

1-1-2010

Medical Image Segmentation Using a Genetic Algorithm

Payel Ghosh
Portland State University

Follow this and additional works at: https://pdxscholar.library.pdx.edu/open_access_etds

Let us know how access to this document benefits you.

Recommended Citation

Ghosh, Payel, "Medical Image Segmentation Using a Genetic Algorithm" (2010). *Dissertations and Theses*. Paper 25.

<https://doi.org/10.15760/etd.25>

This Dissertation is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

Medical Image Segmentation Using a Genetic Algorithm

by

Payel Ghosh

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
in
Electrical and Computer Engineering

Dissertation Committee:
Melanie Mitchell, Chair
Marek A. Perkowski
Dan Hammerstrom
James A. Tanyi
Martin Zwick

Portland State University
©2010

ABSTRACT

Advances in medical imaging technology have led to the acquisition of large number of images in different modalities. On some of these images the boundaries of key organs need to be accurately identified for treatment planning and diagnosis. This is typically performed manually by a physician who uses prior knowledge of organ shapes and locations to demarcate the boundaries of organs. Such manual segmentation is subjective, time consuming and prone to inconsistency. Automating this task has been found to be very challenging due to poor tissue contrast and ill-defined organ/tissue boundaries. This dissertation presents a genetic algorithm for combining representations of learned information such as known shapes, regional properties and relative location of objects into a single framework in order to perform automated segmentation. The algorithm has been tested on two different datasets: for segmenting hands on thermographic images and for prostate segmentation on pelvic computed tomography (CT) and magnetic resonance (MR) images. In this dissertation we report the results of segmentation in two dimensions (2D) for thermographic images; and two as well as three dimensions (3D) for pelvic images. We show that combining multiple features for segmentation improves segmentation accuracy as compared with segmentation using single features such as texture or shape alone.

*To my father
who instilled the love of mathematics in me.*

Acknowledgments

I would first like to thank my advisor, Dr. Melanie Mitchell, not only for her constant guidance and advice, but also her consideration for my personal situation which helped me complete part of this research work off-campus. I would like to express my sincere thanks to Dr. James Tanyi from OHSU. It was due to his immense help that the data from OHSU was acquired in a matter of weeks despite administrative issues. I am thankful to him and Dr. Arthur Hung for providing manual segmentations for the CT and MR data sets. I am also grateful to Dr. Judith Gold from Temple University for providing the second dataset of thermographic images. The dataset proved to be ideal for testing and evaluating the developed algorithm. I would also like to thank all my committee members for their valuable comments and feedback on my work.

Thanks to Intel corp. and the J. S. McDonnell foundation for funding this research. Thanks to the CS and ECE department's staff: Beth Phelps, Kathi Lee, Rene Remillard, Barbara Sabath, and Kelley Gardiner. Thanks to my colleagues Ralf Juengling, Lanfranco Muzi, Martin Cenek, Will Landecker, Mick Thomure and Karan Sharma for insightful discussions during some of the groups meeting that I attended. Finally, I would like to thank my family- my husband, my parents and my sisters, for their constant support and encouragement.

Contents

Abstract	i
Dedication	ii
Acknowledgments	iii
List of Tables	viii
List of Figures	x
List of Abbreviations/Symbols	xvii
1. Introduction	1
1.1 Problem description	3
1.2 Hypothesis.....	6
1.3 Overview of LSGA	7
1.4 Contributions of this dissertation	8
2. Literature Review	9
2.1 Segmentation methods	9
2.1.1 Pixel-based segmentation methods	10
<i>A. Laws' texture based segmentation method.....</i>	<i>12</i>
<i>B. Gabor wavelet transform-based segmentation method.....</i>	<i>15</i>
2.1.2 Shape-based segmentation methods	17
<i>A. Active contour method of segmentation</i>	<i>18</i>
<i>B. Level set methods for medical image segmentation</i>	<i>22</i>
<i>C. The Chan & Vese algorithm.....</i>	<i>23</i>
2.1.3 Incorporating spatial relationships for segmentation	25

A. <i>Fuzzy relative location maps</i>	25
2.2 Optimization methods in image processing.....	28
2.2.1 Traditional optimization methods	28
2.2.2 Stochastic optimization methods	30
2.2.3 Genetic algorithms	31
A. <i>Genotype and phenotype</i>	31
B. <i>Fitness</i>	32
C. <i>Selection</i>	32
D. <i>Crossover (Recombination)</i>	34
E. <i>Mutation</i>	35
F. <i>Termination criteria</i>	35
2.2.4 Evolutionary computation in image processing.....	36
A. <i>GENIE</i>	38
3. Method: LSGA.....	41
3.1 Training stage: Deriving shape model	41
3.2 Training stage: Deriving texture priors	46
3.3 Training stage: Deriving spatial relationships.....	47
3.4 Segmentation stage.....	51
3.4.1 Fitness: Combining shape and texture	52
3.4.2 Fitness: Combining shape, texture and spatial relationships	53
3.4.3 Selection, Crossover and Mutation	54
3.5 Extension to three dimensions	55
3.3.1 Steps for performing 3D segmentation	57

<i>A. Steps for training the 3D-LSGA</i>	57
<i>B. Steps for 3D-LSGA</i>	57
3.4 Summary	58
4. Segmentation of Thermographic Images of Hands.....	59
4.1 Data and problem description	59
4.1.1 Equipment for acquiring images	60
4.2 Experimental setup.....	62
4.3 Evaluation criteria.....	67
4.4 Results and discussion	68
4.5 Summary	74
5. Prostate Segmentation on Pelvic CT/MRI Images	75
5.1 Problem description	76
5.1.1 Acquisition of pelvic images and artifacts.....	77
<i>A. Computed Tomography (CT)</i>	78
<i>B. Magnetic Resonance Imaging (MRI)</i>	80
5.1.2 Margins in prostate cancer radiation-therapy treatment planning	82
5.2 Description of data	83
5.3 Segmentation of pelvic CT/MRI images.....	84
5.3.1 Segmentation using texture and shape priors.....	86
<i>A. 2D segmentation</i>	86
<i>B. 3D segmentation</i>	90
5.3.2 Segmentation using texture, shape and relative location priors.....	92
<i>A. 2DRL segmentation</i>	92

<i>B. 3DRL segmentation</i>	95
5.4 Evaluation criteria	95
5.4.1 Repeatability	95
5.4.2 Dice similarity coefficient and partial Hausdorff distance	96
5.5 Results and discussion	96
5.5.1 Comparison of segmentation algorithms	96
5.5.2 Comparison of LSGA in different modes 2D/2DRL/3D/3DRL.....	103
<i>A. Segmentation of CT images</i>	103
<i>B. Segmentation of MRI images</i>	114
5.6 Summary	126
6. Strengths and Limitations of LSGA and Future Work	127
7. Conclusion	130
References.....	131

List of Tables

2.1 One-dimensional convolution kernels (number 5 shows that it is a five-valued vector)	13
2.2 Convolution matrices derived from Laws' convolution kernels	13
4.1 Shape variability of hands (σ_1 and σ_2 are the two principal eigenvalues)	63
4.2 LSGA parameters	65
4.3 Comparison with ground truth using <i>DSC</i> . The bold values represent the best (highest values) in a row. Note that LSGA did not perform as well in the case of patient 4 due to the limitation in acquiring accurate ground truth as the fingers of this patient were completely invisible.	71
4.4 Comparison with ground truth using <i>H</i> . The bold values represent the best (lowest values) in a row. Note that <i>H</i> represents the point of maximum discrepancy between two contours. It is much lower in the case of LSGA as compared to other methods because it is the only method that tries to outline the fingers on these images	71
4.5 Header file created by the LSGA for a sample test image	72
5.1 LSGA parameters for 2D/2DRL segmentation	89
5.2 LSGA parameters for 3D/3DRL segmentation	90
5.3 Header file created by the LSGA for a sample MRI test image	100
5.4 Performance evaluation of the four protocols for segmenting the prostate on pelvic CT images	102
5.5 Performance evaluation of the four protocols for segmenting the prostate on pelvic MRI images	102

5.6 The header file generated by the 2D-LSGA for a single slice of CT image of patient A	104
5.7 The header file generated by the 2DRL-LSGA for a single slice of CT image of patient A	107
5.8 The header file generated by the 3D-LSGA for a CT image of patient A	111
5.9 The header file generated by the 3DRL-LSGA for a CT image of patient A	112
5.10 The header file generated by the 2D-LSGA for a single slice of MRI image of patient <i>B</i>	115
5.11 The header file generated by the 2DRL-LSGA for a single slice of MRI image of patient <i>B</i>	118
5.12 The header file generated by the 3D-LSGA for a MRI image of patient <i>B</i>	122
5.13 The header file generated by the 3DRL-LSGA for a MRI image of patient <i>B</i>	123
5.14 <i>DSC</i> and <i>H</i> values showing comparison of segmentation outcomes with the ground truth for the LSGA in 2D, 2DRL, 3D, 3DRL modes	125

List of Figures

1.1	Top panel: Thermographic image of the hands of a patient suffering from UEMSD. Bottom panel: Manual segmentation of this image.	4
1.2	Left panel: A 2D pelvic CT scan. The white contour in the center is the prostate that was outlined by an expert. The black region just below the prostate is the rectum. The white structures surrounding the prostate are the bones. Right panel: A 2D pelvic MRI scan with the manually segmented prostate outlined in the center..	5
1.3	The hypothesis for this research is to combine multiple features for segmentation using a GA	6
1.4	Flowchart depicting the sequence of operations of the LSGA	7
2.1	The steps in the segmentation process using Laws' texture measures	13
2.2	An example showing segmentation using Laws' texture measures. Top left: Original image. Top right: Manual segmentation. Bottom left: Linear combination of Laws' output planes. Bottom right: Final classification, white (tank), black (not tank).	15
2.3	An example segmentation using Gabor wavelet transform method. Left: Original image. Right: Final classification, white (tank), black (not tank).	17
2.4	A closed simple curve with normal vectors showing the normal direction of curve evolution	19
2.5	Signed distance representation of a contour with negative distances assigned to pixels inside the contour and positive distances assigned to pixels outside.....	21

2.6 The initial segmenting contour was placed in the center of the image (left panel), Segmentation result shown after 13 iterations of the Chan & Vese algorithm (right panel).	24
2.7 Pseudo-code for the Chan & Vese algorithm	24
2.8 Fuzzy interval f used to define relations such as <i>far</i> and <i>near</i>	26
2.9 Upper left panel: Image with the reference object on top and the target object on bottom. Upper right panel: The fuzzy distance map showing the fuzzy membership values derived from the distance of the target object from the reference object. Lower left panel: The fuzzy angle map derived using the angle from center of the reference object to the any point on the target object. Lower right panel: The intersection of the two maps is the final fuzzy landscape.	27
2.10 A roulette wheel for fitness-proportionate selection showing six individuals.....	34
2.11 Single-point crossover [77].....	35
2.12 Mutation operator applied to a <i>gene</i> (underlined) in a binary string	35
2.13 Pseudo-code depicting a simple genetic algorithm.....	36
2.14 Cloud segmentation on a satellite image using GENIE. Top panel: Original image; Middle panel: Manually marked truth/false (green/red) regions; Bottom panel: Final classification	40
3.1 Segmenting contour (blue contour in the center) and the mesh plot of its signed distance representation.....	42
3.2 (From left) Original contour; x -translated contour; y -translated contour; scaled contour; rotated contour.....	44
3.3 Top panel: Two contours before alignment. Bottom panel: After alignment.....	45

3.4 Pseudo-code showing how the textural segmentation parameters are saved for the GWT method	46
3.5 Pseudo -ode showing how the textural segmentation parameters are saved for the LTM method.....	47
3.6 Pseudo-code for deriving the fuzzy landscape from training images	49
3.7 Reference objects R_1 (above) and R_2 (below) and target object (T_1) in the center	49
3.8 Top panel: Fuzzy landscape (μ_1) derived from R_1-T_1 . Middle panel: Fuzzy landscape (μ_2) derived from R_2-T_1 . Bottom panel: Final fuzzy landscape (μ) created by the union of μ_1 and μ_2	50
3.9 A segmenting contour with the genotype [$\mathbf{w} = [410, 329, 723, 7.4]$, $\mathbf{p} = [2, 6, 1, 6]$]	51
3.10 A sample 3D-LSGA individual with genotype [$\mathbf{w}=[4554, 1595]$, $\mathbf{p}=[5, 12, 1, 1.4, 0, 0, 0]$].....	56
4.1 Thermographic images of hands of four subjects taken before typing (upper panel); after 9 minutes of typing (lower panel). The fingers start to become invisible due to reduced blood flowing in the subjects' hands.....	61
4.2 Mean shape (center). $\pm 3\sigma_1$ variability depicting movement of fingers (left, right). $\pm 3\sigma_2$ variability of the width of fingers (top, bottom)	63
4.3 Variation of the maximum fitness by number of generations of the GA run for the population sizes of 25, 50, and 100	65
4.4 Variation of the maximum fitness by number of generations of the GA run for the mutation rates of 2%, 5%, and 10%	66

4.5 Variation of the maximum fitness by number of generations of the GA run for the crossover rates of 50%, 90%, and 100%	66
4.6 Three candidate segmenting contours in the GA population. Here, F denotes the fitness of each individual	68
4.7 Segmentation outcome on a test image (manual segmentation shown in first from left) using: Gabor wavelet based method (second from left), Laws' texture measures (third from left), Level set-based method of Chan & Vese (fourth from left), LSGA (rightmost).	69
4.8 Maximum fitness of the population plotted over 30 generations for an exhaustive search by the LSGA on a single image.....	70
4.9 Segmentation result from every time interval for each patient.....	73
5.1 The 2D pelvic CT scan (left). The red contour (right) in the center is the prostate outlined by an expert. The white structures surrounding the prostate are the bones.	76
5.2 A 2D pelvic MRI scan (left) and manually segmented (red) contour (right)	77
5.3 Streak artifact present in a pelvic CT image which is caused by beam hardening near bones	79
5.4 Flipping of net magnetization towards the direction of the applied RF field	81
5.5 (Left panel) T_1 weighted image and T_2 weighted image (right panel) from the same patient	81
5.6 A noisy pelvic MRI image.....	82
5.7 A CT image before (top) and after (bottom) the application of an averaging filter to reduce streaking	85

5.8 Mean shape (left panel) and shape variability (right panel) of the prostate derived from training images.....	86
5.9 An MRI image (original image shown in the left panels); (Top right) Texture segmentation using Laws' texture measures; (Bottom right) texture segmentation using Gabor wavelet transform method.....	87
5.10 A CT image (original image shown in the left panels); (Top right) Texture segmentation using Laws' texture measures; (Bottom right) texture segmentation using Gabor wavelet transform method.....	88
5.11 Mean shape (top) and shape variability (bottom) of the prostate derived in three dimensions derived from training images.....	91
5.12 Multiclass texture segmentation of an MRI image using Gabor wavelet transform method on a CT image (top) and MRI image (bottom): Prostate (yellow), Bladder (red), rectum (green), background (dark blue) regions.....	93
5.13 A fuzzy landscape derived from all the training images using the bladder and rectum as the reference objects	94
5.14 Upper left panel: Initial contour placed on top of a test CT image. Upper right panel: Outcome of the Chan & Vese algorithm. Lower left panel: Initial contour placed on top of a test MR image. Lower right panel: Outcome of the Chan & Vese algorithm.	97
5.15 Plot of average fitness versus the number of generations of the sample run of the 2D-LSGA.....	99
5.16 (Left panel) Segmentation result in 2D for a test CT image. (Right panel) Final 2D segmenting contours stacked on top of each other to create a 3D shape.....	101

5.17 3D segmentation result of the LSGA on a test CT image (right panel). A slice of the 3D segmentation generated by the GA (left panel)	101
5.18 Right panel: 3D segmentation result of the LSGA on a test MR image. Left panel: A slice of the 3D segmentation generated by the GA. Since there are only 5 slices in this figure as compared to 10 in the previous figure, therefore the 3D structures look different.....	101
5.19 Slice-by-slice segmentation of pelvic CT scan of patient <i>A</i> using 2D-LSGA	105
5.20 Segmenting contours of 2D-LSGA stacked on top of each other to form a 3D shape	106
5.21 Slice-by-slice segmentation of a CT image of patient <i>A</i> using 2DRL-LSGA	108
5.22 Segmenting contours of 2DRL-LSGA stacked together to form a 3D shape.....	109
5.23 Slice-by-slice segmentation of a CT image of patient <i>A</i> using 3D-LSGA	110
5.24 The 3D segmenting surface of 3D-LSGA	111
5.25 Slice by slice segmentation of a CT image of patient <i>A</i> using 3DRL-LSGA.....	113
5.26 3D segmenting surface of the 3DRL-LSGA.....	114
5.27 Slice-by-slice segmentation of a MRI image of patient <i>B</i> using 2D-LSGA.....	116
5.28 Segmenting contours of 2D-LSGA stacked together to form a 3D shape.....	117
5.29 Slice-by-slice segmentation of a MRI image of patient <i>B</i> using 2DRL-LSGA.....	119
5.30 Segmenting contours of 2DRL-LSGA stacked together to form a 3D shape.....	120
5.31 Slice-by-slice segmentation of an MRI image of patient <i>B</i> using 3D-LSGA.....	121
5.32 3D segmenting surface of the 3D-LSGA.....	122
5.33 Slice-by-slice segmentation of an MRI image of patient <i>B</i> using 3DRL-LSGA.....	124
5.34 Segmenting surface of the 3DRL-LSGA.....	125

5.35 Segmentation result on a MRI image using 3DRL-LSGA. The figure shows that segmentation is inaccurate when the test image is not aligned properly with the training images. Here, the gray contour represents manual segmentation and the white contour shows the segmenting contour derived by LSGA	126
--	-----

List of Abbreviations/Symbols

Abbreviation Key

GA	Genetic Algorithm
GP	Genetic Programming
LSGA	Level Set-Based Genetic Algorithm
CT	Computed Tomography
MR/MRI	Magnetic Resonance/ Magnetic Resonance Imaging
RTP	Radio-Therapy Treatment Planning
IGRT	Image Guided Radiation Therapy
CTV	Clinical Target Volume
2D/2DRL	Two-Dimensional/Two-Dimensional with Relative Location prior
3D/3DRL	Three-Dimensional/Three-Dimensional with Relative Location prior
FLD	Fisher Linear Discriminant
GENIE	GENetic Imagery Exploration
ROI	Region of Interest
UEMSD	Upper Extremity Musculo-Skeletal Disorder
LTM	Laws' Texture Measures
GWT	Gabor Wavelet Transform-Based Segmentation Method
CV	Chan & Vese Algorithm
GATM	Genetic Algorithm Template Matching

Abbreviation Key

NIRS	Near Infra Red Spectroscopy
<i>DSC</i>	Dice Similarity Coefficient
<i>H</i>	partial Hausdorff distance
RF	Radio Frequency
3D-CRT	Three Dimensional Conformal Radiation Therapy
IMRT	Intensity Modulated Radiation Therapy
EBRT	External Beam Radiation Therapy
GTV	Gross Tumor Volume
PTV	Planning Target Volume
PCA	Principal Component Analysis
OHSU	Oregon Health and Science University
GUI	Graphical User Interface

1. Introduction

Target-volume and organ-at-risk delineation on medical images such as computed tomography (CT), magnetic resonance imaging (MRI), and ultrasound, is usually performed manually by an expert physician. However, this process is subjective and the uncertainty and variability in the definition of tumor margins can result in suboptimal treatment of some patients. The development of automated segmentation tools are therefore essential but remain a challenge for several reasons, such as the variability in organ shapes and tissue contrast on medical images. Despite advances in imaging for radiation-therapy treatment planning (RTP), most medical image segmentation algorithms are either semi-automatic or require some form of human intervention to perform satisfactorily [48][69][74]. This is mainly due to the fact that these algorithms do not encode the knowledge of the human anatomy that a physician uses to manually segment an image. Automatic segmentation can be best accomplished if the knowledge of shapes, relative locations, and textures of organs is incorporated into a single algorithmic framework. This dissertation presents a genetic algorithm for combining known representations of shape, texture and relative location to perform automatic segmentation.

Genetic algorithms (GAs) [51][76] simulate the learning process of biological evolution using selection, crossover and mutation. Genetic algorithms are blind optimization techniques that do not need derivatives to explore the search space. Instead they use payoff values, known as fitness, to guide the search. This quality can make GAs

more robust [41] than other local search procedures such as gradient descent or greedy techniques used for combinatorial optimization. GAs have been used for a variety of image processing applications, such as edge detection [45], image segmentation [47], image compression [82], feature extraction from remotely sensed images [46], and medical feature extraction [47]. The image processing problem that has been explored in this dissertation is image segmentation: a technique for delineating a region of interest on an image.

Level set methods have become very popular in the field of medical image segmentation due to their ability to represent boundaries of objects that change with time or are ill-defined [92][83]. In the level set method, a deformable segmenting curve is associated with an energy function. The energy function may consist of region-based terms (such as pixel intensity values, edges, etc.) and contour-based terms (such as curvature and length of the curve). These features are called low-level because they encode the information that can be derived directly from the image. There are many real-world problems that require high-level features for segmentation, such as the prior knowledge of shapes and context information derived from the extrapolations of human perception [42]. Incorporating such information into an explicit energy function term may be difficult or impossible to encode for performing segmentation. A genetic algorithm (GA) solves this difficulty because it eliminates the energy function (and instead uses a fitness function) thereby providing a framework for incorporating high-level features and combining multiple features for segmentation. The level set-based genetic algorithm scheme (LSGA) presented here uses the learned shape, textural properties, and relative location of a known object derived from training images to segment test images. The

individuals of the GA are a vector of parameters of a level set function and are referred to as chromosomes of the GA. The GA optimizes the parameters of the level set function to produce “fit” individuals or good segmentations of the given image using the information derived from training images. The algorithm terminates by converging on the region of interest.

The dissertation is organized as follows: At first a literature review is provided on image segmentation, specifically emphasizing level set methods and algorithms applied to medical images. Then the theory of genetic algorithms is described, and the advantage of using a genetic algorithm over conventional optimization methods is discussed. The LSGA algorithm is then described followed by a comparison with Laws’ texture-based segmentation method [63], the Gabor wavelet-based segmentation algorithm [1], and the shape-based level set method of Chan & Vese [17]. The description of the dataset used and the results achieved from applying the algorithm to segmentation of thermographic images of the hand and prostate segmentation on pelvic CT and MRI images are then discussed. The strengths and limitations of the LSGA and future work are described at the end.

1.1 Problem description

We have tested the LSGA on two different datasets: thermographic images of hands and pelvic CT/MRI images. The challenges faced in these segmentation problems are briefly outlined here. The details specific to each dataset are discussed further in chapters 4 and 5 respectively. In real-world use of these datasets, segmentation is performed manually by a human/expert who uses his prior knowledge of learned

concepts to draw a contour around the region of interest. Figure 1.1 shows a thermographic image of the hand of a patient suffering from Upper Extremity Musculo-Skeletal Disorder (UEMSD) which leads to cold fingertips. The segmentation task is challenging because the fingers are partially or completely invisible in these images. Also, the subjects tend to move their hands during the imaging process. Thus the automatic segmentation method needs to incorporate all of the following to perform successful segmentation: known shape, shape variation, movement and texture of the hands. A manual segmentation (i.e., human-drawn contour) of the same image is shown in the bottom panel of figure 1.1.

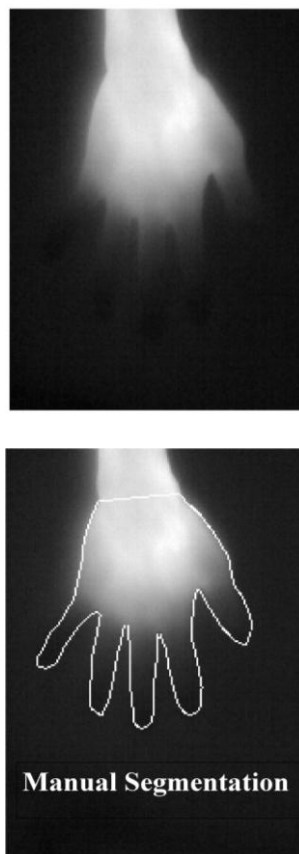


Figure 1.1 Top panel: Thermographic image of the hands of a patient suffering from UEMSD. Bottom panel: Manual segmentation of this image.

Figure 1.2 shows typical pelvic CT and MRIs scans from patients suffering from prostate cancer and undergoing radiation therapy. Manual segmentation of the prostate is typically performed by an expert physician before treatment locations and dosages are determined. The white contours in the center of the images in figure 1.2 show manual segmentation of the images. The prostate is located between the bladder and the rectum also labeled in the figure. Since, the bladder and the rectum are more texturally prominent on these images; the radiologist uses the location of these organs to approximately delineate the prostate on these images. Automating this task therefore involves incorporating the prior knowledge of shape, texture and relative location of organs.

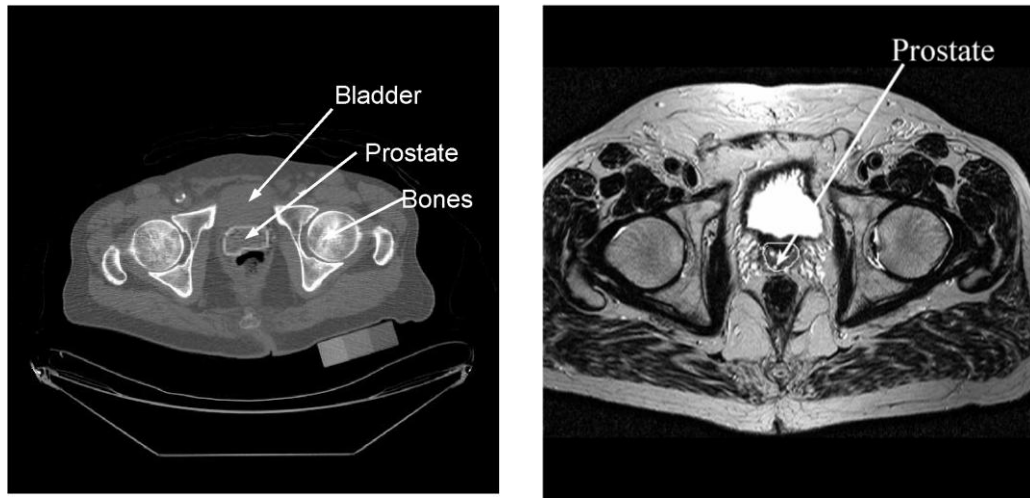


Figure 1.2 Left panel: A 2D pelvic CT scan. The white contour in the center is the prostate that was outlined by an expert. The black region just below the prostate is the rectum. The white structures surrounding the prostate are the bones. Right panel: A 2D pelvic MRI scan with the manually segmented prostate outlined in the center.

1.2 Hypothesis

This research work is based on the following two part hypothesis:

- The combination of pixel-level features (such as textures), object-level features (such as shape and shape variability) and context information (such as spatial relationships between objects) can produce better segmentation than using these features alone.
- A genetic algorithm (GA) can be used to combine these features for performing segmentation. This type of evolutionary optimization scheme is most-suitable for medical images with ill-defined boundaries because it can simultaneously explore multiple optimal solutions of the same problem and can account for the variability in segmentation margins.

Figure 1.3 depicts the hypothesis as a diagram showing the various features incorporated by the GA.

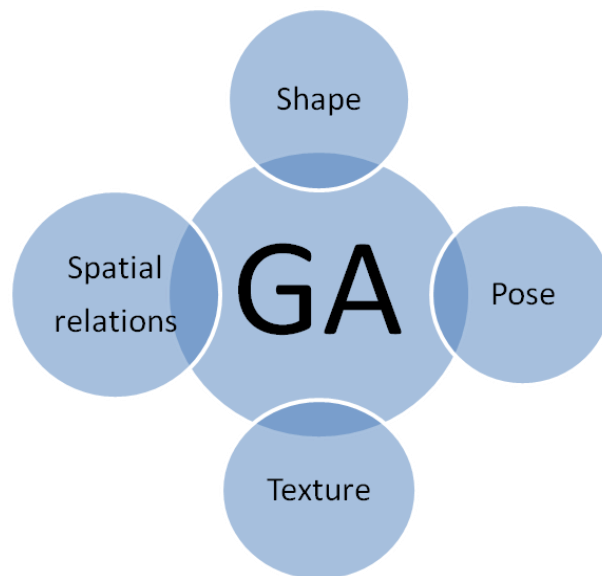


Figure 1.3 The hypothesis for this research is to combine multiple features for segmentation using a GA.

1.3 Overview of LSGA

The flowchart in figure 1.4 presents a very high-level description of the LSGA. The stages during training and segmentation are shown here separated with the dotted line. Shape, texture and location priors are first derived from training images that have been manually segmented. This information is incorporated into the GA to perform segmentation on test images. More details of the method can be found in chapter 3.

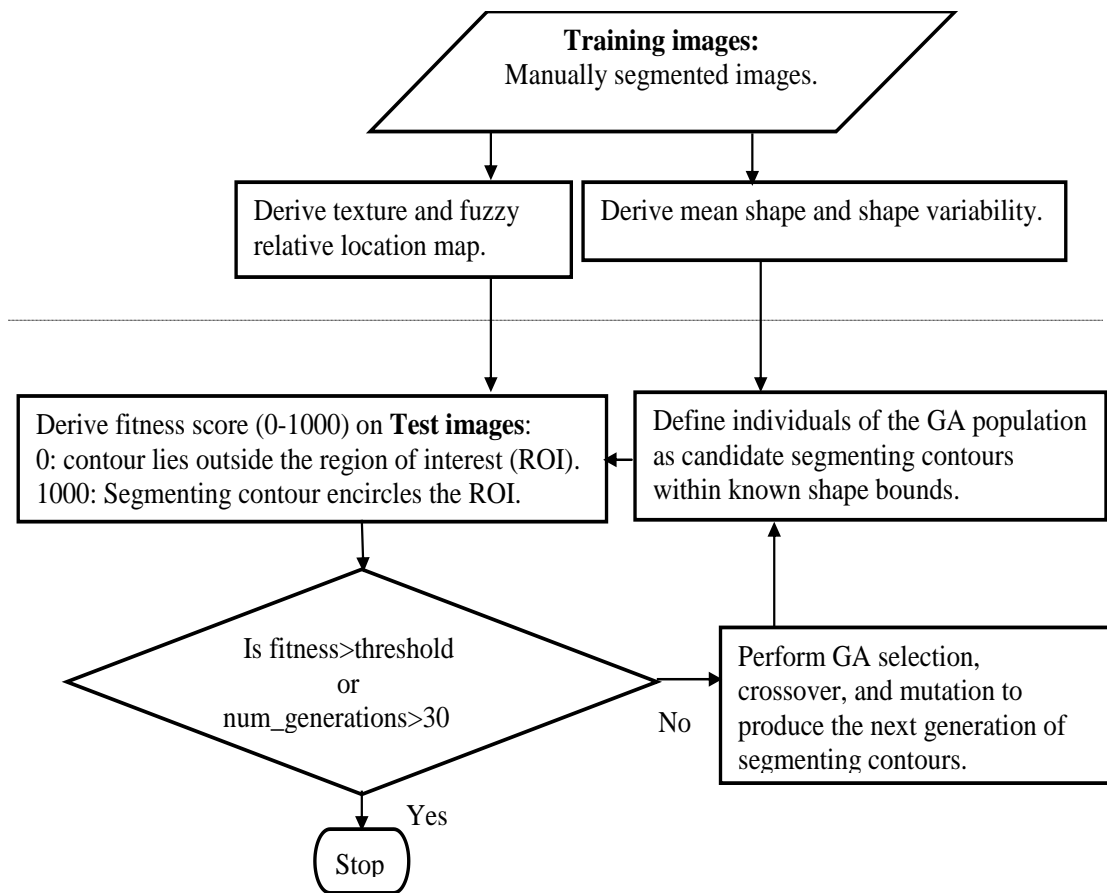


Figure 1.4 Flowchart depicting the sequence of operations of the LSGA.

1.4 Contributions of this dissertation

The research presented in this dissertation is novel in several ways:

- The ability of the LSGA to combine multiple priors for segmentation is the main contribution of this research work. By incorporating derivative-free optimization for level set function optimization, the LSGA creates a framework for combining multiple features such as shape, texture, relative location information for segmentation. This preserves the best of both worlds; the ability of a level set-based flexible contour representation and the ability to incorporate multiple priors for segmentation without increasing the number of derivatives associated with the gradient descent optimization method.
- It is the first attempt to fully automate the segmentation of prostate on pelvic CT/MRI images. Previous approaches for performing such segmentation were template matching approaches such as [26].

2. Literature Review

Medical images are captured from parts of living organisms so that the structural information contained in them can be quantified and analyzed. They can be acquired in several modalities such as Computed Tomography (CT), Ultrasound (US), Magnetic Resonance Imaging (MRI), Spectroscopy and so on [13]. One of the main challenges of medical image processing is that a lot of information is lost during image acquisition, resulting in artifacts. For example CT images have metal streak artifacts; MRI images have geometric distortion artifacts. Another challenge is that the information present in these images is usually of a high-level semantic nature; an effective mechanism for mapping the low-level pixel information to high-level semantic information is needed before this information can be used for accurate diagnosis and treatment planning. Medical image analysis typically involves segmentation, recognition and classification. The goal of this chapter is to provide a general overview of the current state-of-the-art in medical image segmentation and to describe the need to incorporate unconventional optimization techniques such as genetic algorithms for solving medical imaging problems.

2.1 Segmentation Methods

Segmentation is defined as the process of demarcating an object on an image with a closed boundary/contour. Before segmentation can be performed properties of the object that differentiate it from the rest of the image must be determined. These properties can

be image pixel-based properties such as edges, texture, pixel intensity variation inside the object, or object-level properties such as shape, size, orientation, location with respect to other objects, etc. The pixel-based features are referred to as low-level features because they can be inferred using simple image processing routines on an image. For example, edges of an image can be derived using a gradient operator on the image. The object-level features on the other hand are so-called “high-level” features because they require quantifying the semantic concepts that we create in our mind. For example, “size” of an object is a concept that can be represented as the distance between two pixels located in the opposite extremities of the object or by the diameter of a circle enclosing the object.

Pixel-based operations can be used on problems where the object has a prominent edge and markedly different pixel intensity values inside and outside the object. However these techniques alone are not suitable for medical image segmentation and when used are usually performed with a lot of manual/human intervention. Most medical images have a lot of noise and artifacts that appear during the image acquisition process. Also, most medical images have low contrast with broken/diffuse edges around regions of interest. This makes segmentation on medical images a challenging problem. Therefore, often an object-based approach or a combination of pixel and object-based techniques are more suitable for medical image segmentation.

2.1.1 Pixel-based segmentation methods

Pixel-based methods identify local features such as edges and texture in order to extract regions of interest on images. The most commonly used pixel based operation is the edge-detector. Edges are defined as regions on the image with large pixel intensity

variations/contrast. Edge detection methods [43] either use first-order image derivatives (Robert's method), linear filtering techniques (Sobel and Prewitt method), or second-order derivatives (Laplacian method and the Marr-Hildrith transform) on images. However, these techniques can produce broken edges and also include boundaries of other features/details present in an image. Another intensity-based method is the region-growing method [99], which starts from a seed-point (usually placed manually) on the image and performs segmentation by clustering neighborhood pixels using a similarity criterion such as the difference in the gray-level intensity values of two neighboring pixels.

