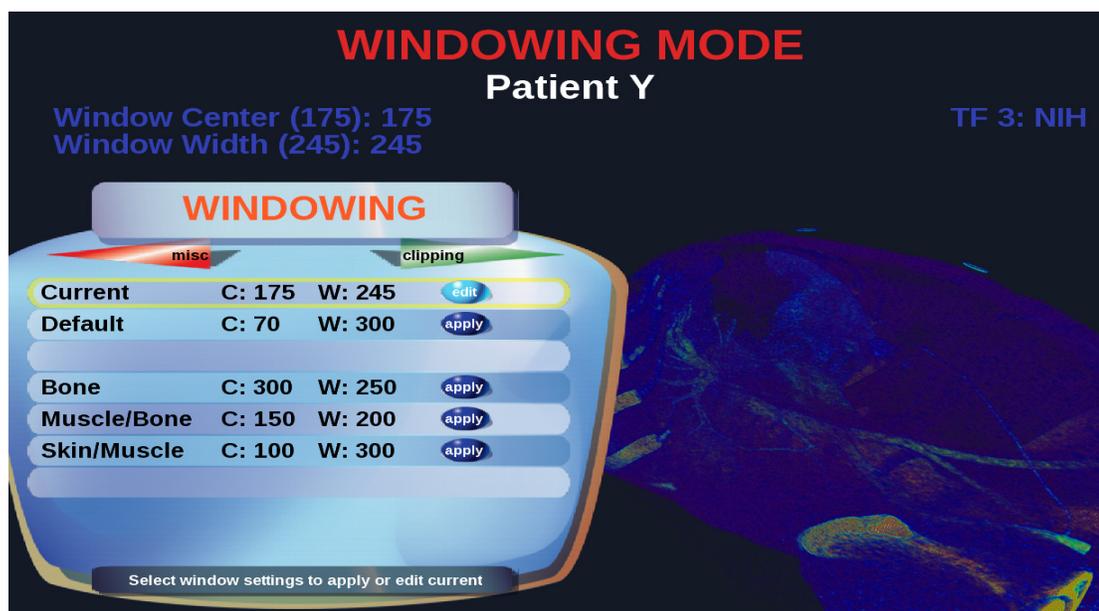


Figure 34: Windowing menu, editing current windowing parameters for manual input.

Users can have access to the windowing feature via the menu interface as well as shortcut button presses. Figures 34 and 35 show the menu interface for windowing, with a variety of preset values for bone, muscle and bone, and skin and muscle.

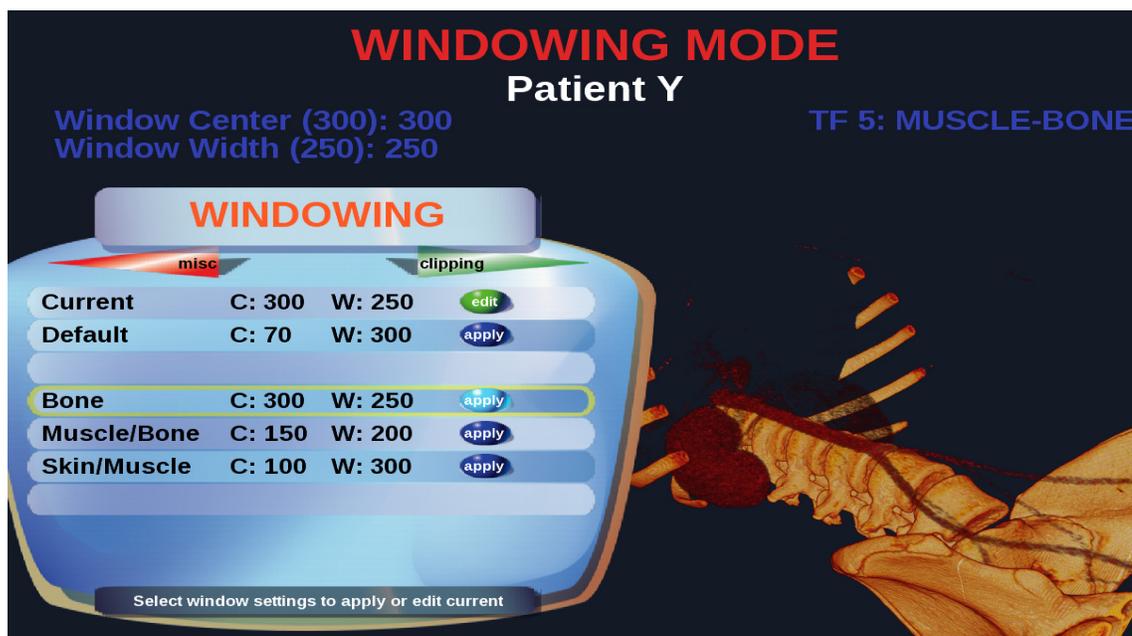


Figure 35: Predefined bone window available as an option in menu.

By using the directional pad (up/down), users can select a specific windowing preset (highlighted by a yellow outline). Pressing Button #2 (activation button) will prompt the application to recalculate and redraw the voxels using the selected windowing preset values. Users can also revert to the default value of windowing parameters as set in the DICOM file or manually edit the current values. When Button #2 is pressed the “edit” function is activated, the up and down arrows on the directional pad will increase or decrease the window center value, while the left and right arrows on the directional pad increases or decreases the window width value. Pressing Button #2 again will apply the new windowing values.

5.4.2 Gamepad Controls and Shortcuts for Windowing Feature

When the application is in windowing mode and the menu is not active, the shortcut controls of the gamepad changes to those shown in Table 6. Similar to manually editing the windowing values in the menu, the directional pad changes the values for the window center and width. Pressing Button #2 recalculates the voxels displayed based on the set window center and width values, while Button #9 resets the windowing parameters to the default values as obtained from the DICOM file. The left joystick, Button #7, and Button #8 remain the same as their functionality in default mode.

Table 6: Gamepad controls for the windowing mode.

Gamepad controls	Function
Left Joystick	Translate model (forward/backward, left/right)
Directional Pad (up/down)	Change window center
Directional Pad (left/right)	Change window width
Button #2	Regenerate voxels based on window values
Button #7	Select previous model
Button #8	Select next model
Button #9	Reset windowing center and width

5.5 Multiple Clipping Planes

Multiple clipping planes can be activated to ‘slice’ into the patient data for a detailed view ‘inside’ the patient as shown in Figure 36. By default, three clippings will be loaded when the clipping mode is activated. The number of clipping planes can be increased or decreased at runtime of the software. Using button presses on the gamepad interface, the user can cycle between all three clipping planes and the joystick allows the user to rotate the clipping plane to any desired angle. A red border identifies the selected

clipping plane, while unselected (but still active) clipping planes are represented by a gray and semi-transparent border. Once at a specific angle, the clipping plane can also be translated along its own normal axis using the up and down arrow buttons on the gamepad.

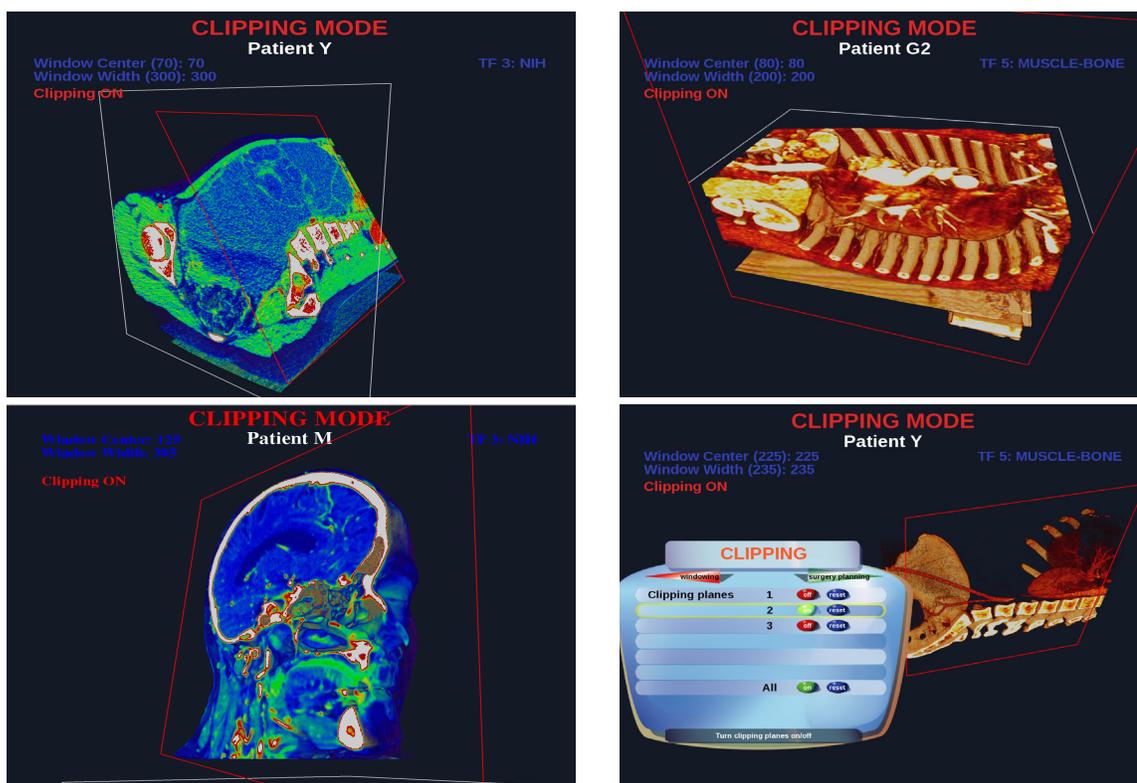


Figure 36: Interactive multiple clipping planes to slice into the 3D volume at any angle.

5.5.1 Clipping Menu

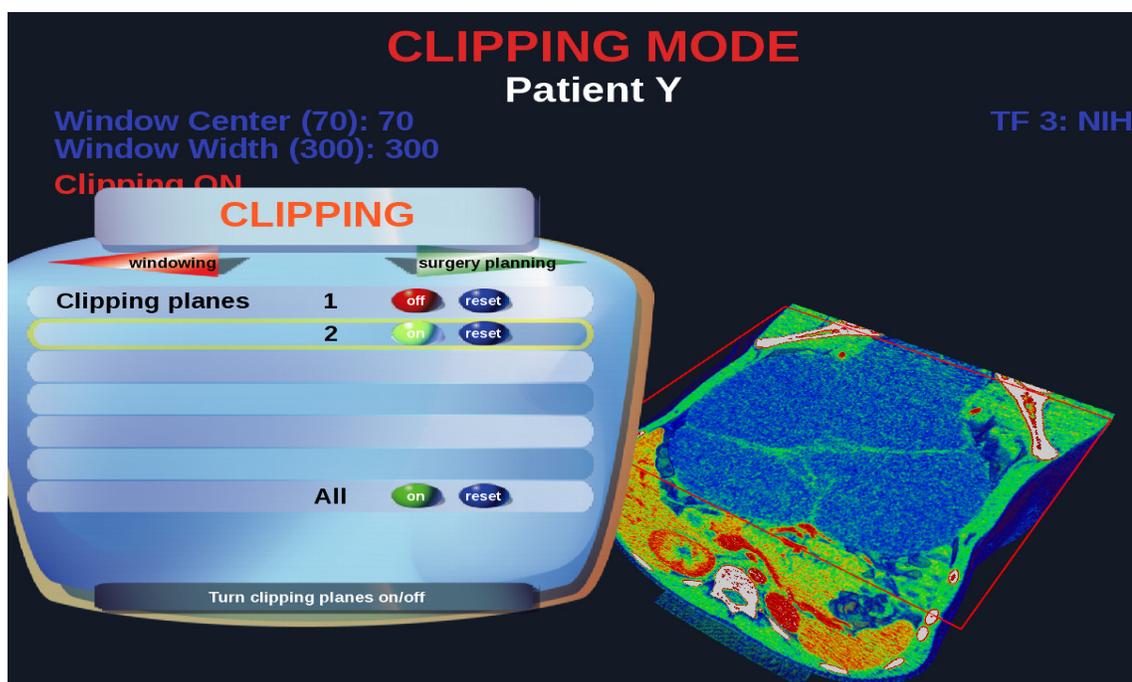


Figure 37: Toggling a specific clipping plane on/off via the option menu.

The menu interface allows the user to toggle a specific clipping plane, as shown in Figure 37. In cases when multiple clipping planes are active in the scene, the clipping plane selected in the menu is highlighted with a red outline while the other clipping planes will have a grey outline. This provides the user with a clear indication as to which clipping plane will be affected by the menu actions.

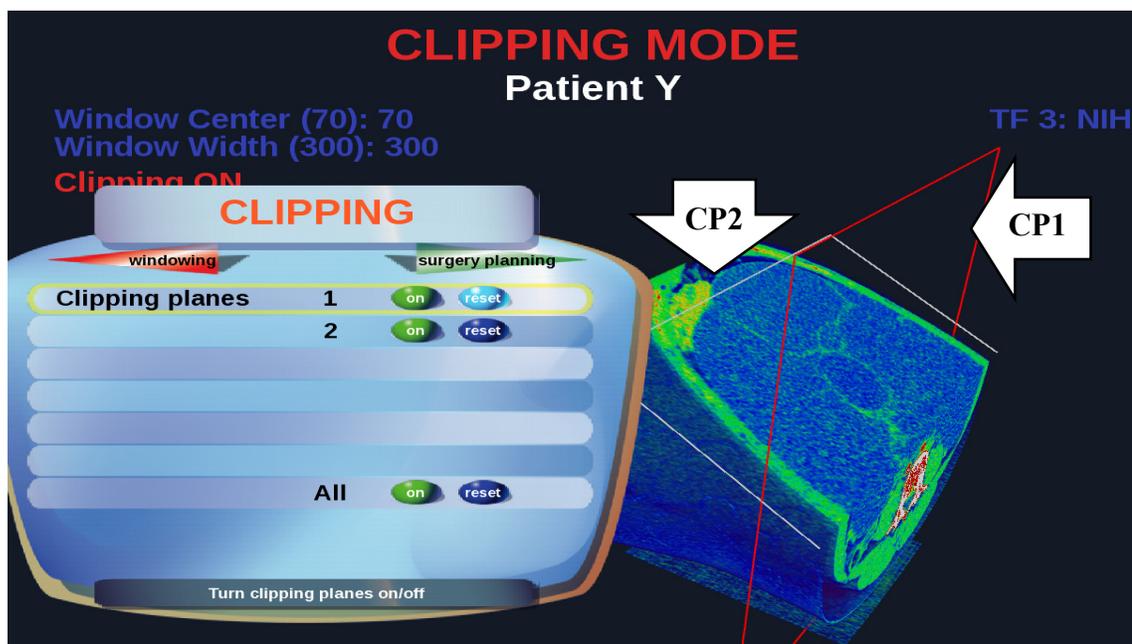


Figure 38: Resetting a specific clipping plane from within the menu.

The user can also reset a specific clipping plane to its initial position, which is in the X-Y plane relative to the volume. The “reset” button on the menu can be selected by pressing the “right” button on the directional pad, and pressing Button #2 resets the selected clipping plane. Pressing “left” on the directional pad selects the toggle button in the menu again. The “All” option will affect all the clipping planes in the scene, whether it is active or not. As the number of clipping planes increase, this option will be a convenient way to globally manage them.

5.5.2 Gamepad Controls and Shortcuts for Clipping Feature

Table 7 outlines the changes to the gamepad controls when in clipping mode. When in clipping mode, the left joystick now rotates the current selected clipping plane, while the up and down arrows on the directional pad moves the clipping plane up and

down its normal axis. Button #2 turns all clipping planes on or off, acting as the activation button for the clipping mode. Instead of cycling between the different models loaded, Buttons #7 and #8 now cycles between the clipping planes, while Button #9 resets the current selected clipping plane to its default location, which is at the center of the model. When the clipping mode is turned off, all gamepad controls will revert to the controls for default mode.

Table 7: Gamepad controls for the clipping mode.

Gamepad controls	Function
Left Joystick	Rotate current selected clipping plane
Directional Pad	Move clipping plane along normal axis
Button #2	Turn clipping planes on/off
Button #7	Select previous clipping plane
Button #8	Select next clipping plane
Button #9	Reset position of selected clipping plane

5.6 Surgery Planning

In a real endoscopic surgery, the trocars are inserted via an incision made on the patient's body. However, once the incision is made and the trocar is inserted, it is crucial that this placement is accurate, since the repositioning of a poorly placed trocar would require additional incisions causing further shock and distress to the patient's body and increasing recovery time. Recent studies based on 1215 minimally invasive surgeries (endoscopic) performed on small infants [120] showed that 11% of those surgeries resulted in complications and as high as 14.8% had to be converted to open surgery due to inexperience and poor planning. Another study based on 180 robotic (endoscopic)

surgeries performed infants and small children [121] also showed that 6.6% of the cases resulted in a conversion to open surgery while 4.4% required re-operations due to poor initial trocar positioning. Thus, there is a need for an effective three-dimensional visualization tool to aid surgeons in the decision making process for surgical planning.

