Prostate Segmentation in Ultrasound Images
Using Image Warping and Ellipsoid Fitting

by

Sara Badiei

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

APRIL 2007

© Sara Badiei, 2007
Abstract

This thesis outlines an algorithm for 2D and 3D semi-automatic segmentation of the prostate from B-mode trans-rectal ultrasound (TRUS) images. In semi-automatic segmentation, a computer algorithm outlines the boundary of the prostate given a few initialization points. The algorithm is designed for prostate brachytherapy and has the potential to: i) replace pre-operative manual segmentation, ii) enable intra-operative segmentation, and iii) be integrated into a visualization tool for training residents.

The segmentation algorithm makes use of image warping to make the 2D prostate boundary elliptical. A Star Kalman based edge detector is then guided along the elliptical shape to find the prostate boundary in the TRUS image. A second ellipse is then fit to the edge detected measurement points. Once all 2D slices are segmented in this manner an ellipsoid is fit to the 3D cloud of points. Finally a reverse warping algorithm gives us the segmented prostate volume.

In-depth 2D and 3D clinical studies show promising results. In 2D, distance based metrics show a mean absolute difference of $0.67 \pm 0.18\text{mm}$ between manual and semi-automatic segmentation and area based metrics show average sensitivity and accuracy over 97% and 93% respectively. In 3D, i) the difference between manual and semi-automatic segmentation is on the order of inter-observer variability, ii) the repeatability of the segmentation algorithm is consistently better than the intra-observer variability, and iii) the sensitivity and accuracy are 97% and 85% respectively. The 3D algorithm requires only 5 initialization points and can segment a prostate volume in less than 10 seconds (approximately 40 times faster than manual segmentation).

The novelties of this algorithm, in comparison to other works, are in the warping and ellipse/ellipsoid fitting steps. These two combine to provide a simple solution that works well even with non-ideal images to produce accurate, real-time results.
Table of Contents

Abstract .......................................................... ii
Table of Contents ................................................ iii
List of Tables ...................................................... vii
List of Figures ...................................................... viii
Acknowledgements ................................................. xv
1 Introduction ...................................................... 1
1.1 Prostate Cancer ............................................... 1
1.2 Prostate Brachytherapy ...................................... 3
  1.2.1 Why Prostate Brachytherapy ........................... 3
  1.2.2 Prostate Brachytherapy Procedure at the BCCA .... 4
1.3 Thesis Motivation: Automatic Prostate Segmentation ... 5
  1.3.1 Why Segment the Prostate Automatically ............ 5
  1.3.2 Difficulties in Automatic Prostate Segmentation .... 6
1.4 Thesis Objective: Semi-Automatic 3D Prostate Segmentation . 6
  1.4.1 Segmentation Criteria at BCCA ....................... 6
  1.4.2 Literature Review .................................... 8
    1.4.2.1 2D Automatic Prostate Segmentation from TRUS images ... 8
    1.4.2.2 3D Automatic Prostate Segmentation from TRUS images ... 9
  1.4.3 General Overview of Approach ........................ 10
1.5 Thesis Overview ................................................. 12
2 Algorithm: 2D Segmentation .................................. 13
  2.1 Introduction to the 6-step Algorithm .................... 13
  2.2 Initialization ............................................... 14
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3</td>
<td>Forward and Reverse Warp</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Forward Warp</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Warping Function</td>
</tr>
<tr>
<td>2.3.2.1</td>
<td>Choosing the ( \sigma ) (Stretch Factor) Automatically</td>
</tr>
<tr>
<td>2.3.3</td>
<td>Reverse Warp</td>
</tr>
<tr>
<td>2.4</td>
<td>Ellipse Fitting</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Equation of an Ellipse</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Fitting an Ellipse to Measurement Points</td>
</tr>
<tr>
<td>2.5</td>
<td>IMMPDA Edge Detector</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Star Kalman - Finding Candidate Edge Points</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Kalman or Extended Kalman Filter - Finding the Prostate Boundary</td>
</tr>
<tr>
<td>2.5.3</td>
<td>Probabilistic Data Association Filter - Finding the Measurement Value</td>
</tr>
<tr>
<td>2.5.4</td>
<td>Interacting Multiple Modes</td>
</tr>
<tr>
<td>2.5.5</td>
<td>System Models</td>
</tr>
<tr>
<td>2.5.5.1</td>
<td>Linear Constant Velocity (Circle) Model</td>
</tr>
<tr>
<td>2.5.5.2</td>
<td>Linear Ellipse Model</td>
</tr>
<tr>
<td>2.5.5.3</td>
<td>Non-linear Ellipse Model - Derivative Method</td>
</tr>
<tr>
<td>2.5.5.4</td>
<td>Non-Linear Ellipse Model - Output Method</td>
</tr>
<tr>
<td>2.5.6</td>
<td>Comparing the Four Models</td>
</tr>
<tr>
<td>2.5.6.1</td>
<td>Comparing the Two Linear Models</td>
</tr>
<tr>
<td>2.5.6.2</td>
<td>Comparing the Two Non-linear Models</td>
</tr>
<tr>
<td>2.5.6.3</td>
<td>Comparing Linear vs Non-linear Model</td>
</tr>
<tr>
<td>3</td>
<td>Study Results and Discussion: 2D Segmentation</td>
</tr>
<tr>
<td>3.1</td>
<td>Study Setup</td>
</tr>
<tr>
<td>3.2</td>
<td>Validation Methods</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Distance Based Metrics</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Area Based Metrics</td>
</tr>
<tr>
<td>3.3</td>
<td>Results</td>
</tr>
<tr>
<td>3.4</td>
<td>Discussion</td>
</tr>
<tr>
<td>4</td>
<td>Algorithm: 3D Segmentation</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction to 8-step Algorithm</td>
</tr>
</tbody>
</table>
### TABLE OF CONTENTS

4.2 Ellipsoid Fitting ........................................... 52  
  4.2.1 Equation of an Ellipsoid ......................... 52  
  4.2.2 Fitting an Ellipsoid to Measurement Points .......... 55  
5 Study Results and Discussion: 3D Segmentation ............. 57  
  5.1 Study Outline ........................................... 57  
    5.1.1 Study Setup ........................................ 58  
    5.1.2 Removing Bias ..................................... 59  
    5.1.3 Segmentation Guidelines for Observers .......... 59  
  5.2 Validation Methods and Results ........................ 60  
    5.2.1 Distance Based Metrics .......................... 60  
      5.2.1.1 Comparing Difference Between Semi-automatic Segmentation and 
              the Gold Standard to Inter-observer Variability .......... 61  
      5.2.1.2 Comparing Repeatability of Semi-automatic Segmentation to Intra- 
              observer Variability .................................. 70  
    5.2.2 Volume Based Metrics ............................ 73  
    5.2.3 3D Segmentation Algorithm Speed .................. 75  
  5.3 Discussion ............................................ 78  
    5.3.1 Performance ....................................... 78  
      5.3.1.1 Potential for Integration at BCCA .............. 78  
      5.3.1.2 Algorithm Limitations ........................ 78  
6 Conclusion and Future Work ................................ 80  
  6.1 Future Work ........................................... 83  
Bibliography .................................................. 85  
A Ellipse/Ellipsoid Reference Frame Conventions ............ 88  
  A.1 Reference Frame Convention in 2D .................... 88  
  A.2 Reference Frame Convention in 3D .................... 89  
B Ellipse/Ellipsoid Parameter Conversions .................... 92  
  B.1 Fitting and Ellipse to Measurement Points in 2D ....... 92  
    B.1.1 Deriving P Parameters from q parameters (q to P) .... 92  
    B.1.2 Deriving q Parameters from P parameters (P to q) .... 93  
    B.1.2.1 Method 1 ....................................... 93
TABLE OF CONTENTS

B.1.2.2 Method 2 ......................................................... 96
B.2 Fitting an Ellipsoid to Measurement Points in 3D ......................... 98
  B.2.1 Deriving $P$ Parameters from $q$ parameters ($q$ to $P$) .............. 98
  B.2.2 Deriving $q$ Parameters from $P$ parameters ($P$ to $q$) .............. 99
    B.2.2.1 Method 1 ..................................................... 99
    B.2.2.2 Method 2 ..................................................... 102
    B.2.2.3 Comparing Method 1 and Method 2 ................................ 103
C Ellipse/Ellipsoid Fitting - Mathematical Proofs .......................... 107
  C.1 LSM Cost Function ................................................. 107
  C.2 Minimizing Cost Function using GEP ................................ 109
D Lagrangian Multipliers ..................................................... 112
E Newton-Raphson Method ..................................................... 114
F Tracking theory and (Extended) Kalman Filters .......................... 116
  F.1 Kalman Filter ...................................................... 116
  F.2 Extended Kalman Filter ............................................ 118
  F.3 Interacting Multiple Modes ......................................... 118
  F.4 Probabilistic Data Association Filter .............................. 122
G Semi-automatic Segmentation Results .................................... 125
## List of Tables

1.1 Overview of best 2D and 3D segmentation algorithms available .......................... 11

2.1 Linear State Space Models ...................................................................................... 30
2.2 Non-linear State Space Models .............................................................................. 31

3.1 Average area and distance based metrics for validation of segmentation algorithm .... 47
3.2 Average times for manual and semi-automatic segmentation ................................. 47

5.1 Pearson correlation coefficients (correlating computer to manual segmentation) for different regions of the prostate and for different observers. .......................... 65

5.2 Comparing the average difference between semi-automatic segmentation and the gold standard ($\Delta_{SA-GS}$) to inter-observer variability (IOV) for different regions of the prostate. .................................................. 69

5.3 Comparing the repeatability of the computer algorithm to intra-observer variability in manual segmentation. .............................................................. 72

5.4 Comparing intra-observer variability to computer repeatability in different regions of the prostate and for different observers. ............................... 73

5.5 Sensitivity and accuracy of 3D semi-automatic segmentation in comparison to the gold standard. .......................................................... 75

5.6 Average times for manual and semi-automatic segmentation of an entire prostate volume composed of approximately 10-13 slices. .......................... 76
List of Figures