More complex pixel-level features are textures. Texture is usually defined in image processing as a region consisting of mutually related pixels that quantify the perceived physical appearance of a surface. Textural segmentation methods can be broadly classified into statistical, spectral and spatial filtering methods, and model-based methods. Statistical approaches such as moment-based methods [101], co-occurrence matrices [97] etc., quantify textures like coarse, grainy, smooth, etc. Spectral and spatial filtering methods try to simulate the human visual system [58] by performing local spectral frequency analysis. One such method, known as *textons*, was developed by Julesz to model elements of texture perception [55][93]. Some other methods that use two-dimensional filtering in the spatial and frequency domain are [18][27][62]. Model-based methods such as Markov random fields (MRF) [22][56] and fractal-based modeling [85] have also been used for texture segmentation.

One major drawback of all pixel-based segmentation algorithms is that they can identify regions outside the object as being part of the object and there is no notion of

shape of a region in these methods. The Laws' texture measures of segmentation and the Gabor wavelet transform-based segmentation method are described here in detail because they have been incorporated into the LSGA.

A. Laws' texture-based segmentation method

A simple texture segmentation method is the Laws' texture energy measures, which can be used to discriminate some basic texture types such as lines, waves, ripples and spots on images. These texture measures are computed by convolving the training images with small integer coefficient masks followed by a non-linear windowing operation to obtain the textural feature planes [63]. The basic one dimensional convolution kernels (usually 5x5) derived by Laws stand for level (L), edge (E), spot (S), wave (W) and ripple (R) texture types respectively (table 2.1). Two-dimensional masks are generated from these vectors by convolving each vector with the transpose of another (table 2.2). To generate the texture energy planes, the training images are first convolved with the each of the 25 two-dimensional masks to obtain 25 grayscale images. When an image is convolved with a mask, the pixel values in the image around 5x5 windows get weighed by the mask parameters, thereby enhancing certain features in the image. For example, if the feature of interest is edges, then convolving with $E_5 \times E_5^T$ enhances the value of each pixel by reducing the weights of its neighboring pixels in 5x5 windows. Thus pixel intensity differences in the image get amplified, making the edges more prominent. A small (15x15) window is then operated on these grayscale images by summing the absolute values of the 225 neighborhood pixels to produce 25 different texture energy planes/maps.

Table 2.1 One-dimensional convolution kernels (number 5 shows that it is a five-valued vector).

L_5	[1 4 6 4 1]
E_5	[-1 -2 0 2 1]
S_5	[-1 0 3 0 -1]
W_5	[-1 2 0 -2 1]
R_5	[1 -4 6 -4 1]

Table 2.2 Convolution matrices derived from Laws' convolution kernels

$L_5 \times L_5^T$	$E_5 \times L_5^T$	$S_5 \times L_5^T$	$W_5 \times L_5^T$	$R_5 \times L_5^T$
$L_5 \times E_5^T$	$E_5 \times E_5^T$	$S_5 \times E_5^T$	$W_5 \times E_5^T$	$R_5 \times E_5^T$
$L_5 \times S_5^T$	$E_5 \times S_5^T$	$S_5 \times S_5^T$	$W_5 \times S_5^T$	$R_5 \times S_5^T$
$L_5 \times W_5^T$	$E_5 \times W_5^T$	$S_5 \times W_5^T$	$W_5 \times W_5^T$	$R_5 \times W_5^T$
$L_5 \times R_5^T$	$E_5 \times R_5^T$	$S_5 \times R_5^T$	$W_5 \times R_5^T$	$R_5 \times R_5^T$

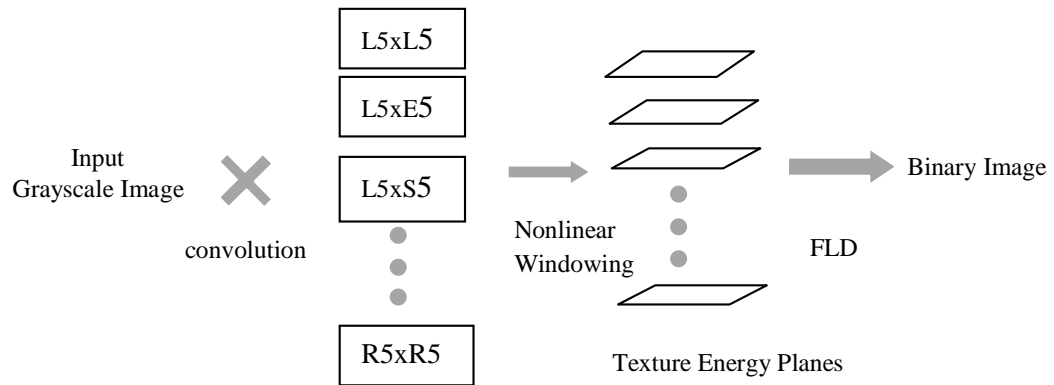


Figure 2.1 The steps in the segmentation process using Laws' texture measures.

Fisher's linear discriminant (FLD) is used to find the weights to linearly combine the texture energy planes and threshold the combination to obtain a binary image with the desired classification. Fisher's linear discriminant is a simple dimensionality reduction approach in which a multi-dimensional data vector \mathbf{x} is projected onto a one-dimensional space such that

$$\mathbf{y} = \mathbf{v}^T \mathbf{x}. \quad (2.1)$$

Although projecting multi-dimensional data along one dimension leads to a loss of information due to significant overlap, the class separation in one dimension can be maximized by adjusting the weight vector \mathbf{v} . This weight vector is a function of the projected class means normalized by the within-class scatter along the direction of \mathbf{v} . For a two-class problem the Fisher criterion is:

$$J(\mathbf{v}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \text{ where, } s_k^2 = \sum_{n \in C_k} (y^n - m_k)^2, k = 1, 2. \quad (2.2)$$

Here m_k is the class mean and s_k is the within-class scatter matrix. Maximizing the function J maximizes the class separation and minimizes the within-class scatter. Figure 2.2 shows an example of Laws' textural method applied to segmenting a tank from background vegetation. Figure 2.2, upper right panel, shows a binary image labeled manually using the 'roipoly' function in MATLAB. The lower left panel shows the linear combination of the texture energy planes obtained by applying Laws' masks on the original image. The weights for combining the texture energy planes were derived using a Fisher linear discriminant that maximizes the separation between feature and non-feature pixels (derived from the binary image). The final segmentation is shown in the lower right panel.

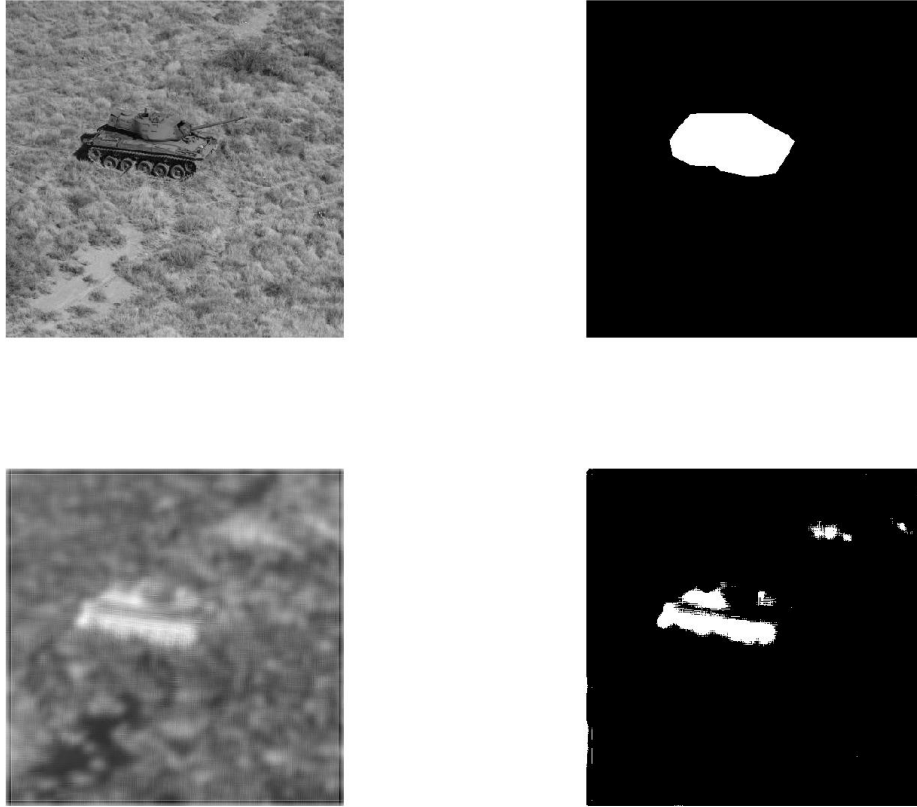


Figure 2.2 An example showing segmentation using Laws' texture measures. Top left: Original image. Top right: Manual segmentation. Bottom left: Linear combination of Laws' output planes. Bottom right: Final classification, white (tank), black (not tank).

B. Gabor wavelet transform-based segmentation method

Frequency-based texture segmentation methods are based on the principle that regions of high pixel intensity variations such as fine details and noise are high frequency components, whereas objects and global features constitute low frequency components of images. However, there is a tradeoff in space and frequency resolution when using traditional frequency domain methods such as Fourier transform methods. The wavelet-transform method for multi-scale texture description was introduced by Mallat [71]. Gabor wavelets are unique because they achieve the maximum joint space-frequency

resolution. This property makes them a good choice for textural feature segmentation where textural features need to be localized both in space and frequency.

Gabor wavelets are based on the Gabor elementary function given by the modulation of the Gaussian with a complex exponential function (equation 2.3).

$$h(x, y) = g(x, y) \exp[j2\pi Wx]$$

$$\text{where, } g(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left\{-\frac{1}{2}\left[\left(\frac{x}{\sigma_x}\right)^2 + \left(\frac{y}{\sigma_y}\right)^2\right]\right\}. \quad (2.3)$$

Here W is the modulation frequency and σ_x, σ_y characterize the bandwidth of the filter.

Gabor wavelets are derived by several translations and dilations of the mother wavelet $h(x, y)$. The method of Gabor wavelets assumes that local texture regions are spatially homogeneous. The mean and standard deviation of the transform coefficients are used to represent regions for classification (for example, figure 2.3). The Gabor wavelets are obtained through the generating function of equation 2.4,

$$h_{mn}(x, y) = a^{-m} H(x', y'),$$

$$a > 1,$$

$$m, n = \text{integers}, \quad (2.4)$$

$$\text{and } x' = a^{-m} \left(x \cos \frac{n\pi}{k} + y \sin \frac{n\pi}{k} \right),$$

$$y' = a^{-m} \left(-x \sin \frac{n\pi}{k} + y \cos \frac{n\pi}{k} \right).$$

Here, k is the number of orientations, H is the Fourier transform of $h(x, y)$, and m and n specify the scale and orientation of the wavelet. Given an image $I(x, y)$, the Gabor wavelet transform is given by equation (2.5) where, h^* is the complex conjugate of h

$$W_{mn}(x, y) = \int I(x_1, y_1) h_{mn}^*(x - x_1, y - y_1) dx_1 dy_1. \quad (2.5)$$

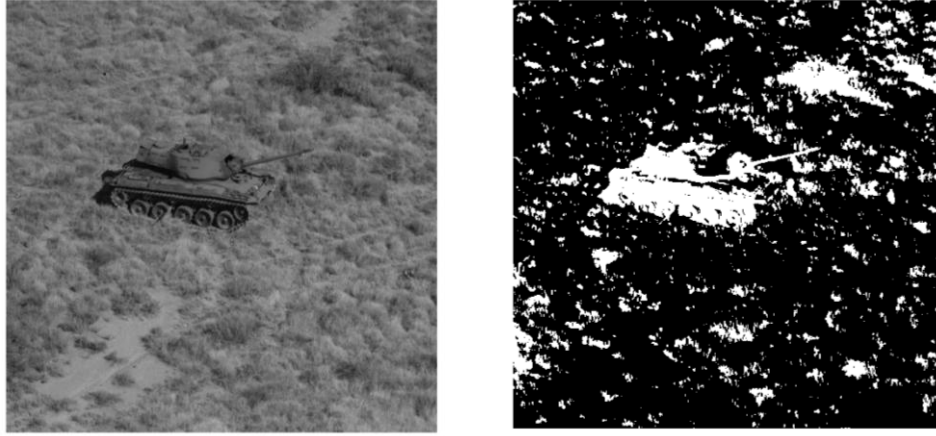


Figure 2.3 An example segmentation using Gabor wavelet transform method. Left: Original image. Right: Final classification, white (tank), black (not tank).

2.1.2 Shape-based segmentation methods

The shape of an object is an abstract concept and is based on human perception which is quite variable. Shapes are generally represented using contours, transforms, or regions [20]. Contour-based methods represent the shape outline either using a set of points on the contour or approximate the curve using a function such as the level-set function. Region-based methods may partition a shape into simpler forms (such as polygons), approximate the shape using a bounding region (such as a bounding rectangle or convex hull), or represent internal features of the shape (e.g., a skeleton). Transform-based representations decompose a shape into one or two dimensional signals (for example Fourier and wavelet transforms are linear transforms, whereas the Hough transform is a nonlinear transform). Transform-domain descriptors of shape can be transform coefficients or transform energy. In this project, shape has been represented using a contour because it can be easily deformed to represent a flexible boundary.

When an object is enlarged, rotated or moved it is still recognizable by a human. This property of an object is called *pose invariance* i.e., the object is identifiable from a different angle or position, or at a different scale. The pose of an object in an image can be changed using an affine transform. Pose is a relative concept and it is usually calculated with respect to the pose of another similar object in an image. Pose can be estimated by deriving the parameters of the affine transform needed to match the two shapes.

A. *Active contour method of segmentation*

Deformable contour models or active-contour models are shape-based procedures where a closed contour deforms using an energy function. This energy function incorporates regional properties of the object, such as edges or mean pixel intensity, and/or object-level features such as curvature of the object and size. During the curve evolution process, the curve is driven towards the boundary of the object. One approach for curve evolution is the marker point method [83] in which a segmenting curve C is parameterized by converting each point on the curve to represent a position vector $\bar{x}(s,t), \bar{y}(s,t)$, where s are points of the curve along a certain orientation (clockwise or counterclockwise), and t is time. The simplest way to evolve the segmenting curve is to move it along its normal vector field. Thus, if C is a simple, closed and smooth initial curve in \mathbb{R}^2 , it moves with a speed F along its normal direction,

$$\begin{aligned} (\dot{x}, \dot{y}) &= F\bar{n} \quad \dot{x} = \frac{dx}{dt}, \dot{y} = \frac{dy}{dt}, \\ \text{where, } \bar{n} &= \frac{(y' - x')}{\sqrt{(x'^2 + y'^2)}} \quad x' = \frac{dx}{ds}, y' = \frac{dy}{ds}, \end{aligned} \tag{2.6}$$

is the unit normal to the curve. This ordinary differential equation is called the “Lagrangian” form of the equation of motion. A standard procedure for computing the evolution of the moving front is to discretize this Lagrangian equation of motion. This is performed by dividing the curve into M equal (or possibly unequal) mesh points $s_i = i\Delta s, i = 0, \dots, M$ of size Δs , and time into n equal intervals of length Δt producing $M+1$ marker points on the curve (or interface). Figure 2.4 shows a discrete parameterization of a curve and normal vectors showing the direction of curve evolution. The front can be interpolated from these marker points as either line segments in two-dimensions (2D) or triangles in three-dimensions (3D). The curve evolution produces points for the $(n+1)^{\text{th}}$ time $(x(i, n+1), y(i, n+1))$ from the previous positions at time n on the interface. One disadvantage of this method is that the curve evolution can lead to two marker points coming closer to each other into a corner and within a few time steps this can lead to oscillations in the curvature, making the output unbounded.

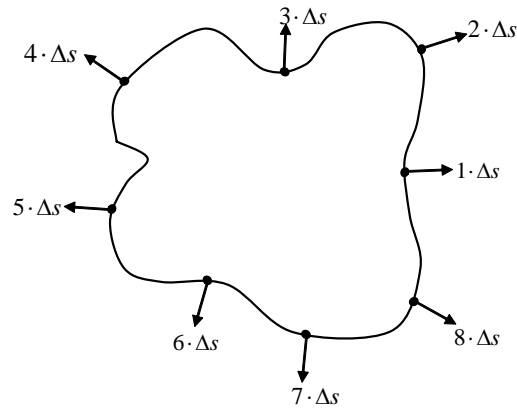


Figure 2.4. A closed simple curve with normal vectors showing the normal direction of curve evolution.

Another approach to active shape modeling is the level-set method introduced by Sethian [70][92]. This methodology became very popular due to its ability to automatically represent changes in the topology of dynamic curves such as the boundaries of soap bubbles, flames and other physical phenomena whose shape changes with time. In this approach the evolving boundary (interface) is represented implicitly as the zero isocontour of some function. For example, the zero isocontour of $\varphi(\vec{x}) = x^2 + y^2 - 1$ is given by a unit circle defined by $\varphi(\vec{x}) = 0$. In this framework, the equation of motion of the front is defined using a simple convection equation such as:

$$\dot{\varphi} + \vec{V} \cdot \nabla \varphi = 0, \quad (2.7)$$

where $\dot{\varphi}$ is the temporal partial derivative of the implicit function, $\vec{V} = \langle u, v, w \rangle$ is the velocity field (u, v, w are components of the velocity field in the x, y and z directions respectively), and ∇ is the spatial gradient operator such that $\vec{V} \cdot \nabla \varphi = u\varphi'_x + v\varphi'_y + w\varphi'_z$. This formulation is called the “Eulerian” approach since the interface is represented by an implicit function as opposed to marker points in the Lagrangian formulation. Equation (2.7) is referred to as the *level set equation*.

The level set function is usually defined in terms of the signed distance function. The signed distance function is an implicit function that takes any point in the plane and returns as its output the Euclidean distance between that point and the closest point on the interface. Pixels outside the interface have positive distance while the pixels inside have negative distance values assigned to them (figure 2.5). The “zero” level set is defined as the interface itself, i.e., the set of all points that are at height zero, or equivalently, whose distance to the interface is zero.

The level set update equation is derived by discretizing the level set equation using the forward Euler time discretization given by:

$$\frac{\varphi(n+1) - \varphi(n)}{\Delta t} + \vec{V}(n) \cdot \nabla \varphi(n) = 0. \quad (2.8)$$

The spatial derivative terms in equation (1) can be expanded as:

$$\frac{\varphi(n+1) - \varphi(n)}{\Delta t} + u(n)\varphi_x(n) + v(n)\varphi_y(n) + w(n)\varphi_z(n) = 0. \quad (2.9)$$

The upwind differencing used for the spatial derivative terms along with the forward Euler time discretization makes the level set update stable. This guarantees that small approximation errors are not amplified with time.

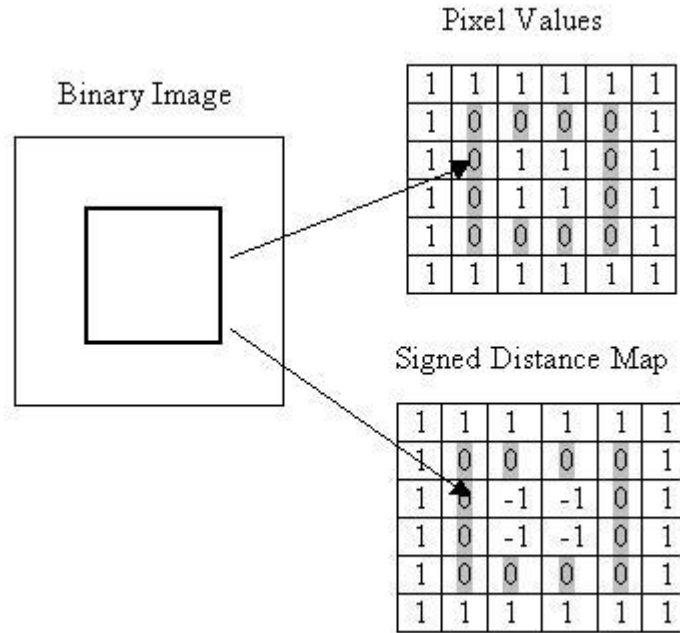


Figure 2.5. Signed distance representation of a contour with negative distances assigned to pixels inside the contour and positive distances assigned to pixels outside.

B. Level set methods for medical image segmentation

Level set methods have been used extensively for medical image segmentation [21]. Some of the popular methods are by Leventon et al.[64], Tsai et al. [99] and Chan and Vese [17]. Leventon et al. introduced the concept of shape representation by principal component analysis (PCA) on signed distance functions. They also incorporated statistical shape priors into their geodesic active-contour model to generate maximum *a posteriori* estimates of pose and shape. They segmented synthetic as well as medical images using their method and compared level-set evolution with and without shape influence. Their segmentation results were within one or two voxels of manual segmentation. However, the initialization point was placed manually on the images.

Tsai et al. derived a shape-based level set function. Tsai et al.'s goal was to find the parameters of this function that produce a good model of the object shape based on priors from the training data. Tsai et al. derived these parameters via an optimization procedure that used statistics defined over local regions in a set of training images. The performance of Tsai et al.'s algorithm thus depended on the particular choice of statistics used to distinguish various regions within a given image. They showed automatic segmentation results on several synthetic images and semi-automatic segmentation on cardiac and pelvic MRI images. The goal of the Tsai et al. algorithm was to optimize parameters of the level set function using the statistics of pixel intensity values in images. The LSGA also optimizes the parameters of the same level set function, but using a GA, allowing high-level texture and spatial relationships to be used for optimization.

Shape priors have also been used with active-contour-based image segmentation by Etyngier et al. [30]. They used diffusion maps to model shapes as finite-dimensional

manifolds. Their segmentation results were accurate but the initial contour was placed manually in the images.

Chan & Vese introduced a region-based energy function based on Mumford-Shah segmentation techniques [80] in order to detect features with diffuse boundaries. The limitation of their model is that it could only detect objects by intensity average values. They mention that other image features such as texture need to be combined with a level-set framework in the future to perform more generalized Mumford-Shah segmentation. The LSGA developed here attempts to address this need. The performance of the LSGA is compared with the Chan & Vese algorithm in a later chapter; here we describe the Chan & Vese algorithm.

C. The Chan & Vese algorithm

The Chan & Vese algorithm is based on the assumption that an image consists of regions with piece-wise constant intensities. If the evolving contour is defined by C then the energy functional that is minimized for curve evolution is given by:

$$F(c_1, c_2, C) = \mu \cdot \text{length}(C) + \nu \cdot \text{Area}(\text{inside}(C)) + \lambda_1 \int_{\text{inside}(C)} |u_0(x, y) - c_1|^2 dx dy + \lambda_2 \int_{\text{outside}(C)} |u_0(x, y) - c_2|^2 dx dy, \quad (2.10)$$

where u_0 is the pixel intensity value, c_1 and c_2 are means of pixel intensity values inside and outside the segmenting contour and $\mu \geq 0, \nu \geq 0, \lambda_1, \lambda_2 > 0$ are fixed parameters.

Thus, the algorithm finds the segmenting contour that maximizes the separation between the mean pixel values inside and outside the segmenting contour. Figure 2.6 shows a sample segmentation using the Chan & Vese method. The pseudo code for the Chan & Vese algorithm is shown in figure 2.7. Here, the initial segmenting contour has been

placed randomly on the image. Minimizing the energy function of equation 2.9 drives the segmenting contour towards the boundary of the object.

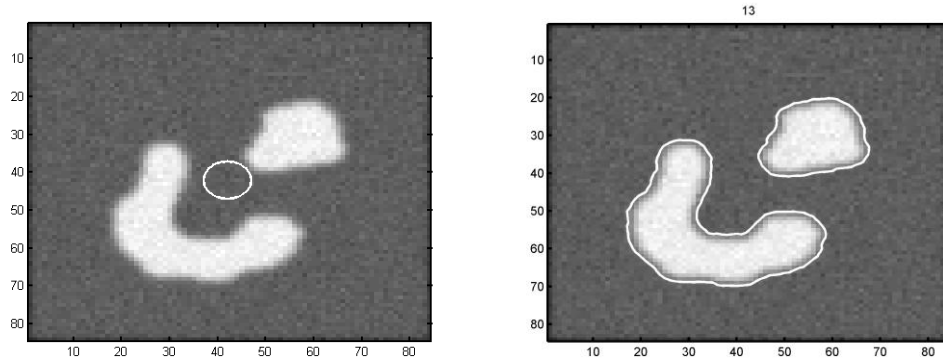


Figure 2.6 The initial segmenting contour was placed in the center of the image (left panel), Segmentation result shown after 13 iterations of the Chan & Vese algorithm (right panel).

```

FUNCTION Chan_Vese_Algorithm
1. Initialize Number_of_generations=0
2. Place one initial segmenting contour in the center of
  the image.
3. Initialize Energy_gradient=0
  //-----iterate-----
4. While Energy_gradient<Threshold
  4.1 Derive regional statistics inside and outside
    the
    segmenting contour;
  4.2 Calculate energy gradient at each point on the
    contour;
  4.3 Evolve the segmenting contour using the energy
    gradient
  End //---- end while-----
END

```

Figure 2.7 Pseudo-code for the Chan & Vese algorithm.

2.1.3 Incorporating spatial relationships for segmentation

Much current research work on image processing is focused on spatial knowledge representation for applications such as medical imaging [9], geographical information systems (GIS) [72], robot tracking [107] etc. Spatial relationship features are described by the relative position of an object in an image with respect to a reference object. Common methods for deriving spatial relationships include topological methods, distance and direction-based methods [105]. Spatial relationships are high-level features because they have the ability to describe scenes [96] on images using concepts such as near/far, inclusion/overlap, above/below etc. Some recent methods for quantifying spatial relations on images are Bayesian network classifiers [86][78], graph matching methods [12], and fuzzy relative location maps [8]. The fuzzy relative location method of Bloch, I. [8][9] is described in detail here because it has been incorporated into LSGA. This method is useful because it can be used to locate objects in images with poor contrast and ill-defined boundaries such as the prostate on pelvic CT/MRI images.

A. Fuzzy relative location maps

Bloch I. [9] developed a method to represent spatial relations between objects as fuzzy subsets of an image. Colliot et al. [19] applied the method to deformable models to improve the segmentation of objects with poor contrast and missing boundaries. They segmented brain sub-cortical structures in T_1 - weighted MRI images with a high degree of accuracy. However, they first construct a manual initial segmentation as the starting point of curve evolution. In this method, fuzzy distance and direction maps are created from the reference object to the target object. The fuzzy maps are created such that

brighter areas (higher gray-level values) represent regions with a higher probability of finding the target object with respect to the reference object.

Fuzzy distance relations such as *near* and *far* are defined using a fuzzy interval f of trapezoidal shape on the space of distances (shown in figure 2.8). For the *near* relation the limits of the fuzzy interval (n_3, n_4) are determined using the largest distance between points in reference object and target object. Similarly, for the *far from* relation, the prior knowledge of the nearest distance between the two objects is used to derive the limits of f (n_1, n_2) such that $0 \leq n_1 \leq n_2 \leq n_3 \leq n_4$.

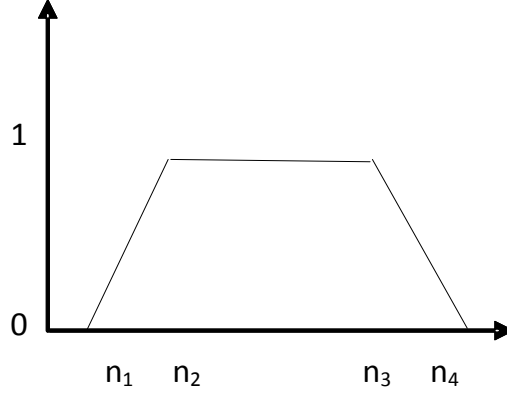


Figure 2.8 Fuzzy interval f used to define relations such as *far* and *near*.

The fuzzy subset of the image space is obtained by applying f to the distance map d_A of the reference object A :

$$\mu_d(P) = f(d_A(P)), \quad (2.11)$$

where P is any point in the image.

To derive the angle maps, a vector joining the point P in the image space to the point Q in the reference object is created. Then the angle between the vectors \mathbf{QP} and \mathbf{u} , and the unit vector in the direction under consideration $\beta(P, Q)$, is computed in the range

$[0, \pi]$. Here Q has been taken as the centroid of the reference object. The fuzzy subset of the image space is then defined as

$$\mu_\alpha(P) = g(\beta(P, Q)), \quad g(\theta) = \max[0, 1 - (2/\pi)\theta]. \quad (2.12)$$

Here, g is a decreasing function which maps the range of angles $[0, \pi]$ to $[0, 1]$. Figure 2.9 shows fuzzy distance and angle maps from the reference object on top to the target object on bottom.

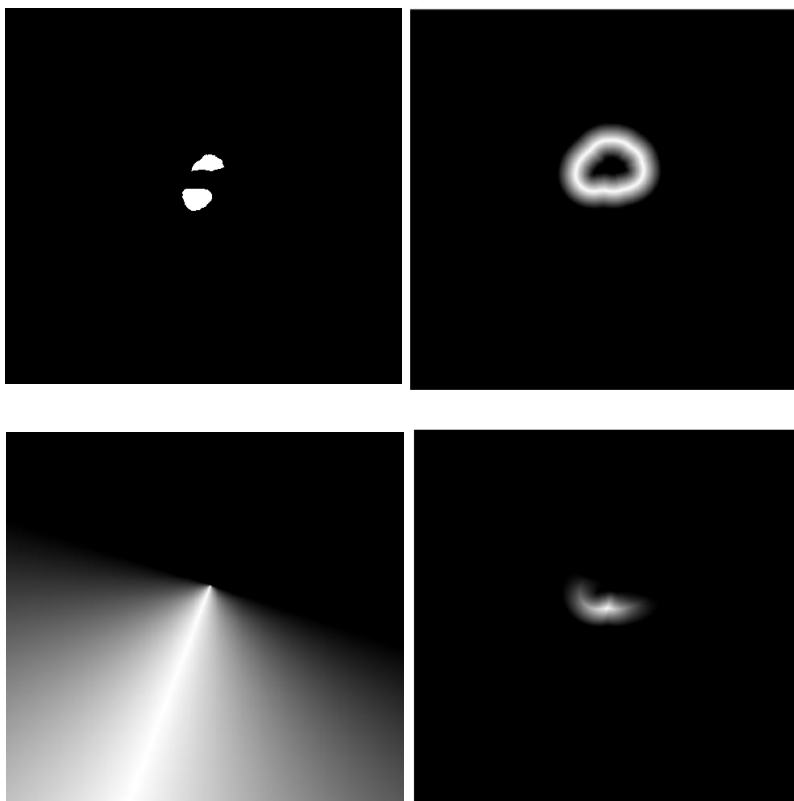


Figure 2.9 Upper left panel: Image with the reference object on top and the target object on bottom. Upper right panel: The fuzzy distance map showing the fuzzy membership values derived from the distance of the target object from the reference object. Lower left panel: The fuzzy angle map derived using the angle from center of the reference object to the any point on the target object. Lower right panel: The intersection of the two maps is the final fuzzy landscape.

2.2 Optimization methods in image processing

Optimization is derived from the word “optimum” and is defined as the process of obtaining the “best” solution to a problem. Optimization problems occur in various disciplines such as engineering, mathematics, economics, physics, etc. [3]. The general approach to optimization is to iteratively generate progressively improved solutions to a given problem. For any optimization problem, a performance criterion known as an objective function or energy function needs to be formulated in terms of some control parameters [65]. Thus, the general problem can be formulated as:

$$\text{minimize } F = f(x) \text{ for } x \in \mathbb{R}^n. \quad (2.13)$$

Here x denotes the n -dimensional parameter space.

Optimization plays an important role in image analysis because of inherent uncertainties in images such as noise, occlusion of objects, and variability in image perception and understanding. One of the first examples of optimization techniques used for image processing is the pioneering work of Roberts [91] for object identification using least squares fitting. Nowadays, optimization methods are being used widely in all aspects of image analysis such as image restoration [52], image compression [90], image segmentation [92], image registration [43], object recognition [31], perceptual grouping [49], stereo and motion estimation [81], etc.

2.2.1 Traditional optimization methods

In traditional methods, optimization is performed by minimizing an energy function that is used to assess the quality of the solution as well as to guide the search towards the optimal solution. The energy function can be represented both explicitly as

well as implicitly. For instance, the gradient descent optimization method and conjugate gradient methods are examples of optimization techniques where the energy function is specified explicitly. They have been used by the deformable contour-based segmentation algorithms discussed in the previous section [57][83]. These methods are useful for convex optimization problems where the minimizing the gradient of the energy function provides sufficient information about the global minima [10]. Such is the case with high-quality images with clearly defined edges and boundaries; in these cases gradient descent methods can be used.

For non-convex optimization problems where the energy function is not smooth, attaining global minima can be a challenge. In such cases heuristic approaches and stochastic optimization techniques [79][102] (such as evolutionary algorithms, simulated annealing, etc.) are more appropriate for searching for the optimal solution. In these techniques, the energy function is represented as an implicit function (for example it may be a function of reward and penalty terms). These techniques are also useful when there is more than one solution to the problem.

Medical images may be considered as non-convex optimization problems because they contain many artifacts, have poor contrast, ill-defined boundaries/edges and their interpretation is quite subjective. Although conventional techniques have been used for interactive analysis of medical images [74], parallel stochastic optimization techniques such as evolutionary computation methods need to be explored so that nearly optimal solutions that model the inherent uncertainties of these images can be derived automatically.

2.2.2 Stochastic optimization methods

Stochastic optimization problems can be used to find nearly optimal solutions to global optimization problems that are generally very difficult to solve using traditional methods [44]. Some popular stochastic optimization methods are Hill-Climbing [75], Simulated Annealing [59], Tabu Search [38] and Evolutionary Algorithms [29]. In the Hill-Climbing technique, an initial solution is randomly chosen from the solution space. The algorithm then searches for a better solution in the neighborhood of the current solution and assigns it as the new solution. This process is iterated until a better solution cannot be found. Simulated annealing is derived from an analogy from the metallurgical process of annealing. It is similar to Hill-Climbing in that the initial solution is chosen at random and successive solutions are chosen from the neighborhood of the current solution. However, the new solution is chosen with a probability that depends on a global parameter T (called temperature) whose value decreases at successive time steps. Tabu search is a heuristic search technique where a list of previous solutions is stored in memory. Here again, the initial solution is chosen at random. A new solution is chosen only if it is not on the Tabu list and is better than the solutions found so far.

Evolutionary computation methods differ from the above mentioned methods because they evaluate multiple solutions in parallel as opposed to a single candidate solution. Therefore, they are less likely to become stuck in a local optimum and are good global optimization techniques. There are several evolutionary computation methods such as Genetic Programming, Evolutionary Programming, Evolutionary Strategies and Genetic Algorithms. Details of these methods can be found in [25]. The following section describes the genetic algorithm in detail.

2.2.3 Genetic algorithms

Genetic algorithms [51][76] simulate the process of biological evolution using the principle of survival of the fittest. GAs have been used for a variety of optimization problems, such as image segmentation [87], feature extraction from remotely sensed images [23], and medical feature extraction [47]. In contrast with traditional optimization methods, a GA uses a stochastic parallel search to reach the optimum solution and so is less likely to become stuck in a local maximum. At each generation a new population is generated and the fitness values of all individuals are evaluated based on their performance in the problem domain. The process of selection, crossover, and mutation is repeated until an offspring with an acceptable fitness value is produced.

The GA stochastically searches the space of candidate solutions to identify the best (or at least an adequate) solution for a given problem. A fitness function is used to evaluate individuals and compare them based on the fitness score. This fitness score is used in the selection process to determine which individuals can reproduce and propagate their good *genes* to future generations using selection, crossover and mutation. This process is iterated over successive generations to achieve fitter or better solutions to a given problem.