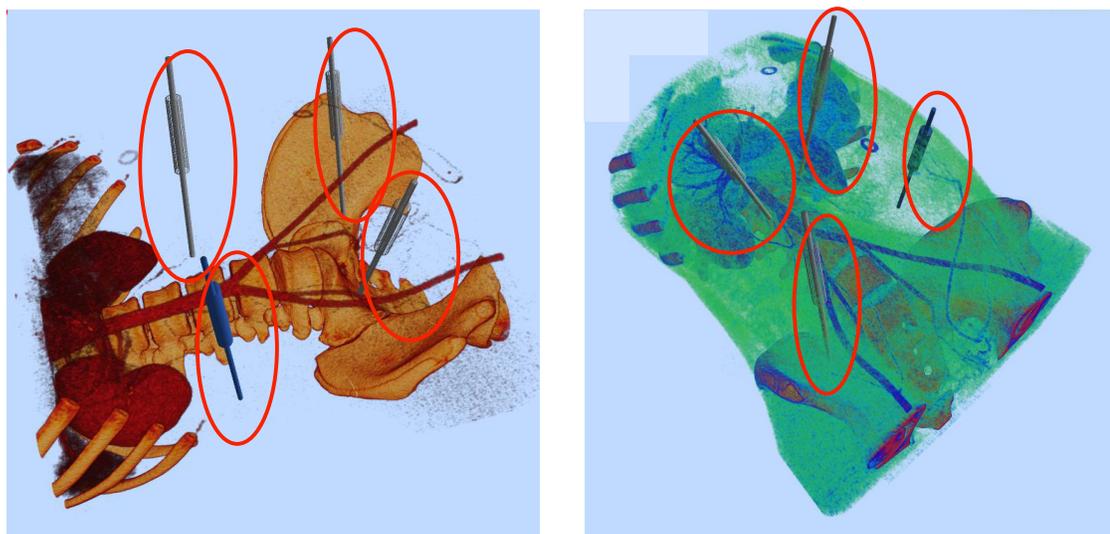


Figure 39: Trocar placement simulator for endo-surgery planning (trocars highlighted in circles).

Graphical simulated trocars are included in the application for use in endoscopic surgery planning. Figure 39 shows multiple trocars (highlighted in circles) positioned within the three-dimensional representation of a patient data. Using this feature, surgeons can accurately plan trocar placement prior to performing a surgical procedure, on the actual patient data. This provides the surgeon crucial information outside the operating room during preparation for surgery. By positioning the trocars at various locations on the patient's body, the surgeon can get an estimate of how far the laparoscopic

instruments can reach inside the body and if these instruments will be in contact with any sensitive organs or blood vessels.

5.6.1 Surgery Planning Menu

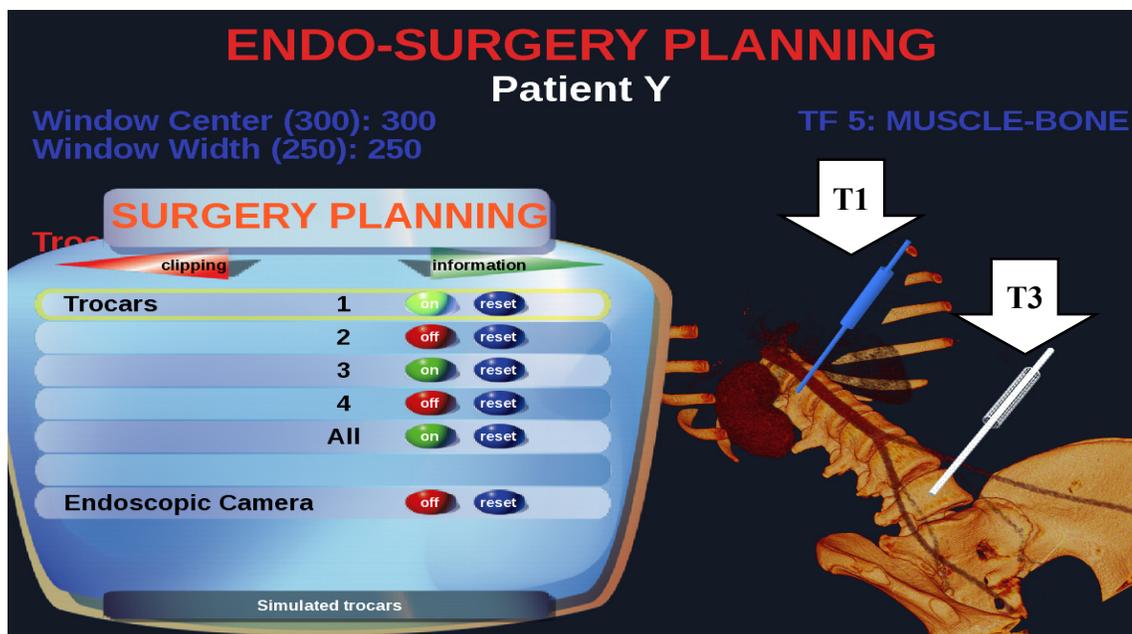


Figure 40: Toggling specific trocars on/off from within the option menu. T1 and T3 represent the first and third trocars that are turned on and currently visible. T1, which is in blue, is the trocar selected from the menu.

The menu interface for the surgery planning feature is similar to the menu interface for managing the clipping planes. There are by default four trocars available in the scene and each trocar is numbered in the menu as shown in Figures 40 and 41. The number of trocars was obtained through discussions from various endoscopic surgeons. As a line corresponding to the trocar is highlighted in the menu, the trocar is highlighted in blue as well, giving a clear indication to the user as to which trocar will be affected by a menu action. Trocars can be toggled on or off using by pressing Button #2 on the

gamepad, or the user can press “right” on the directional pad to highlight the “reset” button on the menu to reset the selected trocar’s position.

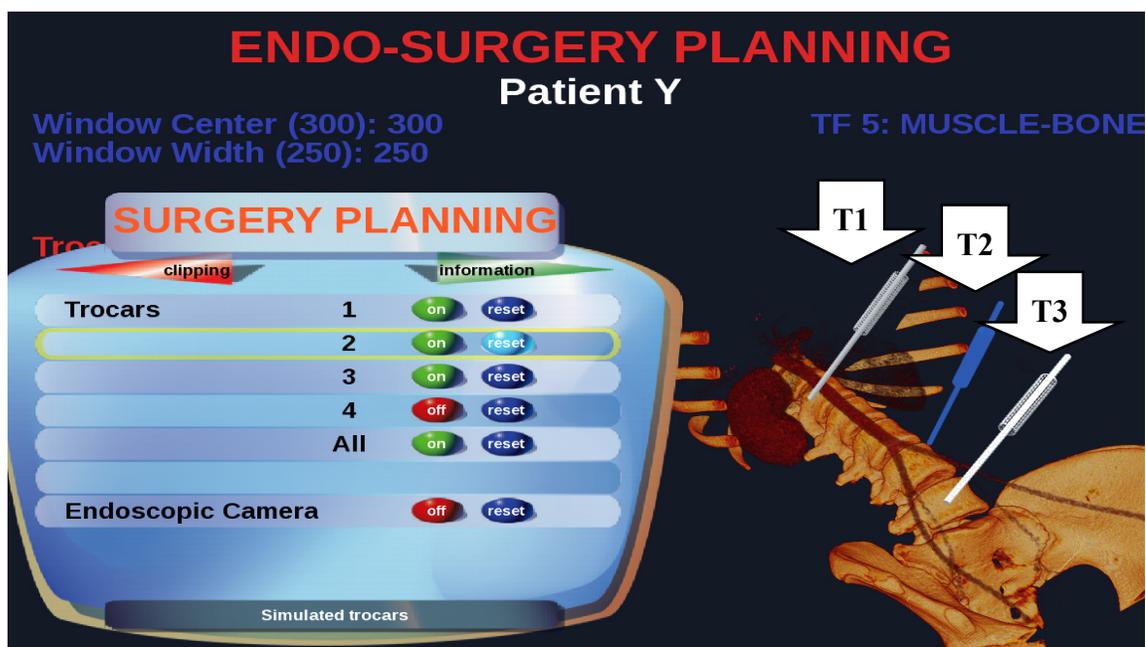


Figure 41: Resetting a specific trocar from within the menu. T1, T2, and T3 represent the first, second, and third trocars that are currently turned on and visible. T2, which is in blue, is the selected trocar from the menu.

Table 8: Gamepad controls for the endo-surgery planning mode.

Gamepad controls	Function
Left Joystick	Pivot selected trocar
Directional Pad	Move selected trocar
Button #2	Turn trocars on/off
Button #7	Select previous trocar
Button #8	Select next trocar
Button #9	Reset position of current selected trocar

The gamepad controls in the endo-surgery planning mode are shown in Table 8. The left joystick now pivots the trocar at a fixed location, while the directional pad repositions the trocar along the body. Similar to turning clipping planes on and off,

Button #2 now turns the endo-surgery planning mode on and off, either showing or hiding all the trocars. When the endo-surgery planning mode is off, the gamepad controls revert to those for the default mode. Buttons #7 and #8 cycle between the four trocars, and Button #9 resets the location of the current selected trocar to the center of the model.

The endoscopic camera mode option in the menu allows the user to view the patient data from the point of view of a simulated endoscopic camera. The joystick controls correspond to the controls of most of today's video games, where the left joystick moves the "player" front/back and left/right, while the right joystick controls the player's viewpoint, rotating it up/down and left/right. With a short learning curve, surgeons can focus on the examination of the patient model instead of struggling to understand the navigational controls of the software.

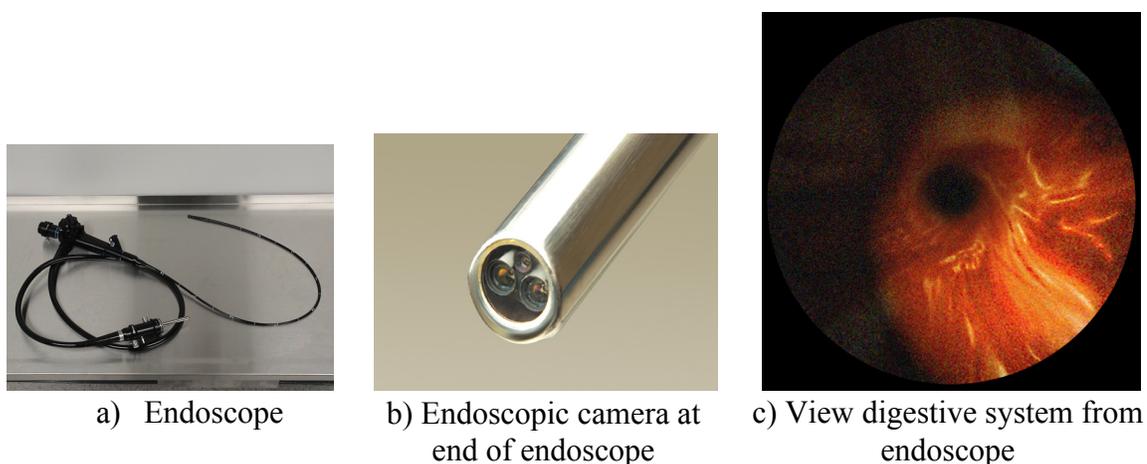


Figure 42: Examples of a) an endoscope, b) the camera at the end tip of the endoscope, and c) a screenshot of the video output from an endoscope inserted into the digestive system.

An endoscope and a screenshot from an endoscope inserted into the digestive system are shown in Figure 42. As seen from Figure 42c, the view from an endoscope has a limited field of view, restricting the amount of visual information available to the

surgeon. When viewed in a virtual reality environment, this software framework puts the surgeon inside the patient model, thus removing the “soda-straw” effect when viewing through a standard camera. This representation is extremely useful as a planning tool to determine optimal placement of an endoscope and other laparoscopic instruments. A case example is provided in the results chapter.

5.6.2 Tumor Inspection

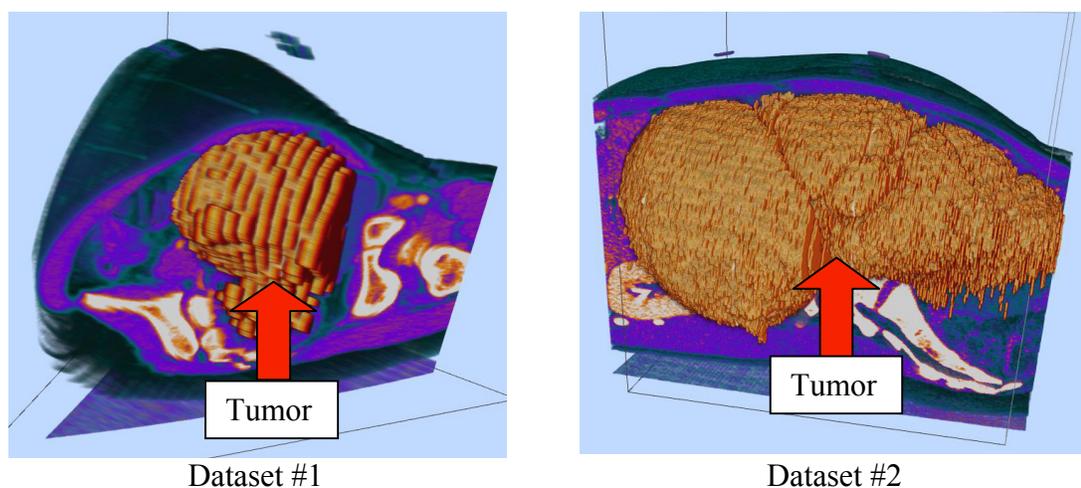


Figure 43: Viewing of automatically segmented tumors in context of patient data.

Tumors segmented using the segmentation methods discussed can also be loaded and viewed in context with the original patient data, for use in diagnostics or surgery planning. When viewed in context with the original patient data, any user interactions such as translation and rotation will affect both models using a “linking” feature implemented into the system. Since the segmented tumor is a 3D model on its own, the user can also choose to “unlink” the models and examine only the tumor. This feature can be used as an effective tool for a surgeon to examine the shape and size of the tumor for

surgical planning or as a diagnosis tool to track the progress of a patient who is undergoing treatment. Datasets that have been processed using the fuzzy segmentation algorithm discussed and the tumors extracted are highlighted as shown in Figure 43.

6 RESULTS AND DISCUSSION

6.1 Information and Properties of Test Datasets

The developed segmentation methods were tested on 10 distinct sets of contrast enhanced CT data, from seven individual patients. Test case #1 is a cystic teratoma, #2 is a mucinousecystadenoma, and #3 is an immature teratoma tumor (courtesy of Dr G. Miyano from Juntendo University School of Medicine, Tokyo, Japan). Test cases #4-#10 are of neuroblastic tumors (courtesy of Dr R.M. Rangayyan from University of Calgary, Alberta, Canada). The CT sets are of varying number of slices and the tumors of interest to be segmented are also of varying difficulty, in terms of being segmented automatically. The difficulty and complexity of automatic tumor segmentation can be loosely categorized at three different levels: tumors with homogeneous tissue densities (category A), tumors with some inhomogeneity and fuzzy edges (category B), and tumors with heterogeneous tissue densities (category C). Tumors with calcifications (calcium build up inside the tumor tissues) can be categorized as either B or C, depending on the degree of calcifications. Tumors from categories C are very challenging segmentation problems since the computer algorithms used for automatic segmentation will be unable to properly identify the tumor tissues without additional intervention from the user. In most cases, even tumors from category C can be difficult to be manually segmented by a human who is not medically trained. Table 9 summarizes the properties and the information about the test cases presented.

Table 9: Summary of information and parameters for the test cases presented.

#	Patient	Tumor description	# of slices	Windowing parameters	Cat
1	P1	Cystic Teratoma on left ovary	38	C:95, W:250	A
2	P2	MucinousCystadenoma on right ovary	301	C:120, W:200	A
3	P3	Immature Teratoma	132	C:110, W:230	A
4	P4	Neuroblastoma, small and diffuse	48	C:110, W:200	B
5	P5	Neuroblastoma	75	C:90, W:280	B
6	P5	Neuroblastoma	70	C:90, W:280	C
7	P6	Neuroblastoma, some calcification	54	C:95, W:220	B
8	P6	Neuroblastoma, some calcification	75	C:90, W:210	C
9	P6	Neuroblastoma, highly calcified	71	C:90, W:210	C
10	P7	Neuroblastoma	69	C:90, W:280	C

Radiologist Dr J.L. Friese from Brigham and Women's Hospital, Boston, Massachusetts provided manual segmentations for test cases #1-#3, and radiologist Dr G.S. Boag from Alberta Children's Hospital, Calgary, Canada provided manual segmentations for test cases #4-#10. The manually segmented region of the tumors will be the gold standard used for comparison with the results obtained from the automatic segmentation algorithm.