1.1 Prostate brachytherapy ................................................................. 1
1.2 Segmentation difficulties due to anatomy ........................................ 7
1.3 Comparing TRUS image quality before and after operation ............. 7

2.1 The proposed semi-automatic segmentation algorithm consists of 6 steps. Steps 1 and 2 show the unwarped (x-mark) and warped (circles) initialization points respectively. Steps 3 and 5 show the best ellipse fit to the warped initialization points and to the IMMPDA edge map respectively. Step 6 shows the final segmentation result ............................................................... 14
2.2 Place every point in the original image (left) into a new position to create the warped image (right) using a radial warping function to get \( r_{new} \) from \( r \) and \( \theta \) .................. 16
2.3 Comparing the original image and original initialization points (a) to the warped image and warped initialization points (b) .................. 16
2.4 The \( \theta \) component (top) and \( r \) component (bottom) of the warping function in 2D ...... 17
2.5 The warping function: \( r_{new}/r = 1 - \sin(\theta) \exp(-r^2/2\sigma^2) \) in 3D. The warped radius, \( r_{new} \), is approximately equal to the unwarped radius, \( r \), for angles, \( \theta \) close to 0\(^\circ\) or 180\(^\circ\) and for large \( r \) values. The warped radius, \( r_{new} \), is less than the un-warped radius, \( r \), for angles, \( \theta \) close to 90\(^\circ\) and small \( r \) values, therefore, causing a stretch. The limit between small and large \( r \) values is set by \( \sigma \) .................. 18
2.6 Compressed prostate boundary (solid line) and rough estimate of un-compressed prostate boundary (dashed line) with \( v \)-components of the initialization points on the \( v \)-axis ................................................................. 19
2.7 For every point on the final ellipse fit (dotted line), we find $r_{\text{new}}$ and $\theta$, and then use the Newton-Raphson method to find the $r$ value that will unwrap that point 21

2.8 (a) Image and contour before reverse warp, and (b) image and contour after reverse warp 22

2.9 Tracking the prostate boundary using Star Kalman/PDA and a seed point 27

2.10 Pixel intensities along the ray shown in figure (a) are recorded and shown in the first subplot in figure (b). The gradients of the pixel intensities are shown in the second subplot and the 5 maximum gradient values are shown in the third subplot in figure (b) 28

2.11 $\hat{z}(k - 1|k - 1)$ on ray $k - 1$ is mapped to $\hat{z}(k|k)$ on ray $k$ using the state space model. Five maxima are chosen on ray $k$ to find $\hat{z}(k|k)$ 32

2.12 Comparison of edge detection results using one mode or two mode IMMs (edge detection is guided along an ellipse (thin solid line) and results are shown for the Circle model (thick dotted line) and Ellipse model (thick solid line)) 42

2.13 Comparison of edge detection results using the Circle model (thick dotted contour) and the Ellipse model (thick solid contour) for four different prostate contours 42

3.1 The radial differences (at angles $\theta_i$) between the manual (dotted line) and semi-automatic (solid line) contour, are used in validating the segmentation algorithm 45

3.2 Pixel by pixel comparison of manual vs semi-automatic contours 46

3.3 Comparison between manual (dotted line) and semi-automatic (solid line) segmentation 48

4.1 Eight steps of the 3D semi-automatic prostate segmentation algorithm 51

5.1 Each of the three observers manually segmented and initialized the initial 10 cases. Two weeks later, each observer manually segmented and initialized 5 of the original 10 cases 58

5.2 A guideline for choosing the location, order and number of initialization points. Points should be chosen somewhere within the grey boxes which are located at the center of the TRUS axis, center of the prostate and right most, bottom most and top most ends of the prostate boundary 60
5.3 Finding points on the secondary curve (*x-marks*) that correspond to points on the reference curve (*o-marks*). The image to right shows the final result in which we have two different curves whose points align radially making radial distance based analysis possible. .............................................................. 61

5.4 A 3D comparison of semi-automatic segmentation (*astrix marks*) to manual segmentation (*circle marks*). .............................................................. 62

5.5 Comparing semi-automatic contour (*solid line*) to the manual contour (*dotted line*) for the entire prostate volume. .............................................................. 63

5.6 Finding the average manual contour (*solid line*) from three manual contours by three different observers (*square marks, x-marks and o-marks*). .............................................................. 64

5.7 Comparing each computer generated contour to the gold standard on a point-by-point basis to evaluate the segmentation algorithm (all units are millimeters). .............................. 66

5.8 Using pearson correlations to compare computer segmentation (initialization by observer 1) to the gold standard. .............................................................. 67

5.9 Finding the inter-observer variability for case c, slice s, point p where $rrm_i$ is $Manual_i(c, s, p)$. Manual segmentation by observers 1, 2 and 3 are represented by (*square marks*), (*o-marks*) and (*x-marks*) respectively. .............................................................. 68

5.10 Degree of inter-observer variability at the base slice of case 1. Manual segmentations by observer 1 (*solid line*), observer 2 (*dashed line*) and observer 3 (*dotted line*). .............................. 69

5.11 Notched box plots comparing the difference between $comp_1$ and $GS$, $comp_2$ and $GS$, $comp_3$ and $GS$ to IOV for different regions of the prostate. .............................................................. 71

5.12 Gold Standard (*dashed line*) vs semi-automatic (*solid line*) segmentation results for observer 3, case 1. (Sensitivity=99.6%, Accuracy=75.6%) .............................................................. 76

5.13 Gold Standard (*dashed line*) vs semi-automatic (*solid line*) segmentation results for observer 3, case 5. In this case sensitivity=95.7% and Accuracy=90.28%. .............................................................. 77

5.14 Gold Standard (*dashed line*) vs semi-automatic (*solid line*) segmentation results for observer 2, case 4. In this case sensitivity=86.95% and Accuracy=84.04%. .............................................................. 77

A.1 Three reference frames for describing an ellipse .............................................................. 90

A.2 Convention for reference frames in 3D .............................................................. 91

A.3 Moving between reference frames in 3D .............................................................. 91
B.1 Two ellipses with different parameters *(left)* can give the same final ellipse shape *(right)* ................................................................. 95
B.2 Correcting the order of the basis vectors in 3D .................................................. 105
D.1 The dashed lines represent the level curves of the function we are trying to optimize. The solid line represents the constraint. The arrows point in the direction of maximum gradient .................................................. 112
E.1 Warping function as a function of \( r \) only with \( \theta = 90^\circ \) and \( \sigma = 40 \). Newton's method is used to find the \( r(i) \) value corresponding to \( r_{\text{new}}(i) \) .................................................. 115
F.1 One iteration of the Kalman Filter [5] .................................................. 119
F.2 One iteration of the Extended Kalman Filter [5]. The bold boxes show the difference between the EKF and KF .................................................. 120
F.3 One cycle of the Interacting Multiple Mode Estimator .................................................. 123
F.4 Tracking aircraft motion using a radar and PDA theory .................................................. 124
G.1 Comparing semi-automatic *(solid line)* to manual segmentation *(dotted line)* for Observer 1, Case 1, initial 10 cases. .................................................. 126
G.2 Comparing semi-automatic *(solid line)* to manual segmentation *(dotted line)* for Observer 1, Case 2, initial 10 cases. .................................................. 126
G.3 Comparing semi-automatic *(solid line)* to manual segmentation *(dotted line)* for Observer 1, Case 3, initial 10 cases. .................................................. 127
G.4 Comparing semi-automatic *(solid line)* to manual segmentation *(dotted line)* for Observer 1, Case 4, initial 10 cases. .................................................. 127
G.5 Comparing semi-automatic *(solid line)* to manual segmentation *(dotted line)* for Observer 1, Case 5, initial 10 cases. .................................................. 128
G.6 Comparing semi-automatic *(solid line)* to manual segmentation *(dotted line)* for Observer 1, Case 6, initial 10 cases. .................................................. 128
G.7 Comparing semi-automatic *(solid line)* to manual segmentation *(dotted line)* for Observer 1, Case 7, initial 10 cases. .................................................. 129
G.8 Comparing semi-automatic *(solid line)* to manual segmentation *(dotted line)* for Observer 1, Case 8, initial 10 cases. .................................................. 129
LIST OF FIGURES

G.9 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 1, Case 9, initial 10 cases. ................................. 130
G.10 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 1, Case 10, initial 10 cases. ................................. 130
G.11 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 1, initial 10 cases. ................................. 131
G.12 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 2, initial 10 cases. ................................. 131
G.13 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 3, initial 10 cases. ................................. 132
G.14 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 4, initial 10 cases. ................................. 132
G.15 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 5, initial 10 cases. ................................. 133
G.16 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 6, initial 10 cases. ................................. 133
G.17 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 7, initial 10 cases. ................................. 134
G.18 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 8, initial 10 cases. ................................. 134
G.19 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 9, initial 10 cases. ................................. 135
G.20 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 10, initial 10 cases. ................................. 135
G.21 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 1, initial 10 cases. ................................. 136
G.22 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 2, initial 10 cases. ................................. 136
G.23 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 3, initial 10 cases. ................................. 137
LIST OF FIGURES

G.24 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 4, initial 10 cases. .................................................. 137
G.25 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 5, initial 10 cases. .................................................. 138
G.26 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 6, initial 10 cases. .................................................. 138
G.27 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 7, initial 10 cases. .................................................. 139
G.28 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 8, initial 10 cases. .................................................. 139
G.29 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 9, initial 10 cases. .................................................. 140
G.30 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 10, initial 10 cases. ................................................. 140
G.31 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 1, Case 2, repeat of 5 cases. ................................................. 141
G.32 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 1, Case 3, repeat of 5 cases. ................................................. 141
G.33 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 1, Case 6, repeat of 5 cases. ................................................. 142
G.34 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 1, Case 8, repeat of 5 cases. ................................................. 142
G.35 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 1, Case 10, repeat of 5 cases. ............................................. 143
G.36 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 2, repeat of 5 cases. ............................................. 143
G.37 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 3, repeat of 5 cases. ............................................. 144
G.38 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 6, repeat of 5 cases. ............................................. 144
LIST OF FIGURES

G.39 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 8, repeat of 5 cases. .................................................. 145
G.40 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 10, repeat of 5 cases. .................................................. 145
G.41 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 2, repeat of 5 cases. .................................................. 146
G.42 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 3, repeat of 5 cases. .................................................. 146
G.43 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 6, repeat of 5 cases. .................................................. 147
G.44 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 8, repeat of 5 cases. .................................................. 147
G.45 Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 10, repeat of 5 cases. .................................................. 148
Acknowledgements

First and foremost I would like to thank my supervisor, Dr. Tim Salcudean, for his knowledge, guidance and support throughout my thesis.