A. *Genotype and phenotype*

Individuals of the GA are candidate solutions and are typically encoded as bit strings or vectors [15] (also known as *chromosomes*) whose interpretation depends on the application. The chromosome is the genetic makeup of an individual and is also known as the *genotype*. The genotypes are uniquely mapped to the solution domain (known as the

phenotype). The GA searches the space of candidate solutions to identify the best solution for a given problem. However, the actual search process operates on the encoding of the candidate solution in the form of real-valued genes.

B. Fitness

The fitness function is used to measure the performance of each individual in the problem domain. A commonly used fitness measure is proportional fitness [41] where the fitness f of each individual is determined as the ratio of the individual's performance relative to the whole population:

$$F(x_i) = \frac{f(x_i)}{\sum_{i=1}^N f(x_i)}, \quad (2.14)$$

where, N is the population size and x_i is the phenotype of the individual. Thus each individual has the probability of reproducing according to its relative fitness value. In multi-objective optimization problems more than one objective functions or fitness functions are simultaneously optimized during the evolutionary search process. Fitness functions are usually specified by the user depending on the application. In the LSGA the fitness function has been derived from [47].

C. Selection

Selection is the process of choosing an individual for reproduction. Selection is performed in a number of different ways. Some of the popular methods are rank selection, fitness proportionate selection, and tournament selection. In rank selection, candidate solutions are sorted according to their fitness score and higher ranked

individuals are more likely to be chosen for crossover than lower ranked individuals. In fitness proportionate selection, the probability of an individual for being selected is given by the ratio of its fitness to the fitness of other members of the population. In tournament selection, two individuals are first chosen randomly from the current population. One of the two individuals is then selected probabilistically, based on fitness.

Genetic algorithms often suffer from premature convergence. This occurs when some individuals in the population are much fitter than others and are the only ones selected for producing the future generations, thereby resulting in the reduction of diversity of the population over successive generations of the GA. This can slow the progress of the GA. Selection procedures such as tournament selection and rank selection can be used instead of fitness proportionate selection to help avoid premature convergence.

Many selection techniques use a probabilistic mechanism known as roulette wheel to select individuals. The roulette wheel is a real-valued interval and can have two types of sizes either the population size of the generation (N) or the sum of fitness values of all individuals in the generation ($\text{sum}(f)$).

This interval is then divided into N subintervals corresponding to each individual. The width of each subinterval is determined by the selection criteria being used. For instance, for fitness-proportionate selection, the interval corresponding to each individual is proportional to its fitness. In tournament selection each individual is given equal priority. Figure 2.10 shows a roulette wheel whose circumference corresponds to the total fitness of the population. Each sector of the roulette wheel corresponds to an individual, with larger sectors assigned to individuals with higher fitness values. This represents

fitness proportionate selection. Thus, when this wheel is spun, the higher fitness individuals are more likely to be selected.

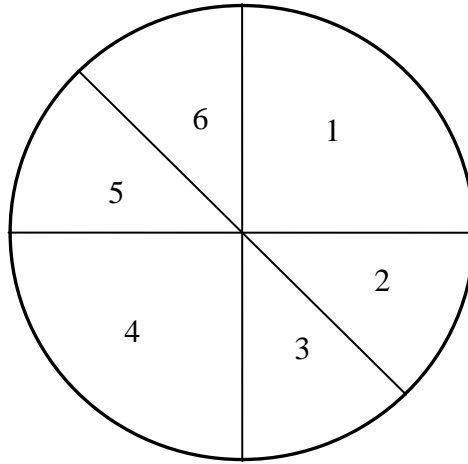


Figure 2.10 A roulette wheel for fitness-proportionate selection showing six individuals.

D. Crossover (Recombination)

After selecting two individuals from the current population, the crossover operator is applied to produce two new offsprings that are members of the next generation (figure 2.11). The crossover operation is not necessarily performed on all individuals in the population. The crossover operator is applied with a probability P_C when pairs of individuals are selected from the previous stage. Single-point crossover is performed using a crossover mask which swaps same length segments of genes between two parents to produce the offspring.

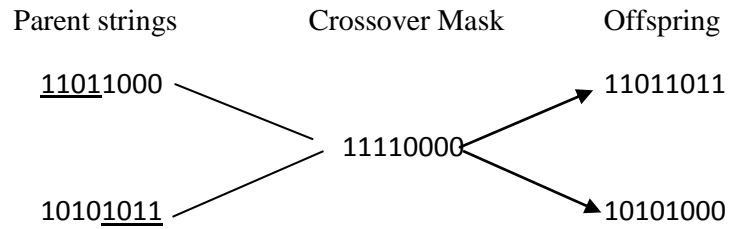


Figure 2.11 Single-point crossover [77].

E. Mutation

The mutation operator is also applied with a probability P_m . This operator chooses a single *gene* at random and changes its value. Figure 2.12 shows the mutation operator applied to a binary string.

$$[1 \underline{1} 000000] \Rightarrow [1 \underline{0} 000000]$$

Figure 2.12 Mutation operator applied to a *gene* (underlined) in a binary string.

F. Termination criteria

Since the GA is a stochastic search method, it can be terminated at any point of time. The GA is usually terminated if the fitness value attained in a generation exceeds a certain threshold or a certain pre-specified number of generations have been produced. Figure 2.13 shows the pseudo code for a simple GA depicting the sequence of operations in a step-by-step fashion.

```

FUNCTION Simple_GA

    5. Initialize Number_of_generations=0
    6. Generate initial population
    //-----Run GA-----
    7. While Number_of_generations<Threshold
        3.1 Derive fitness of each individual in the population
        3.2 Select pairs of individuals from population
        3.3 Reproduce pairs to produce offsprings
        3.4 Increment Number_of_generations
    End //---- end while-----

END

```

Figure 2.13 Pseudo-code depicting a simple genetic algorithm

2.2.4 Evolutionary computation in image processing

Genetic algorithms have been used for segmentation by several groups (e.g., [2][6][4][14][24]). A recent review article on application of genetic algorithms to medical image processing is by Maulik [73]. In [24], the authors present a genetic algorithm template matching (GATM) scheme to automatically detect nodules in CT images. They use lung nodule phantom images along with global nodule intensity distribution for template matching. They achieve a high detection rate and suggest that genetic-algorithm-based methods are very useful for clinical applications.

In [2] the author presents an edge-based segmentation scheme using a genetic algorithm. In [6] the authors formulate the image segmentation problem as an optimization scheme where a genetic algorithm is used to search a hyperspace of segmentation parameter combinations to derive the optimal segmentation. A model-based image analysis technique using a GA is described in [66]. The method uses an evolutionary Hough transform scheme to detect objects with known shapes on images

such as circle and ellipse. Here, the GA population consists of a set of points in the parameter space. In contrast, the LSGA evolves a population of segmenting contours constrained by known shape.

Cagnoni et al. [14] use a GA for segmenting images by evolving parameters of an active contour model called *snakes* [57]. The GA optimizes an energy function based on low-level features such as smoothness of the curve, curvature and image gradient. In contrast, the LSGA framework evolves parameters of a level set function. Unlike the explicit representation of shapes used in [14][66], the level set-based implicit representation of shape used here allows textural and spatial relationship features to be combined in searching the parameter space. Another recent work using GAs for medical image segmentation is by Chabrier et al. [16]. They use a GA to find the optimal combination of information extracted from several different segmentation algorithms.

Another form of evolutionary algorithms is Genetic Programming (GP), which evolves computer programs, a population of which forms the initial gene pool of the algorithm. The goal of this algorithm is to produce a program consisting of an optimal sequence of primitive operators to solve a complex optimization/modeling problem. In [46][47] a GP-based general-purpose image-segmentation system called GENIE (short for GENetic Imagery Exploration) is described that was developed at Los Alamos National Laboratory. Harvey et al. [47] applied GENIE to a medical feature-extraction problem using multi-spectral histopathology images. Their specific aim was to identify cancerous cells on histopathology images of breast tissue. The cancerous cells in these images had distinctly different shape than healthy cells. Their results were not very accurate, since GENIE used only texture-based image operations, and did not have any

object or shape-based operators. Such operators are clearly needed for more accurate medical image segmentation.

A. *GENIE*

GENIE uses genetic programming to evolve image-processing “pipelines”: sequences of elementary image processing operations, including morphological, arithmetic and point operators, filters and edge detectors, among others. Each pipeline, when run on a given multi-spectral image, performs image segmentation by classifying certain pixels as being part of a desired feature or otherwise. The fitness of each pipeline in the population is computed by comparing the final classification output with a set of training images, in which positive and negative examples of the desired feature have been manually highlighted. At the end of a run of GENIE, the fittest pipeline in the population is used to segment the desired feature in new images. A fitness measure similar to that used in GENIE has been adopted in the LSGA developed here because it has distinct payoff terms for reward and penalty.

GENIE uses a set of data planes (from multi-spectral data) and some scratch (answer) planes on which it writes the output of the image processing operators. The image processing operator can either be applied to the data plane or the scratch plane depending on the information encoded in the genes of the pipeline. The results of applying the operator can only be written on scratch planes. These scratch planes are inputs to a Fisher linear discriminant backend that optimally combine the answer planes to separate feature pixels from non-feature pixels. For example, the gene [OPEN rD3 wS0 2 1] applies the image processing operator “open” to one input plane; it reads (‘r’)

from data plane D3 and writes ('w') to scratch plane S0; 2 and 1 are the parameters used for performing the open operation.

GENIE, starting with a population of random pipelines, evaluates the fitness of pipelines in the population, and selects the fittest to produce the next generation, using crossover and mutation to produce an offspring. We obtained an open source version of GENIE from the website of Los Alamos National Laboratory and applied it to segmenting clouds on a multispectral satellite image (see figure 2.14). GENIE perfectly classifies the cloud pixels in this image.

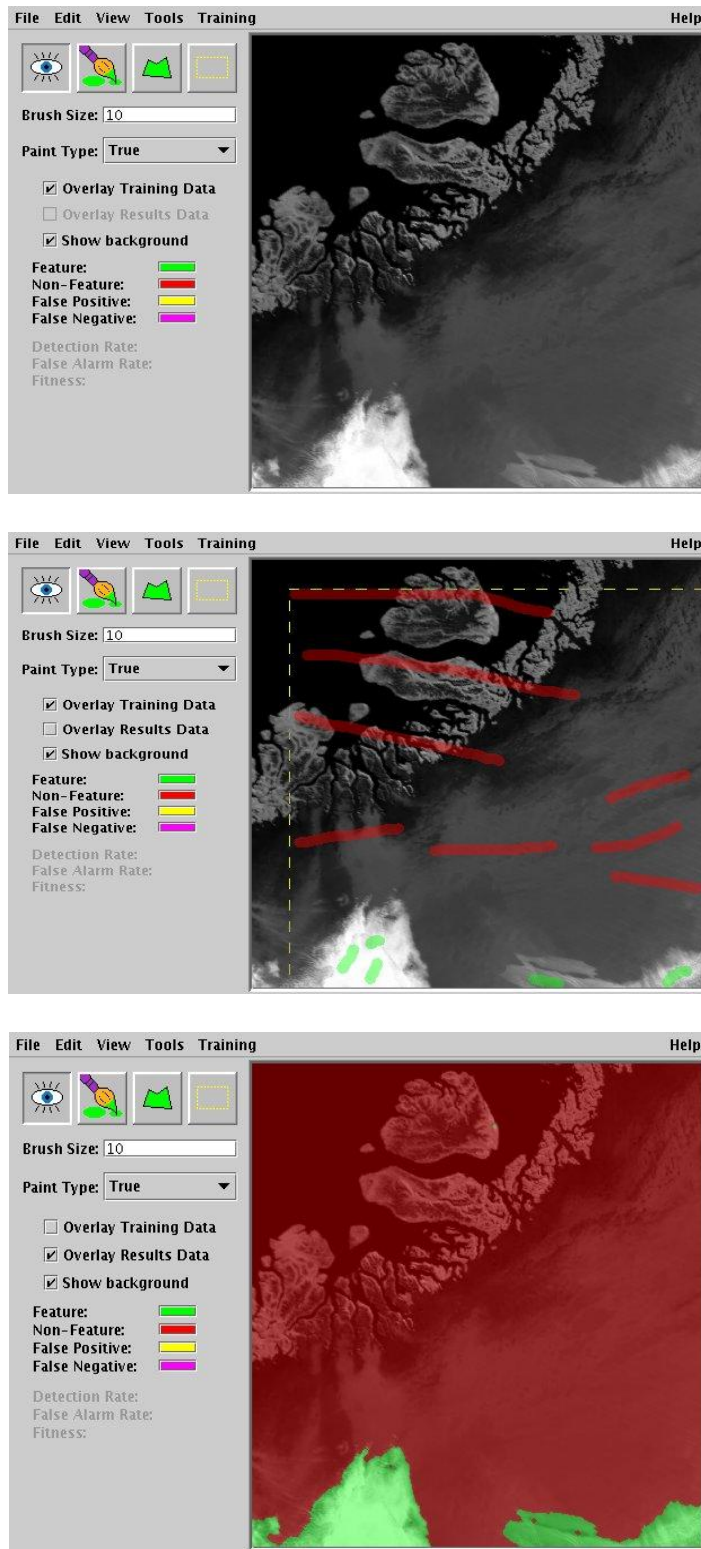


Figure 2.14 Cloud segmentation on a satellite image using GENIE. Top Panel: Original image; Middle Panel: Manually marked truth/false (green/red) regions; Bottom panel: Final classification.

3. Method: LSGA

The GA developed in this research has been called LSGA because it optimizes a level set function using a genetic algorithm to perform segmentation. The LSGA consists of two stages: the training stage and the segmentation stage. In the training stage, shape, shape variability, texture, and relative location information of the region of interest are derived from manually segmented images. The data for the training stage is obtained from a set of training images on which a human has drawn a contour around the object to be segmented, as well as landmark objects in its neighborhood. The set of these training contours provides information about the shape and pose variability of the given object. The textural properties of the object are also derived from the same set of training data. The segmentation phase involves the genetic algorithm evaluating candidate segmenting contours for segmenting the desired object in a new image using a fitness measure, and iterating over successive generations until a stopping criterion is satisfied.

3.1 Training stage: Deriving a shape model

The shape representation is derived from the mean and variance of all manually drawn contours in a training set [99]. The manually drawn contours from the training data are first converted into signed distance functions, Ψ_i ($i = 1$ to n , where n is the number of training contours). Figure 3.1 shows an example segmenting contour and its signed distance representation. Converting the contours to signed distance representations makes it easy to derive the mean shape and shape variability using matrix transformations without finding point correspondences.

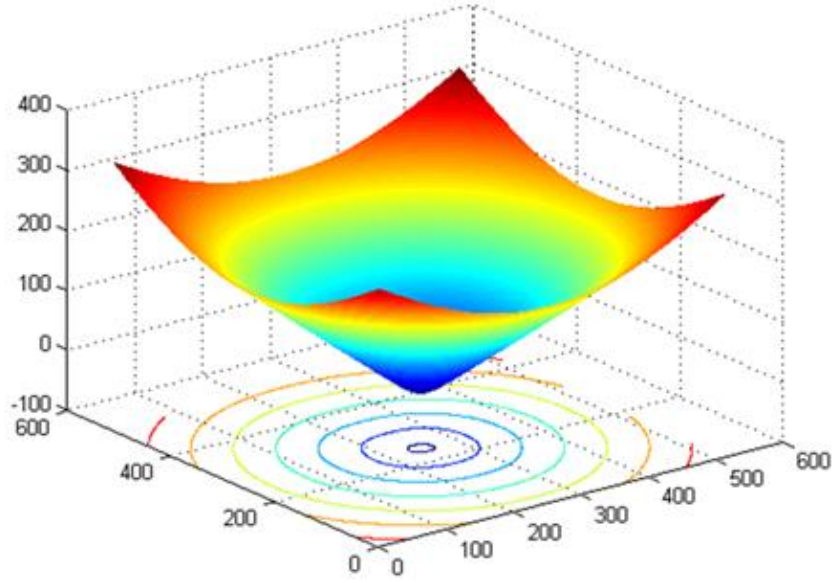


Figure 3.1 Segmenting contour (blue contour in the center) and the mesh plot of its signed distance representation.

The level set function is defined as a linear combination of the mean shape and weighted shape variances in the signed distance domain. The mean shape is defined for n contours as:

$$\bar{\Phi} = \left(\frac{1}{n}\right) \sum_{i=1}^n \psi_i. \quad (3.1)$$

Mean offset functions are derived by subtracting the mean contour from each training contour in the signed distance domain ($\tilde{\psi}_i = \psi_i - \bar{\Phi}$). The columns of the mean offset functions (size $N = N_1 \times N_2$, the same as the training images) are then successively stacked on top of one another to form one large column vector (\mathbf{S}_i) of size $1 \times N$. A new matrix \mathbf{S} (size $N \times n$), called the shape variability matrix, is formed from n such column vectors,

$$\mathbf{S} = [\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_n]. \quad (3.2)$$

The variance in shape is then computed by an eigenvalue decomposition on this shape variability matrix as,

$$\frac{1}{n} \mathbf{S} \mathbf{S}^T = \mathbf{U} \mathbf{\Sigma} \mathbf{U}^T. \quad (3.3)$$

Here \mathbf{U} is an $N \times n$ matrix whose columns represent n orthogonal modes of shape variation and $\mathbf{\Sigma}$ is a diagonal matrix of eigenvalues. By rearranging the columns of \mathbf{U} to form an $N_1 \times N_2$ structure, the n different Eigen shapes $\{\Phi_1, \Phi_2, \dots, \Phi_n\}$ can be obtained.

The mean shape and shape variability are used to define a level set function,

$$\Phi[\mathbf{w}] = \bar{\Phi} + \sum_{j=1}^k w_j \Phi_j. \quad (3.4)$$

Here \mathbf{w} are the weights for linearly combining the k principal Eigen shapes. By incorporating pose parameters into this level set framework a new level set function is obtained [99] that can handle object shapes with different sizes and orientation. Pose is defined using an affine transform $\mathbf{T}[\mathbf{p}]$, which is the product of three matrices, the translation matrix, the scaling matrix and the rotation matrix respectively,

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} h & 0 & 0 \\ 0 & h & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (3.5)$$

Here, $\mathbf{p} = [a, b, h, \theta]$, a, b are x, y translation parameters, h is the scale factor and θ is the angle of rotation. The pixel coordinates of the input image (x, y) are mapped to (\tilde{x}, \tilde{y}) of the affine transformed image. Note that a homogeneous coordinate system is used here. Using the homogeneous coordinate system allows the translation operation in equation (3.5) to be represented with a matrix multiplication. Figure 3.2 shows an original contour

and four affine transformed contours with parameters, $a=10$, $b=10$, $h=1.5$, $\theta = 30^\circ$ (counterclockwise) respectively.

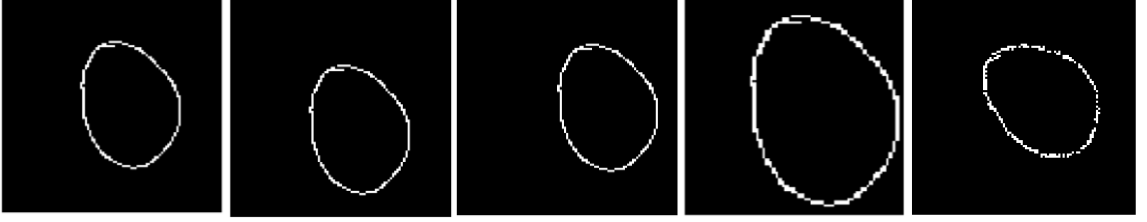


Figure 3.2 (From left) Original contour; x -translated contour; y -translated contour; scaled contour; rotated contour.

The new level set function incorporating pose parameters, derived by Tsai et al., is defined as [99]:

$$\Phi[\mathbf{w}, \mathbf{p}](x, y) = \bar{\Phi}(\tilde{x}, \tilde{y}) + \sum_{j=1}^k w_j \Phi_j(\tilde{x}, \tilde{y}). \quad (3.6)$$

The zero level of this level set function gives the segmenting contour, and its parameters are evolved by the GA. The goal of the Tsai et al. algorithm [99] was to optimize the \mathbf{w} , \mathbf{p} parameters using gradient descent optimization. Therefore, they optimized an energy function which consisted of terms based on regional statistics on pixel intensity values in images. The LSGA also optimizes the \mathbf{w} and \mathbf{p} parameters, but using a GA, which brings the flexibility to use any kind of high-level texture information and to incorporate spatial relationships for segmentation.

Before deriving the mean shapes and shape variances from the training data, the images are aligned for pose. Gradient descent is used to minimize the difference between

pairs of binary images with respect to their pose parameters. The transformed image based on pose is given by:

$$\tilde{\mathbf{I}} = \mathbf{T}[\mathbf{p}] * \mathbf{I}, \quad (3.7)$$

where, $\mathbf{T}[\mathbf{p}]$ is the 2D transformation matrix of equation (3.5). The energy functional used to minimize the difference between two images is given by:

$$E_{align} = \frac{\iint_{\Omega} (\mathbf{I}^i - \mathbf{I}^j)^2}{\iint_{\Omega} (\mathbf{I}^i + \mathbf{I}^j)^2}, i \neq j. \quad (3.8)$$

Here, Ω is the image domain. The denominator term normalizes the area, which prevents the images from shrinking. The initial pose parameters of one of the shapes is kept fixed and the pose parameters of the second image are derived to minimize the pose differences. Figure 3.3 shows the two contours superimposed with each other to show the difference in pose before and after alignment.



Figure 3.3 Top panel: Two contours before alignment. Bottom panel: After alignment.

3.2 Training stage: Deriving texture priors

The textural priors are derived from training images using the Gabor wavelet transform-based texture segmentation method, and the Laws' texture measures described in chapter 2. These priors are saved from the training images and used for textural classification of test images. The textural segmentation on the test image along with the spatial relationship map is used to determine the fitness of an individual segmenting contour.

When the Gabor wavelet transform (GWT) method is used, the mean and standard deviation of wavelet transform coefficients are derived from the training images and saved for performing textural classification on test images. For a two-class problem, the parameters are saved from the two known regions: the region inside and outside an object of interest. For a multi-class problem, the parameters are saved for each known region of the image. Figure 3.4 shows a pseudo code for saving the parameters from the GWT method.

```
FUNCTION Save_GWT_parameters

1. Load training images
2. Load manual segmentations
3. Generate Gabor Wavelets

4. Do (For each training image)
    4.1 Convolve image with each wavelet
    4.2 Save mean and variance of GWT coefficients for
        each known region for each wavelet
5. Average GWT coefficients for each wavelet over all
    training images and save them.

END
```

Figure 3.4 Pseudo-code showing how the textural segmentation parameters are saved for the GWT method.

For the Laws' texture measures (LTM) method, the weights of the Fisher's linear discriminant, which is used to optimally combine the 16 texture energy planes, are derived from training images. Here again, for a two-class problem, the weight parameters are saved from the two known regions inside and outside an object of interest. For a multi-class problem, each pair of classes is treated separately as a two-class problem. The parameters corresponding to each pair are saved from the training images. Figure 3.5 shows a pseudo code for saving the parameters from the LTM method.

```

FUNCTION Save_LTM_parameters
1. Load training images
2. Load manual segmentations
3. Generate convolution matrices
4. Do (For each training image)
    4.1 Convolve image with each convolution matrix to
        create 16 texture energy maps (TEMs)
    4.2 Derive weights for FLD to optimally combine TEMs.
5. Derive average weights of FLD over all
   training images and save them.
END

```

Figure 3.5 Pseudo-code showing how the textural segmentation parameters are saved for the LTM method.

3.3 Training stage: Deriving spatial relationships

Fuzzy spatial relationships are derived only for the pelvic CT/MRI images because the relative position of the prostate with respect to neighboring organs such as the bladder and rectum need to be incorporated for successful segmentation. The fuzzy landscape approach introduced by Bloch [8] and described in chapter 2 has been implemented here. For all training contours, the fuzzy interval limits n_1 , n_2 , n_3 , n_4 (shown

in figure 2.7) are first derived. The final fuzzy distance map (μ_d) is derived from all the training images by setting the fuzzy interval limits to minimum(n_3), maximum(n_4), minimum(n_1), maximum(n_2).

The fuzzy direction map is created by finding the mean direction between the reference object and the target object and applying a decreasing function g to it (μ_a) (for details refer chapter 2). The intersection of the fuzzy distance and direction map is then computed to derive the fuzzy landscape (μ) from the reference object to the target object (equation 3.9).

$$\mu_i = \mu_d \cdot \mu_a, \text{ where, } i = 1, \dots, o \quad (3.9)$$

Here o refers to the total number of reference objects. The pseudo code for deriving the fuzzy landscape is shown in figure 3.6. For multiple reference objects and one target object (for example reference objects: bladder and rectum; and target object: prostate) the union of the fuzzy landscapes from each reference object is derived to compute the final fuzzy landscape (equation 3.10).

$$\mu = \mu_1 + \mu_2 + \dots + \mu_o \quad (3.10)$$

Figure 3.7 shows two reference objects, R_1 (on top of the image) and R_2 (on the bottom of the image); and one target object T_1 (in the center of the image). Figure 3.8 upper panel shows the fuzzy landscape (μ_1) derived from R_1-T_1 . Figure 3.8 middle panel shows the fuzzy landscape (μ_2) derived from R_2-T_1 . Figure 3.8 bottom panel shows the final fuzzy landscape (μ) created by the union of fuzzy landscapes μ_1 and μ_2 .

```

FUNCTION Save_fuzzy_map

1. Load manual segmentation for Prostate, Bladder, and Rectum.
2. Do (For each training image with all three organs)
    2.1 Derive fuzzy landscape  $\mu_1$  between prostate and bladder.
    2.2 Derive fuzzy landscape  $\mu_2$  between prostate and rectum.
    2.3 Derive the fuzzy landscape as  $\mu_1 + \mu_2$ 
3. Derive average fuzzy landscape over all training images and
   save it.
END

```

Figure 3.6 Pseudo-code for deriving the fuzzy landscape from training images.

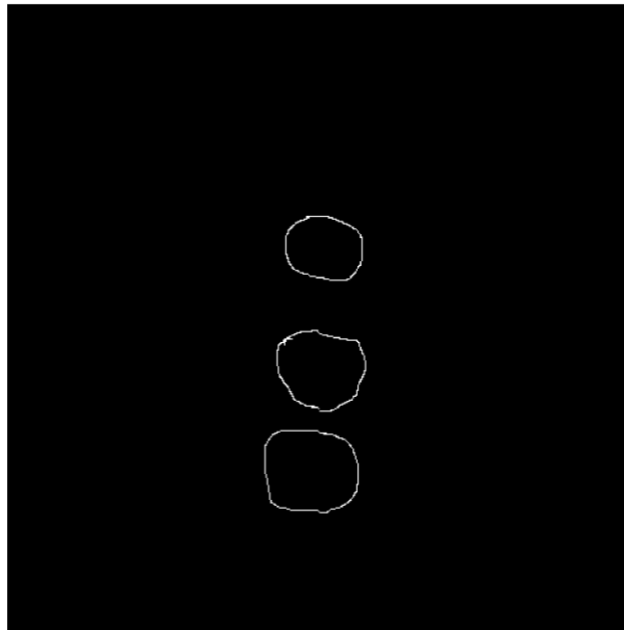


Figure 3.7 Reference objects R_1 (above) and R_2 (below) and target object (T_1) in the center.

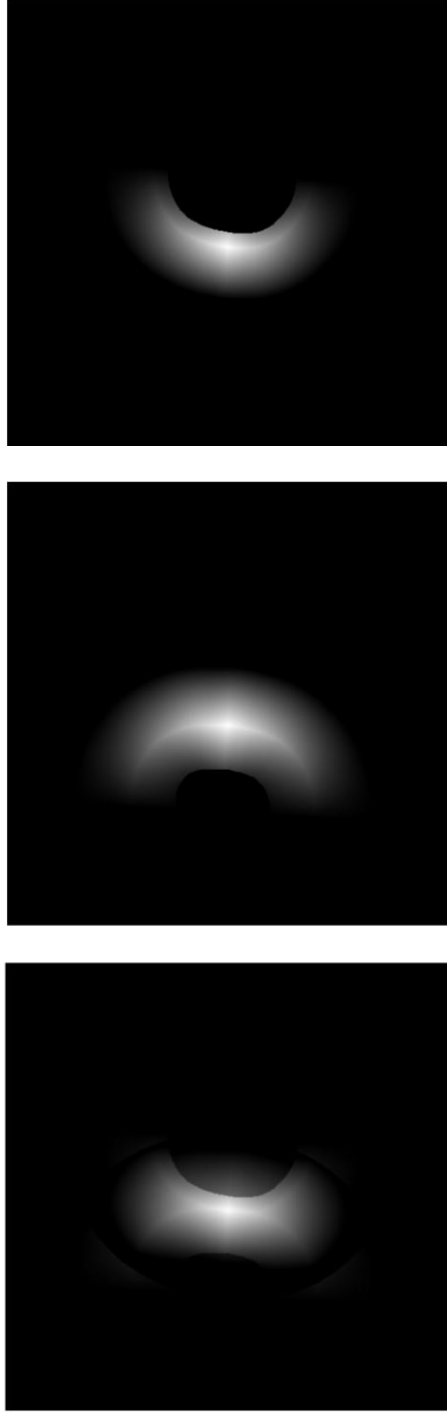


Figure 3.8 Top panel: Fuzzy landscape (μ_1) derived from R_1-T_1 . Middle panel: Fuzzy landscape (μ_2) derived from R_2-T_1 . Bottom panel: Final fuzzy landscape (μ) created by the union of μ_1 and μ_2 .

3.4 Segmentation stage

Segmentation is performed using LSGA by optimizing a population of segmenting contours for shape, texture of enclosed region, relative location, and pose. Each individual in the GA population is defined as a fixed-length *chromosome* of real-valued *genes* which is called its genotype,

$$Individual = [w_1, w_2, w_3, w_4, a, b, h, \theta]. \quad (3.11)$$

The four weight parameters are used for deriving the weighted $\pm\sigma_1$ and $\pm\sigma_2$ variation (where σ_i^2 is the eigenvalues corresponding to i principal Eigen shapes) of the mean shape and a, b, h , and θ are pose parameters as defined in equation (3.5). The pose parameters are chosen randomly from the space of real numbers. Each individual in the population represents a unique segmenting contour which is called its phenotype. Figure 3.9 shows a sample genotype (*chromosome*) and the corresponding phenotype (segmenting contour) of a LSGA individual.

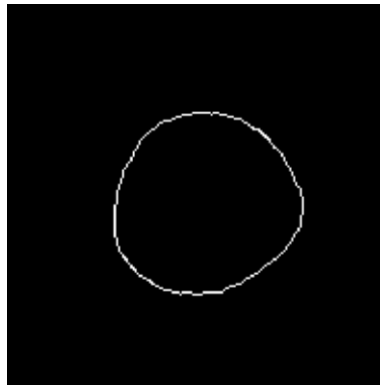


Figure 3.9 A segmenting contour with the genotype $[w = [410, 329, 723, 7.4], p = [2, 6, 1, 6]]$

The fitness of an individual is calculated based on the degree to which the segmenting contour encloses the region of interest (ROI) in a test image (a new image not in the training set). It is calculated differently for thermographic images (because they only combine shape and texture features) and pelvic images (because they combine texture, shape and spatial relationships).

3.4.1 Fitness: Combining shape and texture

The ROI or the “truth plane” in this case is determined by the textural segmentation of the test image. Here, LSGA performs deformable template matching around the textural region of interest. The fitness is calculated by comparing the two binary images \mathbf{B}_1 and \mathbf{B}_2 . The first binary image (\mathbf{B}_1) is obtained from the GA individual being evaluated, by placing the corresponding contour on the test image and classifying all pixels inside the contour as “true” and all other pixels as “false”. Binary image \mathbf{B}_2 is generated by the textural classification of pixels on a test image derived using LTM or GWT methods. Each pixel of the test image is classified as “true” (ROI) or “false” (does not belong to ROI). The fitness is a function of the detection rate (D) and the false alarm rate (F) as:

$$\text{Fitness} = 500(D + (1 - F)). \quad (3.12)$$

$$D = \frac{\begin{array}{l} \text{Count of total pixels inside the segmenting contour } (\mathbf{B}_1) \\ \text{with textural classification } (\mathbf{B}_2) \text{ "true"} \end{array}}{\text{Total number of pixels inside the segmenting contour } (\mathbf{B}_1)}$$

$$F = \frac{\begin{array}{l} \text{Count of total pixels outside the segmenting contour } (\mathbf{B}_1) \\ \text{with textural classification } (\mathbf{B}_2) \text{ "true"} \end{array}}{\text{Total number of pixels outside the segmenting contour } (\mathbf{B}_1)}$$

The detection rate is defined as the fraction of pixels inside the segmenting contour of binary image (\mathbf{B}_1) that have been classified as “true” pixels in the textural classification \mathbf{B}_2 . Note that \mathbf{B}_2 is the so-called “truth plane” used by the fitness function and is not the ground truth (derived from manual segmentation). The ground truth images are only used for evaluating the final segmentation results. The false alarm rate denotes the fraction of pixels outside the segmenting contour \mathbf{B}_1 that are classified as “true” pixels by textural segmentation \mathbf{B}_2 .

The constant 500 scales the fitness so that the maximum fitness score that can be attained using this function is 1000 giving the first few significant digits of the fitness value. A higher fitness score means that more pixels inside (outside) the contour belong to the desired (other) texture type that was derived from the training data.

3.4.2 Fitness: Combining shape, texture, and spatial relationships

When relative location maps are used, a gray-scale image \mathbf{P} is derived by a dot product of the binary texture segmentation on the test image (\mathbf{B}_2) and the fuzzy relative position map derived from training images (\mathbf{RL}). Therefore, each pixel in the texture segmentation is weighed by the fuzzy membership value of finding the prostate derived from the fuzzy relative location map.

$$\mathbf{P} = \mathbf{B}_2 \cdot \mathbf{RL} \quad (3.13)$$

The binary image (\mathbf{B}_1) is obtained in the same way as before, by placing the segmenting contour on the test image and classifying all pixels inside the contour as “true” and all other pixels as “false”. A higher fitness score means that more pixels inside (outside) the contour not only belong to the desired (other) texture type but also satisfy

the spatial relationships that were derived from the training data. Here again, the fitness is a function of the detection rate (D) and the false alarm rate (F):

$$\text{Fitness} = 500(D + (1 - F)), \quad (3.14)$$

$$D = \frac{\begin{array}{c} \text{Sum of pixel values in } \mathbf{P} \text{ that} \\ \text{lie inside the segmenting contour } (\mathbf{B}_1) \end{array}}{\text{Total number of pixels inside the segmenting contour } (\mathbf{B}_1)},$$

$$F = \frac{\begin{array}{c} \text{Sum of pixel values in } \mathbf{P} \text{ that} \\ \text{lie outside the segmenting contour } (\mathbf{B}_1) \end{array}}{\text{Total number of pixels outside the segmenting contour } (\mathbf{B}_1)}.$$

The detection rate is defined as the mean value of pixels in \mathbf{P} that lie inside the segmenting contour of \mathbf{B}_1 . The false alarm rate denotes the mean value of pixels in \mathbf{P} that lie outside the segmenting contour of \mathbf{B}_1 . The detection rate and false alarm rate in this case are determined from the mean of the pixel values in \mathbf{P} that match the segmented contour in \mathbf{B}_1 .

3.4.3 Selection, Crossover and Mutation

The processes in GA evolution: selection, crossover, and mutation is iterated to create new generations until an individual with satisfactory fitness value is created or after a specified number of generations have been produced. Selection is performed either using fitness proportionate or rank selection (details in section 2).

Single-point crossover and mutation described in section 2 were for binary strings. Since, the LSGA individuals consist of real-valued genes, the crossover and mutation operations are modified:

- *Crossover* is implemented by swapping fixed length segments of real-valued *genes* between two individuals.
- *Mutation* is performed by randomly changing the value of a gene with a real number from a pre-specified range of numbers.