To evaluate the accuracy and performance of the developed methods, 10 users performed segmentation of the test cases, with each user performing the procedure four times for every dataset. To ensure medical accuracy when performing these test cases, the users are shown the image of the first slice that the tumor appears with the radiologist's outline of the tumor for use as reference when placing the seed for the probabilistic segmentation method and when selecting the region of interest for the fuzzy segmentation method.

In this chapter, the segmentation results from for the test cases both methods will be presented and discussed. Selected two-dimensional and three-dimensional volume

representations of tumors segmented using both methods will be presented and compared. This is followed by detailed numerical analyses of the segmentation accuracy and repeatability factor of both proposed segmentation methods.

6.2 Hardware and Software

The segmentation method was executed on an Intel Core2 Quad 3.0 GHz processor with 3 GB of memory and 256 MB NVidia 8300GTS graphics card, running Windows Vista Business Edition. The programming of the segmentation method was done in MATLAB v7.4, utilizing its fuzzy logic toolbox and the image processing toolbox.

The immersive viewer was tested on the C6, Iowa State University Virtual Reality Applications Center's six-sided virtual reality projection system. The performance data was obtained by running the application in stereo on the mentioned system, with frame locking on, to sync the multiple graphics threads seamlessly. The C6 VR system is powered by a cluster of 48 computers with Dual AMD Opteron 250 2.4GHz processors and 4GB of memory, each with dual 512 MB NVidia Quadro FX4500 graphics cards configuration, running Linux Red Hat Enterprise 64 bit edition.

6.3 Probabilistic Segmentation of Tumors

In this section, the segmentation results for probabilistic segmentation method will be presented and discussed. In addition, selected two-dimensional and three-dimensional volume representations of the tumors segmented will also be presented and

discussed. This is followed by a detailed numerical analysis of segmentation accuracy and repeatability factor of the proposed segmentation method. Based on the discussion in Chapter 3, refinements were made to the segmentation method and the results from both versions of the probabilistic segmentation method are included in this chapter. Table 10 summarizes the differences between the two versions.

Table 10: Summary of the improvements made to the probabilistic segmentation method.

Version 1	Version 2
Seed is a single pixel	Seed is a 5x5 grid of pixels
Between slices, seed from previous slice is mapped to current slice	Between slices, the centroid of the segmented region from previous slice is used as the seed for current slice
Search region is grown at 10% by default	Rate of change for search region is calculated based on the rate of change for the last two slices

6.3.1 Segmentation Results

When the segmentation algorithm was applied to a well-defined tumor as in test case #1, a Cystic Teratoma tumor on the left ovary, the algorithm performed well as shown in Figure 44. There are misclassification errors, as indicated by the circle in slice #0045, when pixels of non-tumor tissues were selected. These misclassifications are due to the proximity and similarity in surrounding tissue densities. Despite this problem, the method was still successful in segmenting a majority of the tumor pixels and defining the overall shape of the tumor.

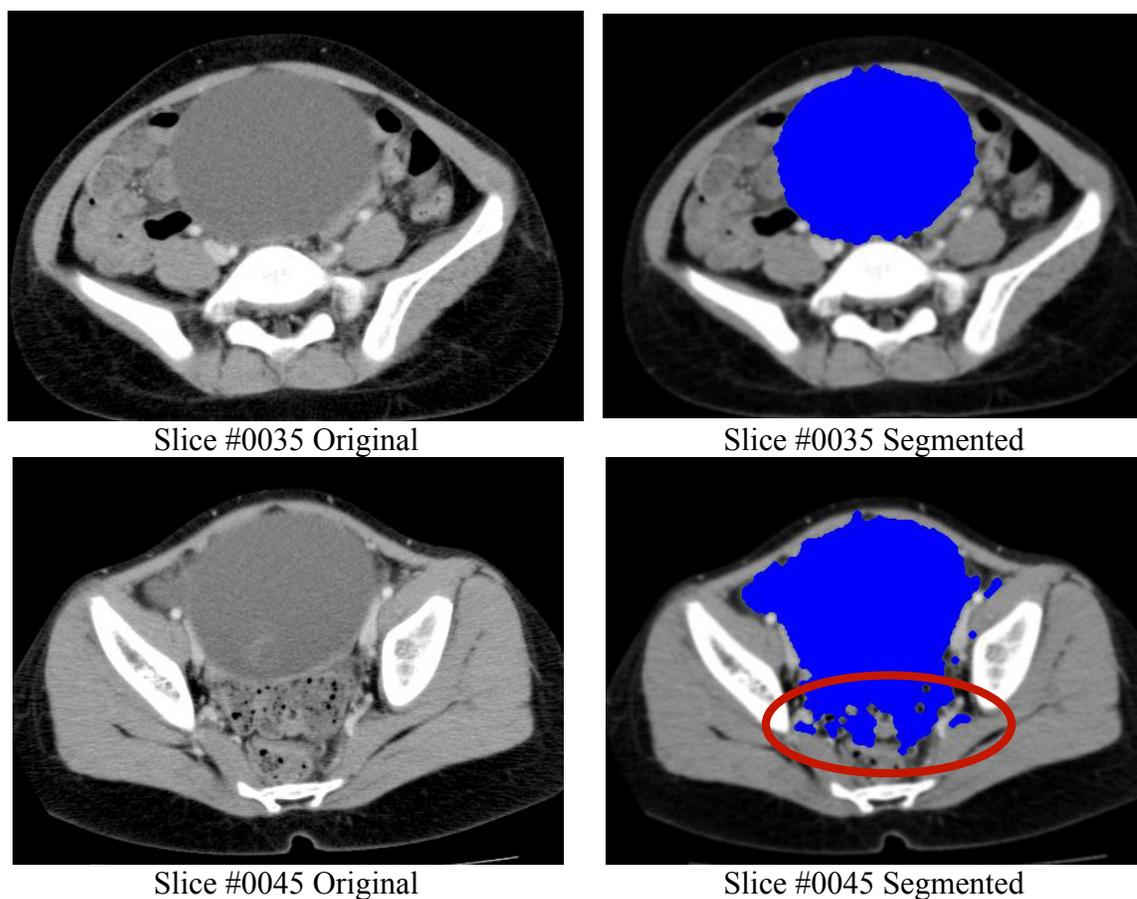


Figure 44: Segmentation results for Test Case #1.

A three-dimensional volume representation of the segmented tumor as well as the manually segmented tumor is presented in Figure 45 for comparison. Segmentation of tissues with similar density as the tumor is circled in the figure. Despite this interference in the segmentation process, the overall shape of the segmented tumor using the probabilistic segmentation method is very much in agreement with the shape obtained from manual segmentation.

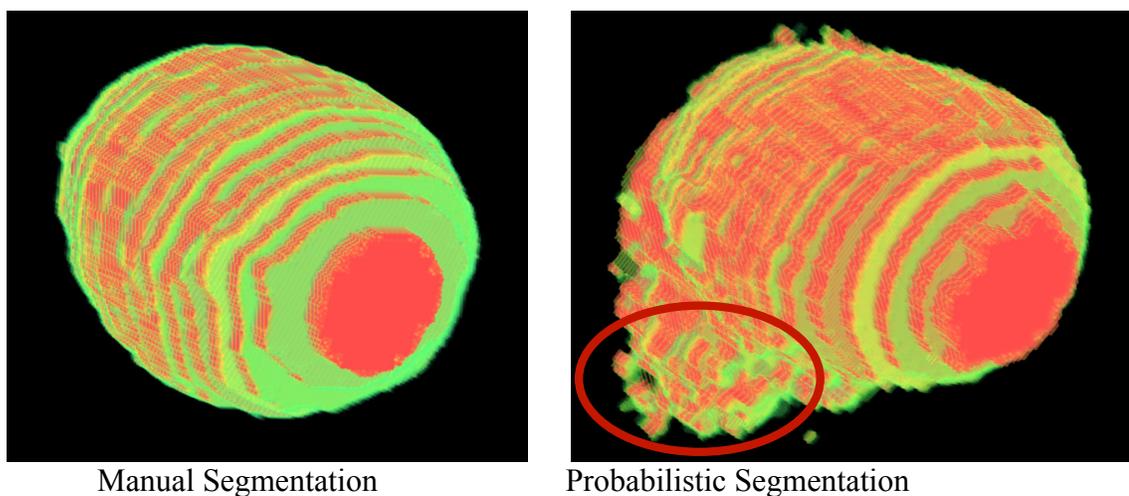
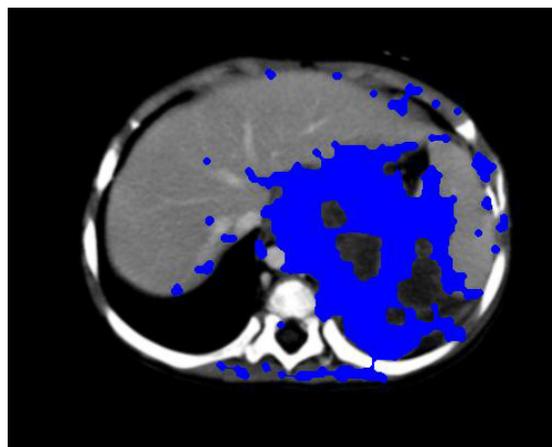


Figure 45: Three-dimensional volume renders of the segmented tumor from Test Case #1, from manual segmentation and probabilistic segmentation.

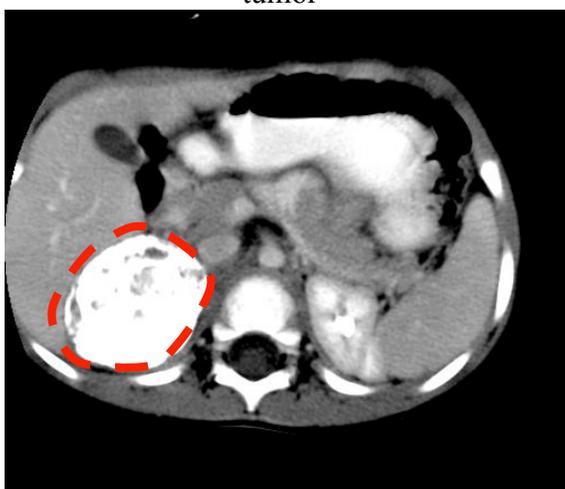
When the probabilistic segmentation algorithm was applied to some of the more diffuse and partially heterogeneous tumors such as test cases #5 and #7, the results have more errors as illustrated in Figure 46. The surrounding healthy tissues and the tumor have very similar densities, thus causing a region leak during the segmentation procedure. In test case #5, the tumor tissues have very similar densities to the tissues around it and in test case #7 the calcifications in the tumor have very similar densities to the bone tissues on the vertebrae.



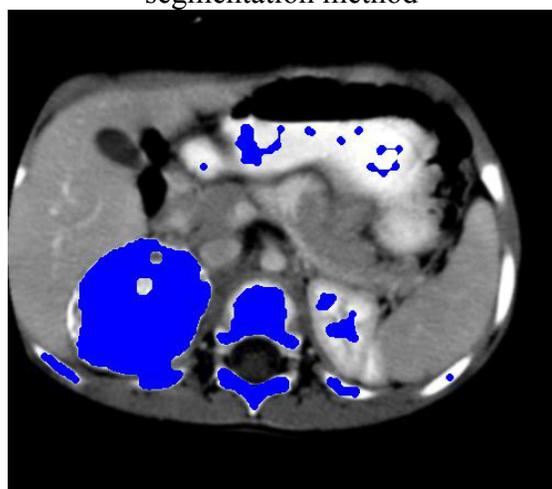
Test Case #5 – Radiologist segmented tumor



Test Case #5 – Tumor from probabilistic segmentation method



Test Case #7 – Radiologist segmented tumor



Test Case #7 – Tumor from probabilistic segmentation method

Figure 46: Probabilistic segmentation results for tumors with mixed tissue densities that are slightly heterogeneous. Actual tumors outlined by radiologist are shown in red dashed lines.

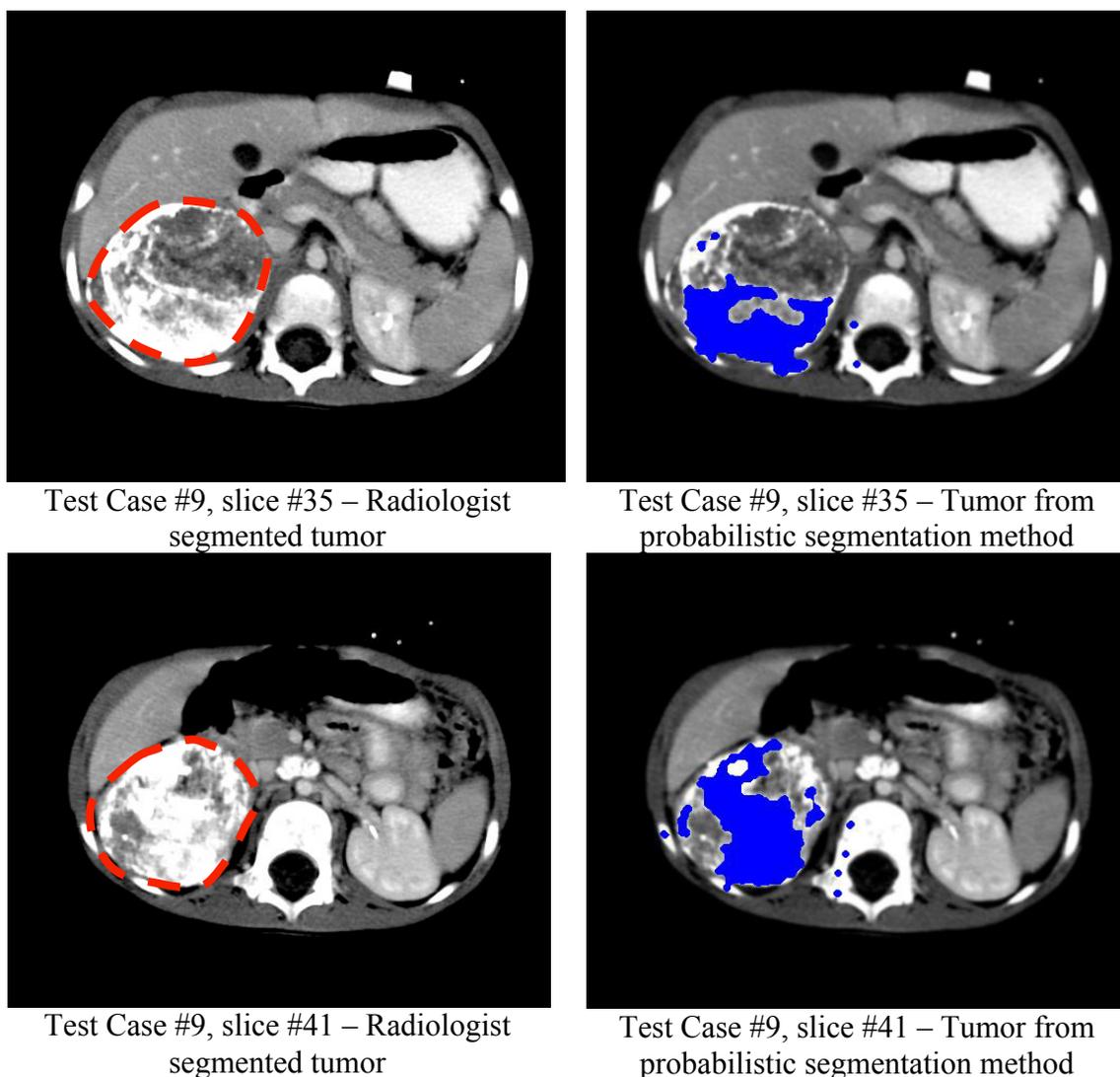


Figure 47: Probabilistic segmentation results for tumors that are highly heterogeneous with calcifications. Actual tumors outlined by radiologist are shown in red dashed lines.

For test case #9, the calcifications in the tumor as well as the mixed tissues caused difficulty for the segmentation method, shown in Figure 46. The segmentation method only managed to segment about half of the tumor, which is expected since the probabilities generated are based on pixel intensities and the intensities defining the

tumor is very heterogeneous. Even for an untrained person without any medical experience, this test case would prove to be difficult to identify and manually segment.

The average time take to process a single slice was approximately 5-11 seconds, depending on the size of the search region. For the smallest test case of 38 slices, the processing time took 3.4 minutes, while the largest test case of 301 slices took 45.2 minutes.