My research would not have been possible without collaboration with the BC Cancer Agency. I would like to thank Dr. William J Morris and Dr. Ingrid Spadinger for their gracious help and insight. I would also like to thank Alex Kruk and Nick Chng for manually segmenting prostate contours for my research.

I am grateful to Dr. James Varah for helping me so patiently with questions regarding ellipse/ellipsoid fitting.

I thank my labmates in the Control and Robotics group. They are my second family and some of my best friends. I don't think anyone would disagree if I said we have the best lab in ECE. In particular I'd like to thank Julian Guerrero for helping me understand Kalman Filters and Danny French for teaching me the basics of Prostate Brachytherapy.

I am ever so thankful for my family. An encouraging phone call, a surprise visit, a few lunch boxes full of food made all the difference in the world.
Chapter 1

Introduction

1.1 Prostate Cancer

The prostate is a gland of the male reproductive system responsible for producing most of the volume of seminal fluid. It is located in front of the rectum, just below the bladder and encloses the prostatic urethra (refer to Figure 1.1a). In older men the prostate is highly variable in size ranging from 15-150cm$^3$.

(a) Male anatomy

(b) Setup for prostate brachytherapy

Figure 1.1: Prostate brachytherapy
Prostate cancer occurs when normal prostate gland cells mutate into cancer cells. If discovered early, prostate cancer is often curable; however, there are very few symptoms at an early stage and tumors can remain undetected until they have spread. In late stages, prostate cancer spreads to nearby organs such as the seminal vesicles, rectum, bladder and bones or travels via the bloodstream and lymphatic system.

Prostate cancer is the most commonly diagnosed cancer in Canadian men affecting one in seven in their lifetime. In 2006, an average of 398 Canadian men will be diagnosed and 81 will die from the disease per week [3].

In recent years, increased prostate cancer awareness and improved diagnostic tools have dramatically increased the number of reported prostate cancer cases. This increase has raised a significant amount of interest in research that aims to improve the detection, treatment and survival rate for prostate cancer patients.

Treatment options for prostate cancer depend on a variety of factors including clinical stage, Gleason scores, pre-treatment PSA levels etc. Some prostate cancers are indolent in their behaviour and 'active surveillance', which refers to regular monitoring without any invasive treatment, is a good option. One common treatment option for early stage, clinically organ confined prostate cancer is 'radical prostatectomy' in which the prostate is removed surgically. Another treatment option is 'radiation therapy' which uses x-rays to irradiate the cancer cells. There are two types of radiation therapy: i) 'External Beam Radiation Therapy (EBRT)' that directs high-energy x-ray beams on the prostate and surrounding tissues and ii) 'Prostate Brachytherapy' in which radiation sources are placed in and around the prostate to irradiate the cancer cells from within. When cancer has spread beyond the prostate it is incurable and the standard treatment is 'hormonal therapy' in which medication or surgical orchietomy is used to block the pathways that produce dihydrotestosterone (DHT), a hormone necessary for the growth of prostate cancer cells. When the disease becomes refractory to hormone therapy, 'Chemotherapy' may be used to postpone disease progression. Sometimes more than one treatment modality is required for optimal success. Treatment complications may include loss of urinary continence, erectile dysfunction and rectal injury.
1.2 Prostate Brachytherapy

1.2.1 Why Prostate Brachytherapy

To be specific, there are two types of prostate brachytherapy: i) High Dose Rate (HDR), and ii) Low Dose Rate (LDR). In HDR brachytherapy, small catheters are inserted into the prostate through the perineum through which radioactive doses are delivered to the prostate. This technique is also called temporary brachytherapy because no radioactive material is left inside the prostate at the end of the procedure. In contrast, LDR brachytherapy is a permanent procedure in which small radioactive capsules are permanently implanted into the prostate. The technique of interest for this thesis is the LDR option and its steps are explained in Section 1.2.2. From this point on a reference to prostate brachytherapy implies LDR brachytherapy.

Of all the treatment options available for locally confined prostate cancer, prostate brachytherapy is the most efficient in terms of recovery time and procedure outcome.

Brachytherapy has several advantages over prostatectomy. It is a minimally invasive procedure (no incisions and very minimal bleeding) that typically takes less than 1 hour to perform in contrast to prostatectomy which could take 2-3 hours. Brachytherapy is done in an out-patient setting meaning that the patient is released from the hospital hours after the procedure and, despite discomfort, is able to walk, sit and resume work immediately. Prostatectomy requires overnight stay at the hospital followed by 8-12 weeks of recovery at home before full activities can be resumed.

Brachytherapy also has several advantages over EBRT. Brachytherapy allows for 50-100% greater local dose intensity than EBRT, and image guidance ensures more accurate placement and less damage to surrounding tissue than with external beam radiation. [26] As well, brachytherapy implants are permanent and the procedure is carried out only once whereas external beam requires daily treatments 5 days per week for 6-8 weeks.

Overall, brachytherapy is an attractive option for patients who do not want to undergo major surgery or endure daily radiation treatment for seven weeks. New developments in image guidance and visualization technology, treatment planning software and improvements in radioisotopes for dose delivery have renewed interest and therefore research in prostate brachytherapy. For this reason, the number of brachytherapy procedures carried out in North America is expected to increase significantly over the next few years. At the BC Cancer Agency (BCCA) there had been 1800 brachytherapy implants between 1998 and 2006 and 300 implants in 2006 alone.
1.2.2 Prostate Brachytherapy Procedure at the BCCA

At the BCCA, a prostate brachytherapy patient undergoes a pre-operative scan approximately 4-8 weeks prior to the actual procedure. During this scan, a TRUS probe is inserted into the rectum, along the long axis of the prostate (coincident with the long axis of the body), and adjusted so that the transverse TRUS images of the prostate are symmetrical about the D column (center column) of the needle guide template (reason for ensuring symmetry is explained in Section 1.4.1). Approximately 9-14 transverse images of the prostate are taken at 5mm increments. The images are referred to as BASE+0.0, BASE+0.5, BASE+1.0 and so on where the 0.5 increment refers to the 5mm increment in slice spacing. Although it seems counter-intuitive, the real base is on slice BASE+0.5 and BASE+0.0 is 5mm superior to the real base (this image is always taken). Since different prostates are of different sizes the apex can be anywhere between BASE+3.5 to BASE+5.5. As with the base, one image is always taken 5mm inferior to the apex of the prostate.

The images are imported into a planning software package by Varian called Vari-Seed. In this software, the oncologists manually outline or segment approximately 70% of the transverse slices. Vari-Seed interpolates in between points on each slice and interpolates contours in between slices to create a 3D representation of the prostate size, shape, location and orientation. This 3D segmented volume is called the Clinical Target Volume (CTV). The oncologists then assign extra margins to the CTV to create a Planning Target Volume (PTV). The PTV is used to design a seed distribution plan to be applied to the patient’s prostate on the day of the procedure.

Four to eight weeks later, during the actual brachytherapy procedure, approximately 60-150 radioactive seeds (Iodine 125 or Palladium 103) are guided trans-perineally, through brachytherapy needles, to specific xyz coordinates within the prostate according to the dosimetric plan. Refer to Figure 1.1b for setup of the brachytherapy procedure and orientation of the xyz coordinate system. The needle guide template helps the oncologist localize the needle in the xy plane, the TRUS images give the z coordinate as well as confirm the xy coordinate and Fluoroscopic images re-confirm the xz coordinates of seeds.

On the day of the implant, the patient has a post-operative CT scan. The CT images are segmented manually to locate the seeds. Knowing the seed locations and the size, shape and orientation of the prostate the radiation dose delivered to the tissue can be calculated to evaluate how closely the final dosimetry matches the pre-implant plan. In a successful implant, the 100%
isodose (lines of equal dose distribution) should cover more than 85% of the CT prostate volume.

1.3 Thesis Motivation: Automatic Prostate Segmentation

1.3.1 Why Segment the Prostate Automatically

Currently oncologists manually select 25-35 points around the prostate boundary and then adjust these points to ensure symmetry and smoothness. When all the slices have been segmented a 3D volume is created. If the smoothness of the resulting 3D volume is un-satisfactory, the oncologists re-visit some slices and re-segment the boundary. Manual prostate segmentation has a very steep learning curve and the times for segmenting a whole volume can range from 20 minutes (non-expert) to 5 minutes (expert). It is clear to see that manual prostate segmentation from TRUS images is a time consuming, and laborious process and it would be ideal if a computer algorithm could segment the prostate boundary automatically.

Currently oncologists only segment the prostate from TRUS images pre-operatively; however, intra-operative prostate segmentation can provide many benefits. For example, it is possible that the prostate shifted and changed shape in the two weeks between the pre-operative scan and the brachytherapy procedure implying that the dosimetric plan (originally designed for the pre-operative scan) is no longer valid. As well it is possible for the prostate to rotate and deform due to needle insertions, and increases in volume due to swelling during the brachytherapy procedure making the dosimetric plan even more inaccurate. Currently, intra-operative updates to the dosimetry to compensate for such deformation are not feasible because it is impossible to carry out intra-operative manual segmentation. Automated prostate segmentation could assist in updating the treatment plan intra-operatively, with minimal user intervention, so that the patient receives the most accurate dose distribution.

Intra-operative prostate segmentation could also provide a 3D, intra-operative, visualization tool. At BCCA, the more experienced oncologists compensate for the deformations discussed previously by inserting the seeds into modified grid locations so that, once the needle is removed, the intended seed drop off location is reached. In order to know how to modify the dosimetric plan, the oncologists must have i) a very good 3D image of the prostate and its surrounding anatomy in their mind, and ii) an understanding of where the prostate is anchored and how it will rotate when punctured at a given location. For this reason, training residents in the 'art' of prostate
brachytherapy is very difficult. It would be desirable to create a software package with an easy to use Guided User Interface (GUI) that integrates the automatic 3D segmentation algorithm, along with an automatic dosimetric planning algorithm and a dynamic model of the prostate's motion. This package would enable real-time intra-operative prostate segmentation which would result in: i) intra-operative updates to the dosimetry, ii) a training tool for residents to help with 3D visualization and motion dynamics of the prostate.