3.5 Extension to three dimensions

The LSGA is extended to three-dimensions by using 3D pose parameters; x, y, z translation (a, b, c), scale (h), yaw (α), pitch (β) and roll (θ).

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} h & 0 & 0 & 0 \\ 0 & h & 0 & 0 \\ 0 & 0 & h & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \mathbf{R}_x * \mathbf{R}_y * \mathbf{R}_z * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.15)$$

\mathbf{R}_x , \mathbf{R}_y , and \mathbf{R}_z are the rotation matrices about the x, y and z axes respectively:

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.16)$$

$$\mathbf{R}_y = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

$$\mathbf{R}_z = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.18)$$

The individuals of the 3D-LSGA population are based on the new pose parameters $p = [a, b, c, h, \alpha, \beta, \theta]$. Thus, the 3D-LSGA individual is given by:

$$Individual = [w_1, w_2, w_3, w_4, a, b, c, h, \alpha, \beta, \theta]. \quad (3.19)$$

Figure 3.10 shows a sample genotype (*chromosome*) and the corresponding phenotype (segmenting surface) of a 3D LSGA individual.

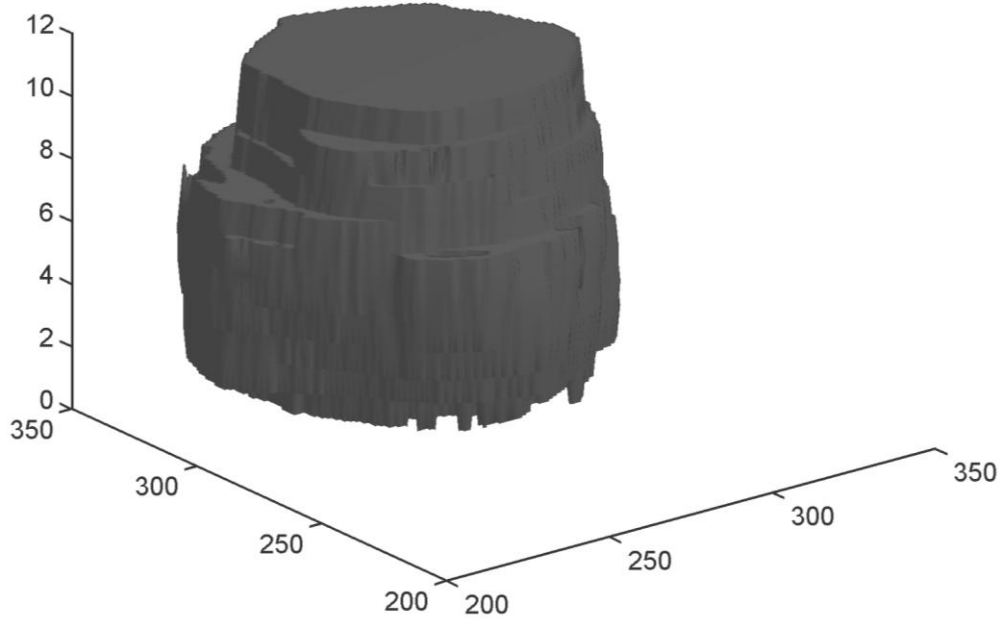


Figure 3.10 A sample 3D-LSGA individual with genotype $[w=[4554, 1595], p=[5, 12, 1, 1.4, 0, 0, 0]]$

The mean shape and shape variability are derived from the 3D images generated by stacking the slices of the CT/MR scans from the training data. The 3D segmenting surface generated by the 3D-LSGA segments all the slices of a test image at once.

3.3.1 The steps for performing 3D segmentation

A. Steps for training the 3D-LSGA

- a. The stacks of 3D training images are aligned with respect to one 3D training image.
- b. 3D mean shape and shape variability are then derived using 3D PCA. Here, the segmenting surfaces are converted into 3D signed distance surfaces. The eigen decomposition is then performed similar to section 3.1 with the only difference that 3D surfaces are used instead of 2D images.
- c. A 3D level set function is then created by incorporating 3D pose of equation 3.12.
- d. Texture segmentations and fuzzy landscapes are then computed for all 2D slices of every 3D training image. The 2D texture parameters and the 2D fuzzy landscapes are then saved as described in sections 3.1 and 3.2. These are not computed in three dimensions because they take impractically long to derive.

B. Steps for 3D-LSGA

- a. An initial population of 3D segmenting surfaces (i.e, stacks of 2D segmenting contours) is generated at first.
- b. For each 3D test image, the following steps are taken to perform segmentation:
 - i. Each 3D segmenting surface is placed on the 3D test image such that every slice of the segmenting surface (a segmenting contour) lies on a separate 2D slice of the test image.
 - ii. The fitness values are derived from the segmentation performance on all 2D slices of the test image. The fitness of the segmenting surface is then

computed as the mean fitness derived from segmentation of all the slices of the test image.

- iii. Selection, crossover and mutation are then performed to create a new population until the stopping criteria is satisfied.

Selection: Selection is performed either using fitness-proportionate or rank selection method to select a pair of individuals from the population.

Crossover: Single-point crossover is performed between each pair of real-valued individuals with a probability P_c .

Mutation: Mutation is performed by randomly changing the value of a *gene* of the 3D-LSGA individual with a probability P_m .

3.4 Summary

The LSGA performs segmentation in two as well as three dimensions. It uses a fitness function, which is an implicit function of payoff values based on the segmentation performance. This makes the level set function optimization for performing segmentation flexible enough to incorporate high-level features which may or may not have derivatives.

4. Segmentation of Thermographic Images of Hands

One of the datasets on which the LSGA has been applied is thermographic images of hands. The images of hands were acquired for studying the relationship between skin temperature of hand and upper extremity musculoskeletal disorders (UEMSDs). The pathophysiology in UEMSDs is largely unknown. However, a component may include reduced blood flow in the upper extremity [11][39][40][60][84][88].

Infrared thermography reveals skin temperature which is largely determined by subcutaneous perfusion. For this study subjects with UEMSD were given a typing challenge and images were taken at periodic time intervals. This segmentation problem is challenging because the fingers of the patients disappear on the thermographic images when they become cold after a typing challenge (a symptom of UEMSD). Successful segmentation of these images involves deriving the prior human knowledge of the shape of the hand, and modeling the movement of the hand and fingers from training images.

4.1 Data and problem description

The data for this analysis has been obtained from Temple University's Ergonomics and Work Physiology Lab where researchers are studying musculoskeletal disorders of distal upper extremity (e.g., tendinitis and carpal tunnel syndrome). Such disorders are widespread and account for 29% of the total workplace illnesses in the United States [7]. UEMSDs have been associated with prolonged keyboard usage and manually intensive clerical jobs [33]. The mechanisms of UEMSD are still not clearly

understood. However, reduction in blood flow has been related to the pathophysiology of disorders such as tendonitis [84], nerve compression syndrome [98], trapezius myalgia [60] etc. Since subcutaneous perfusion plays a major role in skin temperature determination, and is also detectable via infrared thermography, thermal images have been used to examine these disorders. There are very few diagnostic tests that can be used to grade the severity of UEMSDs. Physiological tests, such as thermographic analysis, have the potential to detect UEMSDs in their early clinical phases.

4.1.1 Equipment for acquiring images

For this study far-infrared images of hands were acquired at Temple University using ThermoCAM AM40 thermographic camera (FLIR Systems, Wilsonville, OR) with a sampling rate of 7 Hz. The camera had a resolution of 1.3mrad and a sensitivity of 0.08° at 30° C.

Subjects were given a nine-minute typing challenge in a simulated office environment. The images were acquired at the following times: before typing, after 0-2 minutes of typing, after 3-5 minutes of typing, and after 8-10 minutes of typing.

Figure 4.1 (upper panel) shows the hands of four subjects before starting to type. Figure 4.1 (lower panel) shows the hands of the same patients 9 minutes after typing. The subjects' fingers are partially visible on these images because the hands approach the temperature of the surrounding surface. Also a near infrared spectroscopy (NIRS) probe is adhered to the skin above the first dorsal interosseous muscle of their right hand (between the thumb and index finger) and on some of these images the boundary of the hand touching the probe is not visible.

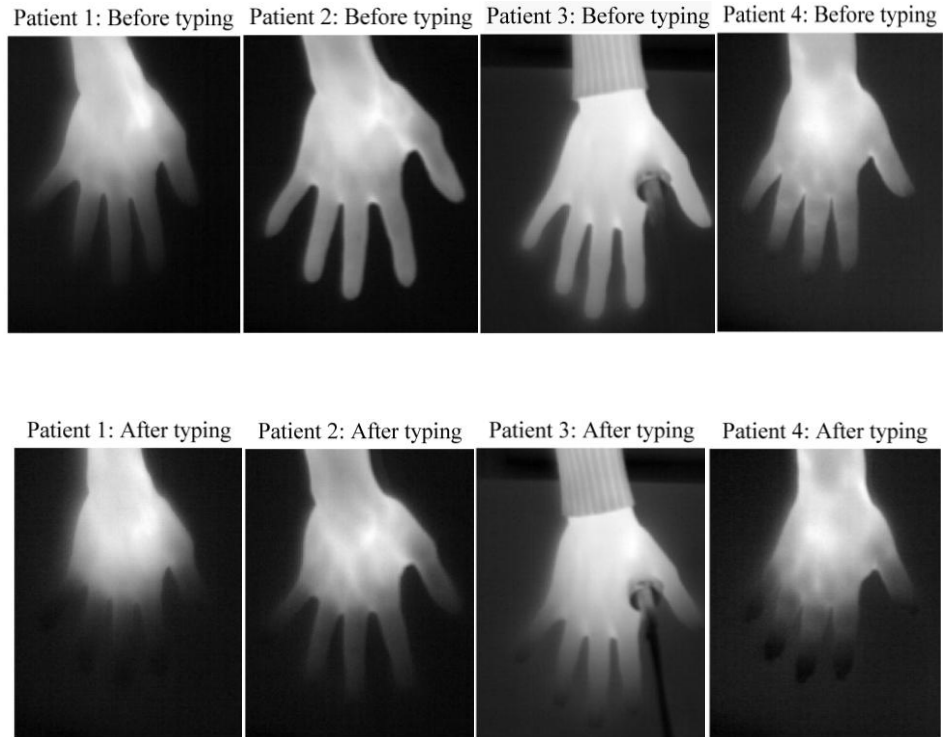


Figure 4.1 Thermographic images of hands of four subjects taken before typing (upper panel); after 9 minutes of typing (lower panel). The fingers start to become invisible due to reduced blood flowing in the subjects' hands.

These images were manually segmented by a human who has a prior knowledge of the shape of the hands and has also looked at the hand images of each subject taken prior to typing. The challenge in the problem is more than mere shape matching because the subjects tend to move their hands and fingers during the imaging process. Therefore, a rigid template matching method is not suitable for this problem. The LSGA performs a deformable template matching within known bounds of mean shape and shape variability (movement of fingers and the hand itself) and the texture of the region it encloses to perform the segmentation task.

Once an image is manually segmented, a score of the mean temperature of the hand is generated and compared with the temperature of the hand at successive time steps to determine if and where the blood flow is changing in the hand with time. The dataset consists of images from four subjects. For each subject there are 500 images for each time range, that is, a total of about 2000 images per subject.

4.2 Experimental setup

The experiments for segmenting the thermographic images of hands were set up in the following way. The images acquired prior to typing were used as the training images. The images taken after the typing challenge at various intervals were used as the test images. A subset of 240 test images (every 25th image was chosen, i.e., about $20 \times 3 = 60$ images per patient) were used for validation purposes. The ground truth in the form of manual segmentation was derived only for the validation set. The segmentation performance for rest of the test images were analyzed visually by a human. A subset of the training images (~ 100) was also manually segmented by a human to derive the model for known shape, texture and movement of the hands of the subjects.

Figure 4.2 shows the mean shape and the variability of the mean shape from one patient. The eigenvalue σ_1 depicts the movement of the fingers and σ_2 the width of the fingers which varies between multiple segmentations (shown in Table 4.1 for all patients). Other eigenvalues affect the length of the fingers, the size of fingertips etc., and are ignored for this study because they are not the principal modes of variation of the shape of hands.



Figure 4.2 Mean shape (center). $\pm 3\sigma_1$ variability depicting movement of fingers (left, right). $\pm 3\sigma_2$ variability of width of fingers (top, bottom).

Table 4.1 Shape variability of hands (σ_1 and σ_2 are the two principal eigenvalues).

	σ_1	σ_2
Patient 1	$2.7 \cdot 10^4$	$1.5 \cdot 10^4$
Patient 2	$1.0 \cdot 10^2$	$0.7 \cdot 10^2$
Patient 3	$1.8 \cdot 10^2$	$1.1 \cdot 10^2$
Patient 4	$2.2 \cdot 10^2$	$1.0 \cdot 10^2$

The parameters used by the GA to evolve the segmenting contour are shown in Table 4.2. A population size of 50 individuals was used for the LSGA. The justification for this choice is described below. The weights for k eigen shapes \mathbf{w} are chosen from a space of integers and then scaled by the variance (σ) to derive the weighted eigen shapes. The rotation parameter is assigned a value in the range of -30 to 30 degrees. Translation parameters are chosen from the range of integers (0-30). The scale parameter is fixed to 1 since the size of the hand remains the same in all the images. Rank selection and fixed-length single-point crossover have been implemented here with a probability of 0.1 and 0.5 respectively. Rank selection is implemented by comparing the fitness of individuals, and making individuals with higher fitness more likely to be selected to produce offsprings. Fixed-length crossover is performed by swapping fixed length segments of *genes* between two individuals. Mutation is performed by randomly changing the value of a gene with another real number within a fixed range of values. Mutation rate is defined as the probability of a single gene to be mutated. Similarly, the crossover rate defines the probability of a crossover to occur between two individuals.

Figures 4.3, 4.4, and 4.5 show how the fitness values change as the GA parameters such as mutation rate, crossover-rate and population size vary. The figures clearly show that the convergence rate of the GA is similar for population sizes of 50 and 100, mutation rates of 5% and 10% and crossover rate of 50%. Therefore, a population size of 50, mutation rate of 10% and a crossover rate of 50% were chosen as the default parameters for performing segmentation on all test images for this dataset.

Table 4.2 LSGA parameters

Population Size	50
Mutation Rate	10 % per gene
Crossover Rate	50% single-point
Selection Criteria	Rank Selection
Weights for eigen shapes, w	(1-5) integers
Translation parameters a, b	Integer (0-30)
Rotation parameter θ	-90° to $+90^\circ$
Scale parameter h	1

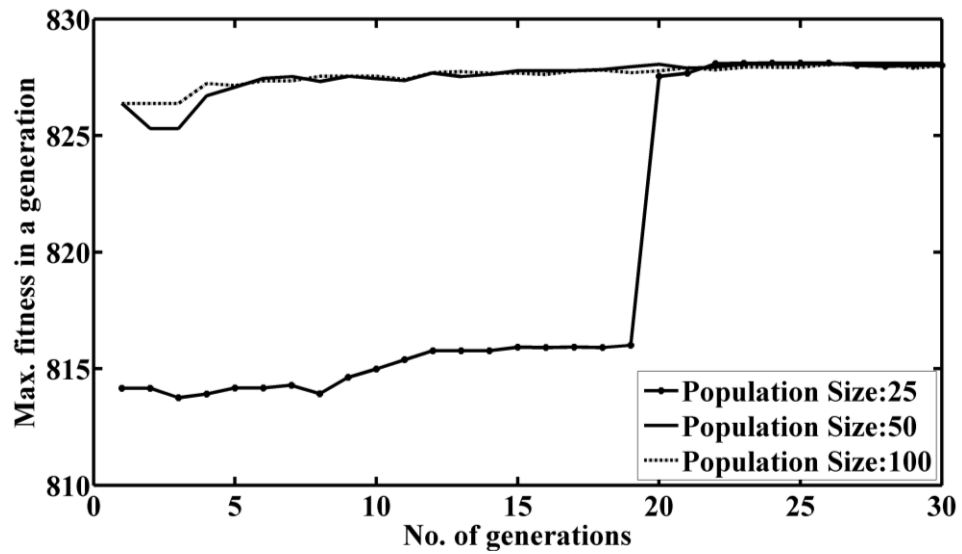


Figure 4.3 Variation of the maximum fitness by number of generations of the GA run for the population sizes of 25, 50, and 100.

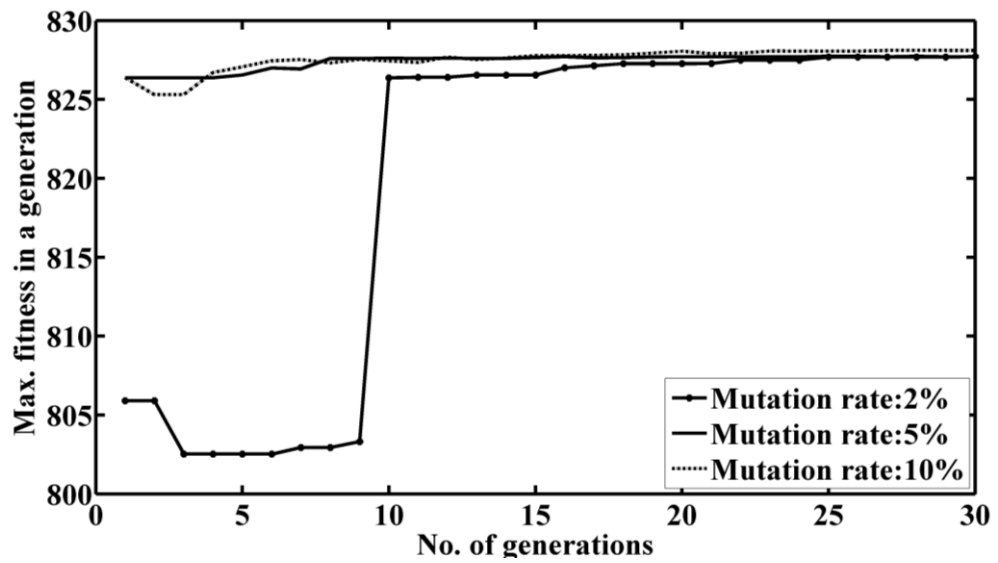


Figure 4.4 Variation of the maximum fitness by number of generations of the GA run for the mutation rates of 2%, 5%, and 10%.

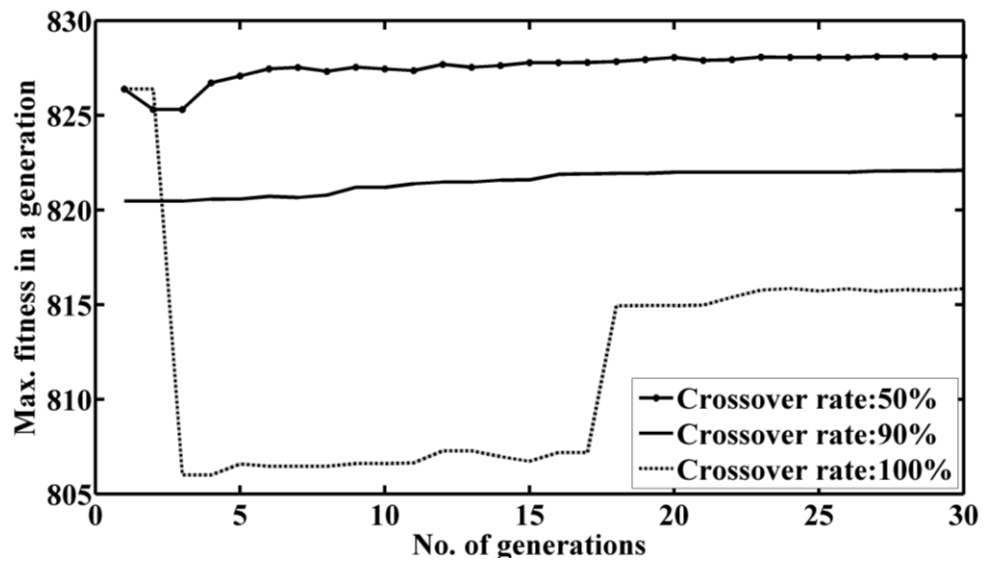


Figure 4.5 Variation of the maximum fitness by number of generations of the GA run for the crossover rates of 50%, 90%, and 100%.

4.3 Evaluation Criteria

For evaluating the performance of the algorithm the definitions of closeness of the segmentation outcome to the truth (here, manual segmentation) were derived using the dice similarity coefficient [44] and the partial Hausdorff distance [23]. Both of these measures have been extensively used for evaluating segmentation algorithms. The ground truth was obtained by averaging over multiple manual delineations.

The dice similarity coefficient provides a measure of the degree of overlap between two segmentations as:

$$DSC(\mathbf{A}, \mathbf{B}) = \frac{2|\mathbf{A} \cap \mathbf{B}|}{(|\mathbf{A}| + |\mathbf{B}|)}. \quad (4.1)$$

Here, \mathbf{A} is the segmenting contour and \mathbf{B} is the ground truth derived from manual segmentation. A DSC of 1 indicates a perfect match and 0 indicates no match.

The partial Hausdorff distance is derived between the boundary points of two contours. If $\mathbf{A} = \{a_1, \dots, a_p\}$ and $\mathbf{B} = \{b_1, \dots, b_q\}$ be finite sets of points on two contours, then the partial Hausdorff distance between them is defined as:

$$H(\mathbf{A}, \mathbf{B}) = \max(h(\mathbf{A}, \mathbf{B}), h(\mathbf{B}, \mathbf{A})),$$
$$h(\mathbf{A}, \mathbf{B}) = \max_{a \in \mathbf{A}} \min_{b \in \mathbf{B}} \|a - b\|. \quad (4.2)$$

The function $h(\mathbf{A}, \mathbf{B})$ takes each point in \mathbf{A} and finds the closest point in \mathbf{B} from that point. It then ranks the points in \mathbf{A} based on the distance values and finds the point with the greatest “mismatch”. Thus, the partial Hausdorff distance is a measure of the distance by which two contours i.e., the final segmentation outcome and the ground truth differ.

4.4 Results and discussion

Segmentation was performed on the test images of each subject using the following methods: LSGA, Gabor wavelet transform-based segmentation algorithm (GWT), Laws' texture measures (LTM) and the Chan & Vese algorithm (CV). Figure 4.7 (first from left) shows a manually segmented hand test image. Since there are only two possible classes: hand and outside the hand for thermographic images, only two-class classification was performed for textural segmentation. The edge derived from the binary outcome of the Gabor wavelet transform-based segmentation algorithm on the sample test image is shown super-imposed with the test image in figure 4.7 (second from left). This method finds only the region of the hand visible by pixel intensity variations on the image. Figure 4.7 (third from left) shows the result of applying the Chan & Vese level set-based algorithm to the same hand image. Since both these methods do not have the notion of a known shape they could not segment the entire hand on the test image.

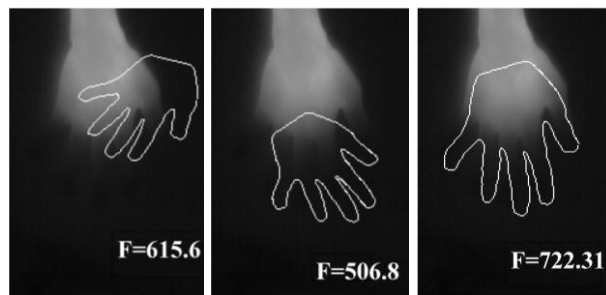


Figure 4.6 Three candidate segmenting contours in the GA population. Here, F denotes the fitness of each individual.

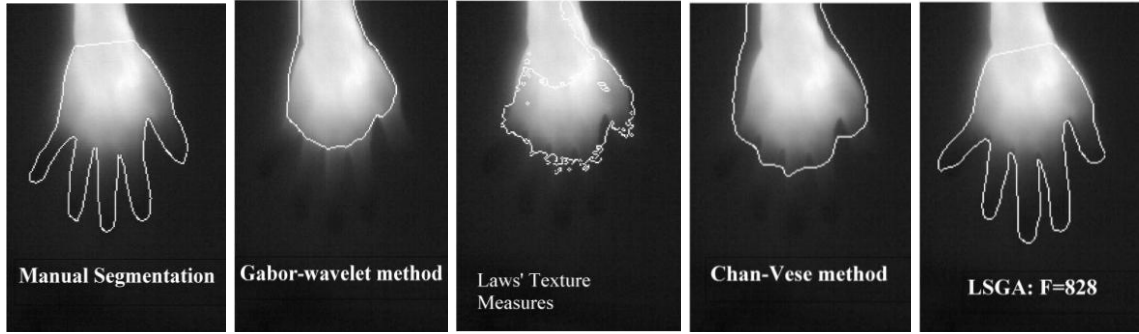


Figure 4.7 Segmentation outcome on a test image (manual segmentation shown in first from left) using: Gabor wavelet based method (second from left), Laws' texture measures (third from left), Level set-based method of Chan & Vese (fourth from left), LSGA (rightmost).

Finally, the LSGA was used to segment the same hand image. Figure 4.6 shows some candidate segmenting contours in the GA population. The LSGA found the optimal location and pose of the hand in the image from a population of 50 segmenting contours (Fig. 4.6 rightmost panel). The segmenting contour is shown on top of the test image to show the segmentation outcome here. The fitness of the final segmenting contour for this image was 828. The figure clearly shows that the LSGA outperforms the other methods. This is also confirmed by visual analysis of the results. Figure 4.8 shows the maximum fitness of the population plotted over 30 generations for an exhaustive search by the LSGA (± 100 pixels around the mean location of the hand) on a single image. The fitness value increases because the GA is based on the principle of survival of the fittest. Figure 4.9 shows the segmentation results from every time interval for each patient. The average *DSC* and *H* were computed from all the segmentation outcomes for each patient. Tables 4.3 and 4.4 show the average values of *DSC* and *H* obtained for all the images in the

validation set for each of the four methods. Table 4.5 shows the header file saved by the GA during for a sample test image.

Note that in the case of patient 4 the *DSC* value attained by the LSGA is low and for patient 1 the *H* is relatively high even though a visual inspection of the results shows satisfactory segmentation. This is due to the limitation in acquiring accurate ground truth as the fingertips are completely invisible in the test images. Also, in case of patient 4 the *DSC* values for LSGA match that of the GWT method. This was expected because the LSGA used the textural classification from Gabor wavelet transform method to derive fitness values. Therefore, it could at least perform as well as the GWT in this case.

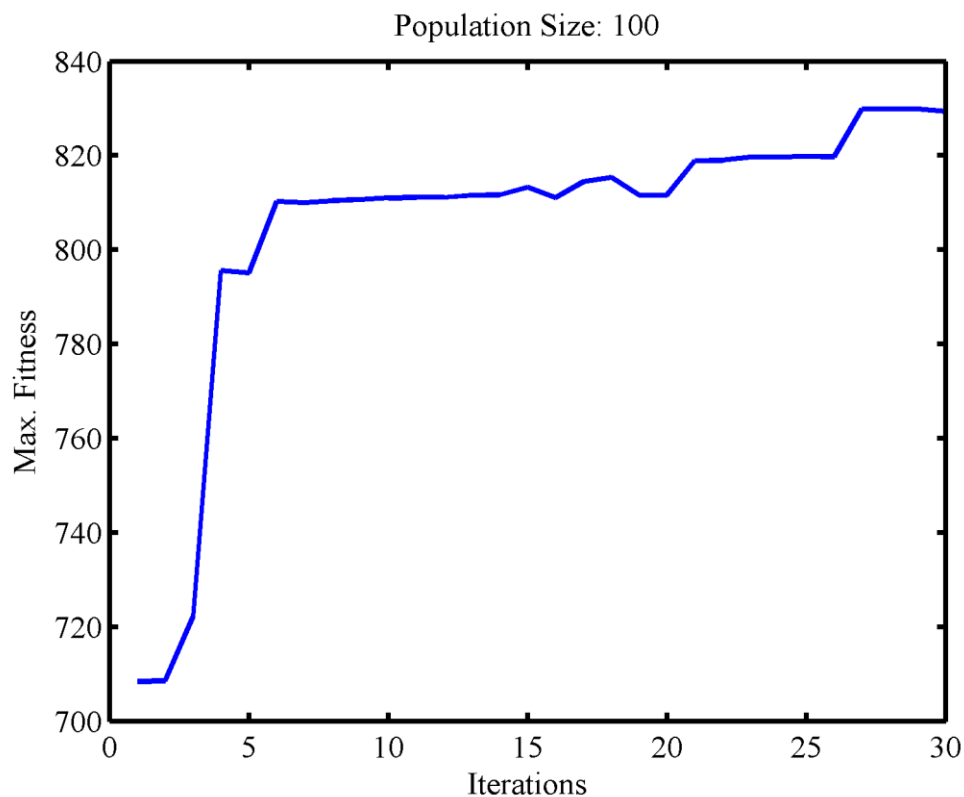


Figure 4.8 Maximum fitness of the population plotted over 30 generations for an exhaustive search by the LSGA on a single image.

Table 4.3 Comparison with ground truth using *DSC*. The bold values represent the best (highest values) in a row. Note that LSGA did not perform as well in the case of patient 4 due to the limitation in acquiring accurate ground truth as the fingers of this patient were completely invisible.

<i>DSC</i>	<i>CV</i>	<i>LTM</i>	<i>GWT</i>	<i>LSGA</i>
Patient 1	0.7±0.008	0.66±0.01	0.67±0.005	0.8±0.24
Patient 2	0.78±0.01	0.66±0.01	0.74±0.002	0.9±0.25
Patient 3	0.76±0.06	0.66±0.05	0.7±0.004	0.85±0.20
Patient 4	0.7±0.003	0.7±0.01	0.6±0.008	0.6±0.41

Table 4.4 Comparison with ground truth using *H*. The bold values represent the best (lowest values) in a row. Note that *H* represents the point of maximum discrepancy between two contours. It is much lower in the case of LSGA as compared to other methods because it is the only method that tries to outline the fingers on these images.

<i>H</i>	<i>CV</i>	<i>LTM</i>	<i>GWT</i>	<i>LSGA</i>
Patient 1	41±4.1	38±2.6	51±2.1	6.7±6.8
Patient 2	30±9.2	45±1.2	42±1.5	3.4±4.8
Patient 3	46±8.08	32±1.53	46±4.04	3.5±1.8
Patient 4	34±1.52	21±11.8	41±0.58	3.5±6.2

Table 4.5 Header file created by the LSGA for a sample test image

Generation	Avg_Fitness	Std_Dev	Best_Fitness
1	607.3	93.3	803.5
2	661.8	88.3	803.5
3	694.7	95.3	803.5
4	728.3	85.1	803.6
5	742.2	77.8	804.7
6	764.2	60.9	804.7
7	762.6	81.8	804.7
8	738.3	108.7	804.7
9	764.5	83.5	804.7
10	789.3	54.7	804.7
11	786.0	61.3	804.7
12	783.7	71.1	804.8
13	799.2	28.4	804.8
14	794.7	43.3	804.9
15	784.0	64.8	810.5
16	799.0	40.8	810.6
17	803.1	19.0	811.6
18	785.5	69.0	811.5
19	794.9	52.4	811.4
20	806.9	15.4	811.4
21	796.4	52.2	811.4
22	785.2	77.0	811.4
23	778.8	88.4	811.4
24	786.6	71.9	811.5
25	795.6	57.6	811.5
26	796.8	56.9	811.5
27	799.9	42.7	811.5
28	788.2	62.5	811.5
29	778.6	83.9	811.5
30	803.0	37.0	828.1

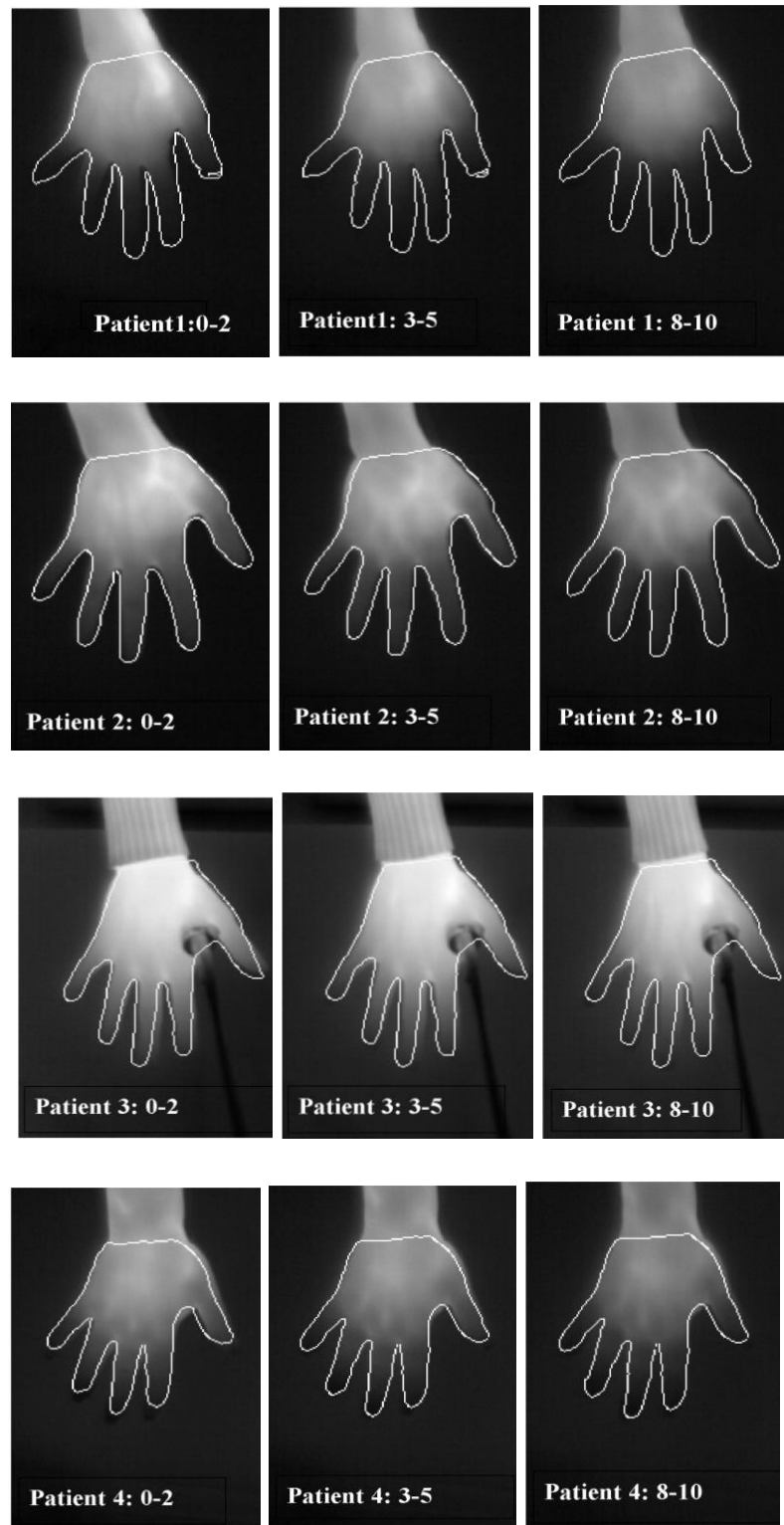


Figure 4.9 Segmentation result from every time interval for each patient.

4.5 Summary

The LSGA performs derivative-free optimization of a level set function for image segmentation. This brings flexibility to the level set curve evolution process by letting the user choose different kinds of features for exploring the fitness landscape. In this chapter two types of features, texture and shape, have been explored for evolving the level sets for segmenting thermographic images of hands. Although these images had visible textural areas separating the hand region from the background, the knowledge of known shape was needed to segment them. The LSGA combined shape with texture to achieve the desired segmentation. Part of this chapter was published as a research paper in the Springer journal, *Evolutionary Intelligence* [34].