6.3.2 Segmentation Accuracy

In order to evaluate the accuracy of the proposed probabilistic segmentation method, the misclassification rates were calculated for each test case. A pixel classified as part of the tumor when it was not in the gold standard is a False Positive (FP), and a pixel identified as tumor based on the gold standard but not selected by the method is a False Negative (FN). Both error rates are calculated as:

$$FP = \frac{V(A) - V(A \cap R)}{V(R)} \times 100\% \quad (18)$$

$$FN = \frac{V(R) - V(A \cap R)}{V(R)} \times 100\% \quad (19)$$

where $V(R)$ is the volume segmented by the radiologist, and $V(A)$ is the volume segmented by the algorithm.

The false positive error rates evaluate the algorithm's ability to correctly classify tumor tissues and ignore healthy tissues. A low FP rate demonstrates that the algorithm is not sensitive to healthy tissues with intensity values similar to the tumor's intensity values, causing classification errors. These errors usually occur when tumor is diffuse

with blurred and poorly defined edges. The false negative error rates evaluate how accurately the algorithm performs in classifying tumor tissues, especially tumors that are highly heterogeneous and/or with calcifications.

Based on the multiple runs performed by 10 users, the average of these classification errors are calculated as the Mean False Positive (MFP) and the Mean False Negative (MFN) and are shown in Table 11.

Table 11: Accuracy evaluation of the probabilistic segmentation algorithm. Mean False Positive (MFP) when non-tumor pixels were selected and Mean False Negative (MFN) when tumor pixels were not selected for all 10 test cases. The results for “Version 1” are for the probabilistic method using the original seed relocation algorithm, while “Version 2” is with the new revised seed relocation and search region algorithm.

#	Patient	Cat	Tumor description	Version 1		Version 2	
				MFP	MFN	MFP	MFN
1	P1	A	Cystic Teratoma on left ovary	3.9%	1.7%	3.1%	1.8%
2	P2	A	MucinousCystadenoma on right ovary	2.4%	3.6%	2.2%	3.6%
3	P3	A	Immature Teratoma	5.3%	4.1%	4.6%	4.3%
4	P4	B	Neuroblastoma, small and diffuse	6.3%	5.6%	5.8%	4.7%
5	P5	B	Neuroblastoma	14.4%	19.8%	13.1%	19.9%
6	P5	C	Neuroblastoma	15.8%	21.3%	13.6%	21.8%
7	P6	B	Neuroblastoma, some calcification	10.8%	13.1%	9.1%	12.6%
8	P6	C	Neuroblastoma, some calcification	28.9%	55.3%	25.2%	53.9%
9	P6	C	Neuroblastoma, highly calcified	8.3%	39.8%	7.1%	40.2%
10	P7	C	Neuroblastoma	16.5%	12.3%	15.4%	12.1%

Based on the results for the revised probabilistic segmentation method, six of the test cases had MFP errors less than 10%; and three of the test cases had MFP errors between 13% and 16%, and one test case with 25% MFP error. Five of the test cases had

MFN errors less than 6%; three had MFN errors between 12% and 22%; and two had MFN error of more than 40%. Comparing the results for Version 1 and Version 2 of the method, there is a significant improvement where the error rates for MFP is reduced in all 10 test cases. This reduction in false positives is from the use of the rate of change parameter, which significantly eliminates the processing of unnecessary pixels. There are also some improvements observed in for some test cases with lower false negative error rates, but a statistical analysis showed that this reduction is not significant. For the rest of this chapter, the discussion of the results will be focused on Version 2 of the probabilistic segmentation method.

The results obtained were compared to a recently published tumor segmentation method by Deglint *et al* [122] to further evaluate the performance of the probabilistic segmentation algorithm. The comparison was possible due to the same test cases used for both methods. Deglint's method removes known structures such as bone, fat, and skin from the image data prior to performing fuzzy connected segmentation on the tumor. Eight of their ten test cases had less than 12% MFN error, while the remaining two test cases had MFN between 20% and 30%. For MFP error rates, Deglint's method was able to successfully segment the tumors on three test cases with less than 20% MFP errors, five test cases had MFP errors between 35% and 50%, and the remaining two test cases had MFP errors of more than 50%. Comparatively, the probabilistic segmentation method had a lower MFP error rate on more test cases and with the worst MFP error rate at only 25%, less than half of what Deglint's worst result (>50%) produced. This demonstrates the probabilistic segmentation algorithm was able to better differentiate between tumor tissues and healthy tissues, even when the dataset is diffuse with blurred edges. The

probabilistic segmentation method did comparatively well on most of the test cases, with MFN error rates similar to Deglint's, except for two test cases with MFN error rates that are between 40% and 54%, 24% more false negatives than the worst results from Deglint's method. However, Deglint's method took a longer time to complete, taking between 1-2 minutes per slice while the probabilistic segmentation method only required 5-11 seconds per slice.

Overall, the method performed well, successfully segmenting the tumor from the healthy tissues in most of the test cases. The low MFN for most of the test cases demonstrated that the method was able to perform without mistaking healthy tissues as tumor tissues. The high MFP for test case #8 meant that more healthy tissues were mistaken as tumor tissues for that one case. This high false positive error rate is most likely a result of poor delineation between the tumor and the surrounding tissues leading to a region leak. Calcification and the inhomogeneity of the tissue density within the tumor caused some pixels to be unselected during the segmentation procedure, resulting in a high MFN values for two test cases.

6.3.3 *Inter- and Intra-User Variability*

This section discusses the repeatability factor of the developed segmentation algorithm. Two factors were considered, the inter-user variability and the intra-user variability. Inter-user variability compares the variability in the segmentation results when performed by multiple users, and intra-user variability compares the variability in the segmentation results when a single user performs the method multiple times.

A paired t-test was performed to determine if the inter- and intra-user variations are significant. The paired t-test is used to evaluate the significance of the differences between two observations, represented by D. The hypotheses used are:

$$H_0: D = 0 \quad (\text{there is no significant difference})$$

$$H_a: D > 0 \quad (\text{there is significant difference})$$

To either reject or accept the null hypothesis (H_0), a t value and its associated p-value is calculated using the following equation:

$$t = (\bar{X} - \bar{Y}) \sqrt{\frac{n(n-1)}{\sum_i^n (\hat{X}_i - \hat{Y}_i)^2}} \quad (20)$$

where \bar{X} and \bar{Y} are the mean values of both observation group, n is the number of items in each observation group (equal for both since this is a paired test) and

$$\begin{aligned} \hat{X}_i &= X_i - \bar{X}_i \\ \hat{Y}_i &= Y_i - \bar{Y}_i \end{aligned} \quad (21)$$

The p-value associated with the calculated t value can be inferred from a standard t-distribution table. If the p-value is low (< 0.05), then there is evidence to reject the null hypothesis. Thus, there is a difference in means across the paired observations.

To test if there is significant inter-user variation, the differences in the segmented tumors for all ten test cases produced by a pair of users were analyzed for statistical significance. For example, User1's segmented tumor volume for test case #1 will be subtracted from User2's segmented tumor volume for the same test case. Any remaining voxels will be the difference in the segmentation result between the two users. This is repeated for all 10 test cases. Since we have 10 test cases and each was performed four

times, $n=40$. If there is no statistically significant difference in the results of a specific pair of users, then the p -value will be greater than 0.05.

Table 12: Statistical analysis of inter-user variability for probabilistic segmentation method.

Testing Pairs	p -value (n=40)
User1-User2	0.2609
User2-User3	0.2897
User3-User4	0.3192
User4-User5	0.3184
User5-User6	0.2425
User6-User7	0.2926
User7-User8	0.2501
User8-User9	0.1844
User9-User10	0.2255
User10-User1	0.2792

The pairing of the users for comparison is listed in Table 12, along with the corresponding t-test p -value result. The results show that all pairs have a p -value > 0.05 , which indicates that there is no significant difference between the pairs of users in terms of the segmentation results obtained.

Testing for the intra-user variation is done in a similar fashion, but instead of pairing different users, the different runs performed by each user are paired and tested, i.e. for User1, run #1 is paired with run #2, run #2 is paired with run #3, and so forth. The statistical analysis to test for intra-user variability also produced a p -value > 0.05 , indicating that there is no significant difference between the multiple segmentation results of the same dataset as performed by the same user.

6.4 Fuzzy Segmentation of Tumors

The same test cases were also segmented using the fuzzy logic method. In this section, the segmentation results for these test cases will be presented and discussed and selected two-dimensional and three-dimensional volume representations of the tumors segmented using the fuzzy rule-based segmentation method will be presented and discussed. This is followed by a detailed numerical analysis of segmentation accuracy and repeatability factor of the proposed segmentation method. Table 13 summarizes the modifications made to the fuzzy segmentation method.

Table 13: Summary of the modifications made to the fuzzy segmentation method.

Version 1	Version 2
Search region grows by 50% at every slice	Search region grows based on a weighted average of the last two slices' rate
Segmentation performed once for a specific window setting	Segmentation performed three times on different window settings

6.4.1 Multi-Window Fuzzy Segmentation Results

As mentioned in Chapter 4.11, the three windowing parameters are defined manually by the user using the preset values provided by the radiologists as a starting point. Table 14 lists the different windowing parameters used during the runs performed to test the multi-window fuzzy segmentation method.

Table 14: Summary of information and parameters for the test cases presented.

#	Windowing parameter 1	Windowing parameter 2	Windowing parameter 3
1	C:65, W:130	C:95, W:250	C:110, W:180
2	C:90, W:150	C:120, W:200	C:135, W:160
3	C:110, W:90	C:110, W:230	C:140, W:160
4	C:76, W:90	C:110, W:200	C:135, W:150
5	C:75, W:70	C:90, W:280	C:115, W:130
6	C:75, W:70	C:90, W:280	C:115, W:130
7	C:88, W:100	C:95, W:220	C:120, W:175
8	C:68, W:85	C:90, W:210	C:105, W:160
9	C:68, W:85	C:90, W:210	C:105, W:160
10	C:75, W:90	C:90, W:280	C:110, W:180

Figure 48 shows the different segmentation results when changing the windowing settings and the final merged segmented region after morphological operations. This new implementation reduced most of the errors due to inhomogeneity in the tissue density. For the rest of this chapter, all the results discussion will be focused on the results produced by the modified version of the fuzzy segmentation method.

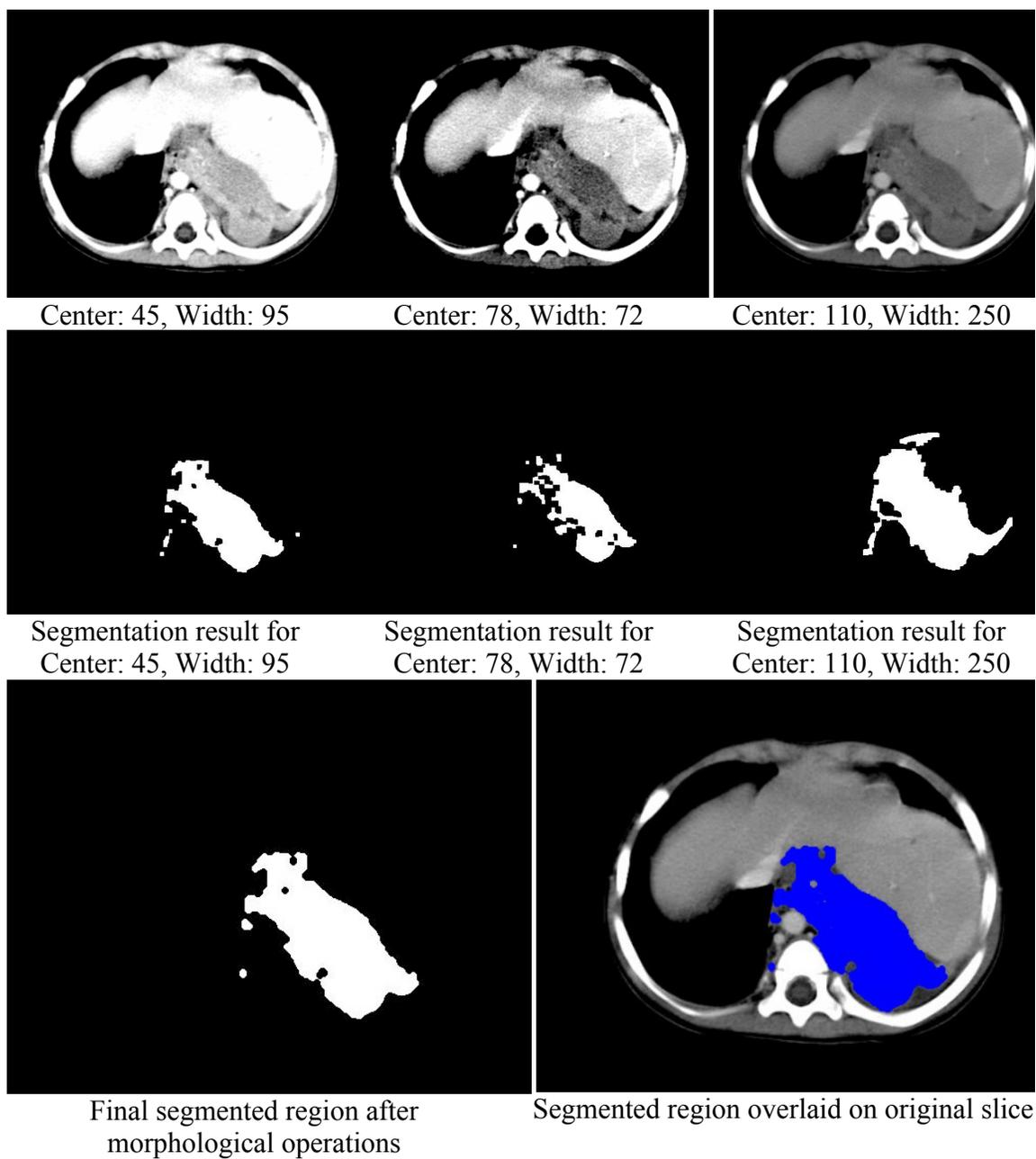


Figure 48: Segmentation results from three different windowing settings of the same slice and the final merged segmented region.

6.4.2 Segmentation Results

The object segmented in Test Case #1 is a Cystic Teratoma tumor on the left ovary, spanning 19 slices. The slice thickness is 5 mm, making the object being segmented approximately 95 mm in length. The results for test case #1 are presented in Figure 49.

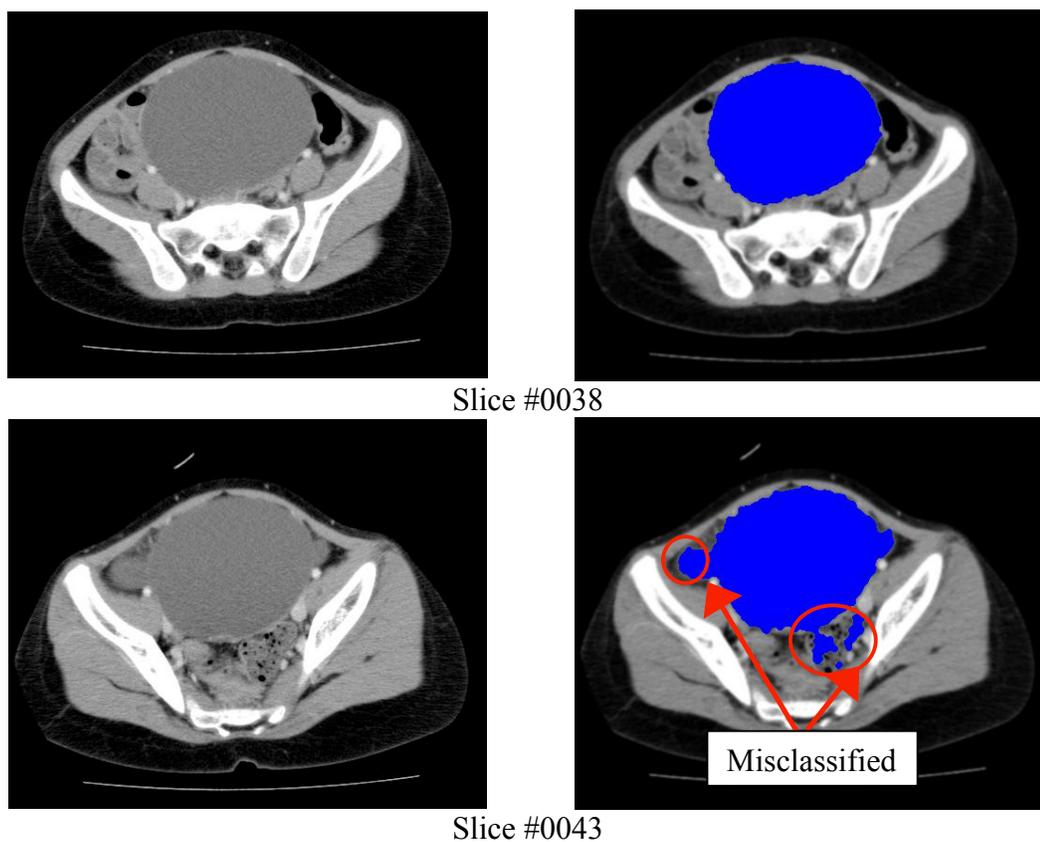


Figure 49: Selected results of the fuzzy segmentation for Test Case #1

A three-dimensional volume representation of the segmented tumor from manual segmentation as well as the results from the proposed fuzzy segmentation method is presented in Figure 50. Segmentation of tissues with similar density as the tumor is circled in the figure. Despite this interference in the segmentation process, the overall

shape of the segmented tumor using the fuzzy segmentation method is very much in agreement with the shape obtained from manual segmentation. Compared to the results from the probabilistic segmentation method, the fuzzy segmentation method performed better by not misclassifying as much healthy tissue as the probabilistic method.