1.3.2 Difficulties in Automatic Prostate Segmentation

The success of automatic prostate segmentation to date has been very limited [2] due to the nature of TRUS images of the prostate. In these images the prostate boundary is unclear and the images suffer from speckle because the acoustic properties of tissues surrounding the prostate are very similar. Figure 1.2 shows transverse ultrasound images being taken of the prostate (gray lines). At the base of the prostate, the boundary becomes blurred with the bladder neck because, i) their acoustic properties are similar and ii) beam widening causes more information to be reflected off of the bladder and thereby producing poor images of the prostate base. The best images are obtained at mid gland where the prostate is free of surrounding organs. At its apex, the prostate merges with the pelvic floor and there, once again, the boundary of the prostate is unclear. In older men 'calcium deposits' surrounding the prostate set up another strong reflecting boundary for the ultrasound waves and cast shadows behind the deposit in the TRUS images. It is common for oncologists to rely on experience and knowledge of physiology to carry out segmentation when the prostate boundary is unclear (which occurs frequently).

At the end of the procedure the prostate is swollen, contains up to 150 seeds (which act as reflectors for the ultrasound waves), has numerous blood trails and an onslaught of platelets and scar tissue attempting to stop the bleeding within the prostate. Therefore, the post-operative TRUS images of the prostate are much worse and the prostate boundary is nearly indistinguishable (Refer to Figure 1.3). In this thesis we focus only on pre-operative TRUS images.

1.4 Thesis Objective: Semi-Automatic 3D Prostate Segmentation

1.4.1 Segmentation Criteria at BCCA

In this thesis we emphasize the following requirements for segmenting the prostate:
1.4 Thesis Objective: Semi-Automatic 3D Prostate Segmentation

Figure 1.2: Segmentation difficulties due to anatomy

(a) Pre-operative prostate TRUS image  (b) Post-operative prostate TRUS image

Figure 1.3: Comparing TRUS image quality before and after operation

1. Smooth, continuous contours with no sharp edges.

(a) The radiation pattern of the radioactive seeds is smooth, symmetrical and spherical. The superposition of the radiation dose from multiple seeds can only create smooth dose patterns and it is difficult to create dosimetry plans for jagged or discontinuous edges. For this reason, and the fact that the prostate has an approximately ellipsoidal shape, we require the prostate contour to be smooth and continuous.
2. Convex shape with no concavities or hourglass shapes.

(a) A dosimetric plan for concave shapes is very hard to obtain. As well, the natural shape of the prostate is convex and physiologically an hourglass shaped prostate is not possible.

At some centers, symmetry is required about the D column on transverse images. The reason for the symmetry is that there are an infinite number of asymmetric images that an operator can take. If such asymmetric images were taken pre-operatively, the oncologists would have difficulty registering the randomly asymmetric pre-operative images to the intra-operative images during the brachytherapy procedure. In addition to reducing intra-operative image registration time, symmetric images would reduce error probability in loading and delivering needles. They would also make treatment planning faster since only half of the prostate needs to be planned on.

In some cases, our segmentation algorithm returns contours that extend further on one side of the D column than the other. If, for example, the contour on the right side of the D column extended further than the left side oncologists at BCCA would extend the contour on the left side to match the right side. Although it is not currently done, we could easily adopt a similar technique in our algorithm to ensure that all segmented prostate contours are symmetric.

1.4.2 Literature Review

1.4.2.1 2D Automatic Prostate Segmentation from TRUS images

The 2D segmentation of the prostate from TRUS images is a challenging problem and has received significant attention in the medical imaging field. Strictly edge-based algorithms such as [1, 2, 21, 25] result in poor segmentation due to speckle noise and poor contrast in TRUS images. One such edge-detection algorithm, introduced by Abolmaesumi et al. [2], uses a spatial Kalman filter along with Interacting Multiple Modes Probabilistic Data Association (IMMPDA) filters to detect the closed contour of the prostate. Although the IMMPDA edge-detector is capable of creating continuous contours, it cannot satisfy the smoothness and symmetry requirements on its own. In order to overcome these issues, deformable models such as those outlined in [20, 22] were proposed. These techniques worked moderately well; however, the segmented shapes were not constrained and hourglass contours, for example, were possible. In order to constrain the prostate contours, groups such as Shen et al. [29] and Gong et al. [16] incorporated a priori knowledge of the possible prostate
1.4 Thesis Objective: Semi-Automatic 3D Prostate Segmentation

shapes into their algorithm. The automatic segmentation procedure of Shen et al. [29] combined a statistical shape model of the prostate, obtained from ten training samples, with rotation invariant Gabor features to initialize the prostate shape. This shape was then deformed in an adaptive, hierarchical manner until the technique converged to the final model. Unfortunately this technique is very expensive computationally, requiring over a minute to segment a single image. Gong et al. [16] modeled the prostate shape by fitting superellipses to 594 manually segmented prostate contours. These a priori prostate models, along with an edge map of the TRUS image, were then used in a Bayesian segmentation algorithm. The edge map was created by first applying a sticks filter, then an anisotropic diffusion filter and finally a global Canny edge-detector. In this technique, the pre-processing steps are computationally expensive and the global Canny edge-detector finds fragmented, discontinuous edges in the entire image.

1.4.2.2 3D Automatic Prostate Segmentation from TRUS images

The 3D segmentation problem is inherently much more difficult (yet more practically useful) than its 2D counterpart, particularly due to poor boundary contrast at the base and apex. For this reason, 3D prostate segmentation has not received as much attention as 2D segmentation and most of the literature presently available for 3D segmentation are extensions of 2D methods. Richard et al. [27] presented a pixel classifier segmentation technique that uses four texture features to assign an 'energy' to each pixel in the image. In the paper they acknowledge that the effect of using texture information is marginal and their algorithm is very computationally extensive. Shao et al. [28] begin with a small sphere (no manual initialization) which is deformed using the level set method. No times or quantitative comparisons are reported and the results seem to suffer from 'boundary leakage' and are sensitive to a parameter chosen experimentally by the user. Ghanei et al. [15] use heavy initialization (5-8 points on 40-70% of the 2D slices) to create an initial ellipsoid mesh. The coordinates of each mesh point are updated iteratively according to i) external forces that use edge detection to propel the point towards the edges and ii) internal forces which enforce smoothness. The algorithm was tested on 10 prostate volumes and took approximately 30s to segment a single volume.

The 2D segmentation algorithm of Ladak et al. [22] has been extended to 3D in [11,12,18,33]. In these papers a slice in which the boundary is visible (i.e., midgland) is manually initialized with 4-8 points to create an initial contour and this contour is refined and updated using the Discrete
1.4 Thesis Objective: Semi-Automatic 3D Prostate Segmentation

Dynamic Contour (DDC). The contours are propagated iteratively through the slices, each time using the previous contour as the initialization and applying the DDC. This technique is prone to propagated segmentation error (especially at the base and apex) and to solve this problem a continuity constraint, using a zero-order AR model [12] is introduced. The speed of this algorithm is less than 10 seconds. As in the 2D case, the contour obtained with the DDC algorithm does not incorporate a priori shape knowledge which results in wandering of the prostate boundary. To correct for this, manual editing tools were introduced but up to 20% of the (parallel) slices need manual editing which adds significant time to the segmentation algorithm. The solution, as in the 2D case, is to introduce a priori shape information and this is done in the 3D extension of the work by Shen et al. [35–37]. In this algorithm 3D texture features obtained from two banks of orthogonal 2D Gabor filters are used to train the Kernal Support Vector Machine (KSVM) which differentiates the prostate from its surrounding tissue. Then a statistical shape model is deformed, as was done in 2D, to find the prostate volume using texture and shape matching. Although the results of this algorithm are quite accurate, it has no real-time capabilities due to its computationally expensive nature (requires approx 3 minutes to segment a single volume). Refer to Table 1.1 for a comparison of the best 2D and 3D segmentation algorithms available to the algorithm presented in this thesis.

1.4.3 General Overview of Approach

The goal of this thesis was to create an algorithm with low computational complexity that satisfies the smoothness and symmetry constraints and is driven by the clinical understanding of prostate brachytherapy. The proposed algorithm requires no computationally expensive pre-processing and is insensitive to the initialization points. The algorithm makes use of a novel image warping technique, ellipse and ellipsoid fitting and locally applied IMMPDA edge-detection. The ellipse fitting procedure outlined in [13,32], and ellipsoid fitting problem outlined in [24] solve a convex problem, are computationally inexpensive, robust and always return an ellipsoidal solution. The IMMPDA edge-detector is applied locally meaning that it finds continuous edges only along the boundary of the prostate instead of along the entire image.
Table 1.1: Overview of best 2D and 3D segmentation algorithms available

<table>
<thead>
<tr>
<th></th>
<th>Fenster et al</th>
<th>Zhan &amp; Shen</th>
<th>Ghanei et al</th>
<th>Gong et al</th>
<th>Badiei et al</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>[9,11,19,23,33]</td>
<td>[35–37]</td>
<td>[15]</td>
<td>[15]</td>
<td>[4]</td>
</tr>
<tr>
<td>Pre-processing</td>
<td>N/A</td>
<td>Extensive</td>
<td>N/A</td>
<td>Extensive</td>
<td>Minimal</td>
</tr>
<tr>
<td>Manual Initialization</td>
<td>4-8 pnts</td>
<td>Non</td>
<td>40+ pnts</td>
<td>2-10 pnts</td>
<td>5 pnts</td>
</tr>
<tr>
<td>A priori shape knowledge</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Contours are always:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symmetrical</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Smooth, Continuous</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No hourglass shapes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Clinical Study:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Volumes</td>
<td>12</td>
<td>6</td>
<td>10</td>
<td>125 2D slices</td>
<td>10</td>
</tr>
<tr>
<td>Number of observers</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Image Resolution</td>
<td>Ideal</td>
<td>Average</td>
<td>Average</td>
<td>Poor (realistic)</td>
<td>Poor (realistic)</td>
</tr>
<tr>
<td>Accuracy</td>
<td>Excellent</td>
<td>Excellent</td>
<td>Excellent</td>
<td>Excellent</td>
<td>Excellent</td>
</tr>
<tr>
<td>Reproducibility</td>
<td>N/A</td>
<td>N/A</td>
<td>Good</td>
<td>N/A</td>
<td>Excellent</td>
</tr>
<tr>
<td>Speed</td>
<td>10 s (C++)</td>
<td>180 s (C++)</td>
<td>30 s (C++)</td>
<td>5 s (C++)</td>
<td>8 s (Matlab)</td>
</tr>
</tbody>
</table>