5. Prostate Segmentation on Pelvic CT/MRI Images

Prostate cancer is one of the most commonly diagnosed malignancies in men over the age of 50 [94]. When diagnosed at an early stage, the disease is curable, and even at later stages treatment can be effective. Treatment options vary depending on the extent of the cancer, and prognosis worsens when diagnosis occurs at an advanced stage. A key ingredient for optimal treatment of prostate cancer is accurate target segmentation on medical images used for treatment planning. Traditionally images are manually segmented by a radiologist or a radiation oncologist to localize the prostate gland within the pelvic anatomy prior to treatment planning. Nevertheless, manual segmentation has its limitations, particularly due to inter-observer and intra-observer segmentation variability. Furthermore, with the rise in the implementation of adaptive radiotherapy, manual segmentation has become a tedious process, necessitating a reliable automated process for expediting prostate localization on CT and/or MRI images.

This procedure cannot be used for diagnosis or staging of prostate cancer. The stage of prostate cancer is usually determined using a combination of clinical findings such as the prostate-specific antigen level and the Gleason score [37] derived from the biopsy of a surgical specimen. Target volumes are determined by multiplying the area of the manually drawn contour with the slice thickness of the 2D CT/MRI images. A major challenge of conformal radiotherapy for treating prostate cancer is the identification of accurate target volumes because the interface between the prostate and its adjacent organs such as the bladder is indistinct and difficult to delineate. The different types of margins used for conformal radiotherapy are discussed in 5.1.2.

5.1 Problem description

The prostate gland is a male reproductive organ located below the bladder and in front of the rectum and is about 3 cm in length along the height of the body. The almond-shaped prostate gland can be deformed by bladder and rectal filling. In addition to this, the size of prostate can vary considerably across patients making automatic segmentation a challenging problem. Figures 5.1 and 5.2 show a single slice of pelvic CT and MRI scan from two patients. The red contour in the center was marked as the prostate by a radiologist. The organ just below the prostate contour is the rectum. The large structures around the prostate (white regions on CT image of figure 5.1) are the bones. Note that the edges near the boundary of the prostate that was marked by the radiologist are not prominent. The radiologist uses prior knowledge of the anatomy of the male pelvis, along with learned texture and shape information, to approximately outline the prostate on these images. These contours are stacked on top of one another to create the three dimensional (3D) shape of the prostate.

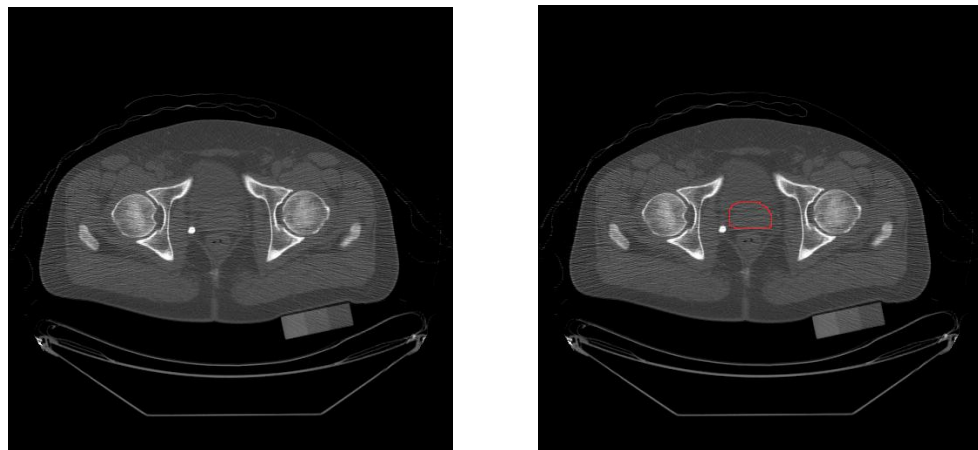


Figure 5.1 A 2D pelvic CT scan (left). The red contour (right) in the center is the prostate outlined by an expert. The white structures surrounding the prostate are the bones.

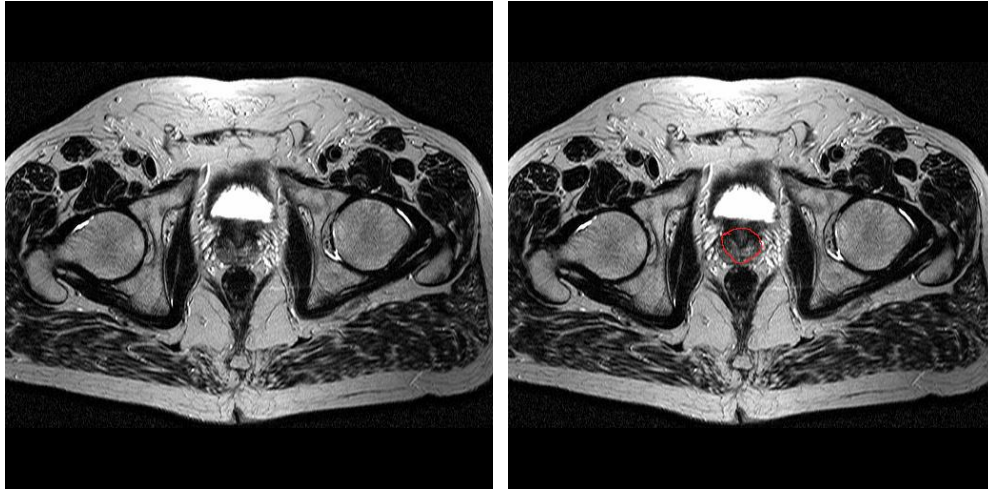


Figure 5.2 A 2D pelvic MRI scan (left) and manually segmented (red) contour (right).

5.1.1 Acquisition of pelvic images and artifacts

To perform treatment planning for prostate cancer, images are usually acquired as CT and MRI images. Information from CT images is useful for treatment planning in the following ways [61]:

- a) They provide a means to determine the size and location of the prostate.
- b) The pixel values in the CT image correspond to the electron density of the imaged tissue, which is used to determine the radiation therapy dose distribution.

However, CT images have poor contrast and there can be significant variability when interpreting structures in CT. MRI images have better contrast and are acquired to complement the treatment plan derived from CT images for prostate cancer patients. The MRI images are co-registered with the CT images using bony landmarks and the final radiation therapy treatment plan occurs through a fusion of information from both types of imaging modalities.

A. Computed Tomography (CT)

In computed tomography, patients are subjected to beams of x-rays at different angles. As the x-ray beams pass through the body of the patient, they get attenuated. This attenuated signal is then detected and reconstructed to produce a CT image. The reconstruction of a 2D image from a set of projections is performed via filtered back-projection. Suppose, the projected signal is given by $p(r, \phi)$, where r is the distance and ϕ is the angle of the x-ray source and detector. Then, each projected signal is Fourier-transformed along r -dimensions to produce $P(k, \phi)$. These projections are then multiplied with $H(k)$ (Fourier-transform of the filter function $h(r)$) to smooth out statistical noise and generate the filtered projections $P_{filt}(k, \phi)$. These projections are then inverse-Fourier-transformed to produce the reconstructed image, $\hat{f}(x, y)$:

$$\hat{f}(x, y) = \sum_{j=1}^n \mathbb{F}^{-1} \{ P_{filt}(k, \phi_j) d\phi \}. \quad (5.1)$$

Here, n is the number of projections and \mathbb{F}^{-1} represents the inverse Fourier transform. Nowadays, CT images are acquired using spiral CT in which, the patient table is passed through a scanner and x-ray source and detectors spirally trace the patient. This produces multi-slice CT images of larger volumes in shorter scan times compared with single-slice CT scanners.

There may be features in the reconstructed image that are not part of the original image. These are known as artifacts. Some common types of artifacts are;

(a) Physics based artifacts such as streaking and aliasing that occur due to inconsistencies or under-sampling of the signal. The streak artifact shown in figure 5.3 occurs due to a

phenomenon known as beam hardening which results from differential photon absorption near dense regions of the body such as bones.

(b) Patient-based artifacts such as metal artifacts and motion artifacts: Metal artifacts occur if patients wear metallic objects such as jewelry or have metallic implants inside their body. Motion artifacts occur due to patient movement during scanning.

(c) Scanner-based artifacts such as ring artifacts occur due to malfunctioning of the scanner itself. More details about artifacts on CT images can be found in [5].

For this research, we analyze images which have only slight to moderate artifacts based on a visual inspection of the images. Datasets with severe artifacts such as the one shown in figure 5.3 have not been analyzed here. Datasets with slight to moderate artifacts were filtered before performing segmentation using an averaging filter that preserves the local mean values in the image.

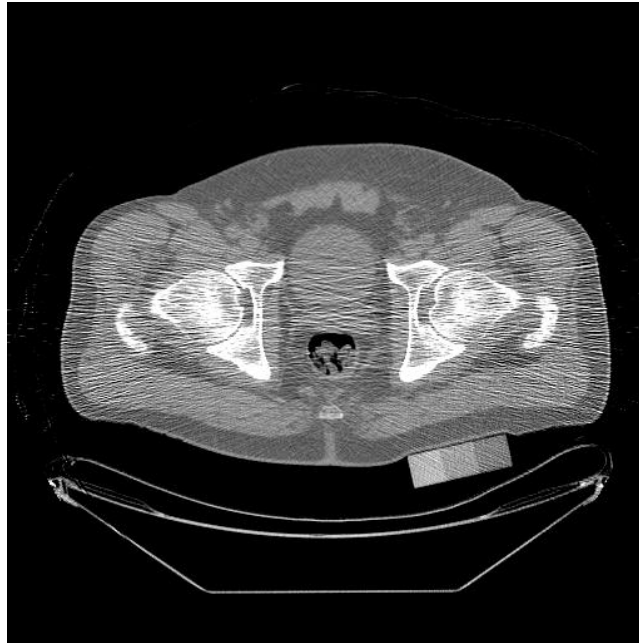


Figure 5.3 Streak artifact present in a pelvic CT image which is caused by beam hardening near bones.

B. Magnetic Resonance Imaging (MRI)

MRI produces excellent soft-tissue contrast and therefore is very useful for diagnosing and treating prostate diseases. MRI is a non-invasive imaging technique which uses magnetic properties of the hydrogen atomic nuclei along with an external magnetic field and radio frequency (RF) pulses to generate detailed images of the body. It is based on the fact that hydrogen atoms are naturally abundant in the human body as part of water molecules and tissues, and have a high magnetic moment. When an external magnetic field (\mathbf{B}_0) is applied to parts of the human body, the hydrogen atomic nuclei align themselves (parallel or anti-parallel) with the external magnetic field. However slightly more number of protons align parallel (low-energy state) than anti-parallel (high energy state) resulting in a net magnetization along \mathbf{B}_0 [31] (shown in figure 5.4).

When an external RF field (\mathbf{B}_1) is applied orthogonal to \mathbf{B}_0 , it has the effect of flipping the net magnetization around the direction of \mathbf{B}_1 (towards the x - y plane) shown in figure 5.4. When the RF pulse is turned off, the net magnetization decays to its original state. The signal generated by the application of specific pulse sequences creates two kinds of images, T_1 and T_2 weighted images. T_1 is the rate constant (decay constant) for the exponential process of net magnetization going back to its thermodynamic equilibrium by losing energy packets to the lattice or its environment, a process called longitudinal relaxation or spin-lattice relaxation. T_2 is the rate constant for the exponential process of net magnetization de-phasing or losing coherence because of exchange of energy packet amongst neighboring nuclei, a process called transverse relaxation or spin-spin relaxation. These two types of relaxation processes occur simultaneously and result in reduction in MRI signal amplitude, but they are sensitive to

the frequency and amplitudes of molecular motions. As a result different parts of the body have different relaxation times and therefore they appear differently on MRI images, resulting in good soft-tissue contrast. Figure 5.5 shows T_1 (left) and T_2 (right) weighted pelvic MRI images of the same patient.

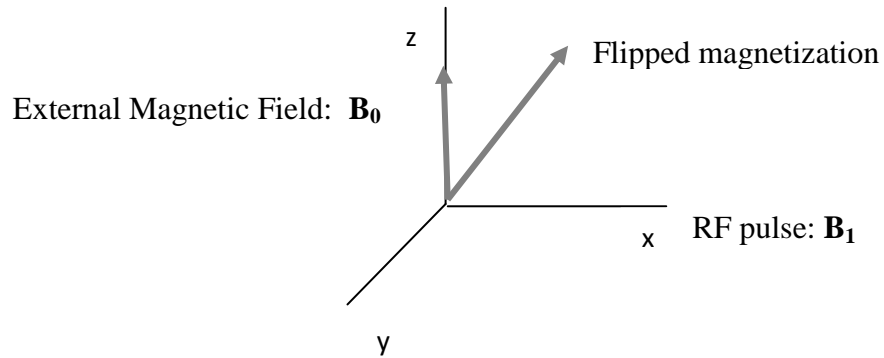


Figure 5.4 Flipping of net magnetization towards the direction of the applied RF field.

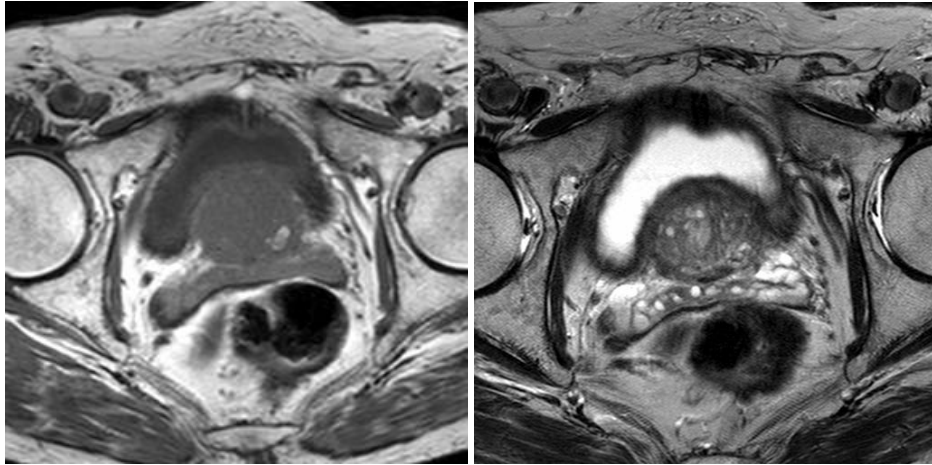


Figure 5.5 (Left panel) T_1 weighted image and T_2 weighted image (right panel) acquired from the same patient.

Some types of artifacts that can occur in MRI images are: RF noise caused due to failure in RF shielding; Motion artifacts caused by movement of the patient; Flow artifacts caused by flow of fluids in the body, etc. Figure 5.6 shows a noisy pelvic MRI image. Again, images with very low signal to noise ratio have been removed from this study because adequate textural features cannot be derived from them.



Figure 5.6 A noisy pelvic MRI image.

5.1.2 Margins in prostate cancer radiation-therapy treatment planning

External beam radiotherapy (EBRT) is a well established standard treatment of prostate cancer. Some common techniques are three-dimensional conformal radiotherapy (3D-CRT) and intensity modulated radiation therapy (IMRT). 3D-CRT has become a standard procedure for treating localized prostate cancer. In this technique, the size, shape and direction of radiation beams are designed to conform to the shape of the target

volume. IMRT is a more complex technique used for cases which require higher dose conformity. It is generally used to improve tumor control and reduce radiotherapy-associated toxicity.

Before EBRT can be performed, tumor margins must be accurately defined to reduce the irradiated volumes of the organs at risk (bladder, rectum) and reduce toxicity. Image-guided radiotherapy (IGRT) using CT and MRI images allow margin reduction by prostate localization before each radiotherapy fraction. The Gross Tumor Volume (GTV) is the region of malignant growth visible on diagnostic tests. The Clinical Tumor Volume (CTV) is defined as the GTV along with a margin that accounts for the uncertainties in defining the GTV. This CTV is usually outlined manually by the oncologist so that the GTV is adequately treated. Another margin known as the Planning Target Volume (PTV) is defined around the CTV and used to select beam sizes and arrangements so that the prescribed dose is absorbed by the CTV [54].

5.2 Description of data

To perform this analysis, images were obtained from a database of 2700 pelvic CT and MRI scans, acquired through collaboration with Oregon Health & Science University (OHSU). CT and MRI images of 10 patients (for each modality) from this database were manually segmented by Dr. Arthur Hung and Dr. James Tanyi, (Dept. of Radiation Medicine, OHSU). The CT and MRI images were acquired using 16-slice big-bore configuration Computed Tomography simulator (Brilliance 190P; Philips Medical Systems, Cleveland, OH, USA) and 3-Tesla whole-body Magnetic Resonance Imaging instrument (Trio; Phillips Medical systems, Malvern, PA, USA) respectively.

Each scan for a patient contains ~15 slices of 2D images. Some patients have multiple CT/MRI scans. The prostate is visible in about 10-12 of these slices; the rest display other organs in the pelvic region such as the bladder and the rectum. The number of slices on which the prostate is visible depends on the resolution of the scans which varies from one patient to another, and is also dependent on the technique used for image acquisition. The prostate has been manually delineated twice on the same set of images by the experts. This provides a database for intra-operator variability. The manually segmented contours derived from the scans of five patients ($\sim 50 \times 2 = 100$ images) have been used as the training data for this analysis. The images from the other five patients (for which the ground truth was available) were used as test images. Note that, the CT and MRI images shown here were taken from different patients.

5.3 Segmentation of pelvic CT/MRI images

Each test image was segmented using all the four methods: Chan & Vese method; Laws' texture measures; the Gabor wavelet transform method; and the LSGA. Segmentation using LSGA was carried out both in two and three dimensions. The preprocessing step and the experiments that were conducted for 2D and 3D segmentation by the LSGA are described below.

As a pre-processing step, the CT images were brightened to improve the visibility of soft-tissue regions using the *imadjust* function in MATLAB. They were then filtered using an averaging filter to smooth out streak artifacts (figure 5.7). Since the MRI images already had good contrast, contrast-enhancement was not performed on them. However some of the images were filtered using an averaging filter to smooth out speckle noise.

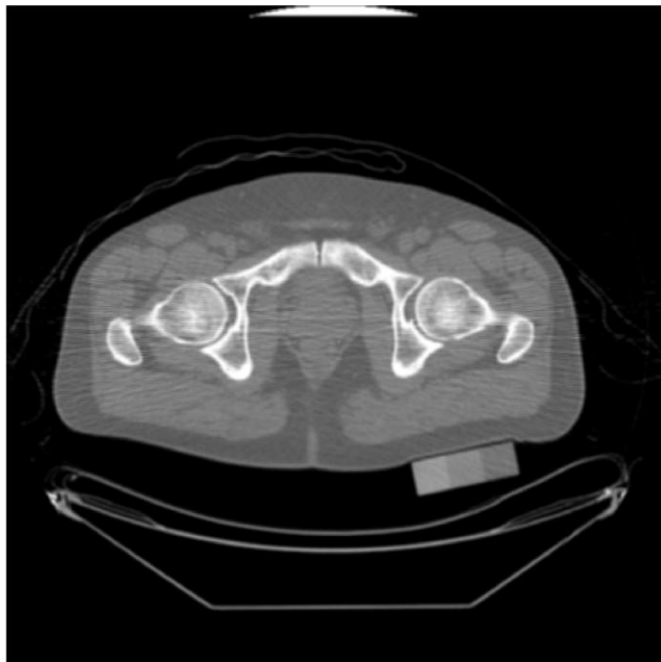


Figure 5.7 A CT image before (top) and after (bottom) the application of an averaging filter to reduce streaking.

5.3.1 Segmentation using texture and shape priors

At first, the LSGA was implemented using only the texture and shape priors for segmenting 2D and 3D CT and MRI images. The steps performed for 2D segmentation using LSGA are listed below.

A. 2D segmentation

1. Shape from training images: The mean shape and shape variability of the prostate was derived using PCA on the training images. Before computing PCA, the 2D manually segmented contours from all the five patients (training images) were aligned with one 2D training contour using pose parameters. Figure 5.8 shows the mean shape and shape variability derived from the 2D training contours.

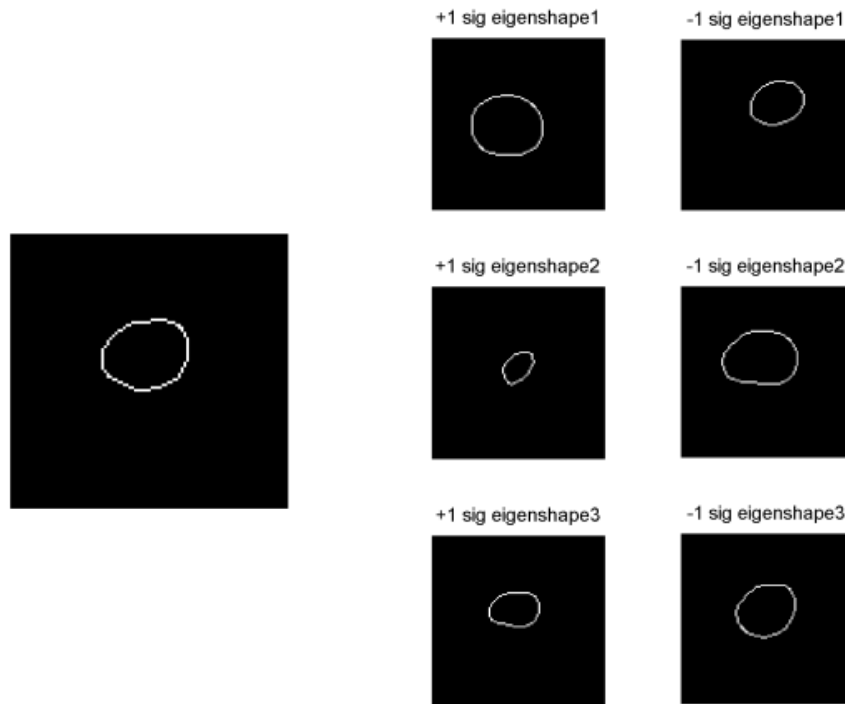


Figure 5.8. Mean shape (left panel) and shape variability (right panel) of the prostate derived from training images.

2. Texture from training images: The texture was derived from training images using two-class classification (prostate vs. background). Laws' texture measures and the Gabor wavelet transform method were used for textural classification on the training images. It was found that Laws' texture measure produced better two-class textural classification for both the CT and MRI images (figures 5.9 and 5.10), therefore it was chosen to be incorporated into the LSGA for segmenting the CT and MRI images.

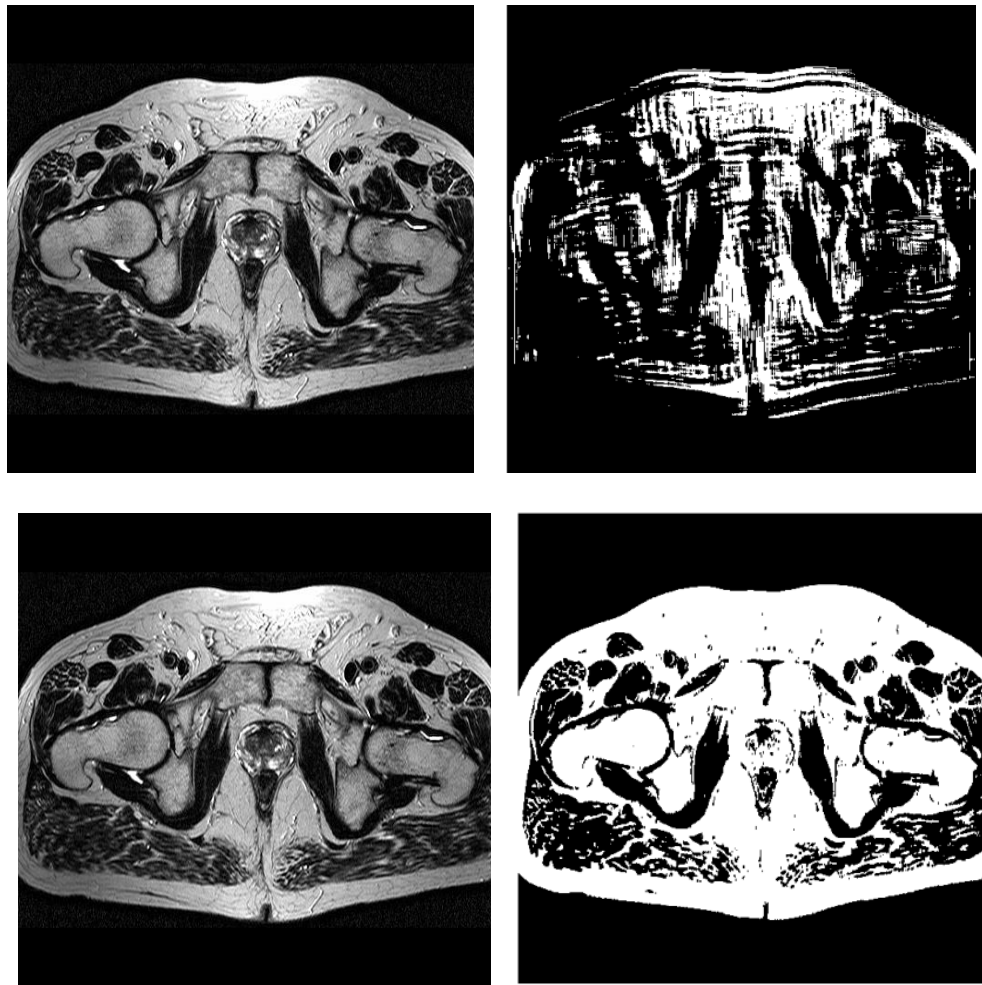


Figure 5.9 An MRI image (original image shown in the left panels); (Top right) Texture segmentation using Laws' texture measures; (Bottom right) texture segmentation using Gabor wavelet transform method.

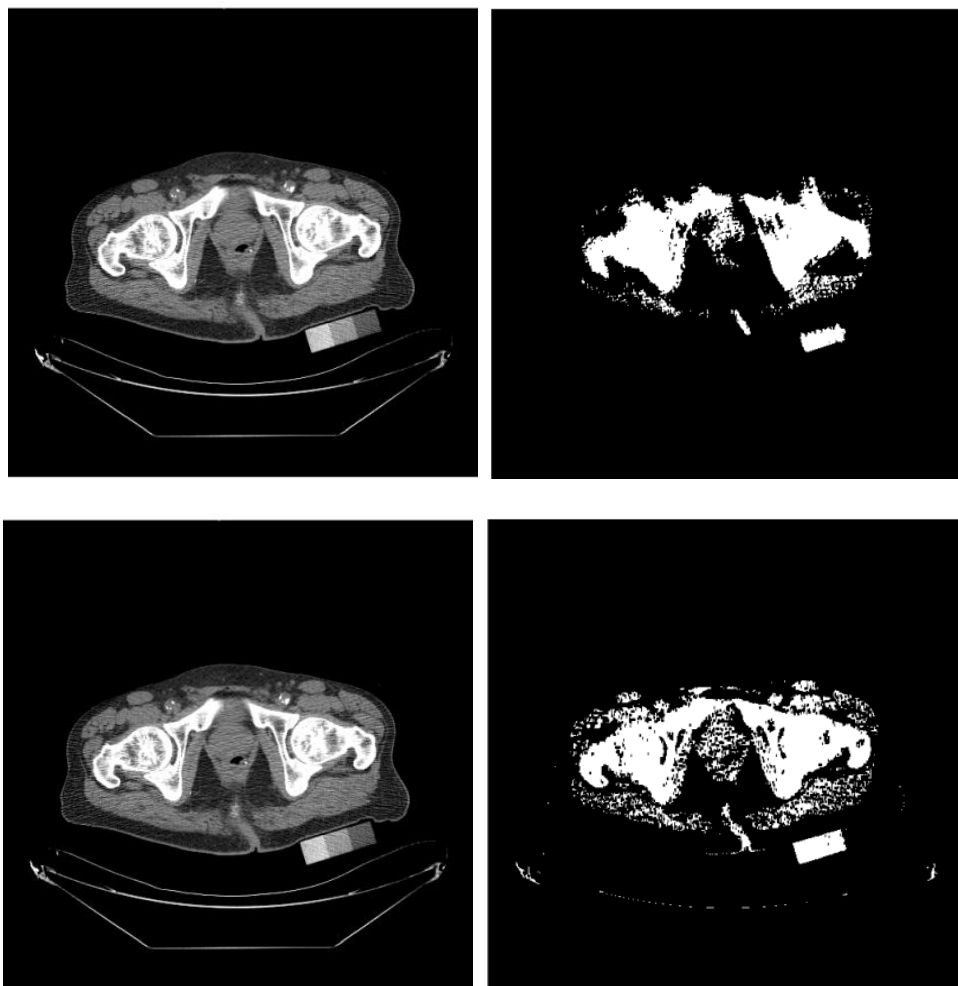


Figure 5.10 A CT image (original image shown in the left panels); (Top right) Texture segmentation using Laws' texture measures; (Bottom right) texture segmentation using Gabor wavelet transform method.

3. Initial GA population: The initial population of the GA was generated using values chosen randomly from the range of parameter values specified in table 5.1. The parameter values were substituted into the level set equation 3.6, to generate segmenting contours as a sum of the mean shape and shape variability of the prostate.

Table 5.1 LSGA parameters for 2D/2DRL segmentation

Population Size	50
Mutation Rate	10 % per gene
Crossover Rate	50% single-point
Selection Criteria	Rank Selection
Weights for eigen shapes, w	$(0- \pm 2)\text{integers} * \sigma_i$
Translation parameters a, b	Integer (0-30)
Rotation parameter θ	-30° to $+30^\circ$
Scale parameter h	(0.5-2)

4. Fitness: The fitness of each individual was computed by comparing the degree of overlap between the segmenting contour and the textural classification on a test image. Here, textural priors generated by Laws' texture measures were used since it narrowed the region of interest more than the Gabor wavelet transform method. The GA used the textural classification map (around the mean location of the prostate) on a test image, to place the final segmenting contour.
5. GA evolution: The GA was iterated by performing selection, cross-over and mutation over 30 generations or until the threshold exceeded the threshold value of 900 (specified by the user).

B. 3D segmentation

The segmentation was performed in three dimensions by the LSGA by simultaneously segmenting all the slices of a test image at once using a segmentation surface. The parameters used by the 3D LSGA are shown in table 5.2. The texture and shape priors derived from the training images were used for segmentation. The mean shape and shape variability were derived in 3D. The same textural segmentation parameters derived for 2D segmentation were used here. Figure 5.11 shows the mean shape and shape variability of the segmenting surface derived using 3D PCA.

Table 5.2 LSGA parameters for 3D/3DRL segmentation

Population Size	25
Mutation Rate	10 % per gene
Crossover Rate	50% single-point
Selection Criteria	Rank Selection
Weights for eigen shapes, w	$(0 - \pm 2)$ integers * σ_i
Translation parameters a, b, c	Integer (0-20)
Rotation parameters α, β, θ	-30° to $+30^\circ$
Scale parameter h	(0.5-2)

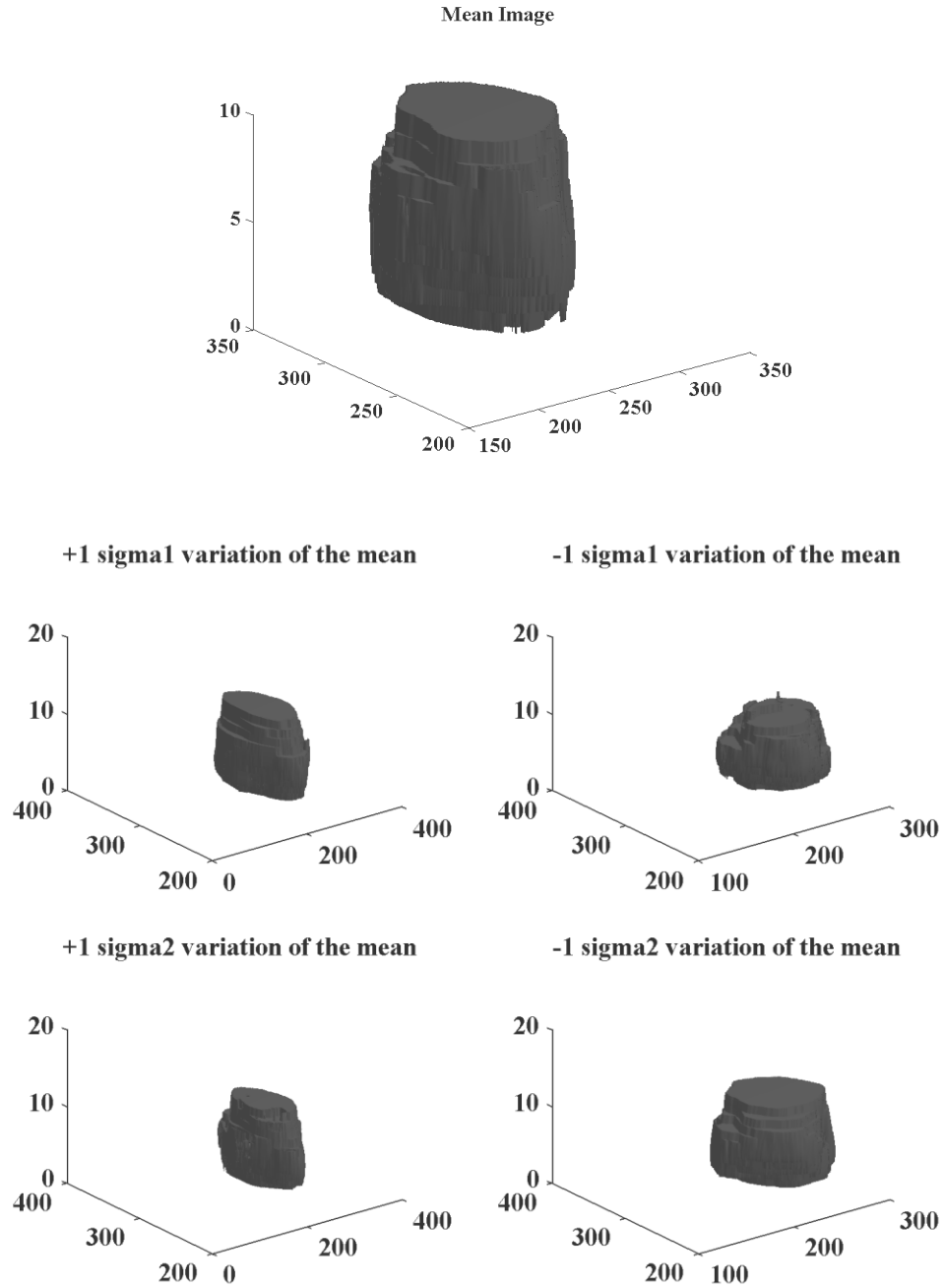


Figure 5.11 Mean shape (top) and shape variability (bottom) of the prostate derived from training images.

5.3.2 Segmentation using texture, shape, and relative location priors.

Next, the LSGA was implemented using texture, shape, and relative location priors for segmenting 2D and 3D CT and MRI images. The steps performed for 2D segmentation using relative location prior (2DRL) by the LSGA are listed below.

A. 2DRL segmentation

1. Shape from training images: Since the training images for the previous as well as this experiment were the same, the mean shape and shape variability derived in the previous experiment were used here for shape representation using the level set function of equation 3.6.
2. Texture from training images: Improved texture segmentation was implemented in this experiment by incorporating multi-class texture segmentation of the following classes: prostate, bladder, rectum and background. The bladder and rectum were manually segmented in all the training images. This step was incorporated to narrow down the textural search space for the LSGA. Figure 5.12 shows multiclass texture segmentation on a CT and MRI image using the Gabor wavelet transform method.
3. Fuzzy relative location map from training images: The fuzzy relative location was derived from the training images using the method described in section 3.3. Here, the bladder and rectum were used as reference objects, and the prostate was the target object. The fuzzy landscape derived from all training images and using both the reference objects is shown in figure 5.13.