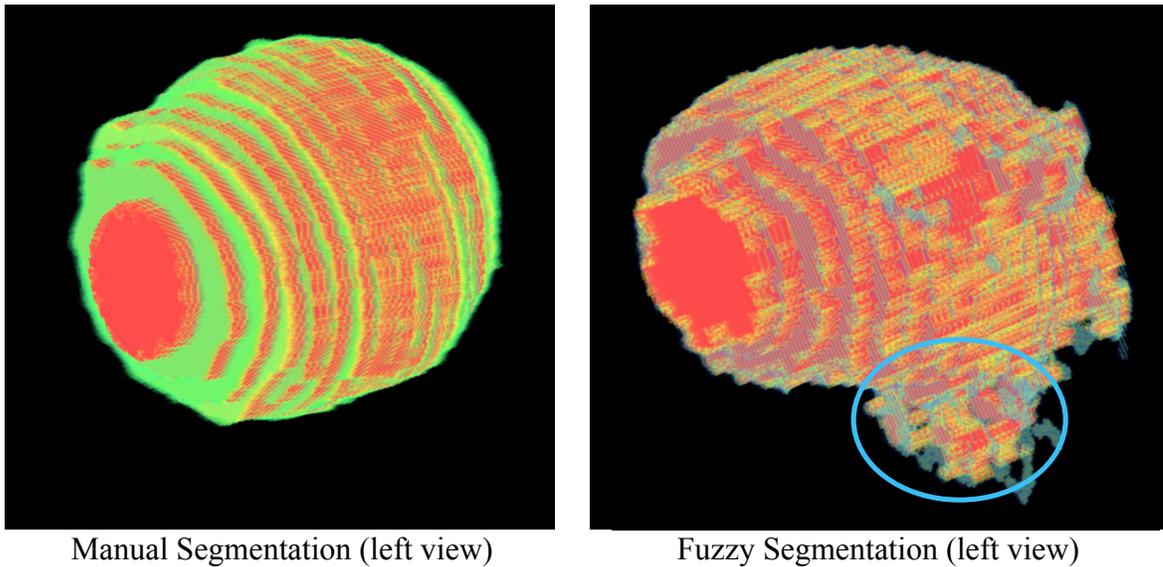
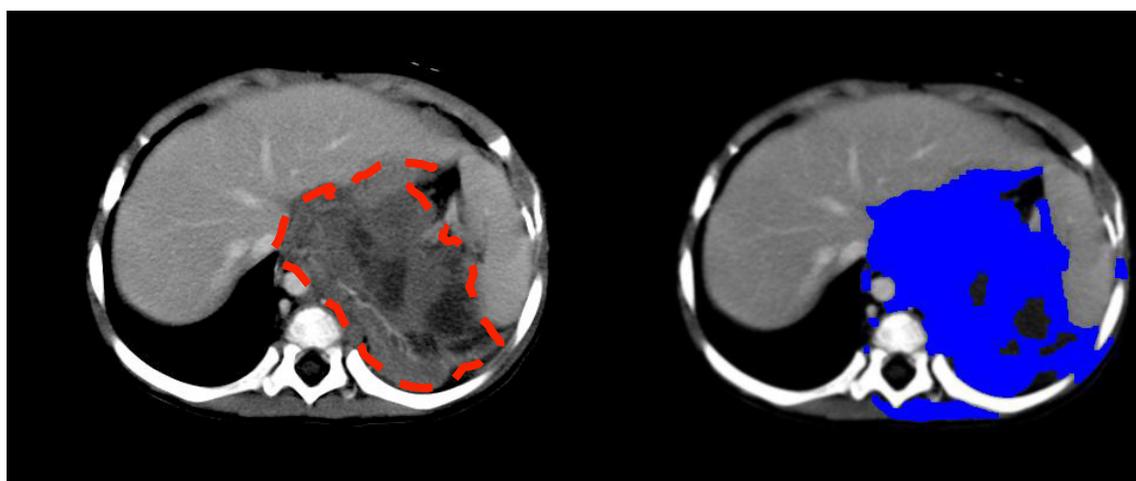


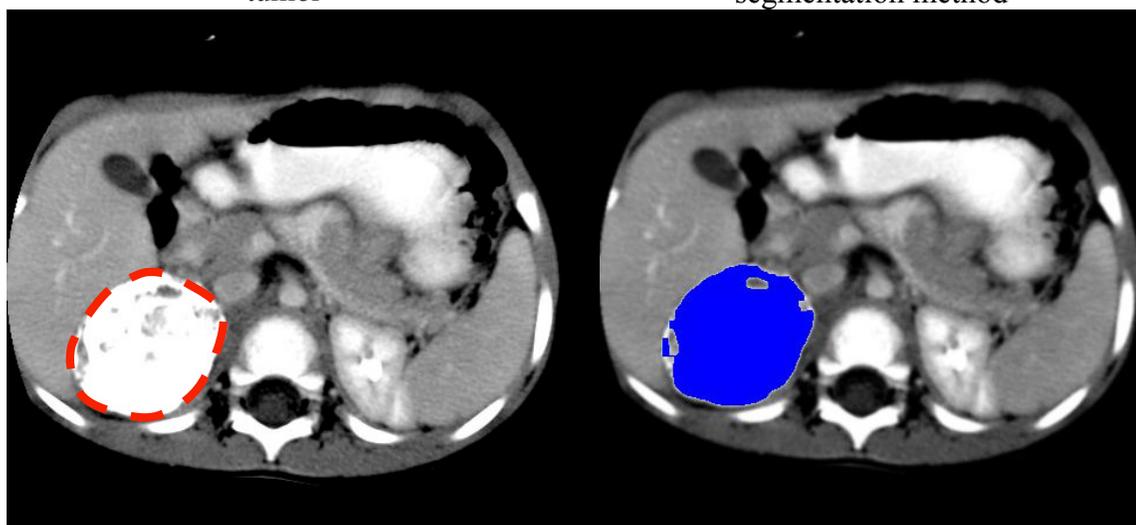
Figure 50: Three-dimensional view of the segmented tumor from Test Case #1 using manual segmentation (left) and from the fuzzy segmentation method (right).

Based on the results in Figures 49 and 50, the method was able to successfully determine the pixels that define the tumor. The shape and size of the tumor were visually defined, even though there are some misclassified pixels, as indicated in slice #0043, when pixels of non-tumor tissues were selected as tumor. This misclassification is due to the similarity in tissue densities and also these tissues are in close proximity to the tumor, resulting in a high similarity value output from the fuzzy inference system.



Test Case #5 – Radiologist segmented tumor

Test Case #5 – Tumor from fuzzy segmentation method



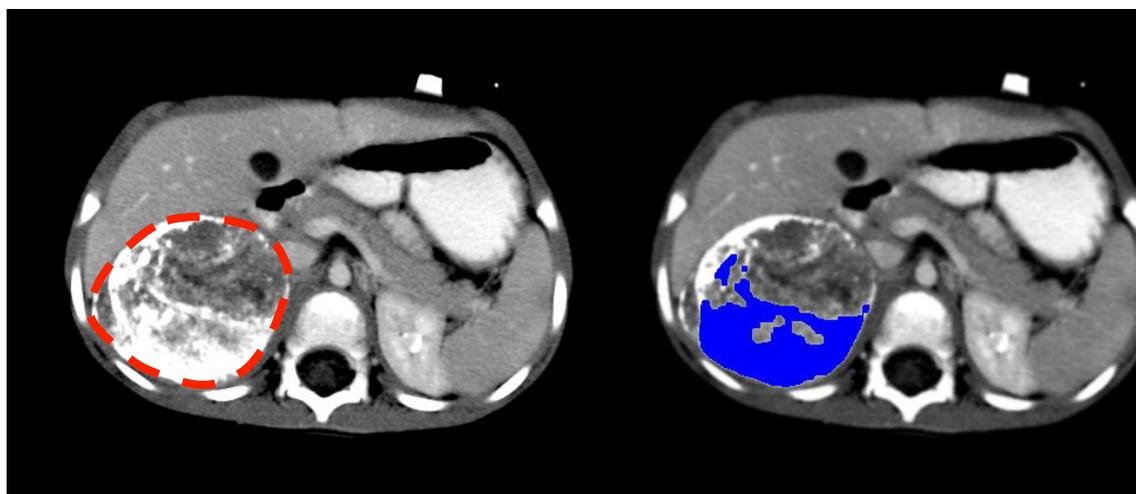
Test Case #7 – Radiologist segmented tumor

Test Case #7 – Tumor from fuzzy segmentation method

Figure 51: Fuzzy segmentation results for tumors with mixed tissue densities that are slightly heterogeneous. Actual tumors outlined by radiologist are shown in red dashed lines.

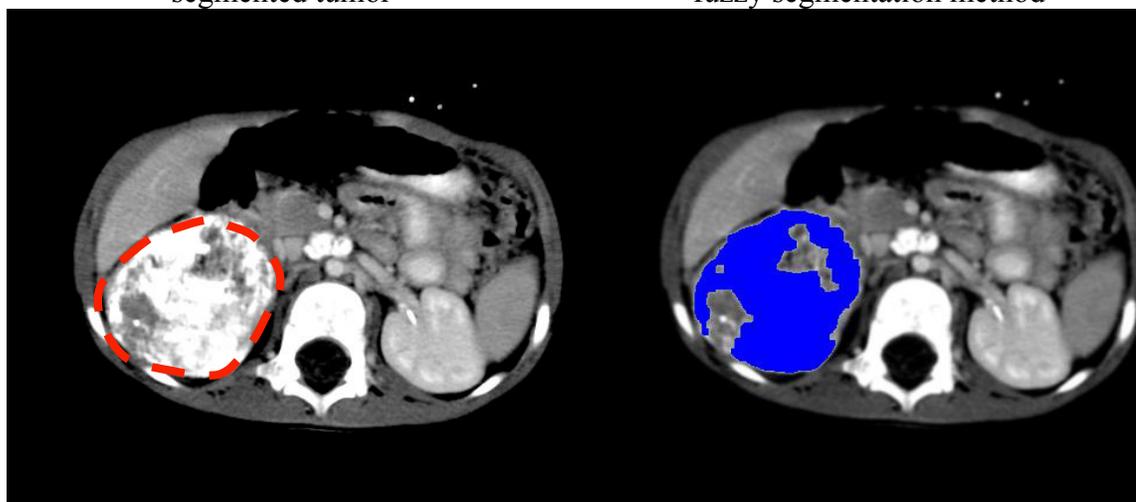
The fuzzy segmentation algorithm was applied to some of the more diffuse and heterogeneous tumors and the results are shown in Figure 51. The results are better than those produced by the probabilistic segmentation method, getting almost all parts of the tumor shown in slice #18 of test case #7. For test case #5, there are some noticeable false

positive errors, when the healthy tissues surrounding the tumor are segmented. The method also did not segment some tissues inside the tumor since they were of a different tissue density, shown as darker pixels.



Test Case #9, slice #35 – Radiologist segmented tumor

Test Case #9, slice #35 – Tumor from fuzzy segmentation method



Test Case #9, slice #41 – Radiologist segmented tumor

Test Case #9, slice #41 – Tumor from fuzzy segmentation method

Figure 52: Segmentation results for tumors that are highly heterogeneous with calcifications. Actual tumors outlined by radiologist are shown in red dashed lines.

As with the probabilistic segmentation method, the calcification in the tumor for test case #9 also caused some difficulty for the fuzzy segmentation method. The method only segmented the calcified part of the tumor, missing most of the non-calcified tissues that also define part of the tumor, shown in Figure 52. Compared to the results obtained from the probabilistic segmentation method, the resulting segmented tumor is more complete, selecting most of the tumor tissues that the probabilistic method missed.

The average time take to process a single slice was approximately 13-38 seconds, depending on the size of the search region. For the smallest test case of 38 slices, the processing time took 8.6 minutes, while the largest test case of 301 slices took 154 minutes. This increased processing time was due to the three different window settings on which the fuzzy segmentation method operated.

6.4.3 Segmentation Accuracy

Overall, the modifications made to the fuzzy segmentation method showed improvements to the results. Both the MFP and MFN errors were lower and a paired statistical t-test measuring the difference between the two versions had a p-value < 0.05 , thus demonstrating that this improvement is significant.

Seven of the test cases had MFP errors less than 10%; two test cases had MFP errors between 10% and 15%; and the remaining test case had an MFP error of 22%. Six of the test cases had MFN errors less than 10%; two had MFN errors between 13% and 16%; and the remaining two had MFN errors over 35%.

Table 15: Accuracy evaluation of the fuzzy segmentation algorithm. Mean False Positive (MFP) when non-tumor pixels were selected and Mean False Negative (MFN) when tumor pixels were not selected for all 10 test cases. The results for “Version 1” are for the fuzzy method using a single window setting, while “Version 2” uses three window settings to generate a merged region.

#	Patient	Cat	Tumor description	Version1		Version2	
				MFP	MFN	MFP	MFN
1	P1	A	Cystic Teratoma on left ovary	3.3%	1.8%	2.9%	1.6%
2	P2	A	MucinousCystadenoma on right ovary	2.8%	3.5%	2.1%	3.0%
3	P3	A	Immature Teratoma	5.2%	3.7%	4.5%	2.9%
4	P4	B	Neuroblastoma, small and diffuse	6.9%	5.6%	5.1%	4.7%
5	P5	B	Neuroblastoma	10.3%	8.2%	8.6%	7.5%
6	P5	C	Neuroblastoma	12.7%	18.4%	10.2%	15.7%
7	P6	B	Neuroblastoma, some calcification	9.3%	9.6%	8.7%	7.2%
8	P6	C	Neuroblastoma, some calcification	23.5%	50.2%	21.9%	48.3%
9	P6	C	Neuroblastoma, highly calcified	5.7%	38.1%	4.1%	35.7%
10	P7	C	Neuroblastoma	14.9%	14.2%	14.3%	13.8%

The high MFP for test cases #8 and #10 meant more healthy tissues were mistaken as tumor tissue. The false positives are most likely a result of the poor delineation between the tumor and the surrounding tissues leading to a region leak as seen in some of the graphical representation of the test case results (notably in test case #8). The high MFN means that some tumor tissues were not segmented as part of the tumor. Calcification and the inhomogeneity of the tissue density within the tumor produced low similarity values for some pixels during the segmentation procedure.

Once again, the segmentation results obtained were compared to the tumor segmentation method developed by Deglint *et al* [120] to evaluate the performance of the fuzzy segmentation algorithm. As mentioned in the results discussion on the probabilistic

segmentation method in Chapter 6.3.2, eight of their ten test cases had less than 12% MFN error, while the remaining two test cases had MFN between 20% and 30%. For MFP error rates, Deglint's method was able to successfully segment the tumors on three test cases with less than 20% MFP errors, five test cases had MFP errors between 35% and 50%, and the remaining two test cases had MFP errors of more than 50%. As with the probabilistic segmentation results, the fuzzy segmentation method also had a lower MFP error rate on more test cases and with the worst MFP error rate at 22% on one test case, which is less than half of what Deglint's worst result (>50%) produced. The fuzzy segmentation method did just as well as Deglint's method on the MFN error rates for most of the test cases. For the fuzzy segmentation, eight test cases had MFN errors below 16%, while Deglint had eight test cases with error rates less than 12%. In the two test cases for the fuzzy segmentation, the MFN error rates are between 35% and 48%, 14% more false negatives than Deglint's method. Similar to the probabilistic segmentation method, the time taken to process a single slice using the fuzzy segmentation method is also significantly faster than the time taken by Deglint's method.

Based on the statistical results and the comparison with an established method, the fuzzy segmentation method performed very well, successfully segmenting the tumor from the healthy tissues in most of the test cases. The high MFP for test case #6 meant that more healthy tissues were mistaken as tumor tissues for that one case. This high false positive error rate is most likely a result of poor delineation between the tumor and the surrounding tissues leading to a region leak. The low MFN for most of the test cases demonstrated that the method was able to perform without mistaking healthy tissues as tumor tissues. Calcification and the inhomogeneity of the tissue density within the tumor

caused some pixels to be unselected during the segmentation procedure, resulting in a high MFN values for two test cases.