Pros and Cons

- Fenster: Images used seem to be of better quality than typically encountered. The DDC does not incorporate a priori shape knowledge which is necessary when dealing with non-ideal images. Approx 20% of parallel slices require manual editing which adds approx 60 s to segmentation time.
- Zhan: Very computationally expensive
- Ghanei: Too many initialization points
- Gong: Algorithm is for 2D images only. If extended to 3D, the algorithm would be slow (approx 60 s). In-depth clinical studies showed excellent results. Clinical integration is feasible.
- Badiei: Excellent performance despite low resolution images. Can be easily integrated into clinical setting. Extensive 3D clinical study was carried out with excellent results. Upon optimization and transfer to C++ this will be the fastest available algorithm. Algorithm performance at the base and speed of image warping could be improved.
1.5 Thesis Overview

The purpose of this thesis was to design and test an algorithm for 3D prostate segmentation from 2D TRUS images. The first step was to create a 2D semi-automatic segmentation algorithm and test and validate it. The second step was naturally to extend this 2D algorithm to 3D and again test and validate it. The proceeding 5 chapters are broken down as follows:

Chapter 2 - Algorithm: 2D Segmentation. This chapter presents the 6-step 2D segmentation algorithm: i) initialization, ii) forward warp, iii) initial ellipse fit, iv) Interacting Multiple Modes Probabilistic Data Association (IMMPDA) edge detector, v) final ellipse fit and vi) reverse warp. The details of how the initialization points are chosen, how the image and initialization points are forward and reverse warped, how an ellipse is fit to a set of measurement points and how the edge detector works are outlined in detail in this chapter.

Chapter 3 - Study Results and Discussion: 2D Segmentation. A study carried out on 17 midgland transverse TRUS images of the prostate is presented to evaluate the 2D semi-automatic segmentation algorithm using manual segmentation as the gold standard. Results and simulation times are presented and discussed.

Chapter 4 - Algorithm: 3D Segmentation. This chapter outlines the eight steps of the 3D segmentation algorithm: i) initialization on midgland only, ii) forward warp of all slices, iii) 2D segmentation only on midgland, iv) 2D segmentation on remaining slices with initialization propagating from previous slice's contour, v) initial ellipsoid fit, vi) contouring base and apex, vii) final ellipsoid fit, and viii) reverse warp. The details of this 8-step algorithm and how to fit an ellipsoid to a set of measurement points are presented in this chapter.

Chapter 5 - Study Results and Discussion: 3D Segmentation. A study was carried using 10 prostate volumes (each with 10-14 slices) and 3 observers (2 non-expert and 1 expert). The difference between computer segmentation and the gold standard (average manual segmentation) was compared to inter-observer variability. The repeatability of the computer algorithm was compared to the intra-observer variability. Various distance and volume based metrics were used to evaluate the segmentation algorithm. Results and simulation times are presented and discussed.

Chapter 6 - Conclusion and Future Work. An overview of contributions and future work are summarized in this chapter.
Chapter 2

Algorithm: 2D Segmentation

2.1 Introduction to the 6-step Algorithm

In 2D semi-automatic prostate segmentation, the goal is to create an algorithm that finds the prostate boundary in TRUS images given a few initialization points. The proposed computer algorithm has six steps. In the first step, the user selects 5 initialization points (initialization outlined in Section 2.2) and in the second step, the image along with the initialization points are warped (warping algorithm described in Section 2.3). The warped prostate should look like an ellipse so that, in the third step, we can make use of the extensive literature on ellipse fitting to fit an initial ellipse to the warped initialization points. The details of how to fit an ellipse to a set of points are outlined in Section 2.4. In step four, this initial ellipse fit is used to confine and locally guide the IMMPDA edge-detector (IMMPDA described in Section 2.5) along the boundary of the warped prostate to prevent the trajectory from wandering. In step five, a second ellipse is fit to the IMMPDA edge points, which provides a more accurate estimate of the warped prostate boundary. Finally, in the last step, inverse warping is applied to the second elliptical fit to obtain the final segmented prostate contour (reverse warping discussed in Section 2.3). The only preprocessing is a computationally cheap median filter ($5 \times 5$ kernel) applied locally along the path of the edge-detector. The 6-step segmentation procedure outlined above is presented in Figure 2.1.
Chapter 2

Algorithm: 2D Segmentation

2.1 Introduction to the 6-step Algorithm

In 2D semi-automatic prostate segmentation, the goal is to create an algorithm that finds the prostate boundary in TRUS images given a few initialization points. The proposed computer algorithm has six steps. In the first step, the user selects 5 initialization points (initialization outlined in Section 2.2) and in the second step, the image along with the initialization points are warped (warping algorithm described in Section 2.3). The warped prostate should look like an ellipse so that, in the third step, we can make use of the extensive literature on ellipse fitting to fit an initial ellipse to the warped initialization points. The details of how to fit an ellipse to a set of points are outlined in Section 2.4. In step four, this initial ellipse fit is used to confine and locally guide the IMMPDA edge-detector (IMMPDA described in Section 2.5) along the boundary of the warped prostate to prevent the trajectory from wandering. In step five, a second ellipse is fit to the IMMPDA edge points, which provides a more accurate estimate of the warped prostate boundary. Finally, in the last step, inverse warping is applied to the second elliptical fit to obtain the final segmented prostate contour (reverse warping discussed in Section 2.3). The only pre-processing is a computationally cheap median filter (5 × 5 kernel) applied locally along the path of the edge-detector. The 6-step segmentation procedure outlined above is presented in Figure 2.1.
2.2 Initialization

In the first step of the 6-step segmentation algorithm, the user is presented with the TRUS image and asked to initialize the image. The 5 initialization points must be selected at the following locations: i) the center of the TRUS probe, ii) the center of the prostate and iii) 3 points along the boundary of the prostate at the right most edge, bottom most edge and top most edge. Figure 2.3 a) shows an example of an initialized TRUS image.

Using the 3 initialization points around the boundary of the prostate, 3 more points are interpolated at the left most edge of the prostate and midway between the left and top and midway between the right and top initialization points (as shown with \(\times\)-marks in Figure 2.1). The interpolation is done by making use of the symmetry of the prostate shape. In step 2 of the 6-step algorithm these initialization points are warped and in step 3 an initial ellipse is fit to them. The warping algorithm is described in detail in the following section.
2.3 Forward and Reverse Warp

As mentioned previously, the prostate is a walnut shaped organ with a roughly elliptical cross section. During image acquisition the TRUS probe is pressed anteriorly against the rectum wall which causes the prostate to look deformed in 2D transverse image. The goal of the forward warping algorithm is to undo the compression effect caused by the TRUS probe so that the 2D prostate boundary looks elliptical. The goal of the reverse warping algorithm is to bring the prostate contour and prostate image back into the original (compressed) form.

2.3.1 Forward Warp

We assign a polar coordinate system to the 2D TRUS images of the prostate with origin on the TRUS probe axis, angle $\theta = 0^\circ$ along the horizontal, and $r$ extending radially. The assumption in our warping algorithm is that the TRUS probe causes radial compression; therefore, all of the warping will be done along the radial direction. Referring to Figure 2.2, our goal is to take every point $(u, v)$ at angle $\theta$ and radius $r$, in the original image $I_o$, and place it into a new warped position $(x, y)$ at angle $\theta$ and radius $r_{new}$, to create the warped image, $I_{fwd}$. The pseudo-code for doing this is as follows:

\[
\begin{align*}
\text{for } \theta = 0 : \pi \\
& \text{for } r = R_{min} : R_{max} \\
& \quad u = u_o + r \cos(\theta) \\
& \quad v = v_o - r \sin(\theta) \\
& \quad r_{new} = f(r, \theta) \\
& \quad x = u_o + r_{new} \cos(\theta) \\
& \quad y = v_o - r_{new} \sin(\theta) \\
& \quad I_{fwd}(y, x) = I_o(v, u)
\end{align*}
\]
Figure 2.2: Place every point in the original image (left) into a new position to create the warped image (right) using a radial warping function to get $r_{new}$ from $r$ and $\theta$

As you can see from the pseudo-code above, we need to come up with a function, $r_{new}$, to describe the radial compression as a function of $\theta$ and $r$. This function is derived in the next subsection. Refer to Figure 2.3 to see the effects of warping on an image and its initialization points.

Figure 2.3: Comparing the original image and original initialization points (a) to the warped image and warped initialization points (b)
2.3 Forward and Reverse Warp

2.3.2 Warping Function

We can undo the effect of the anterior compression caused by the TRUS probe by: i) stretching the image maximally around $\theta = 90^\circ$ and minimally around $\theta = 0^\circ$ and $\theta = 180^\circ$, and ii) stretching the image maximally for small $r$ values and minimally for large $r$ values. These requirements suggest a $\sin(\theta)$ dependence in the angular direction and a Gaussian dependence in the radial direction. The resulting warping function is:

$$r_{\text{new}} = r - r \sin(\theta) \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad (2.1)$$

where $r_{\text{new}}$ represents the new warped radius, $r$ is the un-warped radius, $\theta$ ranges from $0^\circ$ to $180^\circ$ and $\sigma$ is a variable that represents the amount of stretch in the radial direction. Refer to Figure 2.4 and 2.5 for 2D and 3D graphs of the warping function. Small $\sigma$ values indicate less radial stretch and can be used for prostate shapes that are already elliptical and/or have experienced little deformation. Larger $\sigma$ values indicate greater radial stretch and can be used for prostate shapes that have experienced more deformation by the TRUS probe. Using the initialization points for each TRUS image, we can design a small algorithm that will choose the optimal $\sigma$ automatically. This is discussed in detail in Section 2.3.2.1.