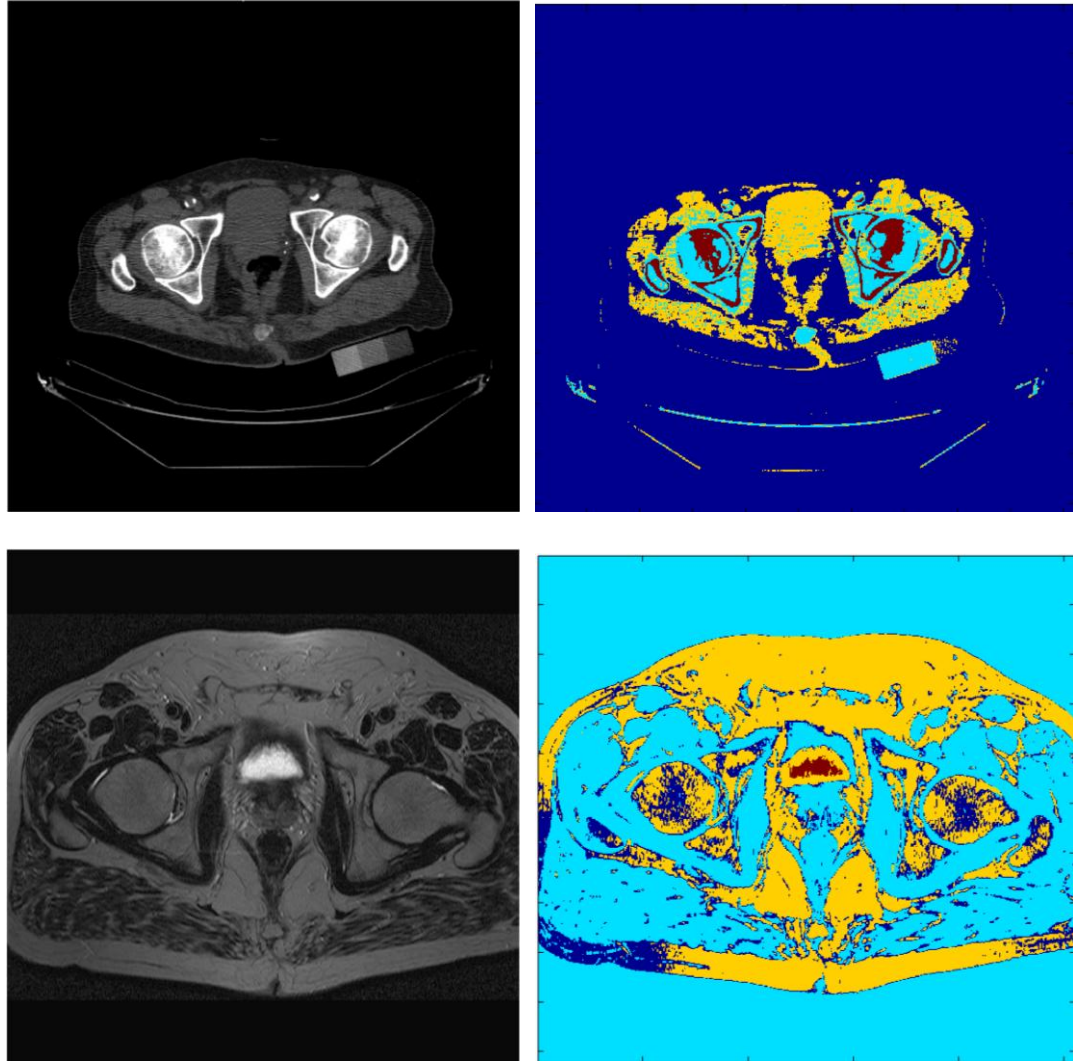


Figure 5.12 Multiclass texture segmentation of an MRI image using Gabor wavelet transform method on a CT image (top) and MRI image (bottom): Prostate (yellow), Bladder (red), rectum (green), background (dark blue) regions.

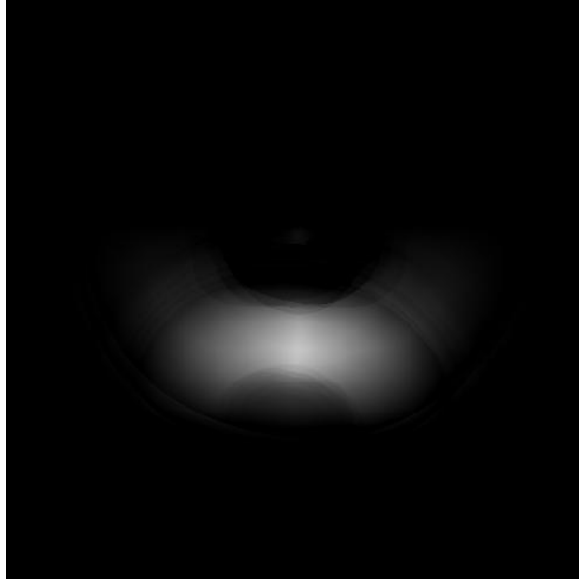


Figure 5.13 The fuzzy landscape derived from all training images using the bladder and rectum as the reference objects.

4. Initial GA population: The initial population of the GA was generated in the same manner as 5.3.1, by using values chosen randomly from the range of parameter values specified in table 5.1.
5. Fitness: The fitness of each individual was computed using the method described in section 3.4.2. Here, the multi-class textural priors generated by Gabor wavelet transform were used. The GA used the combination of the textural classification map on a test image, and the fuzzy landscape to place the final the segmenting contour on the test image.
6. GA evolution: The GA was iterated by performing selection, crossover and mutation over 30 generations or until the threshold exceeded the threshold value of 900 (specified by the user).

B. 3DRL segmentation

Three dimensional segmentation using relative location prior (3DRL) segmentation was performed by the LSGA by simultaneously segmenting all the slices of a test image at once using a segmentation surface (a 3DRL LSGA individual). Here, texture, shape, and relative location priors were derived from the training images were used for segmentation. The mean shape and shape variability derived in 3D from section 5.3.1 were used here. The multi-class textural segmentation parameters derived for 2D segmentation were used by the 3DRL LSGA.

5.4 Evaluation criteria

5.4.1 Repeatability

The segmentation performance of various segmentation algorithms has been evaluated using a measure derived from Udupa et al. [103] known as *repeatability*. The *repeatability* of a given segmentation method is determined by applying the same algorithm multiple times on the same image and then comparing the independent binary outcomes. The (\cap) operator signifies the region common to two binary outcomes. The (\cup) operator signifies the union of the two binary outcomes

$$Repeatability(R) = \frac{\text{outcome}_1 \cap \text{outcome}_2}{\text{outcome}_1 \cup \text{outcome}_2}. \quad (5.2)$$

5.4.2 Dice similarity coefficient and partial Hausdorff distance

Since the Dice Similarity Coefficient measures the degree of overlap between two regions, the precision of each segmentation method was evaluated by comparing the binary segmentation outcomes (**B**) with the ground truth (**GT**).

To make a fair comparison of the various modes of LSGA, the segmentation was performed in each mode using multi-class texture segmentation with the GWT technique. The fuzzy map was used for 2DRL-LSGA (two-dimensional segmentation using the relative location map) and 3DRL-LSGA (three-dimensional segmentation using the relative location map). The final segmentation outcomes for each method was then compared with the manually segmented ground truth contours using the *DSC* and *H* measures defined in section 4.3.

5.5 Results and discussion

5.5.1 Comparison of segmentation algorithms

As an initial test, all the test images were segmented using the four different methods. At first, the level set based segmentation algorithm of Chan & Vese was used to segment the test images. The initial contour was placed in the center of each test image (left panels of figure 5.14). The upper right panel on figure 5.14 shows the outcome of the algorithm for a CT image. Figure 5.14 (lower right panel) shows the result of the algorithm on a MRI image. In both the cases, the algorithm found boundaries between the regions with markedly different pixel intensity values. The obtained result was expected because the algorithm is designed to find regions with different pixel intensity values inside and outside the contour. The result is the original image superimposed by the

evolved curve. The original curve divides into many curves surrounding several regions; therefore, it is not possible to obtain a binary image from the final outcome. Since, the evaluation of the algorithm performance requires a binary image; the evaluation was performed by stopping the Chan & Vese algorithm after five iterations and deriving the binary outcome from the diverging segmenting contour at the end of the fifth iteration.

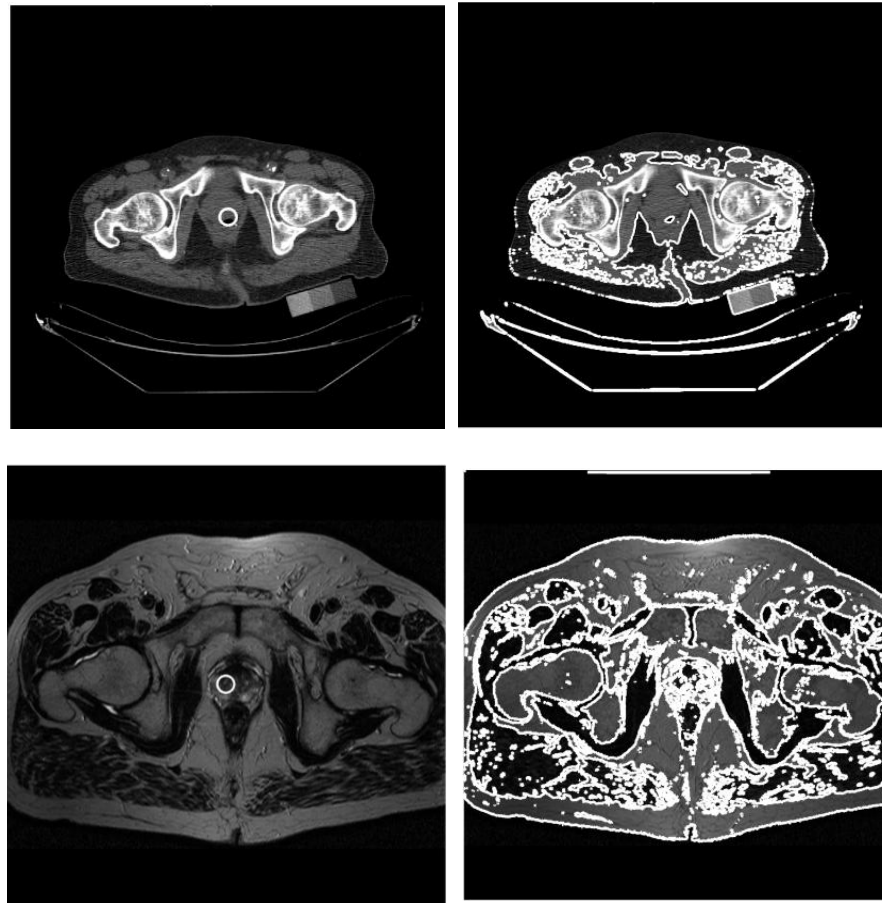


Figure 5.14 Upper left panel: Initial contour placed on top of a test CT image. Upper right panel: Outcome of the Chan & Vese algorithm. Lower left panel: Initial contour placed on top of a test MR image. Lower right panel: Outcome of the Chan & Vese algorithm.

The test images were also segmented using Laws' texture measures and the Gabor wavelet transform-based segmentation algorithm. Figures 5.9, 5.10, and 5.12 already show examples of two-class and multi-class segmentation using the two methods, therefore, they have not been shown here again. For evaluation, only two-class segmentation outcomes from both the methods have been used. Tables 5.4 and 5.5 show the evaluation of segmentation outcomes using *repeatability* and *DSC* measures for the two-class textural classification.

As an initial test, the LSGA with texture and shape priors alone was used for segmenting CT and MRI images in two as well as three dimensions using the parameters mentioned in section 5.3. In this initial analysis, the two-class Laws' texture classification was performed on the test images as mentioned in section 5.3.1. For all the five test images (for each modality), the LSGA was iterated for 30 (10) generations for 2D (3D) segmentation because the maximum fitness threshold was never attained. The sample header file from a 2D segmentation on a test MR image is shown in table 5.3. The fitness values attained by the maximum fit individuals in this LSGA are low (maximum fitness that can be attained is 1000) because it was derived from the texture segmentation of the entire image and not just the prostate region. Figure 5.15 shows the plot of average fitness versus the number of generations. The fitness values increase with each generation because the GA evolution is based on the principle of the survival of the fittest. Figure 5.16 (left panel) depicts a sample 2D segmentation on a CT image. Figure 5.16 right panel shows the all the 2D segmenting contours for all slices of the CT image, stacked on top of each other to create a 3D shape. The segmentation contours are not aligned with respect to each other because there was no information exchange across slices for

performing segmentation. The segmentation result on a CT image with 3D LSGA is shown in figure 5.17 (right panel). Figure 5.17 (left panel) shows a slice from the 3D segmentation result obtained on a test CT image. Figure 5.18 (left panel) shows a slice from the 3D segmentation result obtained on a test MR image. The segmentation result on a MR image with 3D LSGA is shown in figure 5.18 (right panel). Note that the final segmenting contours are much better aligned with respect to each other in case of 3D segmentation than 3D segmentation. This occurs because the 3D LSGA segments all slices of the test image at once. So, the segmenting contours remain aligned with each other. These 3D structures appear hollow because they only consist of contour boundaries. The detailed segmentation results using each of the LSGA modes 2D/2DRL/3D/3DRL are described in the next section.

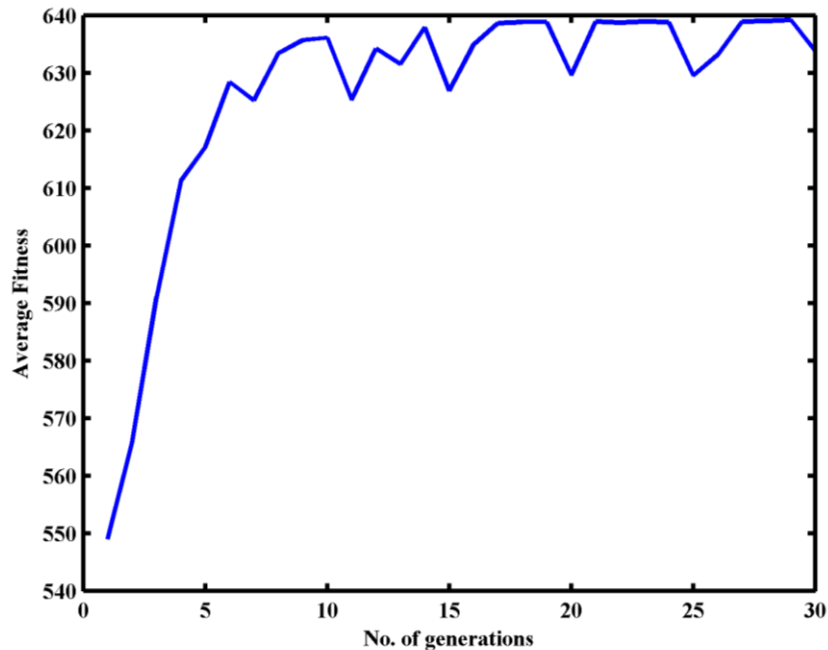


Figure 5.15 Plot of average fitness versus the number of generations of the sample run of the 2D-LSGA.

Table 5.3 Header file created by the LSGA for a sample MRI test image

Generation	Avg_Fitness	Std_Dev	Best_Fitness
1	549.0	37.0	636.7
2	565.6	49.0	636.7
3	591.0	38.9	636.7
4	611.2	25.1	636.7
5	617.1	34.0	638.1
6	628.4	7.1	638.1
7	625.2	34.6	638.1
8	633.4	8.4	638.1
9	635.7	3.2	638.3
10	636.1	2.8	638.8
11	625.3	46.0	638.8
12	634.2	19.4	638.8
13	631.5	34.7	638.8
14	637.9	1.9	638.8
15	626.9	41.7	638.9
16	634.9	19.7	638.9
17	638.6	0.6	638.9
18	638.8	0.3	638.9
19	638.8	0.1	639.2
20	629.6	36.0	639.2
21	638.9	0.3	639.2
22	638.7	1.2	639.2
23	638.9	0.3	639.2
24	638.8	1.2	639.2
25	629.5	35.5	639.2
26	633.1	32.8	639.2
27	638.9	1.3	639.2
28	639.0	0.8	639.2
29	639.2	0.0	639.2
30	633.8	29.0	639.2

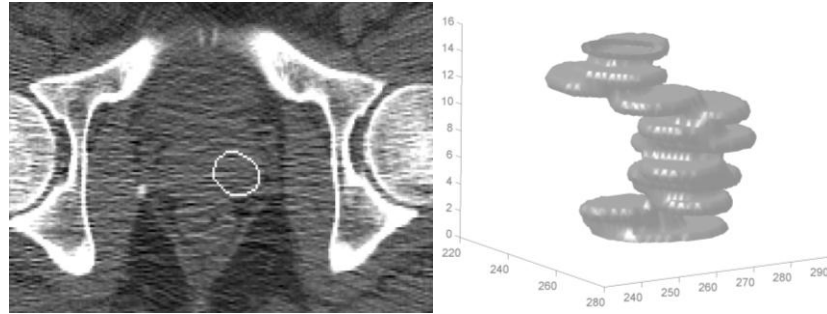


Figure 5.16 (Left panel) Segmentation result in 2D for a test CT image. (Right panel) Final 2D segmenting contours stacked on top of each other to create a 3D shape.

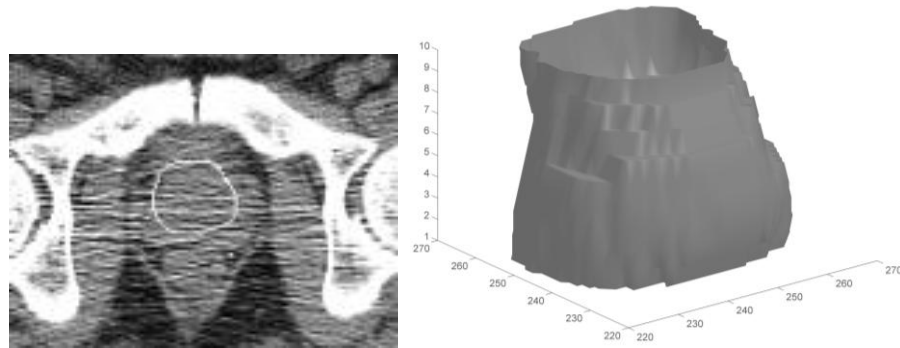


Figure 5.17 3D segmentation result of the LSGA on a test CT image (right panel). A slice of the 3D segmentation generated by the GA (left panel).

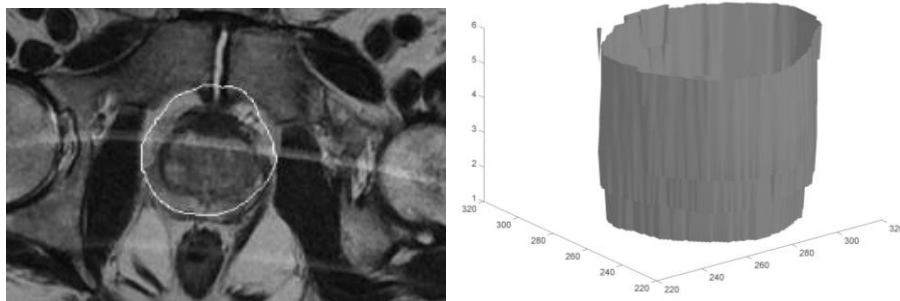


Figure 5.18 Right panel: 3D segmentation result of the LSGA on a test MR image. Left panel: A slice of the 3D segmentation generated by the GA. There are only 5 slices in this figure as compared to 10 in the previous figure, therefore the 3D structures look different.

Tables 5.4 and 5.5 show the performance evaluation by comparing the binary results obtained from the four different methods: 3D-LSGA, LTM, GWT, and CV averaged over all test images. As mentioned before, the final outcome of the Chan & Vese algorithm is not a binary image; therefore the algorithm has been evaluated by using the diverging segmenting contour generated at the fifth iteration of the algorithm. Also, the texture segmentation algorithms have been evaluated for performance based on two-class classification of the prostate versus the background. The *repeatability* (R) measure was derived from all the test images. The segmentation result of the 3D-LSGA has been used for evaluation purposes.

Table 5.4 Performance evaluation of the four protocols for segmenting the prostate on pelvic CT images.

Protocol	R	DSC
3D-LSGA	0.42 ± 0.01	0.67 ± 0.04
LTM	1 ± 0	0.07 ± 0.04
GW	1 ± 0	0.05 ± 0.03
CV	1 ± 0	0.21 ± 0.1

Table 5.5 Performance evaluation of the four protocols for segmenting the prostate on pelvic MRI images.

Protocol	R	DSC
3D-LSGA	0.4 ± 0.01	0.55 ± 0.3
LTM	1 ± 0	0.04 ± 0.02
GW	1 ± 0	0.05 ± 0.02
CV	1 ± 0	0.23 ± 0.06

The tables clearly show that the repeatability of the LSGA (incorporating texture and shape priors) is very poor. Genetic algorithms are stochastic optimization methods and

find satisfactory solutions close to the ground truth. These solutions may not be the same for every run of the GA. In contrast, the LTM, GWT, and CV methods are deterministic and produce the same segmentation outcomes every time.

The LSGA searches around the mean location of the prostate using the textural segmentation to find an approximate location of the prostate in the pelvic images. The *DSC* values for 3D-LSGA are much better than the other methods because the LSGA combines shape and texture priors for segmentation. In contrast the LTM and GWT methods only use texture features and the CV method uses an evolving contour for segmentation.

5.5.2 Comparison of LSGA in the following modes 2D/2DRL/3D/3DRL

The detailed comparison of segmenting the CT and MRI images using the LSGA in the four different modes is shown here for CT and MRI images separately. Segmentation outcome for each modality has been shown for one patient (referred to as “patient A” for CT and “patient B” for MRI) here. Table 5.10 shows the *DSC* and *H* values averaged over all the five test subjects.

A. Segmentation of CT images

Figure 5.19 shows the segmentation outcome of 2D-LSGA for each slice of a CT image of patient A (white contour). The gray contour on the same image depicts the manual segmentation performed on the same image by a radiation oncologist. The 3D structure created by stacking all the segmenting contours on top of each other is shown in

figure 5.20. Table 5.6 shows a sample header file generated from the 2D-LSGA segmentation of a single slice of a CT image of patient A.

Table 5.6 The header file generated by the 2D-LSGA for a single slice of CT image of patient A.

Generation	Avg_Fitness	Std_Dev	Best_Fitness
1	529.1	18.8	585.4
2	541.4	20.9	588.2
3	556.2	20.8	588.2
4	560.1	26.1	597.0
5	567.9	25.0	599.2
6	579.0	21.3	599.2
7	579.5	27.1	599.2
8	587.1	18.4	602.0
9	591.1	10.1	600.0
10	592.2	12.9	600.0
11	586.3	24.3	600.0
12	592.9	17.0	601.8
13	591.4	20.3	602.0
14	597.6	4.7	603.1
15	594.6	16.5	604.8
16	595.7	14.6	604.8
17	595.8	18.8	605.0
18	596.1	18.3	605.0
19	596.7	19.7	605.0
20	595.8	22.0	605.6
21	597.0	20.3	606.0
22	592.3	27.2	606.0
23	595.6	23.3	606.0
24	595.0	25.5	606.0
25	601.7	13.7	606.0
26	601.4	17.2	606.2
27	604.5	4.3	606.2
28	601.2	12.5	606.2
29	599.0	18.8	607.5
30	599.7	18.7	607.5

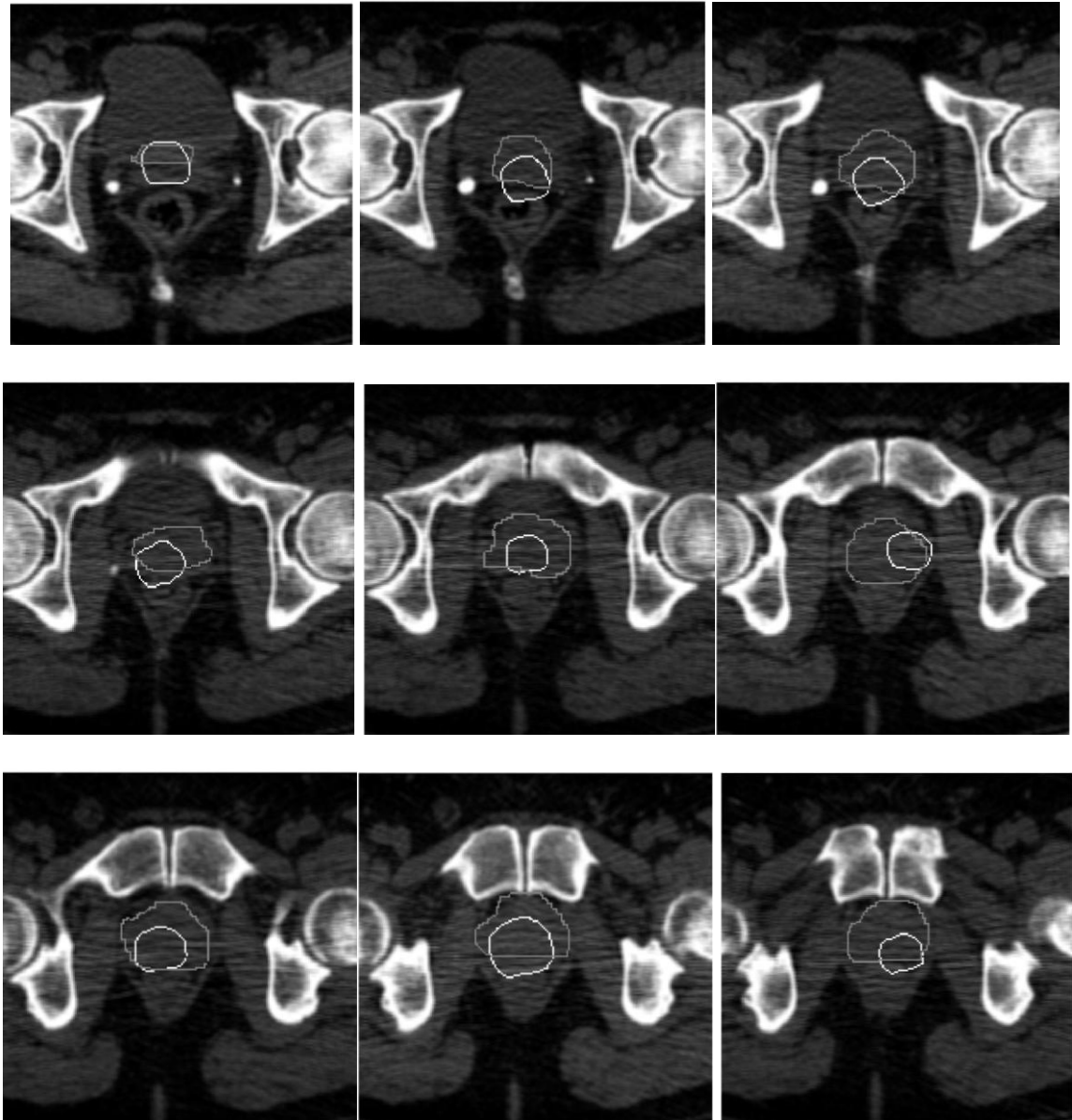


Figure 5.19 Slice-by-slice segmentation of pelvic CT scan of patient *A* using 2D-LSGA.

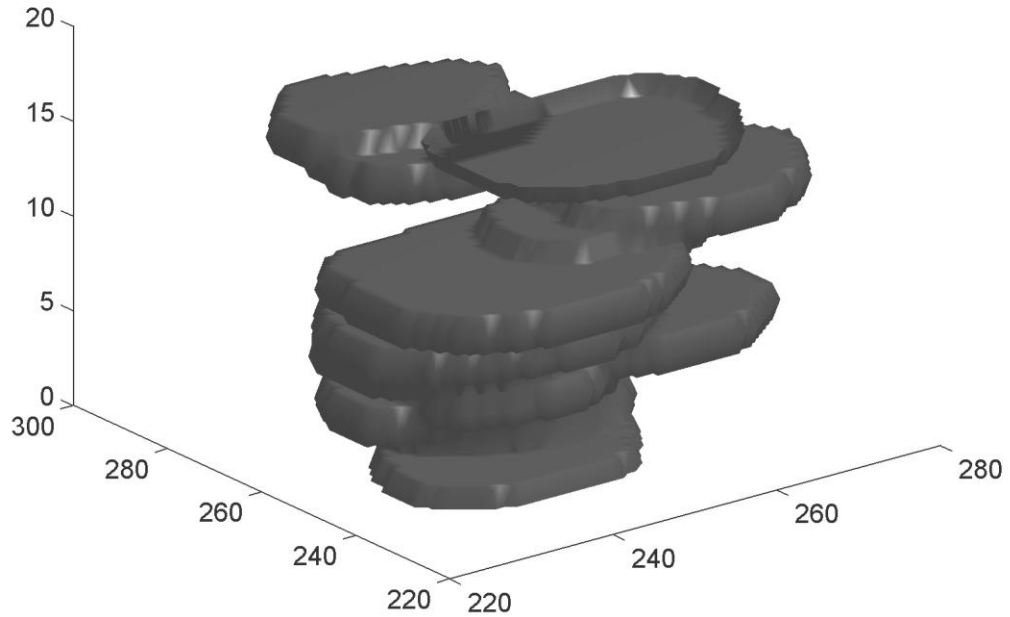


Figure 5.20 Segmenting contours of 2D-LSGA stacked on top of each other to form a 3D shape.

Figure 5.21 shows the segmentation outcome of 2DRL-LSGA for each slice of a CT image of patient A (white contour). The gray contour on the same image depicts the manual segmentation performed on the same image by a radiation oncologist. The 3D structure created by stacking all the segmenting contours on top of each other is shown in figure 5.22. Table 5.7 shows a sample header file generated from the 2D-LSGA segmentation of a single slice of a CT image of patient A.

Table 5.7 The header file generated by the 2DRL-LSGA for a single slice of CT image of patient A.

Generation	Avg_Fitness	Std_Dev	Best_Fitness
1	515.6	6.5	531.2
2	518.4	7.6	535.6
3	520.5	8.5	535.6
4	526.4	8.4	537.5
5	529.4	6.6	537.5
6	533.1	3.6	537.5
7	534.4	4.7	538.1
8	533.5	5.0	538.1
9	535.3	2.2	539.7
10	535.7	2.9	539.7
11	535.9	5.2	549.2
12	535.7	7.0	548.0
13	537.6	6.5	550.1
14	537.5	7.7	550.1
15	539.5	5.5	550.0
16	541.0	7.7	550.0
17	543.9	6.3	550.7
18	545.1	8.3	551.4
19	548.5	3.0	551.4
20	549.7	1.3	551.9
21	548.4	7.0	551.9
22	548.2	6.9	553.3
23	549.8	7.2	553.5
24	548.0	10.7	553.5
25	548.6	9.2	553.5
26	550.1	6.9	553.5
27	548.7	8.9	553.5
28	549.2	7.5	553.5
29	549.0	8.5	554.1
30	549.8	8.6	554.2

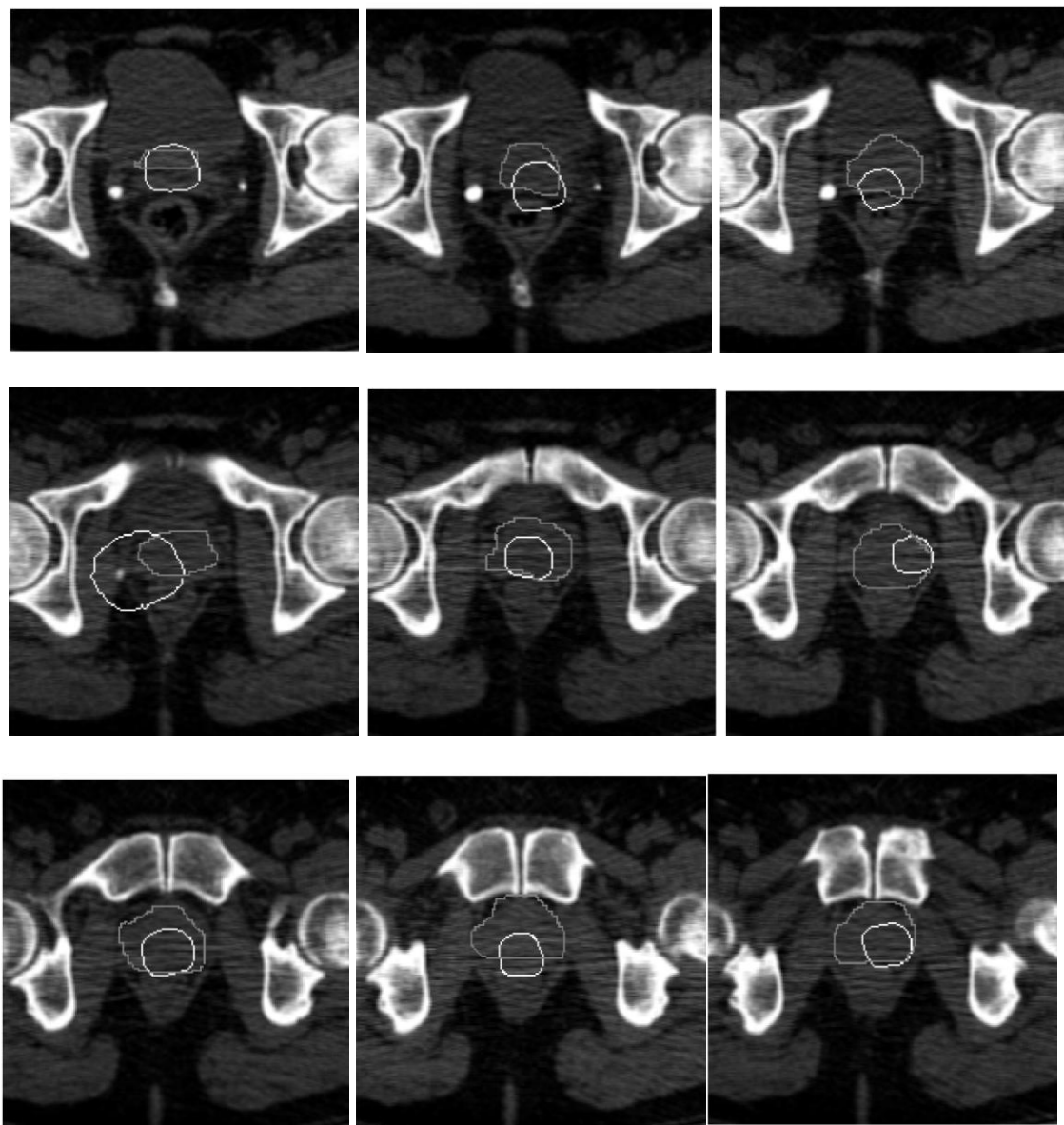


Figure 5.21 Slice-by-slice segmentation of a CT image of patient A using 2DRL-LSGA.