6.4.4 *Inter- and Intra-User Variability*

This section discusses the repeatability factor of the fuzzy segmentation algorithm. Similar to the statistical testing of variability for the probabilistic segmentation method, the segmentation results of selected pairs were compared. The results in Table 15 show that all pairs have a p-value > 0.05 , indicating that there is no significant difference between the pairs of users in terms of the segmentation results obtained. The statistical analysis to test for intra-user variability also produced a p-value > 0.05 , indicating that there was no significant difference between the multiple segmentation results of the same dataset as performed by the same user.

Table 16: Statistical analysis of inter-user variability for fuzzy segmentation method.

Testing Pairs	p-value (n=4)
User1-User2	0.2319
User2-User3	0.3157
User3-User4	0.3792
User4-User5	0.2904
User5-User6	0.1625
User6-User7	0.3933
User7-User8	0.3271
User8-User9	0.3614
User9-User10	0.3735
User10-User1	0.2912

6.5 Comparison of Both Segmentation Methods

Both methods discussed in this dissertation have some common similarities in that both are based on the spatial and intensity features to segment tumors. The difference lies in how both methods utilize this feature information to generate values for the segmentation process.

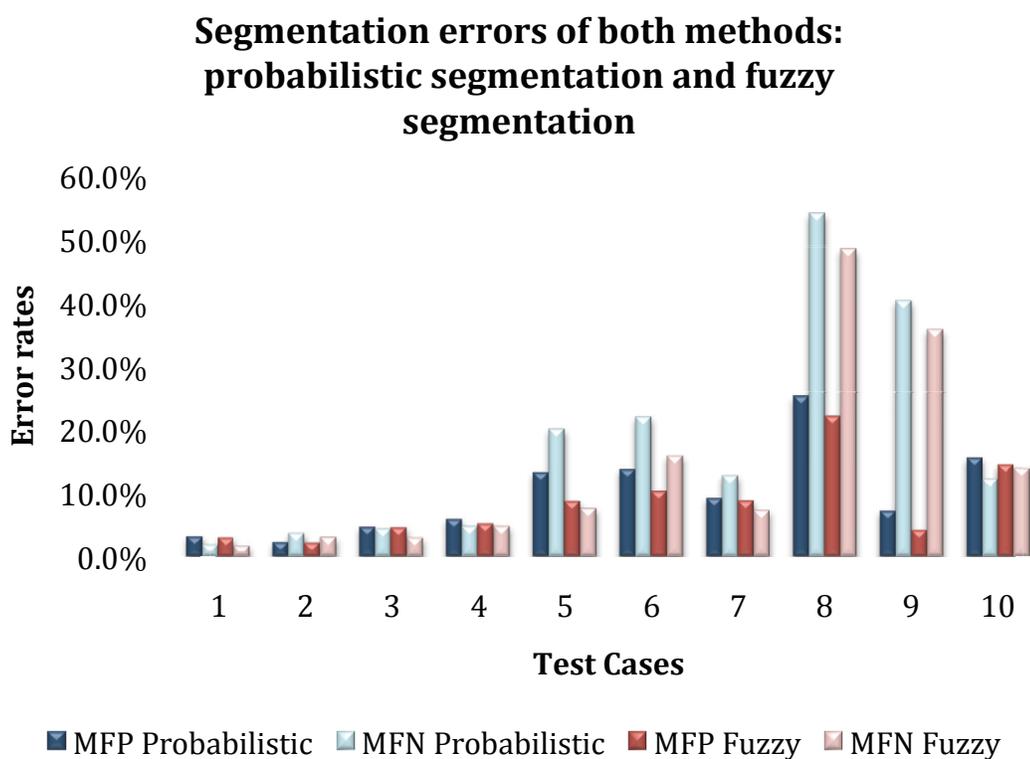


Figure 53: Performance comparison of both segmentation methods: MFN and MFP errors for probabilistic segmentation method, and MFN and MFP errors for fuzzy segmentation method.

The results of these two methods are also very different and the chart in Figure 53 shows that for most of the test cases, the fuzzy segmentation method performed with better accuracy when compared to the probabilistic segmentation method. The MFP errors for every test case were either slightly or significantly lower, the MFN errors also

demonstrate the same behavior, except for test cases #6 and #10 where the MFN errors for the fuzzy segmentation method were higher than that of the probabilistic segmentation method.

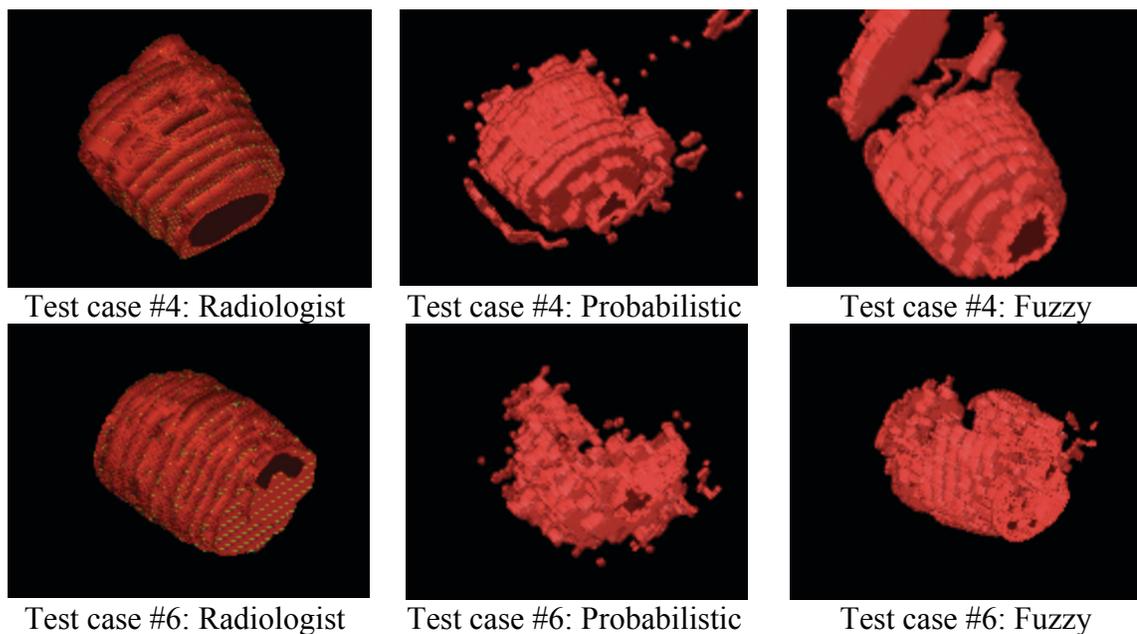


Figure 54: Comparison of both segmentation results with radiologist's manual segmentation for test cases #4 (category B) on the top row and #6 (category C) on the bottom row.

Figure 54 compares the visual results of both segmentation methods with the manually segmented results by a radiologist. For test case #4 shown in the top row of the figure, there are minor improvements shown by the fuzzy segmentation algorithm with approximately 1% improvement in both MFP and MFN error rates. The segmentation results for both methods agree with the manually segmented tumor by a radiologist visually as all three volume renders are very similar in both shape and size. The improvements demonstrated by the fuzzy segmentation algorithm (shown in the bottom row of the figure) is more apparent for test case #6, which is a category C tumor, with

approximately 4% improvements in the MFP error rate and 7% improvement in the MFN error rate. In the volume renders of the probabilistic and fuzzy segmentation results, the result of the fuzzy segmentation was able to segment the tumor more completely. The overall shape and size of the segmented tumor is very similar to that segmented by the radiologist, except for some false negative errors on the top of the tumor, where parts of were not segmented properly. Automatic segmentation on a category C tumor is a challenging problem but the improvements demonstrated by the fuzzy segmentation show promising results for further investigation and research.

6.6 Surgery Planning in Isis

6.6.1 Tumor Inspection

As previously mentioned, tumors segmented using the segmentation methods discussed can also be loaded and viewed in context with the original patient data for use in diagnostics or surgery planning. Dr Thom Lobe, a pediatric surgeon from Blank Children's Hospital at Des Moines, Iowa, was invited to qualitatively evaluate the functionality of the software application.

Combined with the other features built into Isis, multiple three-dimensional views of the tumor can be obtained and examined. In Figure 54, the location of the tumor relative to the spinal cord and pelvic bone can be clearly seen, which can be crucial visual information for surgeons.

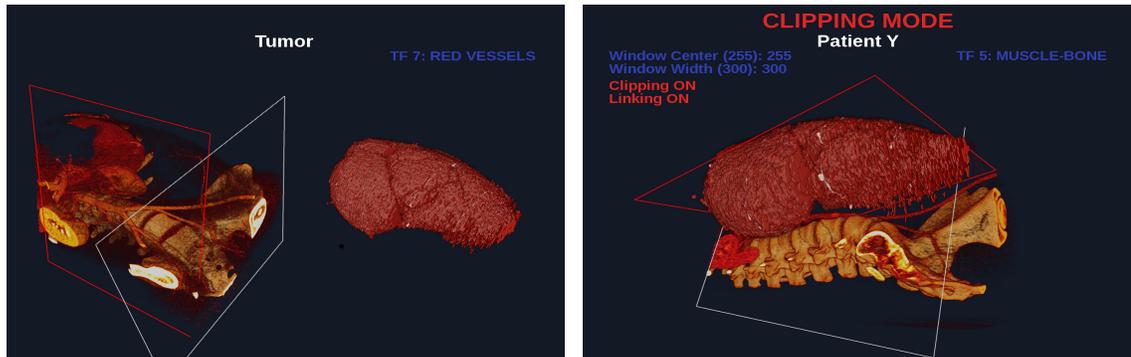


Figure 55: Tumor volume and patient data viewed separately (left), and tumor model linked to a windowed patient model (right).

When viewing the tumor model on its own, Dr Lobe was able to inspect the approximated shape and size of the tumor. For example, Figure 55 shows how linking a tumor to a patient model that has been windowed down to bone and muscle tissues allowed Dr. Lobe to examine the aortic valves and vascular structure in close proximity to the tumor, which he said would be a critical part to a surgical procedure on this patient. This information would otherwise be extremely difficult to conceptualize in a two-dimensional slice of the patient dataset. Viewing this tumor-patient combination model in a three-dimensional environment provides the surgeon with multiple vantage points and spatial information. Dr. Lobe's response was enthusiastic that Isis could become an invaluable resource for surgical planning. Based on the meeting, plans for planning actual surgeries using Isis are currently being developed.

6.6.2 *Simulated Endoscope Camera View*

Figure 56 shows a patient model as viewed from the simulated endoscopic camera mode, where the surgeon can navigate using the gamepad controls for a close-up view inside the patient model. This view is designed to simulate an endoscopic camera view,

where the surgeon can rotate the “camera” around a fixed position for a 360° view or reposition the camera anywhere inside the patient model. Figure 55 shows the camera placed in the abdomen, pointed towards the aortic bifurcation and spine.

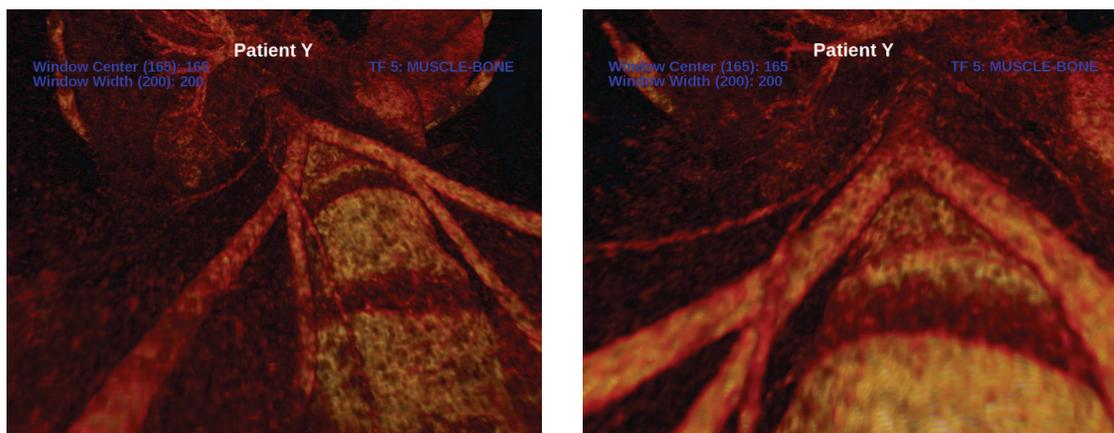


Figure 56: Simulated endoscopic camera inside the patient model.

The joystick controls correspond to the controls of most of today’s video games, where the left joystick moves the “player” front/back and left/right, while the right joystick controls the player’s viewpoint, rotating it up/down and left/right. With a short learning curve, surgeons can focus on the examination of the patient model instead of struggling to understand the navigational controls of the software.

When viewed in a virtual reality environment, this software framework puts the surgeon inside the patient model, thus removing the “soda-straw” effect when viewing through a standard camera. This representation is extremely useful as a planning tool to determine the optimal placements of an endoscope and other laparoscopic instruments. If camera placement is not correct, the critical organs may not be in the field of view of the surgeon, requiring additional incisions and more time for the procedure. Using Isis,

surgeons can virtually test out different endoscopic camera positions on actual patient data to better ensure correct placement.

6.7 Performance Evaluation of Isis

To evaluate the performance of the Isis visualization software, two measures were considered: 1) loading time based on different dataset sizes, and 2) frame rate measures relative to dataset size for user interaction. These measures represent the speed and efficiency of the application in handling the complex and large amounts of medical image data during interaction and visualization.

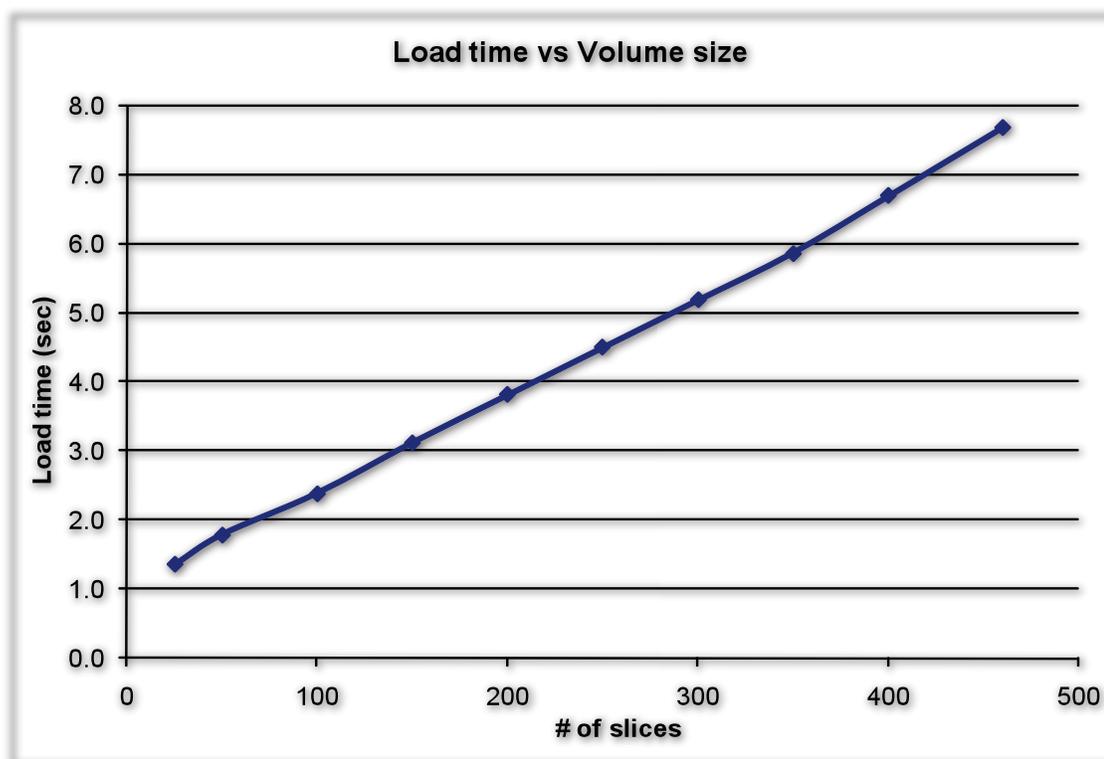


Figure 57: Loading time of different volume sizes.

The average load time for datasets of varying sizes were obtained and shown in Figure 57. These timings shown only measure the time it takes for the application to read in the DICOM files and generate a three-dimensional volume model on a single desktop machine. As the size of the dataset increases, the loading time increases linearly as well. For the largest dataset (460 slices) tested, the average load time was approximately 7.84 seconds. The application will take longer to launch on a clustered computer system because many 48 processors require time to cache the entire volume dataset onto memory. The bottleneck of the cluster network connection increases the loading time since large amount of data are being transferred.