![Figure 2.4: The $\theta$ component (top) and $r$ component (bottom) of the warping function in 2D](image)
2.3 Forward and Reverse Warp

Figure 2.5: The warping function: $r_{\text{new}}/r = 1 - \sin(\theta) \exp\left(-r^2/2\sigma^2\right)$ in 3D. The warped radius, $r_{\text{new}}$, is approximately equal to the unwarped radius, $r$, for angles, $\theta$ close to $0^\circ$ or $180^\circ$ and for large $r$ values. The warped radius, $r_{\text{new}}$, is less than the un-warped radius, $r$, for angles, $\theta$ close to $90^\circ$ and small $r$ values, therefore, causing a stretch. The limit between small and large $r$ values is set by $\sigma$

2.3.2.1 Choosing the $\sigma$ (Stretch Factor) Automatically

In order to find the stretch factor automatically, we first solve equation 2.1 for $\sigma$ as follows:

$$\sigma = \sqrt{-\frac{r^2}{2\ln\left(\frac{1-r_{\text{new}}/r}{\sin(\theta)}\right)}}$$  (2.2)

To find $\sigma$ we need an $r$ value and an $r_{\text{new}}$ value that are unique to the image we are trying to warp. These values can be found from the initialization points. Referring to Figure 2.6 and recalling from the previous section, the observers place their initialization points on the prostate image in the following order: i) $(u_T, v_T)$ at the TRUS center, ii) $(u_S, v_S)$ at the prostate center, iii) $(u_1, v_1)$ on the right most point of the prostate boundary, iv) $(u_2, v_2)$ on the bottom most point of the prostate boundary, and v) $(u_5, v_5)$ on the top most point of the prostate boundary. The point $(u_3, v_3)$ is obtained by reflecting $(u_1, v_1)$ about the sagittal plane (which is also the D column on the need needle guide template) and $(u_4, v_4)$, which is the mirror image of $(u_6, v_6)$ is placed on the prostate boundary half way between $(u_3, v_3)$ and $(u_5, v_5)$. In this manner the initialization points are symmetric about the prostate's sagittal plane.
In Figure 2.6, the dotted ellipse is the approximate shape of the prostate when the TRUS probe is not inserted into the rectum. During radial compression, the maximum stretch occurs along $\theta = 90^\circ$ where the point $(u'_2, v'_2)$ gets pushed up to $(u_2, v_2)$; therefore, we assume that if we find $\sigma$ there then everything else will follow. When the TRUS probe is not in the rectum, we can assume the prostate is symmetrical so that:

$$v_S - v_5 = v'_2 - v_S$$  \hspace{1cm} (2.3)

Solving this equation for $v'_2$ we get:

$$v'_2 = 2v_S - v_5$$  \hspace{1cm} (2.4)

In the polar coordinate system of Figure 2.6, with origin at $(u_T, v_T)$, the radial distance from the TRUS center to the compressed prostate bottom is:

$$r = v_T - v_2$$  \hspace{1cm} (2.5)
and the radial distance from the TRUS center to the uncompressed prostate bottom is:

\[ r_{new} = v_T - v'_2 \]  

(2.6)

Substituting equation 2.4 into the above equation we get:

\[ r_{new} = v_T - 2v_3 + v_5 \]  

(2.7)

Now we have both \( r \) and \( r_{new} \) in terms of the initialization points at \( \theta = 90^\circ \) and we can use equation 2.2 to find \( \sigma \):

\[ \sigma = \sqrt{\frac{-(v_T - v_2)^2}{2ln\left(\frac{2-v_2-v_5}{v_T-v_2}\right)}} \]  

(2.8)

Therefore, given any image and its initialization points, we can always find an optimal \( \sigma \) (stretch factor) for that image.

### 2.3.3 Reverse Warp

If we recall, the 6-step semi-automatic segmentation algorithm has the following steps: i) initialization, ii) forward warp, iii) initial ellipse fit, iv) IMMPDA edge detection, v) final ellipse fit and vi) reverse warp. Referring to Figure 2.7, our goal is to take every warped point on the final ellipse fit, \((x, y)\) at angle \( \theta \) and radius \( r_{new} \), and place it into a new un-warped position, \((u, v)\) at angle \( \theta \) and radius \( r \), to create the final semi-automatically segmented prostate contour. The pseudo-code for doing this is as follows:

```plaintext
for i = 1 : length(x)
% Cycle through every point on the final ellipse fit
    r_{new}(i) = sqrt((x(i) - w_T)^2 + (y(i) - v_T)^2)
    \theta(i) = atan((v_T - y(i))/(x(i) - w_T))
    r(i) = f(r_{new}(i), \theta(i), \sigma)
    u(i) = w_T + r(i) \cos(\theta(i))
```

\[ v(i) = v_T - r(i) \sin(\theta(i)) \]

end

As you can see from the pseudo-code above, for every point \((x, y)\) at angle \(\theta\) and radius \(r_{\text{new}}\) on the ellipse we need to find \(r\) to be able to un-warp that point. Since a closed form solution for \(r\) cannot be obtained from equation 2.1, the Newton-Raphson method is used as described in E. Refer to Figure 2.8 to see the effect of the reverse warping on a prostate contour in 2D.

2.4 Ellipse Fitting

2.4.1 Equation of an Ellipse

We would like to come up with a general equation, in two dimensions, that represents an ellipse that has been both translated and rotated. Before we go on we refer to reader to Appendix A where the reference frame notations that will be used throughout the rest of this section are outlined.

Let's first consider an ellipse centered at the origin, with no rotation as in Figure A.1a. There

Figure 2.7: For every point on the final ellipse fit (dotted line), we find \(r_{\text{new}}\) and \(\theta\), and then use the Newton-Raphson method to find the \(r\) value that will unwrap that point.
2.4 Ellipse Fitting

(a) (b)

Figure 2.8: (a) Image and contour before reverse warp, and (b) image and contour after reverse warp

are three ways to represent this ellipse. The parametric equation of this ellipse is:

\[
\begin{bmatrix}
5x \\
5y
\end{bmatrix} = \begin{bmatrix}
\cos(\theta_t) \\
\sin(\theta_t)
\end{bmatrix}
\]

(2.9)

where \( [5x \ 5y]^T \) are points in the final prostate frame. Here, \( a \) is the width of the ellipse along the \( 5x \) axis, \( b \) is the width of the ellipse along the \( 5y \) axis and \( \theta_t \) is a sweep variable. The canonical form of the same equation is:

\[
\left(\frac{5x}{a}\right)^2 + \left(\frac{5y}{b}\right)^2 = 1
\]

(2.10)

and the quadratic form is:

\[
P_1''5x^2 + P_3''5y^2 + P_6'' = 0
\]

(2.11)

Next, we incorporate the 2D rotation and translation outlined in Appendix A into the above 3 equations. The parametric equation becomes:
2.4 Ellipse Fitting

\[
\begin{bmatrix}
1_x \\
1_y
\end{bmatrix} = 
\begin{bmatrix}
x_o \\
y_o
\end{bmatrix} + 
\begin{bmatrix}
\cos (\theta) & -\sin (\theta) \\
\sin (\theta) & \cos (\theta)
\end{bmatrix} 
\begin{bmatrix}
a \cos (\theta_i) \\
b \sin (\theta_i)
\end{bmatrix} \tag{2.12}
\]

and the canonical form becomes:

\[
\left(\frac{1_x - x_o}{a} \cos (\theta) + \frac{1_y - y_o}{b} \sin (\theta)\right)^2 + \left(-\frac{1_x - x_o}{a} \sin (\theta) + \frac{1_y - y_o}{b} \cos (\theta)\right)^2 = 1 \tag{2.13}
\]

where \([x_o \ y_o]^T\) represents the center of the ellipsoid with respect to frame 1. In order to get the quadratic form of a rotated and translated ellipse we use the equation of a general conic in 2D:

\[
F(V, P) = V \cdot P \tag{2.14}
\]

\[
= P_1 1_x^2 + P_2 1_x 1_y + P_3 1_y^2 + P_4 1_x + P_5 1_y + P_6
\]

\[
= \begin{bmatrix} 1_x & 1_y \end{bmatrix} A \begin{bmatrix} 1_x \\
1_y
\end{bmatrix} + B \begin{bmatrix} 1_x \\
1_y
\end{bmatrix} + c
\]

where \(V = [1_x^2 \ 1_x 1_y \ 1_y^2 \ 1_x \ 1_y \ 1 \ y \ 1]\), \(P = [P_1 \ P_2 \ P_3 \ P_4 \ P_5 \ P_6]\), \(A = [P_1 \ \frac{P_2}{2} \ \frac{P_3}{2}]\), \(B = [P_4 \ P_5]\) and \(c = P_6\). It should be noted that \(P'' = \begin{bmatrix} P''_1 \ P''_3 \ P''_6 \end{bmatrix}\) in equation 2.11 denotes the \(P\) parameters in the unrotated and untranslated reference frame and \(P'' \neq P_1\). The general conic can represent hyperbolas, parabolas, ellipses and degenerate cases; however, in the case of ellipses the \(A\) matrix is positive definite. A positive definite matrix has a positive determinant and therefore we can say that the subset of \(P\) vectors that represent an ellipse must satisfy the constraint \(4P_1 P_3 - P_2^2 > 0\).

Referring to equations 2.12, 2.13, and 2.14, we can see that there are two sets of parameters that uniquely define our ellipse. We define the \(q\) parameters from equations 2.12, 2.13 as:

\[
q = [a \ b \ x_o \ y_o \ \theta] \tag{2.15}
\]

and the \(P\) parameters from equation 2.14 as:
2.4 Ellipse Fitting

\[ P = [P_1 \ P_2 \ P_3 \ P_4 \ P_5 \ P_6] \]  
(2.16)

It should be noted that any scalar multiple of a given \( P \) vector produces the same ellipse due to a free scaling factor. In other words, in order to define an ellipse we only need 5 parameters and we can fix one of the parameters, for example \( P_6 \), to be a constant value.