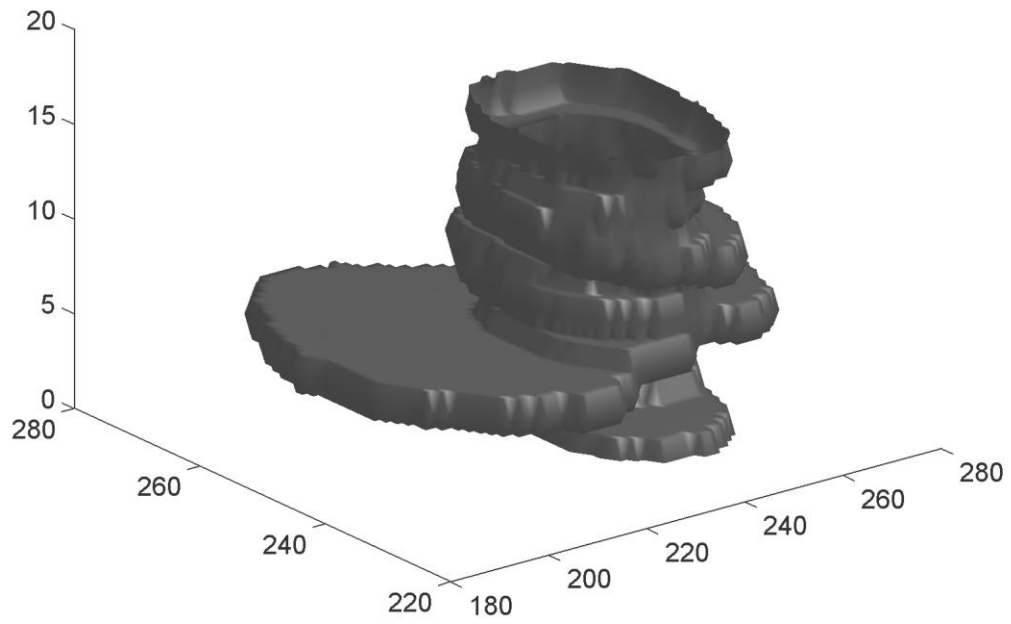


Figure 5.22 Segmenting contours of 2DRL-LSGA stacked together to form a 3D shape.

Figure 5.23 shows the segmentation outcome of 3D-LSGA for each slice of a CT image of patient A (white contour). The gray contour on the same image depicts the manual segmentation performed on the same image by a radiation oncologist. The 3D structure created by stacking all the segmenting contours on top of each other is shown in figure 5.24. Table 5.8 shows a sample header file generated from the 3D-LSGA segmentation for the CT image of patient A. Note that the 3D-LSGA was stopped after 10 generations.

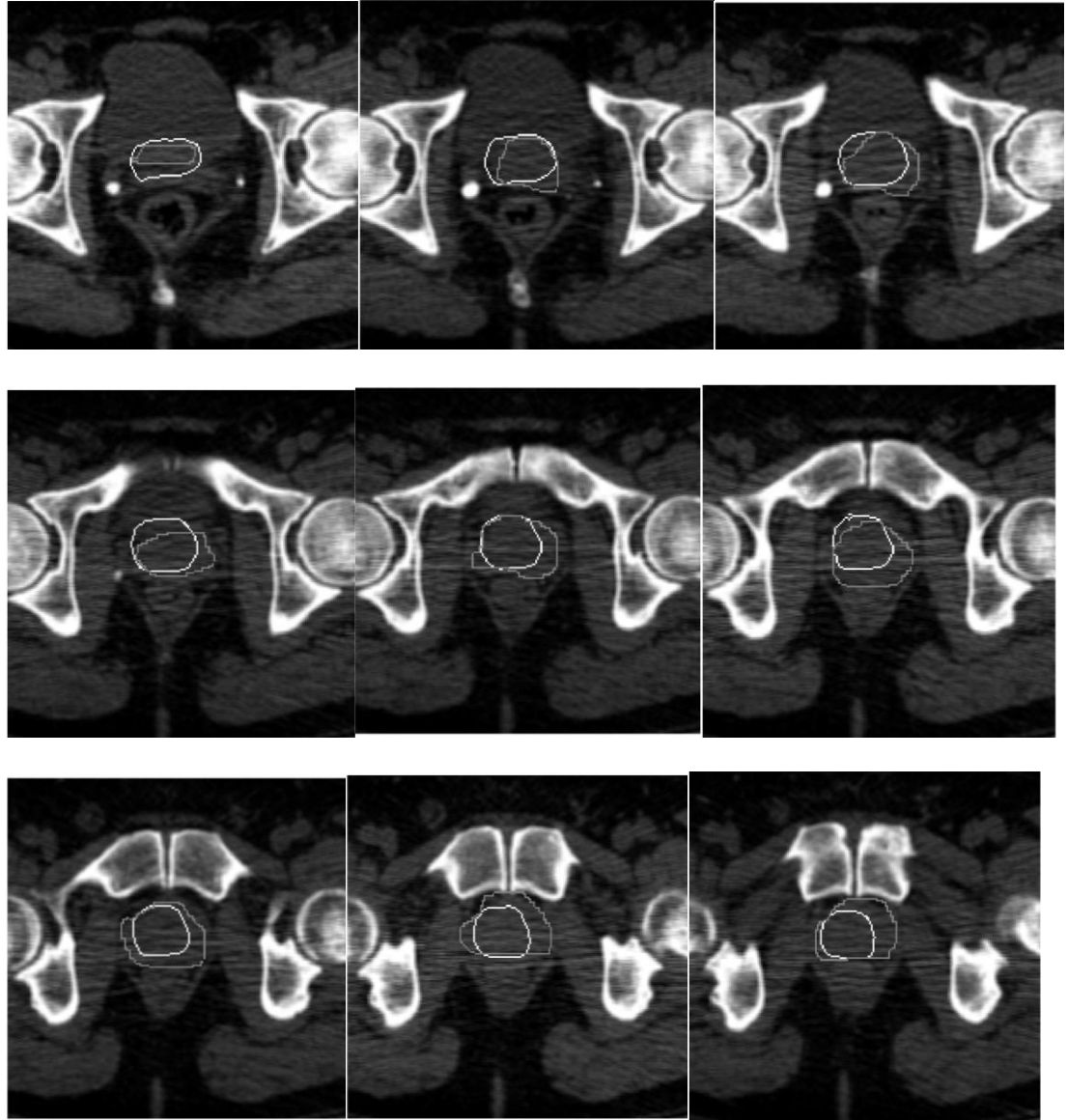


Figure 5.23 Slice-by-slice segmentation of a CT image of patient *A* using 3D-LSGA.

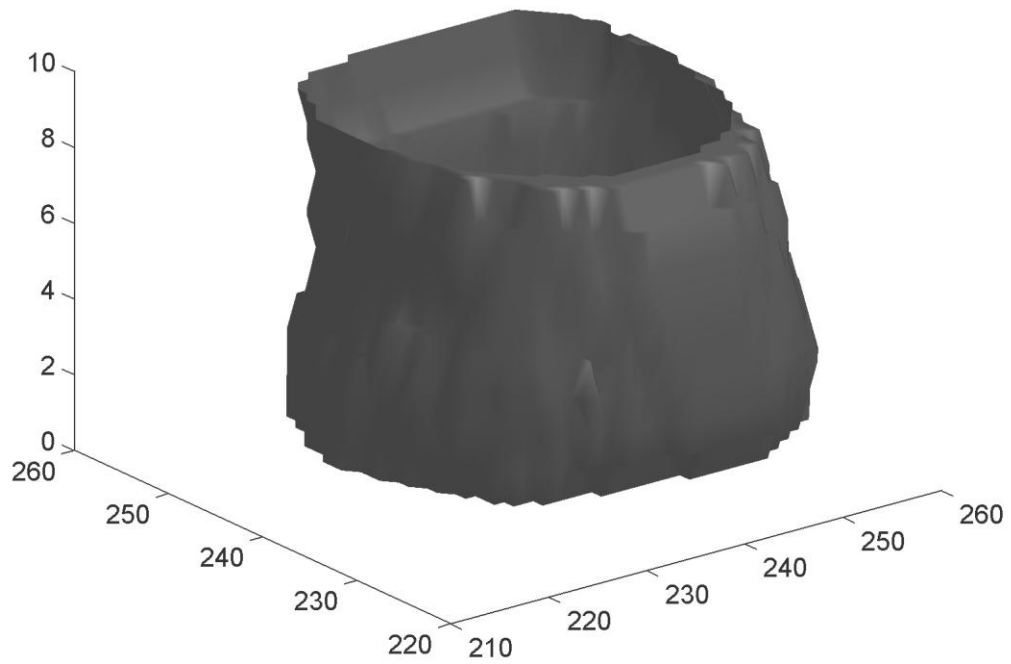


Figure 5.24 The 3D segmenting surface of 3D-LSGA.

Table 5.8 The header file generated by the 3D-LSGA for a CT image of patient A.

Generation	Avg_Fitness	Std_Dev	Best_Fitness
1	506.8	0.4	507.4
2	506.9	0.3	507.4
3	507.1	0.2	507.4
4	507.0	0.5	507.5
5	507.0	0.5	507.5
6	507.3	0.3	507.6
7	507.2	0.2	507.5
8	507.2	0.3	507.6
9	507.3	0.3	507.7
10	507.3	0.3	507.6

Figure 5.25 shows the segmentation outcome of 3DRL-LSGA for each slice of a CT image of patient A (white contour). The gray contour on the same image depicts the manual segmentation performed on the same image by a radiation oncologist. The 3D structure created by stacking all the segmenting contours on top of each other is shown in figure 5.26. Table 5.9 shows a sample header file generated from the 3DRL-LSGA segmentation for the CT image of patient A. Note that the 3DRL-LSGA was stopped after 10 generations.

Table 5.9 The header file generated by the 3DRL-LSGA for a CT image of patient A.

Generation	Avg_Fitness	Std_Dev	Best_Fitness
1	531.2	3.6	536.5
2	534.4	4.2	538.5
3	537.2	2.0	539.2
4	537.8	0.8	539.3
5	537.9	1.2	539.3
6	537.6	2.3	539.3
7	538.6	0.9	539.7
8	538.1	1.7	539.3
9	537.5	1.5	539.3
10	538.2	1.0	540.1

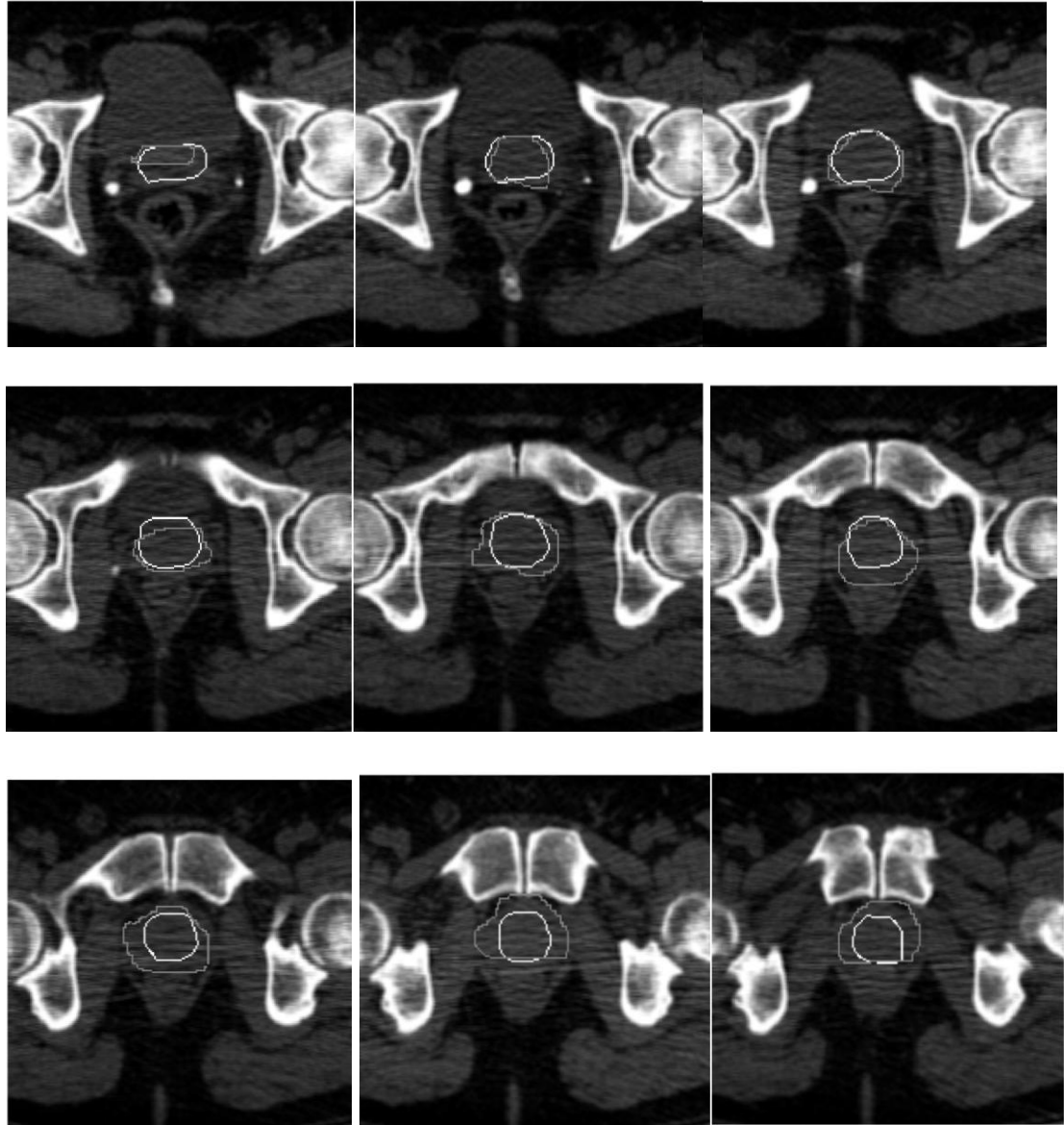


Figure 5.25 Slice-by-slice segmentation of a CT image of patient *A* using 3DRL-LSGA.

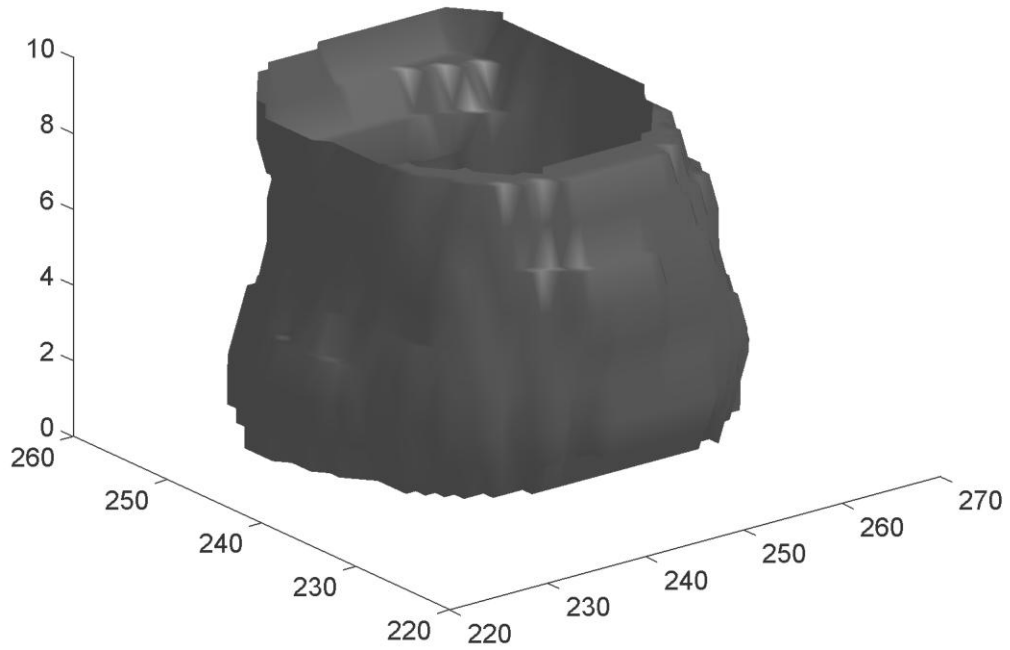


Figure 5.26 3D segmenting surface of the 3DRL-LSGA.

B. Segmentation of MRI images

Figure 5.27 shows the segmentation outcome of 2D-LSGA for each slice of a MRI image of patient *B* (white contour). The gray contour on the same image depicts the manual segmentation performed on the same image by a radiation oncologist. The 3D structure created by stacking all the segmenting contours on top of each other is shown in figure 5.28. Table 5.10 shows a sample header file generated from the 2D-LSGA segmentation of a single slice of a MRI image of patient *B*.

Table 5.10 The header file generated by the 2D-LSGA for a single slice of MRI image of patient *B*.

Generation	Avg_Fitness	Std_Dev	Best_Fitness
1	432.5	30.7	497.6
2	448.6	29.2	544.3
3	473.9	40.1	566.0
4	499.1	43.2	566.0
5	515.9	38.8	566.0
6	532.8	33.8	566.0
7	540.4	30.7	576.7
8	547.1	29.1	577.0
9	564.3	15.7	582.1
10	568.8	7.0	579.6
11	564.2	30.2	582.8
12	573.1	14.3	582.8
13	576.9	9.9	582.1
14	577.7	9.0	598.8
15	578.6	21.2	598.8
16	578.2	28.1	598.8
17	569.3	41.4	599.1
18	575.6	30.6	598.9
19	589.4	7.7	599.1
20	593.3	7.0	599.1
21	593.5	8.3	598.9
22	588.6	27.4	598.9
23	594.9	6.3	598.9
24	583.6	33.8	598.9
25	587.9	29.5	598.9
26	594.5	11.5	598.9
27	596.6	4.9	598.9
28	592.5	27.1	598.9
29	592.4	25.9	598.9
30	588.2	33.6	598.9

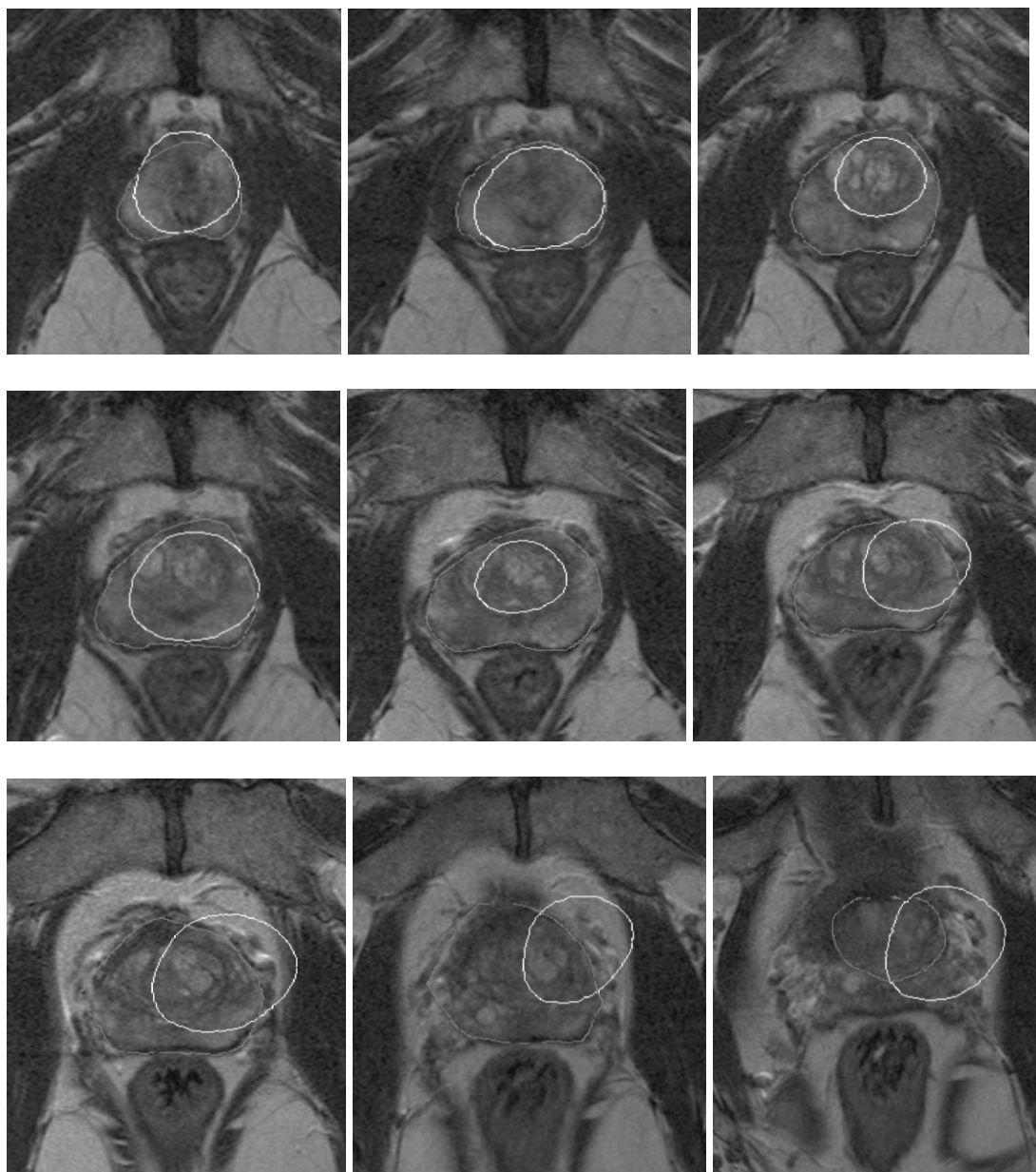


Figure 5.27 Slice-by-slice segmentation of a MRI image of patient B using 2D-LSGA.

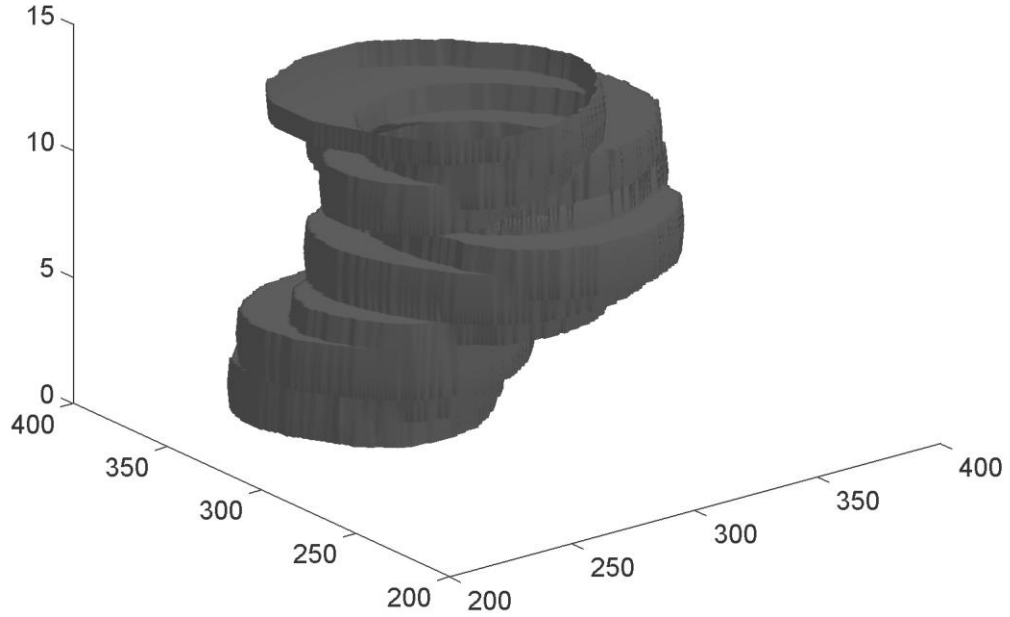


Figure 5.28 Segmenting contours of 2D-LSGA stacked together to form a 3D shape.

Figure 5.29 shows the segmentation outcome of 2DRL-LSGA for each slice of a MRI image of patient *B* (white contour). The gray contour on the same image depicts the manual segmentation performed on the same image by a radiation oncologist. The 3D structure created by stacking all the segmenting contours on top of each other is shown in figure 5.30. Table 5.11 shows a sample header file generated from the 2D-LSGA segmentation of a single slice of a MRI image of patient *B*.

Table 5.11 The header file generated by the 2DRL-LSGA for a single slice of MRI image of patient *B*.

Generation	Avg_Fitness	Std_Dev	Best_Fitness
1	590.0	29.1	661.0
2	604.0	28.1	663.3
3	625.0	27.7	692.6
4	641.0	18.5	667.6
5	651.7	18.1	667.6
6	654.2	21.3	667.6
7	656.6	21.8	667.6
8	661.7	10.6	669.3
9	665.6	11.1	699.5
10	669.3	8.9	702.5
11	668.8	21.6	703.3
12	676.8	14.6	705.2
13	677.5	28.4	707.6
14	691.2	17.3	704.8
15	698.0	8.9	706.8
16	699.6	13.6	706.8
17	703.4	1.6	706.8
18	703.2	2.4	707.6
19	699.8	23.0	707.6
20	705.0	2.0	707.8
21	705.9	2.4	708.8
22	704.4	9.3	708.8
23	702.6	23.4	708.4
24	702.7	12.9	708.4
25	702.3	22.0	708.5
26	704.9	10.0	709.3
27	707.0	6.0	709.3
28	702.8	17.9	709.3
29	707.7	1.5	709.3
30	706.3	7.4	709.3

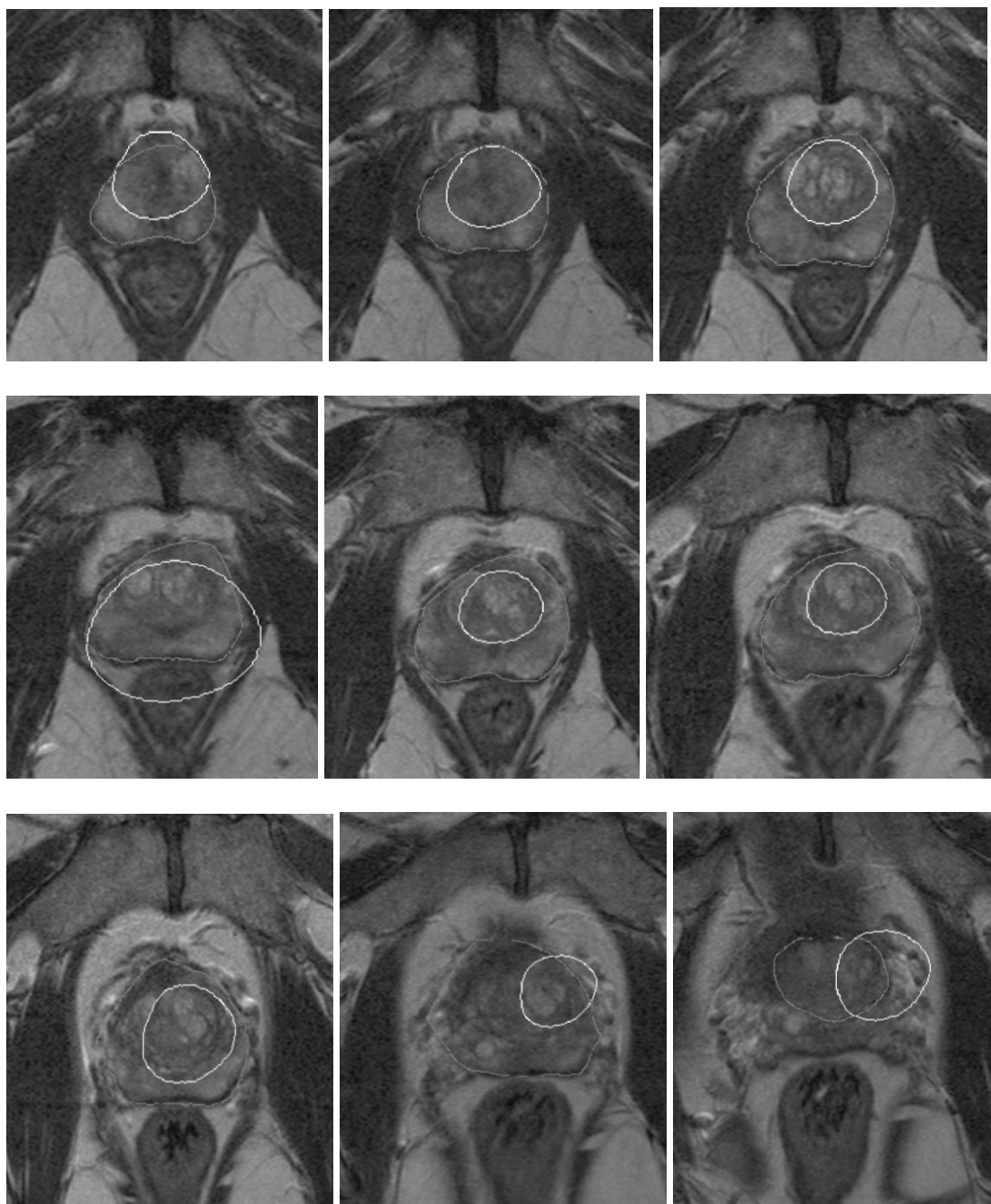


Figure 5.29 Slice-by-slice segmentation of a MRI image of patient B using 2DRL-LSGA of patient B .

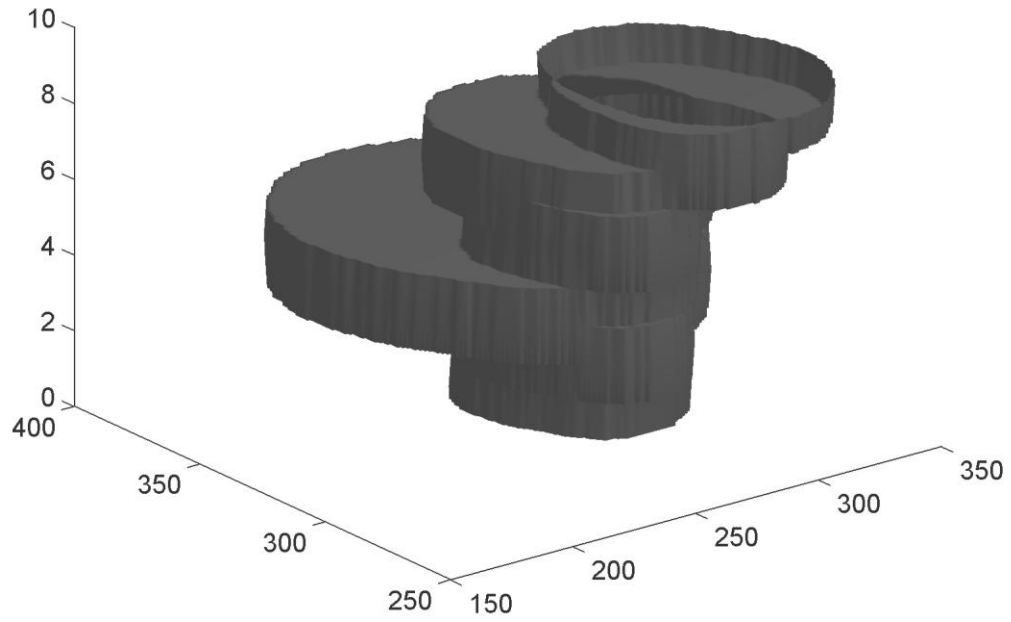


Figure 5.30 Segmenting contours of 2DRL-LSGA stacked together to form a 3D shape.

Figure 5.31 shows the segmentation outcome of 3D-LSGA for each slice of a MRI image of patient *B* (white contour). The gray contour on the same image depicts the manual segmentation performed on the same image by a radiation oncologist. The 3D structure created by stacking all the segmenting contours on top of each other is shown in figure 5.32. Table 5.12 shows a sample header file generated from the 3D-LSGA segmentation for the MRI image of patient *B*. Note that the 3D-LSGA was stopped after 10 generations.

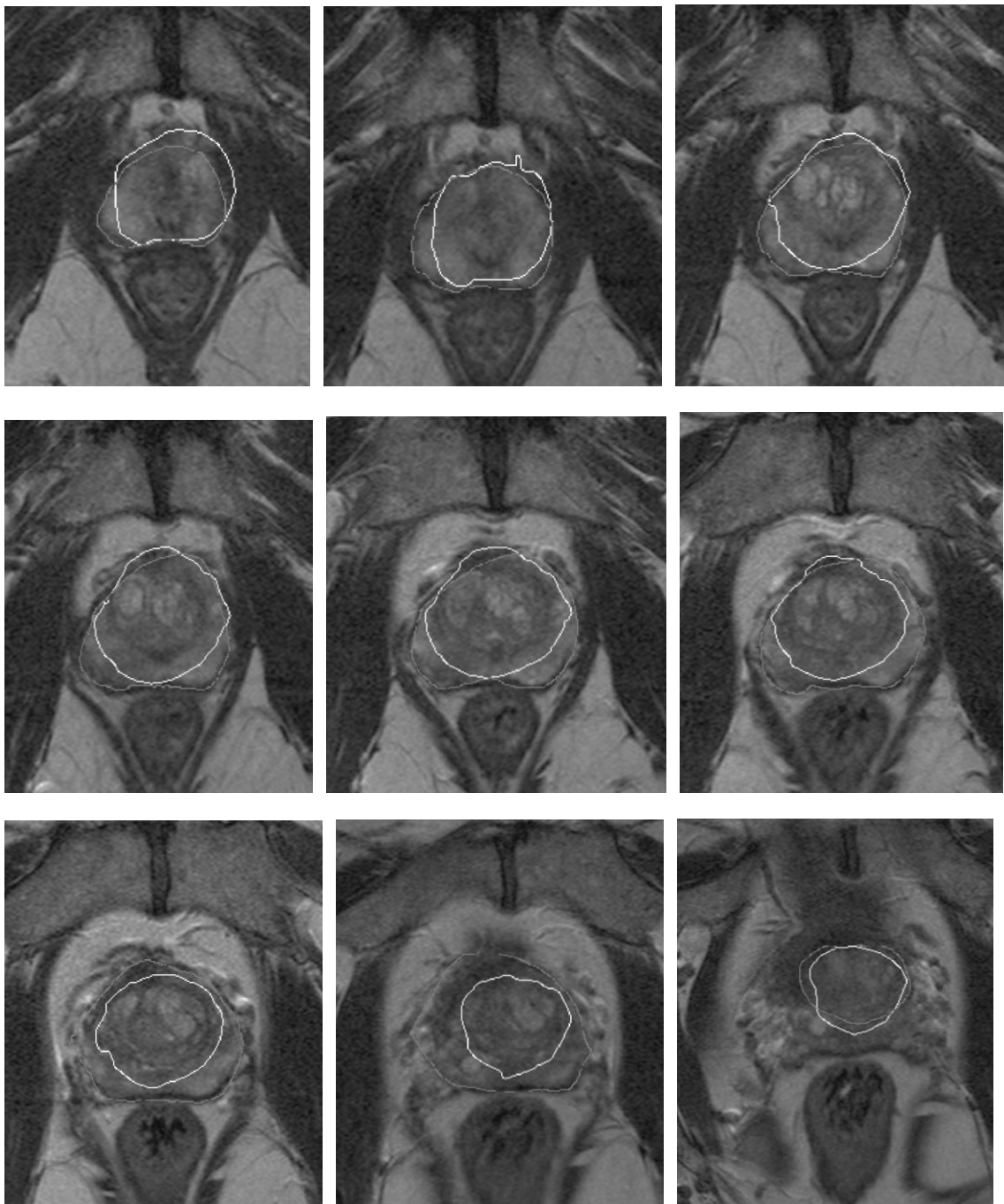


Figure 5.31 Slice-by-slice segmentation of a MRI image of patient B using 3D-LSGA.