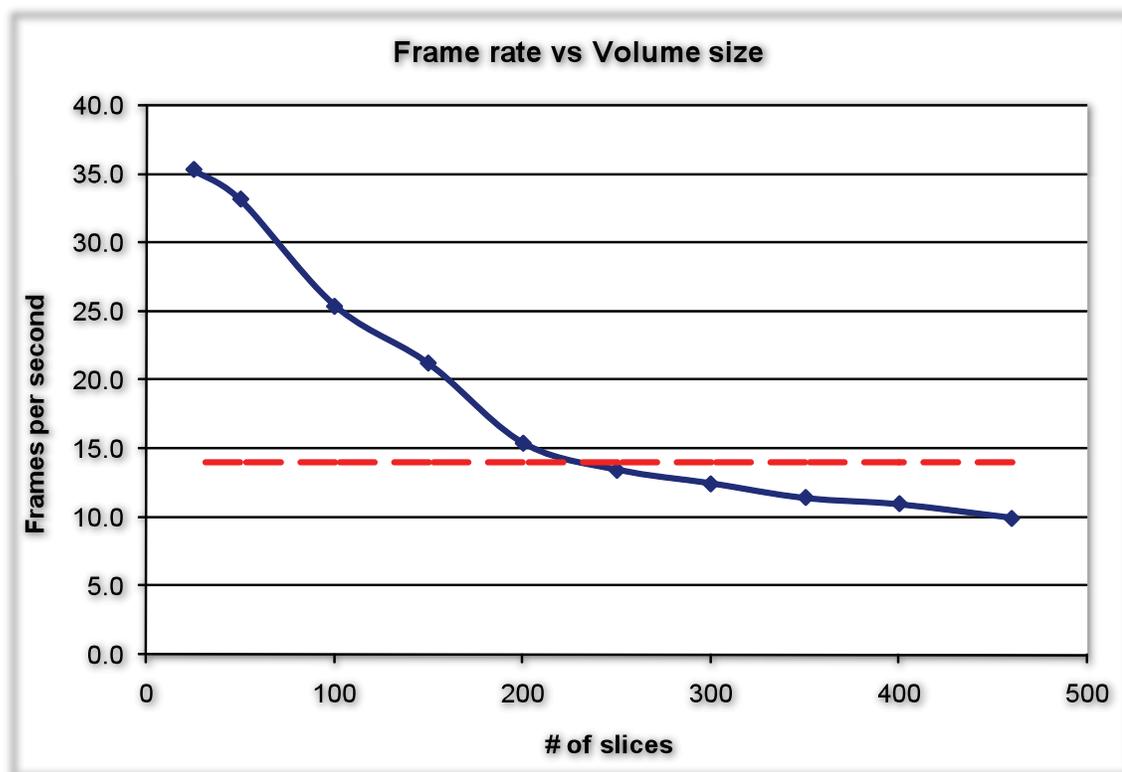


Figure 58: Plot of the frame rate (in stereo) as the number of slices being loaded increases. Minimum requirement for user interaction shown as dashed red line at 14 frames per second.

Figure 58 presents a graph of the frame rates when data of different sizes are loaded into the application and viewed in stereo. As expected, there is a tradeoff in effect when the frame rate decreases as the size of the data increases. According to Burdea and Coiffet [123], users will start to experience discomfort when the frame rate drops below 14 frames per second in an interactive immersive environment, indicated by the red dashed line in Figure 58. This threshold is crossed when approximately 230 slices are loaded into the application. The frame rate continues to decrease at a steady rate as the size of the dataset increases and when the largest dataset was loaded, at 460 slices (approximately 120 million voxels), the frame rate decreased to 9.93 frames per second. At a refresh rate below 10hz, a user will experience “jitter” when interacting with the model in a virtual environment [124].

7 CONCLUSIONS AND FUTURE WORK

7.1 Summary and Conclusions

A new framework for improving patient diagnosis and treatment has been developed, which includes:

- A probabilistic tumor segmentation algorithm, with automatic seed relocation.
- A fuzzy logic based tumor segmentation algorithm, with adaptive membership functions and fuzzy rules.
- A visual system for examining and manipulating large-scale medical image data in an immersive virtual environment, with full interaction using a wireless gamepad.

The first segmentation method is a probabilistic segmentation process to allow tumor extraction from a CT medical study. The segmentation process implements a probabilistic selection process developed using the selection process from the Simulated Annealing optimization algorithm as a foundation. This process allows pixels to be segmented as the object of interest based on a probability formulation. In addition, an automated seed and search region selection process was developed to allow processing of multiple image slices automatically as an object's size, shape, and location changes between subsequent slices. Apart from the first slice in the dataset, where the user manually selects the seed and search region for segmentation, the method performs automatically for all other slices. From the test cases presented, the automated seed

selection process was efficient in searching for new seed locations, as the object changed size, location, and orientation in each slice of the study. From the results, the resulting objects segmented were successfully identified and selected. Furthermore, the method requires information from only a two-dimensional image data at a time to accommodate performance on a regular personal computer.

The second segmentation method developed is based on a fuzzy logic system, which successfully extracted tumors from three-dimensional CT image data automatically. This method required input from the user only for the first slice of the dataset, where a region of interest was selected to initialize the segmentation process. For subsequent slices in the set, the method automatically updates the ROI for the current slice based on the segmented region of the previous slice to segment the tumor from the current slice. The membership functions used for fuzzy inference were set up to adapt to changes in the size and intensity of the tumor being segmented, between consecutive slices within the dataset. The method segments the dataset three times, using three different window settings and the results are then combined to generate a final merged segmented region. Based on the results obtained, the extracted object is visually acceptable with the shape and size of the tumors clearly defined.

Analysis of the segmentation results showed that both algorithms were successful in segmenting the tumor from seven of the ten CT datasets with less than 10% false positive errors and five test cases with less than 10% false negative errors. The consistency of the segmentation results statistics also showed a high repeatability factor, with low values of inter- and intra-user variability for both methods. Using information based on spatial and intensity features alone was sufficient to accurately segment some of

the more circumscribed tumors with relatively homogeneous tissue densities. Although both methods had more than 40% false negative errors on two test cases, this was comparable to the results obtained from current published methods. In addition, these two test cases are very difficult to segment due to the mixed tissue densities within the tumor and the presence of calcifications. When compared to current segmentation methods, the developed methods performed faster with comparable accuracy in segmentation results. The efficiency from only segmenting the data one slice at a time allows a user to run both segmentation algorithms on any current laptop and desktop computers.

Based on the current research reviewed, the developed interactive immersive visualization system is unique and new. The ability to interact and visualize actual patient data in an immersive virtual environment can be used as an effective tool for information gathering prior to surgery, an aid in diagnostics, or patient education. The ease of use of the gamepad controls provides an intuitive interaction between the user and the virtual environment and a menu system provides sufficient information to any new to properly operate Isis. Pre-programmed shortcut buttons are also available for the experience users. Combined with the segmentation results obtained from either one of the proposed segmentation methods, this framework will be a valuable tool in assisting surgeons during diagnosis as well as pre-operative planning of tumor removal procedures. The ability to inspect the patient data, segmented tumor, or both in an immersive environment will be advantageous from a visual information standpoint.

7.2 Future Work

For future development of the segmentation methods, the focus will be on improving the algorithm to reduce the error rates when the methods are used on tumors that have mixed tissues and those that are highly heterogeneous with fuzzy edges. One approach would be to first exclude organs and structures that could potentially interfere with the segmentation procedure, thus reducing the errors caused by region leaks [120]. By first segmenting and removing commonly known structures such as bone, skin, and air from the dataset improves the delineation of the tumor and prevents these structures from interfering with the segmentation procedure. Connectivity properties between pixels, such as fuzzy connectedness methods, to influence the segmentation process could also be explored. These connectivity properties will help reduce the false negative error rates from under-segmentation of tissues inside the tumor due to mixed tissue densities. The use of level set or deformable methods to include shape information will help define the resulting segmented tumors better, especially on tumors that have blurred edges and are heterogeneous. Another possible improvement would be to convert the process to perform in a parallel computing environment to reduce computational time so that multiple pixels could be examined simultaneously.

The visualization application is still currently in the development stage, with several additional features and functionalities planned. Current development work includes improvements to the surgical planning components, such as additional controls and interactions with the simulated endoscopic camera and the ability to segment the colon or a selected region of the airway for fixed-path navigation as a full virtual

endoscopy tool. User studies will also be performed to evaluate the effectiveness and usability of the gamepad controls and the menu interface. In addition, the usability of Isis compared to similar desktop medical image data visualization applications such as OsiriX and VolView will also be performed and evaluated. Currently, there are plans and discussions with Dr Thom Lobe, pediatric surgeon from Blank's Children Hospital in Des Moines, Iowa, to use Isis as an aid to his surgical planning. This application is intended to foster the widespread use of advanced visualization in preoperative planning, surgical training, and diagnostic assistance.

7.3 Acknowledgements

The work presented in this dissertation would not have been possible without the support and assistance from many people, to whom I am grateful and indebted to. I would like to take this opportunity to express my gratitude and thanks to everyone involved.

First and foremost, I'd like to thank my family for their love, understanding, and support all these years of graduate school. And to my advisor, mentor, and friend, Dr Eliot Winer, thank you for your guidance in this research and for giving me the strength and confidence to reach my goals and ambitions.

To Dr Thom Lobe, pediatric surgeon from Blank Children's Hospital at Des Moines, Iowa, and Dr Go Miyano, pediatric surgeon from Juntendo University, School of Medicine, Tokyo, Japan, thank you for your valuable time and input, which has been essential to the development of the segmentation methods and Isis.

To Dr Raj Rangayyan and his research lab at University of Calgary, Alberta, Canada, thank you for providing the neuroblastoma CT datasets, manually segmented contours of the tumor by radiologist Dr Graham Boag, as well as the segmentation results of their developed method. Also a special thanks to Dr Jeremy Friese from Brigham and Women's Hospital, Boston, Massachusetts for providing the manual contours of tumors from three CT datasets.

To the faculty and staff of the Virtual Reality Applications Center (VRAC) at Iowa State University, thank you for your never-ending support in keeping the lab running and meeting our everyday computing needs.

And last but not least, my research colleagues Vijay Kalivarapu, Andrew Koehring, Kenny Kopecky, Marisol Martinez, Brett Nekolny, Brandon Newendorp, Christian Noon, Catherine Peloquin, Alex Renner, and Levi Swartzentruber, who have been helpful in running the various test cases required for the segmentation algorithms.

REFERENCES

- [1] "How Does CT Work?" from *Imaginis - Medical Procedures*. http://imaginis.com/ct-scan/how_ct.asp (Available April 20 2008)
- [2] Joseph P. Hornak. "The Basics of MRI" <http://www.cis.rit.edu/htbooks/mri/> (Available April 20 2008)
- [3] "The DICOM Standard" from *NEMA - National Electrical Manufacturers Association*. <http://medical.nema.org/dicom> (Available April 20 2008)
- [4] Sonka M, Hlavac V, Boyle R: *Image Processing, Analysis, and Machine Vision*. PWS Publishing, Pacific Grove, 1999.
- [5] Gonzalez RC, Woods RE: *Digital Image Processing*. Addison-Wesley Publishing Company, Reading, MA, USA, 1993.
- [6] Pham DL, Prince JL, and Dagher AP: An Automated Statistical Characterization of Brain Tissues in Magnetic Resonance Imaging. *International Journal of Pattern Recognition and Artificial Intelligence* 11:1189-1211, 1997.
- [7] Rajapakse JC, Giedd JN, and Rapoport JL: Statistical Approach to Segmentation of Single-Channel Cerebral MR Images. *IEEE Trans. Med. Imaging* 16:176-186, 1997.
- [8] Neumann A and Lorenz C: Statistical Shape Model Based Segmentation of Medical Images. *Computer Medical Image Graphics*, 22:133-143, 1998.
- [9] Vincken KL, Koster A, and Viergever MA: Probabilistic multiscale image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(2) 1997
- [10] Kupinski MA and Giger ML: Automated seeded lesions segmentation on digital mammograms. *IEEE Trans. Med. Imaging*, 17(4) 1998.
- [11] Qian Q, Titterton DM, "On the use of Gibbs Markov Chain Models in the Analysis of Images Based on Second-Order Pairwise Interactive Distributions," *J. Applied Statistics*, 16(2):267-281, 1989.
- [12] Figueiredo MT, Leita JM, "Bayesian Estimation of Ventricular Contours in Angiographic Images," *IEEE Trans. Med. Img.*, 11(3):416-429, September 1992.
- [13] Bell ZW, "A Bayesian Monte Carlo Segmentation Method for Images Dominated by Gaussian Noise," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(9):985-990, September 1989.
- [14] Bouman CA, Shapiro M, "A Multiscale Random Field Model for Bayesian Image Segmentation," *IEEE Trans. Img. Proc.* 3(2):162-177, March 1994.
- [15] Dubes RC, Jain AK, Nadabar SG, Chen CC, "MRF Model Based Algorithms for Image Segmentation," *Proc. Of 10th IEEE Pattern Recognition 1990*, pp. 801-814, 1990.
- [16] Bello MG, "A Combined Markov Random Field and Wave Packet Transform Based Approach for Image Segmentation," *IEEE Trans. Img. Proc.* 3(6):834-846, 1994.
- [17] Tu Z, Zhu SC, "Image Segmentation by Data-Driven Markov Chain Monte Carlo," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(5):657-673, May 2002.
- [18] Zhang J, Modestino JW, "Maximum-Likelihood Parameter Estimation for Unsupervised Stochastic Model-Based Image Segmentation," *IEEE Trans. Img. Processing*, 3(4):404-420, July 1994
- [19] Zhang J, Modestino JW, "A Model-fitting Approach to Cluster Validation with Application to Stochastic Model-Based Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(10):1009-1017, October 1990.
- [20] Chou P, Brown CM, "The Theory and Practice of Bayesian Image Labeling," *Intl. J. Computer Vision*, 4(3):185-210, June 1990.
- [21] Marroquin JL, Santana EA, Botello S, "Hidden Markov Measure Field Models for Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(11):1380-1387, November 2003.

- [22] Ruan S, Moretti B, Fadili J, Bloyet D, "Fuzzy Markovian Segmentation in Application of Magnetic Resonance Images," *Computer Vision and Image Understanding*, 85(1):54-69, 2002.
- [23] Wilson R, Li CT, "A Class of Discrete Multiresolution Random Fields and its application to Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(1):42-56, January 2003.
- [24] Leemput KV, Maes F, Vandermuehlen D, Suetens P, "A Unifying Framework for Partial Volume Segmentation of Brain MR Images," *IEEE Trans. Med. Img.*, 22(1):105-119, January 2003.
- [25] Cheng H, Bouman CA, "Multiscale Bayesian Segmentation Using a Trainable Context Model," *IEEE Trans. Img. Proc.* 10(4):511-525, April 2001.
- [26] Chen JL, Gunn SR, Nixon MS, Gunn RN, "Markov Random Field Models for Segmentation of PET Images," *Image Processing and Medical Imaging 2001*, pp. 468-474, 2001.
- [27] D'Elia C, Poggi G, Scarpa G, "A Tree-Structure Markov Random Field Model for Bayesian Image Segmentation," *IEEE Trans. Img. Proc.*, 12(10):1259-1273, October 2003.
- [28] Choi H, Baraniuk RG, "Multiscale Image Segmentation Using Wavelet-Domain Hidden Markov Models," *IEEE Trans. Img. Proc.* 10(9):1309-1321, September 2001.
- [29] Poggi G, Ragozini ARP, "Image Segmentation by Tree-Structured Markov Random Fields," *IEEE Signal Processing Letters*, 6(7):155-157, July 1999.
- [30] Shaffrey CW, Kingsbury NG, Jermyn IH, "Unsupervised Image Segmentation via Markov Trees and Complex Wavelets," *Proc. Image Processing 2002*, pp. 801-804, 2002.c
- [31] Li J, Najmi A, Gray RM, "Image Classification by a Two-Dimensional Hidden Markov Model," *IEEE Trans. Signal Proc.*, 48(2):517-533, February 2000.
- [32] Melas DE, Wilson SP, "Double Markov Random Fields and Bayesian Image Segmentation," *IEEE Trans. Signal Proc.* 50(2):357-365, February 2002.
- [33] Zhang Y, Brady M, Smith S, "Segmentation of Brain MR Images Through a Hidden Markov Random Field Model and the Expectation-Maximization Algorithm," *IEEE Trans. Med. Img.* 20(1):45-57, January 2001.
- [34] Davatzikos CA, and Prince JL: An active contour model for mapping the cortex. *IEEE Trans. Med. Img.* 14(1):65-80, 1995.
- [35] Lobregt S, and Viergever MA: A discrete dynamic contour model. *IEEE Trans. Med.Img.* 14(1):12-24, 1995.
- [36] Paulus D, Wolf M, Meller S, and Nieman H: Three-dimensional computer vision for tooth restoration. *Medical Image Analysis*, 3(1):1-19, 1999.
- [37] Caselles V, Kimmel R, and Sapiro G: Minimal surfaces based object segmentation. *IEEE Trans Patt. Anal. And Mach. Intel.* 19(4), 1997.
- [38] McInerney T, and Terzopoulos D: Topology adaptive deformable surfaces for medical image volume segmentation. *IEEE Trans. Med. Img.* 18(10):840-850, 1999.
- [39] Xu C, and Prince JL: Snakes, shapes, and gradient vector flow. *IEEE Trans. Img. Proc.* pp. 359-369, 1998.
- [40] Varrett WA, and Mortensen EN: Interactive live-wire boundary extraction. *Med. Img. Anal.* 1(4):331-341, 1996.
- [41] Liang J, McInerney T, Terzopoulos D: United snakes (towards an more perfect union of active contour techniques.) *IEEE Seventh Intl. Conf. on Computer Vision*, Kerkyra, Greece, 1999.
- [42] Montagnat J, and Delingette H: Volumetric medical image segmentation using shape constrained deformable models. *Joint Conf. Computer Vision, Virtual Reality and Robotics in Medicine and Medical Robotics and Computer-Assisted Surgery*, Grenoble, France. 1997.
- [43] Leemput KV, Maes F, Vandermeulen D, and Suetens P: Automated Model-Based Tissue Classification of MR Images of the Brain. *IEEE Transactions on Medical Imaging*, 18(10):897-908, 1999.
- [44] Aboutas GB, and Dawant BM: Automatic Brain Segmentation and Validation: Image-based versus Atlas-based Deformable Models. *SPIE Proc. Med. Imaging*, 3034:299-310, 1997.
- [45] Sandor S, and Leahy R: Surface-based Labeling of Cortical Anatomy Using A Deformable Atlas. *IEEE Trans. Med. Imaging*, 16:41-54, 1997.
- [46] Anbeek P, Vincken KL, van Osch MJP, Bisschops RHC, and van der Grond J: Probabilistic segmentation of white matter lesions in MR imaging. *NeuroImage*, 21:1037-1044, 2004.