As will be shown in Section 2.4.2, there is an elegant way to find the \( P \) vector for an ellipse that best fits a set of points. Once we have found this \( P \) vector, we would like to draw out the ellipse contour on the prostate image. It is very difficult to draw an ellipse in Matlab\textsuperscript{TM} given the \( P \) vector; however, it is very easy to do so using equation 2.12 and the \( q \) parameters. For this reason we need to derive a relation between the \( q \) and the \( P \) parameters. Appendix B.1.1 shows how to derive the \( P \) Parameters in terms of the \( q \) parameters and Appendix B.1.2 shows two methods of finding the \( q \) parameters in terms of the \( P \) parameters.

2.4.2 Fitting an Ellipse to Measurement Points

Our goal is to fit an ellipse to a set of measurement points (in this case the points around the boundary of the prostate). The following equation gives a measure of how far away a point \( [^1x_j \ ^1y_j] \) is from the ellipse defined by \( P \):

\[ F(V_j, P) = V_j \cdot P \]
\[ = P_1 x_j + P_2 y_j + P_3 ^1x_j y_j + P_4 ^1y_j x_j + P_5 ^1y_j + P_6 \]
(2.17)

The best ellipse fit is found by finding the \( P \) vector that minimizes the sum of squared distances (also known as Least Squares Minimization - LSM), to a set of points \( [^1x_i \ ^1y_i] \), \( (i = 1...n) \) as follows:

\[ \text{Cost} = \sum_{i=1}^{n} (F(V_i, P))^2 \]
\[ = \sum_{i=1}^{n} (V_i \cdot P)^2 \]
\[ = \|SP\|^2 \]
(2.18)
where \( S = [V_1^T \ V_2^T \ V_3^T \ \cdots \ V_n^T] \) is called the Design Matrix [13]. Refer to Appendix C.1 for the proof of why and how \( \sum_{i=1}^{n} (V_i \cdot P)^2 \) equals \( ||SP||^2 \). Since we are interested in elliptical fits only, we must place the constraint \( 4P_1P_3 - P_2^2 > 0 \) on the set of \( P \) vectors returned by this LSM.

Instead of the inequality \( 4P_1P_3 - P_2^2 > 0 \), we will use the constraint \( 4P_1P_3 - P_2^2 = 1 \). This is because we could use any positive number, \( \gamma \), on the right hand side of this equality; however, dividing both sides by \( \gamma \) would result in a \( P \) vector that represents the same ellipse due to the free scaling factor.

We can re-write the constraint \( 4P_1P_3 - P_2^2 = 1 \) as \( P^TNP = 1 \), where:

\[
N = \begin{bmatrix}
0 & 0 & 2 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\] (2.19)

Note that \( N \) is a singular matrix and therefore its inverse does not exist. Now the ellipse fitting problem reduces to minimizing \( \text{Cost} = ||SP||^2 \) subject to the constraint \( P^TNP = 1 \). Using Lagrangian Multipliers, which are explained in more detail in Appendix D, we let \( f(P) = ||SP||^2 \) and \( h(P) = P^TNP - 1 \) and find their gradients as follows:

\[
\text{grad}(f(P)) = \lambda \text{grad}(h(P)) \quad (2.20)
\]

\[
\frac{\partial}{\partial P} \left( P^T (S^T S) P \right) = \lambda \frac{\partial}{\partial P} \left( P^TNP - 1 \right) \quad (2.21)
\]

where \( ||SP||^2 = P^T (S^T S) P \). If we use the following matrix differentiation rule:

\[
\frac{\partial}{\partial x} (x^T A x) = x^T (A^T + A) \quad (2.22)
\]

equation 2.21 becomes:

\[
(N - \mu S^T S) P = 0 \quad (2.23)
\]
where $\mu = 1/\lambda$. The above equation can be solved using the Generalized Eigenvalue Problem (GEP) with $\mu$ representing the eigenvalues and $P$ representing the eigenvectors. If we recall, we are attempting to find the $P$ vector that represents an ellipse that best fits a set of measured points; however, the GEP gives a set of $P$ vectors. How do we know which of the eigenvectors is the right $P$ vector?

In Appendix C.2 we prove that: i) there exists only one positive eigenvalue, and ii) the $P$ vector that minimizes the cost function is the eigenvector corresponding to this unique positive eigenvalue.

### 2.5 IMMPDA Edge Detector

This section describes the edge detection algorithm used in the 4th step of the 6-step 2D segmentation algorithm. We model the distance from the center of the prostate to the prostate boundary using four models (two linear and two non-linear) which are derived and presented in section 2.5.5. Before presenting the derivation for each model we give a brief overview of the Star Kalman (section 2.5.1), the (Extended) Kalman Filter (section 2.5.2), Probabilistic Data Association (section 2.5.3), and Interacting Multiple Modes (section 2.5.4). In section 2.5.6 we compare the performance of the four models and choose the best one for segmenting the prostate boundary.

#### 2.5.1 Star Kalman - Finding Candidate Edge Points

We begin the edge detection algorithm by choosing a point at the center of the prostate in the TRUS image and we call this the seed point (not to be confused with the implanted seeds). From the seed point we project rays outwards, at angular increments $\Delta \theta$, for all 360 degrees (thus the name Star Kalman) to discretize the prostate boundary. Next the gradient of the image intensity is found along each ray according to:

$$\text{Gradient}(r, \theta) = \frac{1}{3} \{I(r + 2 \Delta r, \theta) + I(r + \Delta r, \theta) + I(r, \theta) - I(r - \Delta r, \theta) - I(r - 2 \Delta r, \theta) - I(r - 3 \Delta r, \theta)\}$$

(2.24)

where $I(r, \theta)$ is the image intensity at radius $r$ and angle $\theta$ using polar coordinates (with respect to the seed point). It should be noted that the gradient intensity is not found along the entire length
of the ray but from a value $R_{min}$ to $R_{max}$. Along any ray $j$, $R_{min}$ and $R_{max}$ are approximately ±20 pixels on either side of the initial ellipse fit (3rd step of the 6-step algorithm). This means that the initial ellipse fit confines the search area for the edge detector to avoid picking up erroneously high gradient values from structures such as the pubic bone, calcium deposits etc.

Along each ray we choose the highest $E$ (here $E = 5$) gradients and call them candidate edge points. Referring to Figure 2.9 we have $E$ candidate edge points located along ray $k$ at a radial distance of $r_1(k), r_2(k)\ldots r_E(k)$ from the seed point. Figure 2.10b shows the pixel intensities, gradient of pixel intensities and 5 gradient maxima along a single ray in Figure 2.10a.

Indeed, from Figure 2.9 and Figure F.4 we can see that there is a perfect analogy between the prostate boundary and the aircraft trajectory since the aircraft's radial position with respect to the radar, at time $k$, could be either $r_1(k)$, or $r_2(k)$ or $r_E(k)$. The accurate prediction of the prostate boundary is made harder due to the speckle noise in the image similar to the random noise signals in the air.

### 2.5.2 Kalman or Extended Kalman Filter - Finding the Prostate Boundary

A Kalman Filter (KF) or Extended Kalman Filter (EKF) can be adopted to track the prostate boundary. The KF equations for estimating the states of a system (outlined in more detail in

![Figure 2.9: Tracking the prostate boundary using Star Kalman/PDA and a seed point](image-url)
2.5 IMMPDA Edge Detector

Figure 2.10: Pixel intensities along the ray shown in figure (a) are recorded and shown in the first subplot in figure (b). The gradients of the pixel intensities are shown in the second subplot and the 5 maximum gradient values are shown in the third subplot in figure (b)

Appendix F.1) are:

\[
\begin{align*}
\hat{x}(k|k) &= \hat{x}(k|k-1) + W(k) \{z(k) - \hat{z}(k|k-1)\} \\
\hat{x}(k|k-1) &= F(k-1)\hat{x}(k-1|k-1) + G(k-1)u(k-1) \\
\hat{z}(k|k-1) &= H(k)\hat{x}(k|k-1)
\end{align*}
\] (2.25)

The EKF equations for estimating the system states (outlined in more detail in Appendix F.2) are:

\[
\begin{align*}
\hat{x}(k|k) &= \hat{x}(k|k-1) + W(k) \{z(k) - \hat{z}(k|k-1)\} \\
\hat{x}(k|k-1) &= f(k-1, \hat{x}(k-1|k-1), u(k)) \\
\hat{z}(k|k-1) &= h(k, \hat{x}(k|k-1))
\end{align*}
\] (2.26)

where in both sets of equations at time $k$, $\hat{x}(k|k)$ is the estimate of the system states, $W(k)$ is the filter gain, $z(k)$ is the system measurement, $\hat{z}(k|k-1)$ is the measurement prediction, $u(k)$ is the
input into the system, and \( F(k) \) (\( f \) in the EKF) and \( H(k) \) (\( h \) in the EKF) represent the model of the time-varying system.

Here, the time varying system is the distance between the seed point and the prostate boundary. This distance varies with each angle increment, \( \Delta \theta \), and we can represent this spatial discretization as temporal discretization by letting \( \Delta \theta \) equal \( k \). Tables 2.1 and 2.2 summarize two linear and two non-linear models for this time-varying system and section 2.5.5 outlines the detailed derivation along with pros and cons of each model. It should be noted that none of the models have an input term, \( u(k) \).

At any time \( k \) we need to obtain a measurement value, \( z(k) \), from the TRUS image. Recalling from the previous section, there are \( E \) candidate edge points at any time \( k \) and we use these candidate points in a Probabilistic Data Association (PDA) filter, outlined in more detail in Appendix F.4, to find one measurement value, \( z(k) \).