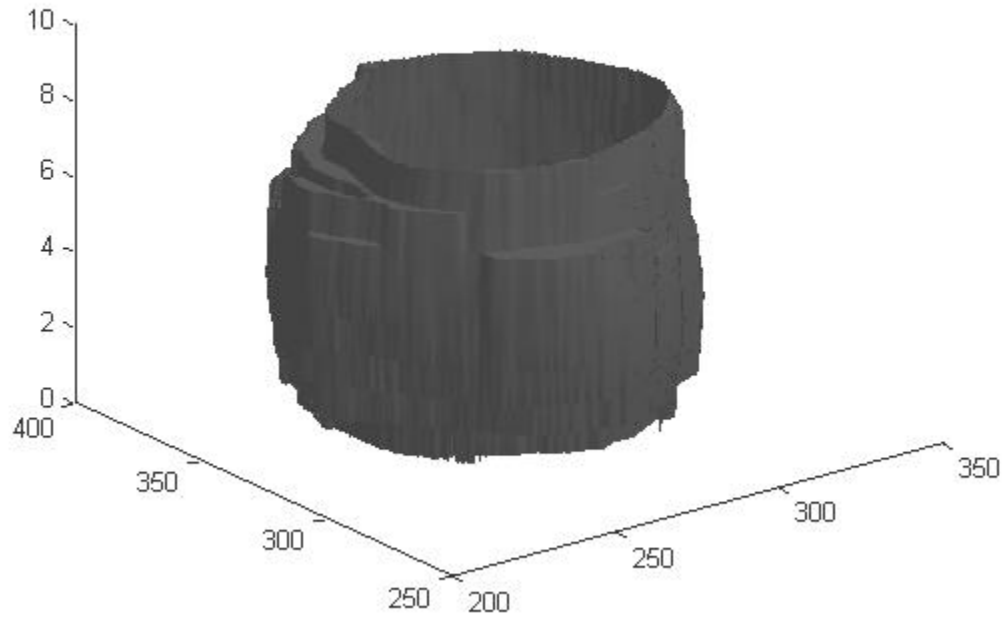


Figure 5.32 3D segmenting surface of the 3D-LSGA.

Table 5.12 The header file generated by the 3D-LSGA for a MRI image of patient *B*.

Generation	Avg_Fitness	Std_Dev	Best_Fitness
1	501.3	0.8	502.4
2	501.4	0.7	502.4
3	501.7	0.7	502.6
4	501.6	0.9	502.5
5	501.5	0.9	502.5
6	501.5	0.8	502.5
7	500.9	0.9	502.6
8	501.2	0.8	502.5
9	501.3	0.8	502.4
10	501.9	0.7	502.4

Figure 5.33 shows the segmentation outcome of 3DRL-LSGA for each slice of a MRI image of patient *B* (white contour). The gray contour on the same image depicts the manual segmentation performed on the same image by a radiation oncologist. The 3D structure created by stacking all the segmenting contours on top of each other is shown in figure 5.34. Table 5.13 shows a sample header file generated from the 3DRL-LSGA segmentation for the MRI image of patient *B*. Note that the 3DRL-LSGA was stopped after 10 generations.

Table 5.14 shows the *DSC* and *H* values computed using LSGA in all the four modes. Note that the *DSC* and *H* values for the MRI images are much higher than CT images. This is because the MRI images consisted of part of the pelvic image instead of the entire 2D image and therefore were more difficult to align with respect to training images. Figure 5.35 shows a sample MRI image on which the segmentation is inaccurate because the test image was not aligned properly with the training images.

Table 5.13 The header file generated by the 3DRL-LSGA for a MRI image of patient *B*.

Generation	Avg_Fitness	Std_Dev	Best_Fitness
1	644.5	2.9	647.4
2	646.3	1.4	648.0
3	647.3	1.9	651.4
4	646.9	1.1	648.9
5	647.2	0.6	648.0
6	647.4	1.2	649.7
7	647.2	1.3	650.2
8	647.2	1.7	650.2
9	647.9	1.1	650.2
10	648.3	0.9	650.2

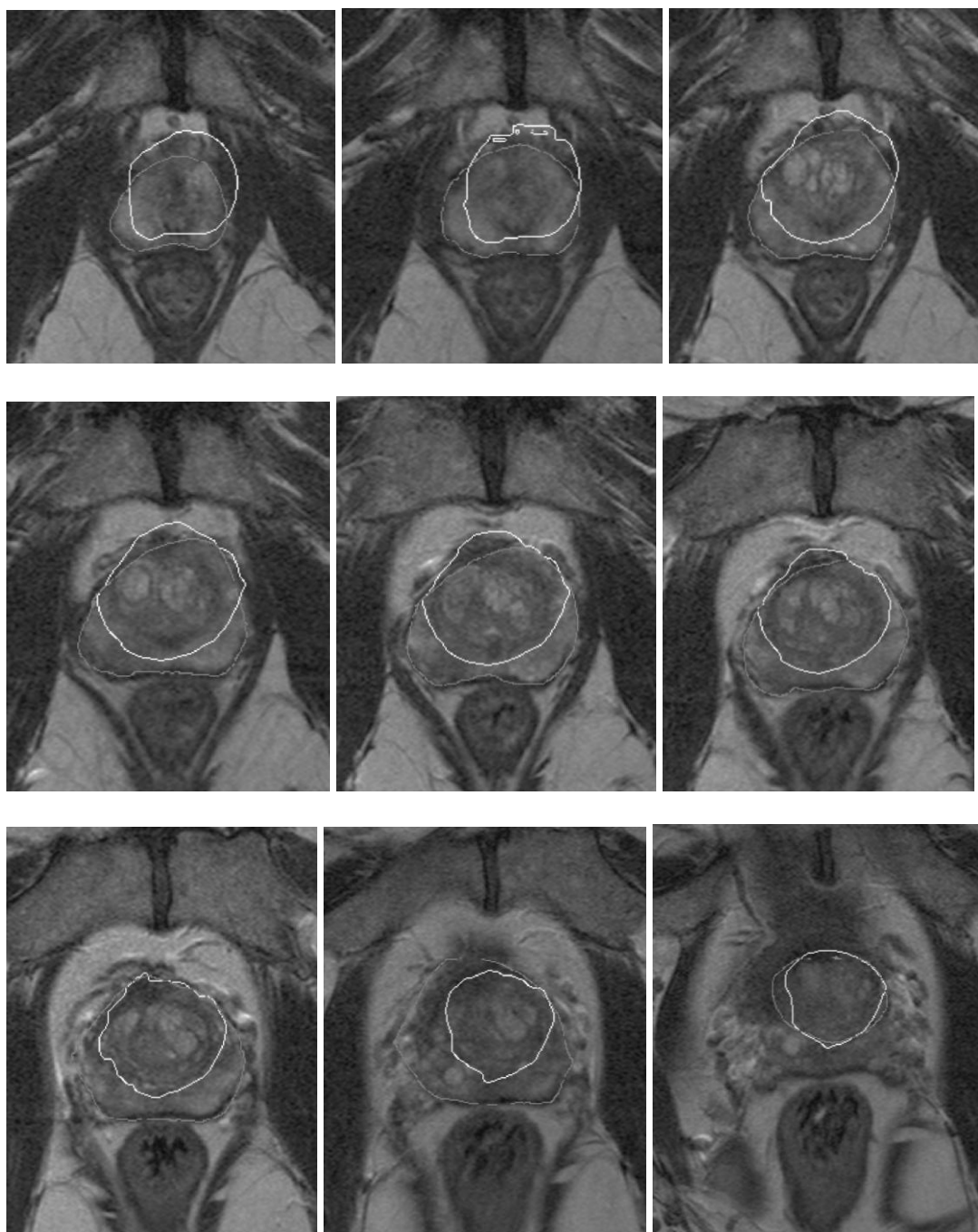


Figure 5.33 Slice-by-slice segmentation of an MRI image of patient B using 3DRL-LSGA.

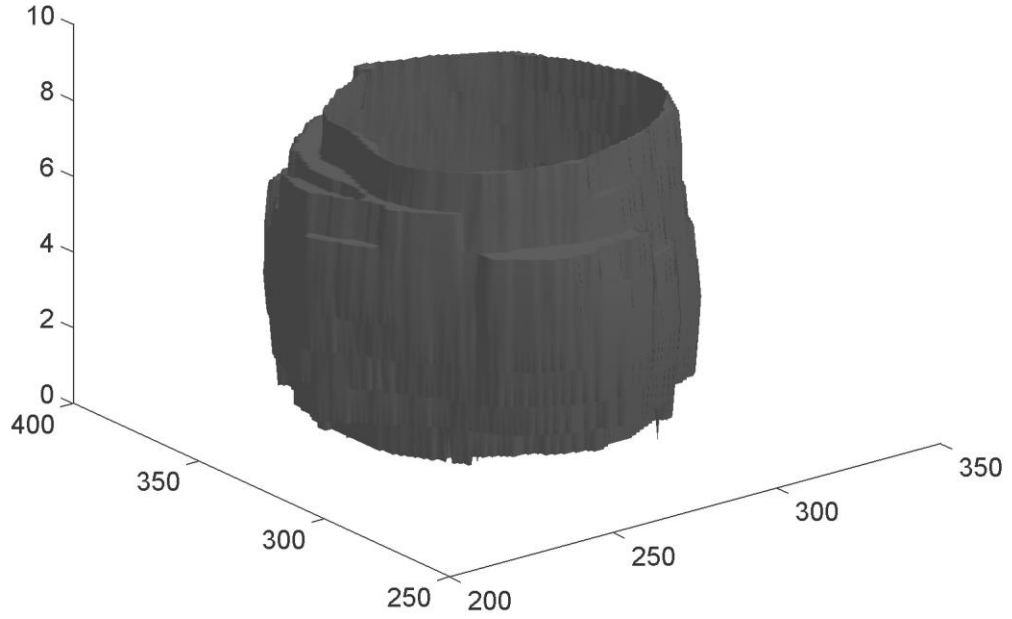


Figure 5.34 Segmenting surface of the 3DRL-LSGA.

Table 5.14 DSC and H values showing comparison of segmentation outcomes with the ground truth for the LSGA in 2D, 2DRL, 3D and 3DRL modes.

LSGA	DSC CT	H CT	DSC MRI	H MRI
2D	0.37 ± 0.2	18.4 ± 7.9	0.32 ± 0.2	58.1 ± 33.5
2DRL	0.45 ± 0.01	14.14 ± 0.4	0.35 ± 0.2	52.66 ± 32.5
3D	0.67 ± 0.04	5.8 ± 2.7	0.55 ± 0.3	40.8 ± 43.1
3DRL	0.69 ± 0.01	5.5 ± 3.1	0.54 ± 0.3	47.55 ± 44.5

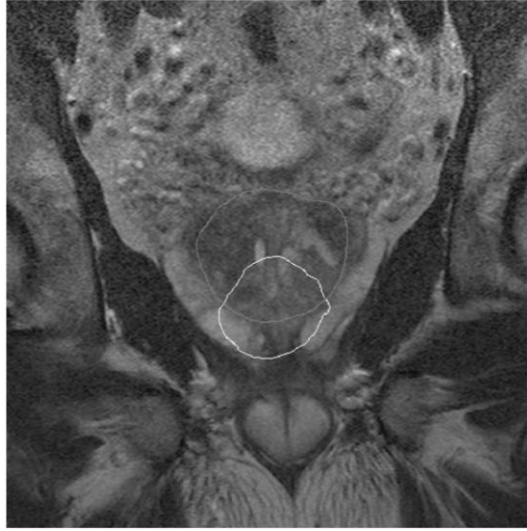


Figure 5.35 Segmentation result on a MRI image using 3DRL-LSGA. The figure shows that the segmentation is inaccurate when the test image is not aligned properly with the training images. Here, the gray contour represents manual segmentation and the white contour shows the segmenting contour derived by LSGA.

5.6 Summary

The visual analysis of the results confirms that the LSGA performs satisfactory segmentation of the pelvic CT and MRI images. Performing segmentation in three dimensions improves the alignment of the segmenting contours with respect to each other. Incorporating spatial relationships into the LSGA helps place the initial segmenting contour randomly on the images as opposed to the mean location for segmentation without the spatial relationships. It also improves fitness of the individuals of the 3D-LSGA, to more accurately represent the “goodness” of a segmenting contour. Part of this work has been published as a chapter [35] in the book, *Medical Image Analysis and Machine Learning Technologies: Algorithms and Techniques* by IGI global publishers.

6. Strengths and Limitations of LSGA and Future Work

The advantages and disadvantages of incorporating a genetic algorithm for level set segmentation are discussed here. Some advantages of the LSGA over traditional level set based segmentation methods are:

1. Flexibility

The genetic algorithm optimizes an implicit fitness function instead of an explicit energy function term. This makes it flexible enough to incorporate any kind of feature for performing segmentation without modifying the fitness function.

2. Ability to generalize

Genetic algorithms, like any other evolutionary optimization technique, typically have more generality than traditional optimization methods, in that they can be applied to a large set of features, and may be able to solve different kinds of segmentation problems by combining various types of features. Techniques to enhance the current LSGA, to perform such generalization can be explored in future. Other application domains where LSGA can be applied in future are: Iris segmentation from images of the eye, Liver segmentation on abdominal images, and a variety of other image retrieval problems that require combining multiple types of prior information.

3. Parallel

The LSGA evaluates multiple candidate solutions in parallel for performing segmentation. This is quite different from existing curve evolution based

segmentation techniques that evolve only one segmenting contour. Parallelization makes LSGA converge to a global minimum/maximum on the fitness landscape.

The limitations of the LSGA are the following:

1. Computational complexity

At each generation of the LSGA, the process of GA fitness evaluation, selection and crossover is repeated until the desired solution is achieved. The computational time of the LSGA is thus, a linear function of the population size (S) and the number of generations (G) and is given by $O(S \cdot G)$. This makes the GA slow when the population size is large and the GA is run for a large number of generations. One method to speed up the LSGA and thus make it practically viable is to implement it using a multi-thread approach and running each thread on parallel workstations (e.g., parallelism [67]). This will be explored in future.

2. Precision

As discussed in the previous section, a genetic algorithm can never produce exactly similar results in two runs. This is a disadvantage of all evolutionary computation methods. However, in cases where the global minimum is very difficult to find, deriving a “satisfactory” outcome using a genetic algorithm is worth exploring. For example, for the case of pelvic CT and MR images, it is impossible to derive the exact outline of the prostate (this is difficult for human experts too). However, deriving an approximate outline by modeling the uncertainties in tumor boundaries can significantly speed up the treatment planning process.

3. Dependence on registration

For prostate segmentation, all test images need to be registered with a training image so that the fuzzy relative location map can be used. Therefore, images need to be well-registered before segmentation can be performed using LSGA. The performance of the GA for the pelvic dataset thus depends to a certain extent on the registration accuracy.

4. Preference for smaller segmenting contours.

Although the fuzzy landscape helps localize the LSGA search to a small area on the pelvic images, using the fuzzy landscape makes the LSGA biased towards smaller segmenting contours centered on the fuzzy landscape. This can be overcome if the fuzzy angle maps are computed from all points on the reference object instead of just the center of the reference object. This is a subject of future work and can be explored to overcome the bias of the LSGA.

5. User interface

Currently the LSGA does not have a graphical user interface (GUI) that can be directly used by physicians. A GUI needs to be created that can allow a physician to upload any dataset and manually segment organs of interest and landmark regions on a subset of the data. The GUI also needs to integrate the LSGA with a variety of built-in feature extraction methods to segment test images. Such a framework needs to be developed in future to make the LSGA practically useful.

7. Conclusion

The LSGA performs derivative-free optimization of a level set function for image segmentation. Representing candidate solutions of the GA as segmenting contours and assessing their performance using a fitness function eliminates the need for defining an energy function (and the associated derivatives) and simplifies the optimization procedure needed for image segmentation.

This makes the level set function optimization flexible and enables the user to choose different kinds of features for exploring the fitness landscape. In this dissertation three types of features, texture, shape, and relative location have been explored for performing segmentation. The LSGA has been used for successfully segmenting thermographic images of hands, and for segmenting the prostate on pelvic CT and MRI images with a reasonable amount of success.

References

- [1] Ahmadian, A., & Mostafa, A. (2003). An efficient texture classification algorithm using Gabor wavelet. In *Proceedings of the 25th Annual International Conference of the IEEE EMBS*, 930-933, Washington, DC: IEEE press.
- [2] Ali, Y. M. B. (2009). Edge-based segmentation using robust evolutionary algorithm applied to medical images. *Journal of Signal Processing Systems*, 54(1), 231 – 238.
- [3] Antoniou, A., & Wu-Sheng, L. (2007). *Practical Optimization Algorithms and Engineering Applications*. New York: NY: Springer.
- [4] Ballerini, L. (1999). Genetic Snakes for Medical Images Segmentation. In Poli, R. et al. (Eds.), *Proceedings of the First European Workshops on Evolutionary Image Analysis, Signal Processing and Telecommunications*, Lecture Notes In Computer Science, 1596, 59-73, London, UK: Springer-Verlag.
- [5] Barrett, J. F., & Keat, N. (2004). Continuing Medical Education: Artifacts in CT: Recognition and Avoidance. *Radiographics*. 24,1679-1691.
- [6] Bhanu, B., Lee, S., & Ming, J. (1995). Adaptive image segmentation using a genetic algorithm. *IEEE Transactions on Systems Man and Cybernetics*, 25(12), 1543-1567.
- [7] Bureau of Labor Statistics. (2008). Nonfatal occupational injuries and illnesses requiring days away from work. *USDL*, 08-1716, U.S. dept. of labor, Washington.
- [8] Bloch, I. (1999). Fuzzy relative location between objects in image processing: a morphological approach. *IEEE trans. on Patt. Ana. Mach. Intell.*, 21(7), 657-664.
- [9] Bloch, I. (2009). Fuzzy methods in medical imaging. In N. Ayache, J. Duncan and N. Paragios (Eds.), *The Handbook of Biomedical Image Analysis*. Heidelberg, Germany: Springer.
- [10] Boyd, S. P., & Vandenberghe, L. (2004). *Convex Optimization*. Cambridge, UK: Cambridge University Press.
- [11] Brunnekreef, J. J., Oosterhof, J., Thijssen, D. H. J., Colier, W. N., & Van Uden, C.J. (2006). Forearm blood flow and oxygen consumption in patients with bilateral repetitive strain injury measured by near-infrared spectroscopy. *Clinical Physiology and Functional Imaging*, 26: 178-84.
- [12] Bunke, H. (2000). Graph matching for visual object recognition. *Spatial Vision*, 13, 335–340.
- [13] Buxton, R. (2003). *Introduction of Functional Magnetic Resonance Imaging, Principles and Techniques*. Cambridge, UK: Cambridge University Press.
- [14] Cagnoni, A., Dobrzeniecki, A. B., Poli, R., & Yanch, J. C. (1999). Genetic algorithm-based interactive segmentation of 3D medical images. *Image and Vision Computing*, 17(12), 881-895.

- [15] Caruana, R. A. & Schaffer, J. D. (1988). Representation and Hidden Bias: Gray vs. Binary Coding. *Proc. 6th Int. Conf. Machine Learning*, 153-161.
- [16] Chabrier, S., Rosenburger, C., Emile, B., & Laurent, H. (2008). Optimization based image segmentation by genetic algorithms. *EURASIP Journal on Video and Image Processing*, 1-23.
- [17] Chan, T., & Vese, L. (2001). Active contours without edges. *IEEE Trans. on Image Proc.*, 10, 266-277.
- [18] Coggins, J. M., & Jain, A.K. (1985). A spatial filtering approach to texture analysis. *Pattern Recognition Letters*, 3, 195-203.
- [19] Colliot, O., Camara, O., & Bloch, I. (2006). Integration of Fuzzy Spatial Relations in Deformable Models - Application to Brain MRI Segmentation. *Pattern Recognition*, 39(8), 1401-1414.
- [20] Costa, L.F., & Cesar, R. M. Jr. (2001). *Shape analysis and classification theory and practice*. Boca Raton, FL: CRC press.
- [21] Cremers, D., Rousson, M., & Deriche, R. (2007). A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *Intl. J. of Comp. Vis.*, 72(2), 195-215.
- [22] Cross, G. C., & Jain, A. K. (1983). Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5, 25-39.
- [23] Daida, J. M., Hommes, J. D., Bersano-Begey, T. F., Ross, S. J., & Vesecky, J. F. (1996). Algorithm discovery using the genetic programming paradigm: Extracting low-contrast curvilinear features from SAR images of Arctic ice. In P. J. Angeline & K. E. Kinnear Jr. (Eds.), *Advances in Genetic Programming 2*, 417-442. Cambridge, MA: MIT press.
- [24] Dehmeshki, J., Ye, X., Lin, X., Valdivieso, M., & Amin, H. (2007). Automated detection of lung nodules in CT images using shape-based genetic algorithm. *Computerized Medical Imaging and Graphics*, 31(6), 408-417.
- [25] De Jong, K. A. *Evolutionary Computation: A Unified Approach*. Cambridge, MA: MIT Press.
- [26] Dowling, J., Fripp, J., Greer, P., Ourselin, S., & Salvado, O. (2009). Automatic atlas-based segmentation of the prostate: A MICCAI 2009 Prostate Segmentation Challenge entry. *Medical Image Computing and Computer Assisted Intervention Conference*, 2009.
- [27] Dunn, D., Higgins, W.E. (1995). Optimal Gabor filters for texture segmentation. *IEEE Transactions on Image Processing*, 4(7), 947-964.
- [28] Elfadel, I. M., Picard, R.W. (1993). Gibbs random fields, co-occurrences and texture modeling. *IEEE Trans. Patt. Anal. and Mach. Intell.*, 16(1), 24-37.

- [29] Engelbrecht, A. P. (2007). *Computational Intelligence: an Introduction*. 2nd Edition. Hoboken, NJ: Wiley Publishing.
- [30] Etyngier, P., Segonne, F., & Keriven R. (2007). Active contour-based image segmentation using machine learning techniques. In *Proceedings of the Medical Image Computing and Computer-Assisted Intervention Conference*, 891-900, Heidelberg, Germany: Springer.
- [31] Faulkner, W. M. (1996). *Basic Principles of MRI*. www.e-radiography.net/mrict/Basic_MR.pdf
- [32] Friedland, N. S. & Rosenfeld, A. (1992). Compact object recognition using energy-function based optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:770-777.
- [33] Gerr, F., Monteilh, G. F., Marcus, M. (2006). Keyboard use and musculo-skeletal outcomes among computer users. *J. Occup Rehabil*, 16, 265-277.
- [34] Ghosh, P., Mitchell, M., & Gold, J. (2010). LSGA: Combining level-sets and genetic algorithms for segmentation. *Evolutionary Intelligence*, 3, 1-11.
- [35] Ghosh, P., Mitchell, M., Tanyi, J., & Hung, A. (2009). A genetic algorithm-based level-set curve evolution for prostate segmentation on Pelvic CT and MRI Images. In Gonzalez, F., Romero, E. R. (Eds.), *Biomedical Image Analysis and Machine Learning, Applications and Techniques*, Chap. 6. Hershey, PA: IGI Global.
- [36] Ghosh, P., Mitchell, M. (2006). Segmentation of medical images using a genetic algorithm. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, 1171-1178. New York, NY: ACM.
- [37] Gleason, D. F., Mellinger, G. T., and the Veterans Administration Cooperative Urological Research Group. (1974). Prediction of prognosis for prostatic adenocarcinoma by combined histologic grading and clinical staging. *J. Urol.* 111, 58-64.
- [38] Glover, F. (2006). Parametric tabu-search for mixed integer programs. *Comput. Oper. Res.* 33, 9, 2449-2494.
- [39] Gold, J. E., Cherniack, M., & Buchholz, B. (2004). Infrared thermography for examination of skin temperature in the dorsal hand of office workers. *European Journal of Applied Physiology*, 93(1-2), 245-251.
- [40] Gold, J. E., Cherniack, M., Hanlon, A., Dennerlein, J. T., Dropkin, J. (2009). Skin temperature in the dorsal hand of office workers and severity of upper extremity musculoskeletal disorders. *Int. Arch. Occup. Env. Health*, 82(10): 1281-1292.
- [41] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: 1st Addison-Wesley Longman Publishing Co Inc.
- [42] Goldberger, J., & Greenspan, H. (2006). Context-based segmentation of image sequences. *IEEE Trans. on Patt. Ana. and Mach. Intell.*, 28(3), 463-468.

- [43] Gonzales, R. C., & Woods, R. E. (2002). *Digital Image Processing*, second edition. Upper Saddle River, NJ: Prentice Hall.
- [44] Gray, P., Hart, W., Painton, L., Philips, C., Trahan, M., & Wagner, J. (1997). *A survey of global optimization methods*. Sandia National Laboratories, <http://www.cs.sandia.gov/opt/survey/>.
- [45] Harris, C., & Buxton, B. (1996). Evolving edge detectors. *Research Note RN/96/3*, University College, Dept. of Computer Science, London.
- [46] Harvey, N. et al. (2002). Comparison of GENIE and conventional supervised classifiers for multispectral image feature extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 40(2), 393-404.
- [47] Harvey, N., Levenson, R.M., & Rimm, D.L. (2003). Investigation of automated feature extraction techniques for applications in cancer detection from multi-spectral histopathology images. *Proc of SPIE*, 5032, 557-566.
- [48] He, L., Peng, Z., Everding, B., Wang, X., Han, C. Y., Weiss, K. L., & Wee, W. G. (2008). A comparative study of deformable contour methods on medical image segmentation. *Image and Vision Computing*, 26(2), 141-163.
- [49] Herault, L. and Horaud, R. (1993). Figure-ground discrimination: A combinatorial optimization approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15, 899-914.
- [50] Hill, A., & Taylor, C. (1992). Model-based image interpretation using genetic algorithms. *Image and Vision Computing*, 10(5), 295-300.
- [51] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI, University of Michigan Press.
- [52] Hung, Y. P., Cooper, D. B., and Cernuschi-Frias, B. (1991). Asymptotic Bayesian surface estimation using an image sequence. *International Journal of Computer Vision*, 6(2), 105-132.
- [53] Huttenlocher, D., Klanderman, G., & Rucklidge, W. (1993). Comparing images using the hausdorff distance. *IEEE Trans. Patt. Anal. and Mach. Intell.*, 15, 850-863.
- [54] ICRU. (1993). *Prescribing, recording, and reporting photon beam therapy. Report #50*, Bethesda, MD., International Commission on Radiation Units and Measurements.
- [55] Julesz, B. (1986). Texton gradients: The texton theory revisited. *Bio. Cybern.*, 54, 245-251.
- [56] Kashyap, R. L., Chellappa, R., & Ahuja, N. (1981). Decision rules for the choice of neighbors in random field models of images. *Computer Graphics and Image Processing*, 15, 301-318.

- [57] Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1, 321-331.
- [58] Kulikowski, J. J., Marcelja, S. & Bishop, P. (1982). Theory of spatial position and spatial frequency relations in the receptive fields of simple cells in the visual cortex. *Biol. Cybern*, 43,187-198.
- [59] Laarhoven, P. J., & Aarts, E. H. (Eds.) (1987). *Simulated Annealing: Theory and Applications*. Norwell, MA: Kluwer Academic Publishers.
- [60] Larsson, R., Öberg, P. Å., & Larsson, S-E. (1999). Changes of trapezius muscle blood flow and electromyography in chronic neck pain due to trapezius myalgia. *Pain*, 79, 45-50.
- [61] Lattanzi, J., McNeeley, S., Pinover, W., Horwitz, E., Das, I., Schultheiss, T. E., Hanks, G. E. (1999). A comparison of daily CT localization to a daily ultrasound-based system in prostate cancer. *International Journal of Radiation Oncology, Biology and Physics*, 43(4), 719-725.
- [62] Laws, K. I. (1980). *Texture Image Segmentation*. Ph.D. dissertation, Univ. of Southern California.
- [63] Laws, K. I. (1980). Rapid texture identification. *SPIE Image Processing for Missile Guidance*, 238, 376-380.
- [64] Leventon, M., Grimson, E., & Faugeras, O. (2000). Statistical shape influence in geodesic active contours. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 1, 316-323, IEEE Press.
- [65] Li, S.Z. (2009). *Markov Random Field Modeling in Computer Vision*, 3rd Ed., New York, NY: Springer.
- [66] Louchet, J. (2007). Model-based image analysis using evolutionary computation. In: *Genetic and Evolutionary Computation for Image processing and Analysis*, New York, NY: Hindawi.
- [67] Luque, G., Alba, E., & Dorronsoro, B. (2005). Parallel genetic algorithms. In E. Alba (Ed.), *Parallel metaheuristics: A new class of algorithms*, 107-126. Hoboken, NJ: John Wiley & Sons Inc.
- [68] MacEachern, L., & Manku, T. (1998). Genetic algorithms for active contour optimization. *IEEE Proceedings of the Intl Symp on Circuits and Systems*, 4, 229-232.
- [69] Mahr, A., Levegrun, S., Bahner, M.L., Kress, J., Zuna, I., & Schlegel, W. (1999). Usability of semiautomatic segmentation algorithms for tumor volume determination, *Invest. Radiolog.*, 34, 143-150.
- [70] Malladi, R., Sethian, J. A., Vemuri, B. C. (1995). Shape modeling with front propagation: A level set approach. *IEEE Trans on Patt Anal and Machine Intell*, 17(2), 158-175.

- [71] Mallat, S. (1989). A theory for multiresolution signal decomposition: The Wavelet representation. *IEEE Trans. Pattern Analysis & Machine Intell.*, 11(7), 674-693.
- [72] Mark, D. M., & Egenhofer, M. J. (1994). Modeling spatial relations between lines and regions: combining formal mathematical models and human subjects testing. *Cartography and Geographic Information Systems*, October 1994, 21(4), 195-212.
- [73] Maulik, U. (2009). Medical image segmentation using genetic algorithms. *Trans. Info. Tech. Biomed.* 13(2), 166-173.
- [74] McInerney, T., & Terzopoulos, D. (1996). Deformable models in medical image analysis: A survey. *Medical image analysis*, 1(2), 91-108.
- [75] Michalewicz, Z., & Fogel, D. B. (2000). *How to Solve It: Modern Heuristics*. Berlin, Germany, Springer-Verlag.
- [76] Mitchell, M. (1996). *An introduction to genetic algorithms*. Cambridge, MA: MIT Press.
- [77] Mitchell, T. M. (1997). *Machine learning*. Singapore: McGraw-Hill International Editions.
- [78] Miyajima, K., & Ralescu, A. (1994). Spatial organization in 2D segmented images: Representation and recognition of primitive spatial relations. *Fuzzy Sets and Systems*, 65(2/3), 225-236.
- [79] Motwani, R. & Raghavan, P. (1995). *Randomized Algorithms*. Cambridge, UK: Cambridge University Press.
- [80] Mumford, D., Shah, J. (1989). Optimal approximation by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math*, 42, 577-685.
- [81] Murray, D. and Buxton, B. (1987). Scene segmentation from visual motion using global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8, 220-228.
- [82] Nordin, P., Banzhaf, W. (1996). Programmatic compression of images and sound. In Koza, J. R. et al. (Eds.), *Proceedings of the 1st Annual Conference on Genetic Programming*. Morgan Kaufmann, San Francisco, CA.
- [83] Osher, S.J., & Fedkiw, R.P. (2002). *Level set methods and dynamic implicit surfaces*. New York, NY: Springer.
- [84] Oskarsson, E., Gustafsson, B. E., Pettersson, K., & Piehl, Aulin K. (2007) Decreased intramuscular blood flow in patients with lateral epicondylitis. *Scandinavian Journal of Medicine and Science in Sports*, 17, 211-215.
- [85] Pentland, A. P. (1984). Fractal-based description of natural scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6), 661-674.

- [86] Pham, T. V., & Smeulders, A. W. M. (2006). Learning spatial relationships in object recognition. *Pattern Recognition Letters*, 27, 1673-1684.
- [87] Poli, R., & Cagoni S. (1997). Genetic programming with user-driven selection: Experiments on the evolution of algorithms for image enhancement. In *Proceedings of the 2nd Annual Conference on Genetic Programming* (pp. 269-277). San Francisco, CA: Morgan Kaufmann.
- [88] Pritchard, M. H., Pugh, N., Wright, I., Brownlee, M. (1999). A vascular basis for repetitive strain injury. *Rheumatology*, 38, 636-639.
- [89] Pusey, E., Lufkin, R. B., Brown, R. K., Solomon, M. A., Stark, D. D., Tarr, R. W. & Hanafee, W. N. (1986). Magnetic resonance imaging artifacts: mechanism and clinical significance. *Radiographics*. 6, 891-911.
- [90] Rabbani, M. & Jones, P. W. (1991). *Digital Image Compression Techniques*. SPIE Tutorial Text Vol. TT07, Bellingham, WA: SPIE Publications.
- [91] Roberts, L. G. (1965). Machine perception of three-dimensional solids, In J. T. Tippett, et al. (Ed.), *Optical and Electro-Optical Information Processing*. Cambridge, MA: MIT Press.
- [92] Sethian, J. A. (1999). *Level set methods and fast marching methods*. New York, NY: Cambridge University Press.
- [93] Shi, J. & Malik, J. (2000). Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 888-905.
- [94] Shibuya, K., Mathers, C.D., Boschi-Pinto, C., Lopez, A. D., Murray, C. J. L.(2002) Global and regional estimates of cancer mortality and incidence by site: II. Results for the global burden of disease 2000. *BMC Cancer*, 2:37.
- [95] Shotton, J., Winn, J., Rother, C., & Criminisi, A. (2006). Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: *European Conference on Computer Vision*, 1-15.
- [96] Skubic, M., Blisard, S., Bailey, C., Adams, J., & Matsakis, P. (2004). Qualitative analysis of sketched route maps: translating a sketch into linguistic descriptions. *IEEE Trans. Syst. Man, Cybern.- Part B: Cybern.* 34(2), 1275-1282.
- [97] Sonka, M., Hlavac, V., Boyle, R. (1994). *Image processing, analysis and machine vision*. London, UK: Chapman & Hall.
- [98] Sugimoto, H., Miyajim, N., & Ohsawa, T. (1994). Carpal tunnel syndrome: evaluation of median nerve circulation with dynamic contrast-enhanced MR imaging. *Radiology*, 190, 459-466.
- [99] Tremeau, A., Borel, N. (1997). A Region Growing and Merging Algorithm to Color Segmentation. *Pattern Recognition*, 30(7), 1191-1203.

- [100] Tsai, A., Yezzi, A., Wells, W., Tempany, C., Tucker, D., Fan, A., Grimson, E., & Willsky A. (2003). A shape-based approach to the segmentation of medical imagery using level sets. *IEEE Transactions on Medical Imaging*, 22, 137-154.
- [101] Tuseryan, M. (2004). Model based texture segmentation, *Pattern Recognition Letters*, 15, 659-668.
- [102] Tuy, H. (1998). *Convex Analysis and Global Optimization*, volume 22 of *Nonconvex Optimization and Its Applications*. Alphen aan den Rijn, Netherlands: Kluwer.
- [103] Udupa, J. K., LeBlanc, V. R., Zhuge, Y., Imielinska, C., Schmidt, H., Currie, B. E., Hirsch, L. M., & Woodburn, J. (2006). A framework for evaluating image segmentation. *Computerized Medical Imaging and Graphics*, 30, 75-87.
- [104] Van Rijsbergen, C.J. (2004). *The Geometry of Information Retrieval*. Cambridge, UK: Cambridge University Press.
- [105] Vieu, L. (1997). Spatial representation and reasoning in artificial intelligence. In Stock, O. (Ed.), *Spatial and Temporal Reasoning*, 5-41. Alphen aan den Rijn, Netherlands: Kluwer.
- [106] Xu, C., Pham, D. L., & Prince, J. L. (2000). Medical Image Segmentation Using Deformable Models. In Fitzpatrick, J.M. & Sonka, M. (Eds.), *Handbook of Medical Imaging- Volume 2: Medical Image Processing and Analysis*, 129-174. Bellingham WA: SPIE Press.
- [107] Yang, M., Wu, Y., & Hua, G. (2009). Context-aware visual tracking. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 31(7), 1195-1209.