- [47] Lorenzo-Valdes M, Sanchez-Ortiz GI, Mohiaddin R, and Rueckert D: Segmentation of 4D cardiac MR images using a probabilistic atlas and the EM algorithm. *Medical Image Computing and Computer-Assisted Intervention - MICCAI*, pp. 440-450, 2003.
- [48] Antonie M, Zaiane OR, and Coman A: Application of Data Mining Techniques for Medical Image Classification. Proceedings of the Second International Workshop on Multimedia Data Mining (MDM/KDD'2001). San Francisco, CA, USA. August 26, 2001.
- [49] Kupinski MA and Giger ML: Automated seeded lesions segmentation on digital mammograms. *IEEE Trans. Med. Imaging*, 17(4) 1998.
- [50] Wang Y, Adali T, Kung S, and Szabo S: Quantification and segmentation of brain tissues from MR images: A probabilistic neural network approach. *IEEE Trans. Image Processing*, 7(8) 1998.
- [51] Wells WM, Grimson WEL, Kikinis R, Jolesz FA: Adaptive Segmentation of MRI Data. *IEEE Trans. Medical Imaging*, 15(4) 1996.
- [52] de Ridder D, Duin RPW, Verbeek PW, et al.: The applicability of neural networks to non-linear image processing. *IEEE Trans. Pattern Anal. Appl.* 2(2):111-128, 1999.
- [53] Leemput KV, Maes F, Vandermeulen D, and Suetens P: Automated Model-Based Tissue Classification of MR Images of the Brain. *IEEE Transactions on Medical Imaging*, 18(10):897-908, 1999.
- [54] Kecman V: *Learning and Soft Computing*, Chap 4 Single Layer Networks, pp. 193-254, MIT Press, Cambridge, MA, USA, 2001.
- [55] Ma F, Wang W, Tsang WW, Tang Z, Xia S, and Tong X: Probabilistic segmentation of volume data for visualization using SOM-PNN classifier. *Symposium on Volume Visualization*, pp. 71-78, 1998.
- [56] Reddick WE, Glass JO, Cook EN, Elkin TD, and Deaton RJ: Automated Segmentation and Classification of Multispectral Magnetic Resonance Images of Brain Using Artificial Neural Networks. *IEEE Trans. Med. Imaging*, 16:911-918, 1997.
- [57] Zadeh LA: Fuzzy sets. *Information and Control* 8:338-353, 1965.
- [58] Kecman V: *Learning and Soft Computing*, Chap 6 Fuzzy Logic Systems, pp. 365-420, MIT Press, Cambridge, MA, USA, 2001.
- [59] Udupa J, and Samarasekera S: Fuzzy connectedness and object definition - theory, algorithms and applications in image segmentation. *Graphical Models and Image Processing*, 58(3):246-261, May 1996.
- [60] Saha PK and Udupa JK: Fuzzy Connected and Image Segmentation. *Proceedings of IEEE* 91(10):1649-1669, 2003.
- [61] Lei T, Udupa JK, Saha PK, and Odhner D: Artery-vein separation via MRA-An image processing approach. *IEEE Trans. Medical Imaging*, 20(8):689-703, 2001.
- [62] Liu J, Udupa JK, Odhner D, Hackney D, and Moonis G: A system for brain tumor volume estimation via MR imaging and fuzzy connectedness. *Computerized Medical Imaging and Graphics*, 29(1):21-34, 2005.
- [63] Zhuge Y, Udupa JK, and Saha PK: Vectorial scale-based fuzzy-connected image segmentation. *Computer Vision and Image Understanding*, 101(3):177-193, 2006.
- [64] Herman GT and Carvalho BM: Multiseeded segmentation using fuzzy connectedness. *IEEE Trans. Pattern Anal. and Machine Intel.* 23(5):460-474, 2001.
- [65] Pednekar AS, and Kakadiaris IA: Image segmentation based on fuzzy connectedness using dynamic weights, *IEEE Trans. Image Processing*, 15(6):1555-1562, 2006.
- [66] Nyul LG, Falcao AX, and Udupa JK: Fuzzy-connected 3D image segmentation at interactive speeds. *Graphical Models*, 64(5):259-281, 2002.
- [67] Dunn JC: "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", *Journal of Cybernetics* 3: 32-57, 1973
- [68] Zhu C, and Jian T: Multicontext fuzzy clustering for separation of brain tissues in magnetic resonance images. *NeuroImage*, 18(3):685-696, 2003.
- [69] Cinque L, Foresti G, and Lombardi L: A clustering fuzzy approach for image segmentation. *Pattern Recognition*, 37(9):1797-1807, 2004.

- [70] Szilagyi L, Benyo Z, Szilagyi SM, and Adam HS: MR brain image segmentation using an enhanced fuzzy C-means algorithm. Proc. 25th Intl. Conf. of IEEE Engineering in Medicine and Biology Society, pp. 724-726, Sept 2003.
- [71] Velthuizen RP, O'Hall L, Clarke LP, Bensaid AM, Arrington A, and Silbiger ML: Unsupervised fuzzy segmentation of 3D magnetic resonance brain images. Proc. SPIE Biomedical Image Processing and Biomedical Visualization, pp. 627-635, July 1993.
- [72] Herndon RC, Lancaster JL, Toga AW, and Fox PT: Quantification of white and gray matter volumes from T1 parametric images using fuzzy classifiers. J. Magnetic Resonance Imaging, 6(3):425-435, 1996.
- [73] Yao J, Miller M, Franaszek M, and Summers RM: Colonic polyp segmentation in CT colonography-based on fuzzy clustering and deformable models. IEEE Trans. Medical Imaging, 23(11):1344-1352, 2004.
- [74] Pham DL, and Prince JL: Adaptive fuzzy segmentation of Magnetic Resonance Images. IEEE Trans. Med. Img. 18(9):737-752, 1999.
- [75] Hachouf F, and Mezhoud N: An improved fuzzy rules-based segmentation system. Proc. 7th Intl Signal Processing and Applications, pp. 533-536, 2003.
- [76] Hata Y, Kobashi N, Kamiura N, and Ishikawa M: Fuzzy logic approach to 3D magnetic resonance image segmentation. Infor. Proc. In Medical Imaging, 1230:387-392, 1997.
- [77] Wirth MA, Lyon J, and Nikitenko D: A fuzzy approach to segmenting the breast region in mammograms. IEEE Annual Meeting of Fuzzy Information, pp. 474-479, June 2004.
- [78] Tolia YA, and Panas SM: On applying spatial constraints in fuzzy image clustering using a fuzzy rule-based system. IEEE SP Letters, 5(10):245-247, 1998.
- [79] Colliot O, Camara O, Dewynter R, Bloch I: Description of brain internal structures by means of spatial relations for MR image segmentation. Proc. of SPIE, 5370:444-455, 2004.
- [80] Toledo R, Xavier O, Petia R, et al.:Eigensnakes for vessel segmentation in Angiography. 15th Intl. Conf. on Pattern Recog. 2000.
- [81] Ackerman MJ: The Visible Human Project. Proc of IEEE, 86(3):504-511, 1998.
- [82] Amira: Visualize-Analyze-Present, <http://www.amiravis.com/> (Available April 29, 2008)
- [83] VitalImages: Vitrea Software, http://www.vitalimages.com/Solutions/Vitrea_Software.aspx (Available April 29, 2008)
- [84] Rosset A, Spadola L, and Ratib O: OsiriX: An Open-Source Software for Navigating in Multidimensional DICOM Images. J. of Digital Imaging, 17(3):205-216, 2004.
- [85] KitWare Volview, <http://www.kitware.com/products/volview.html> (Available April 19 2008)
- [86] Haluck RS, and Krummel TM: Computers and Virtual Reality for Surgical Education in the 21st Century. ARCH/SURG 135:786-793, 2000.
- [87] Meissner M, Lorensen B, Zuiderveld K, Simha V, Wegenkittl R: Volume Rendering in Medical Applications. IEEE Visualization 2002, Oct 27 - Nov 1, 2002, Boston, MA, USA.
- [88] Ferrucci JT: Colon Cancer Screening with Virtual Colonoscopy: Promise, Polyps, Politics. American Jour. Radiology 177:975-988, 2001.
- [89] Pescatore P, et al: Diagnostic accuracy and interobserver agreement of CT Colonoscopy. GUT 47:126-130, 2000.
- [90] Cotton PB, et al: Computer Tomographic Colonography: A Multicenter Comparison with Standard Colonoscopy for Detection of Colorectal Neoplasia. Jour. American Med. Assoc. 291(14):1713-1719, 2004.
- [91] Bartz D. Virtual Endoscopy in Research and Clinical Practice. Computer Graphics Forum, 24(1):111-126, 2005.
- [92] Immersion Corp Product List, <http://www.immersion.com/medical/products/laparoscopy/> (Available April 20 2008)
- [93] RealSims Systems Product List, <http://www.realsimsystems.com/products.htm> (Available April 20 2008)
- [94] The MedicalSim Group - Computer Enhanced Laparoscopic Training System, www.medicalsim.org/celts.htm (Available April 20 2008)

- [95] SimBionix - The LAPMentor, http://www.simbionix.com/LAP_Mentor.html (Available April 20 2008)
- [96] Mentice Corp - Minimally Invasive Surgical Trainier in Virtual Reality (MIST-VR), <http://www.mentice.com> (Available April 20 2008)
- [97] Schijven M, and Jakimowicz J: Virtual Reality Surgical Laparoscopic Simulators. *J. of Surgical Endoscopy* 17:1943-1950, 2003.
- [98] Gallagher AG, and Satava RM: Virtual Reality as a metric for the assessment of laparoscopic psychomotor skills. *J. of Surgical Endoscopy*, 16:1746-1752, 2002.
- [99] Gallagher AG, Lederman AB, McGlade K, Satava RM, and Smith CD: Discriminative validity of the Minimally Invasive Surgical Trainer in Virtual Reality (MIST-VR) using criteria levels based on expert performance. *J. of Surgical Endoscopy*, 2004.
- [100] Ahlberg G, Heikkinen T, Iselius L, Leijonmarck CE, Rutqvist J, and Arvidsson D: Does Training in a Virtual Reality Simulator Improve Surgical Performance? *J. Surgical Endoscopy*, 16:126-129, 2002.
- [101] Grantcharov TP, Kristianse VB, Bendix J, Bardram L, Rosenberg J, and Funch-Jensen P: Randomized Clinical Trial of Virtual Reality Simulation for Laparoscopic Skills Training. *British J. of Surgery* 91:146-150, 2004.
- [102] Bowman DA, and Hodges L: Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments. *J. Visual Languages and Computing*, 10(1):37-53, 1999.
- [103] Bowman DA, and Wingrave CA: Design and Evaluation of Menu Systems for Immersive Virtual Environments, *Proc Virtual Reality 2001 Conference*, 2001.
- [104] Jacoby R, and Ellis S: Using Virtual Menus in a Virtual Environment, *Visual Data Intepretation*, 1992.
- [105] Angus I, and Sowizral H: Embedding the 2D Interaction Metaphor in a Real 3D Virtual Environment, *SPIE Stereoscopic Displays and Virtual Reality Systems*, 1995.
- [106] Lindeman R, Sibert J, and Hahn J: Hand-held Windows - Towards Effective 2D Interaction in Immersive Virtual Environments, *IEEE Virtual Reality 1999*.
- [107] Gerber D, and Bechmann D: The Spin Menu - A Menu System for Virtual Environments. *Proc. IEEE Virtual Reality 2005*, pp. 271-272, 2005.
- [108] Larimer D, and Bowman DA: VEWL - A Framework for Building a Windowing Interface in a Virtual Environment. *Proc of Human-Computer Interaction- INTERACT 2003*, pp. 809-812, 2003.
- [109] Lucas JF, Kim JS, and Bowman DA: Resizing Beyond Widgets - Object Resizing Techniques for Immersive Virtual Environments. *Proc of Human Factors in Computing Systems*, pp. 1601-1604, 2005.
- [110] White M, Petridis P, Liarokapis F, and Plecinckx D: Multimodal Mixed Reality Interfaces for Visualizing Digital Heritage. *Special Issue Intl J. Architectural Computing (IJAC) in Cultural Heritage*, 5(2):322-337, June 2007.
- [111] Norman D: *The Design of Everyday Things*, MIT Press, 1998.
- [112] Shneiderman B: *Designing the User Interface*, 3rd ed., Addison-Wesley, 1998.
- [113] Kirkpatrick S, Gelatt Jr CD, and Vecchi MP: Optimization by Simulated Annealing. *Science*, 220(4598):671-680, May 1983.
- [114] Verdegay J, Nachtgeael M: *Fuzzy Sets Based Heuristics for Optimization*. Springer-Verlag, Aug 2003.
- [115] Mamdani EH, and Assilian S: An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *Intl J of Man-Machine Studies*, 7(1):1-13, 1975.
- [116] OFFIS Open Source DICOM Toolkit DCMTK, <http://dicom.offis.de/dcmtk.php.en> (Available April 20 2008)
- [117] The VRJuggler Suite, <http://www.vrjuggler.org> (Available April 20 2008)
- [118] Systems in Motion 3D Graphics Development Tool: Coin3D, <http://www.coin3d.org> (Available April 20 2008)
- [119] Systems in Motion 3D Graphics Development Tools: SimVoleon Plugin for Coin3D, <http://www.coin3d.org> (Available April 20 2008)

- [120] Sinha CK, Paramalingam S, Davenport M, Patel S, and Ade-Ajayi N: Feasibility of Minimally Invasive Surgery in Neonates. *Proceedings from International Pediatric Endosurgery Group Annual Congress for Endosurgery in Children*, Cannes, Frances, June 12-15 2008.
- [121] Meehan JJ, and Phearman LA: Robotic Surgery for Infants and Small Children: Is There Room For This? *Proceedings from International Pediatric Endosurgery Group Annual Congress for Endosurgery in Children*, Cannes, Frances, June 12-15 2008.
- [122] Deglint HJ, Rangayyan RM, Ayres FJ, Boag GS, and Zuffo MK: Three-Dimensional Segmentation of the Tumor in Computed Tomographic Images of Neuroblastoma. *J. Digital Imaging*, 20(1):72-87, March 2007.
- [123] Burdea G and Coiffet P: *Virtual Reality Technology*. New York, John Wiley & Sons, 1994.
- [124] Stuart R: *Design of Virtual Environments*, Barricade Books, August 2001.