### 2.5.3 Probabilistic Data Association Filter - Finding the Measurement Value

In the PDA filter, we assign each candidate boundary point a Gaussian weight function based on the magnitude of the gradient at that point (higher magnitudes receive higher weight) and their distance away from \( \hat{z}(k|k - 1) \).

\[
\begin{align*}
    p_i(k) &= \frac{Mag^2(k)}{\sqrt{2\pi S(k)}} \exp \left( -\frac{(r_i(k) - \hat{z}(k|k - 1))^2}{2S(k)} \right) \\
    z(k) &= \sum_{i=1}^{E} \left( \frac{p_i(k)}{\sum_{i=1}^{E} p_i(k)} r_i(k) \right) \quad i = 1...E, \quad k = 1...N_r
\end{align*}
\]

(2.27)

In the above equation we can see that points closer to \( \hat{z}(k|k - 1) \) receive a higher weight since \( (r_i(k) - \hat{z}(k|k - 1)) \) will be smaller. Then \( z(k) \) is found by finding the weighted average of the candidate boundary points [5]. Refer to Figures 2.11 for a pictorial explanation.

### 2.5.4 Interacting Multiple Modes

The time-varying model for the KF is:
### 2.5 IMMPDA Edge Detector

#### Interacting Multiple Modes

<table>
<thead>
<tr>
<th>Mode 1:</th>
<th>Mode 2:</th>
<th>Transition prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R = 20$</td>
<td>$R = 20$</td>
<td>$P_k = [0.75, 0.25, 0.25]$</td>
</tr>
<tr>
<td>$Q = 10^7$</td>
<td>$Q = 10^7$</td>
<td><em>Start in Mode 1</em></td>
</tr>
</tbody>
</table>

#### Covariance

| $Q_{x-1} = Q_o$ | $R_{x-1} = R_o$ |

#### State Space Model

| $z_k = \begin{bmatrix} 1 \Delta \theta \n_k \end{bmatrix}$ | $z_k = \begin{bmatrix} 0 \Delta \theta \n_k \end{bmatrix}$ |

#### State Variables

| $x_k = \begin{bmatrix} \theta_k \n_k \end{bmatrix}$ | $x_o = \begin{bmatrix} 0 \end{bmatrix}$ |

#### Constant Velocity
| Linear State Space Models |

| $x_k = \begin{bmatrix} \theta_k \n_k \end{bmatrix}$ | $x_o = \begin{bmatrix} 0 \end{bmatrix}$ |

### Table 2.1: Linear State Space Models
<table>
<thead>
<tr>
<th>Non-linear State Space Models</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ellipse - Derivative Based</strong></td>
</tr>
<tr>
<td>$x_k = \begin{bmatrix} r_k &amp; a_k &amp; b_k &amp; \theta_k \end{bmatrix}^T$</td>
</tr>
<tr>
<td>$x_o = \begin{bmatrix} r_o &amp; a_o &amp; b_o &amp; \theta_o \end{bmatrix}^T$</td>
</tr>
<tr>
<td>$z_k = \begin{bmatrix} r_k \end{bmatrix}<em>{k-1} + \begin{bmatrix} \Delta \theta \end{bmatrix}</em>{k-1} + \begin{bmatrix} \eta_{k-1}^T \end{bmatrix}$</td>
</tr>
<tr>
<td><strong>Ellipse - Output Based</strong></td>
</tr>
<tr>
<td>$x_k = \begin{bmatrix} a_k &amp; b_k &amp; \theta_k \end{bmatrix}^T$</td>
</tr>
<tr>
<td>$x_o = \begin{bmatrix} a_o &amp; b_o &amp; \theta_o \end{bmatrix}^T$</td>
</tr>
<tr>
<td>$z_k = r_k + \begin{bmatrix} \eta_{k-1}^T \end{bmatrix}$</td>
</tr>
</tbody>
</table>

$$r = \frac{ab}{\sqrt{b^2 \cos^2(\theta_k - \theta) + a^2 \sin^2(\theta_k - \theta)}}$$

**Covariance**

$Q_{k-1} = \begin{bmatrix} Q_{oo} & 0 & 0 & 0 \\
0 & Q_{oo} & 0 & 0 \\
0 & 0 & Q_{oo} & 0 \\
0 & 0 & 0 & Q_{oo} \end{bmatrix}$

No IMM

$R_{k-1} = R_o$
Figure 2.11: $\hat{z}(k-1|k-1)$ on ray $k-1$ is mapped to $\hat{z}(k|k-1)$ on ray $k$ using the state space model. Five maxima are chosen on ray $k$ to find $\hat{z}(k|k)$

$$x(k + 1) = F(k)x(k) + G(k)u(k) + \Gamma(k)V(k)$$
$$z(k + 1) = H(k + 1)x(k + 1) + w(k + 1)$$

(2.28)

Similarly, the time-varying model for the EKF is:

$$x(k + 1) = f(k, x(k), u(k)) + \Gamma(k)V(k)$$
$$z(k + 1) = h(k + 1, x(k + 1)) + w(k + 1)$$

(2.29)

where $V(k)$ and $w(k)$ are sequences of white Gaussian noise with covariance $Q(k)$ and $R(k)$ representing the process and measurement noise respectively. Generally, large $Q(k)$ and small $R(k)$ implies confidence in the ultrasound image, resulting in high curvature boundary segmentation, whereas small $Q(k)$ and large $R(k)$ implies confidence in the model, resulting in smooth boundary segmentation. In our model of the time-varying system we would like to have both options of large $Q(k)$ with small $R(k)$ and small $Q(k)$ with large $R(k)$ at different times.

The Interacting Multiple Modes (IMM) allows us to have multiple models running at once with different process and measurement covariances. The states and outputs of the models are then
mixed and combined according to a pre-defined set of probabilities. The IMM enables the user to have access to an overall model which allows for both high curvature and smooth boundary segmentation. For more detail on the IMM refer to Appendix F.3. In the following section, the specifications of the IMMs are given for each system model.

2.5.5 System Models

In the following sections we present the state variables, system models, measurement and process noise covariances and IMM specifications for two linear models and two non-linear models that we tried. At the end we will discuss the merits of each model and specify the model that was used for the final segmentation algorithm.

2.5.5.1 Linear Constant Velocity (Circle) Model

The Constant Velocity Model is the most basic model. It was derived initially by Abulmaesumi et al in [2] and was based on work by Bar-Shalom et al in [6]. The assumption here is that the prostate boundary is approximately circular and the rate of change of the radius remains constant at each angular increment (hence the name Constant Velocity Model).

State Variables:

In this model we define our state variables as:

\[
\begin{align*}
x(k) &= [r(k) \ \dot{r}(k)]^T \\
x_o &= [a_o \ 0]^T
\end{align*}
\]  

(2.30)

where \( r(k) \) is the radial distance between the seed point and the actual prostate boundary and \( \dot{r}(k) \) is the rate of change of this distance. The initial value of \( r \) is \( a_o \) and is found from the user initialization points and the initial value of \( \dot{r} \) is zero.
State Space Model:

If $r(k)$ is the radial distance between the seed point and the prostate boundary at angular increment (analogous to time) $k$, then it follows from the definition of a derivative that:

$$\dot{r}(k-1) \approx \frac{r(k) - r(k-1)}{\Delta \theta}$$

(2.31)

where $k$ and $k - 1$ are separated by $\Delta \theta$. If the above equation is solved for $r(k)$ then we have the 1st order Taylor Expansion of $r(k)$ at $k$ about $k - 1$. For a better approximation of $r(k)$ we find the 2nd order Taylor Expansion and its derivative as follows:

$$\begin{bmatrix} r(k) \\ \dot{r}(k) \\ x(k) \end{bmatrix} = \begin{bmatrix} 1 & \Delta \theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r(k-1) \\ \dot{r}(k-1) \\ x(k-1) \end{bmatrix} + \begin{bmatrix} \frac{\Delta \theta^2}{2} \\ \Delta \theta \\ \Gamma(k-1) \end{bmatrix} \frac{\dot{r}(k-1)}{V(k-1)}$$

(2.32)

If the goal is to represent the prostate boundary as a Constant Velocity Model ($\dot{r}(k) = \dot{r}(k-1)$) then the second term on the right hand side of the equation above, $\Gamma(k-1)V(k-1)$, represents the process noise with covariance $\Gamma(k-1)Q(k-1)\Gamma'(k-1)$ (refer to Figure F.1 in Appendix F.1). The output (measurement) of the system will simply be:

$$\begin{bmatrix} r(k) \\ x(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ H(k-1) \end{bmatrix} \begin{bmatrix} r(k-1) \\ \dot{r}(k-1) \\ x(k-1) \end{bmatrix} + w(k-1)$$

(2.33)

where $w(k-1)$ is the measurement noise with covariance $R(k-1)$.

Noise Covariances:

It is assumed that $\dot{r}(k-1)$ is a zero mean, Gaussian random variable with variance $Q_o$. This would imply that the process noise covariance is:
\[ \Gamma(k - 1)Q(k - 1)\Gamma'(k - 1) = \begin{bmatrix} \frac{\Delta \theta^2}{2} \\ \Delta \theta \\ \frac{\Delta \theta^2}{2} \end{bmatrix} Q_o \begin{bmatrix} \frac{\Delta \theta^2}{2} & \Delta \theta \\ \frac{\Delta \theta^2}{2} & \Delta \theta^2 \\ \frac{\Delta \theta^2}{4} & \frac{\Delta \theta^2}{2} \end{bmatrix} \]

\[ = Q_o \begin{bmatrix} \frac{\Delta \theta^4}{4} & \frac{\Delta \theta^3}{2} \\ \frac{\Delta \theta^3}{2} & \Delta \theta^2 \\ \frac{\Delta \theta^2}{2} & \Delta \theta^2 \end{bmatrix} \] (2.34)

As well, it is assumed that \( w(k - 1) \) is a zero mean, Gaussian random variable with variance \( R_w \).

**Interacting Multiple Modes:**

The prostate boundary was represented with two Constant Velocity models (modes). Mode 1 had a large \( R(k) \) and small \( Q(k) \) and enabled the edge detector to extract smooth boundary contours (trusting the model more than the measurements). Mode 2 had a small \( R(k) \) and large \( Q(k) \) enabling the edge detector to pick up on fine resolution (trusting the measurements more than the model). In Mode 1 \( R(k) = 20 \) and \( Q(k) = 10 \) and in Mode 2 \( R(k) = 20 \) and \( Q(k) = 10^5 \). The transition probability from a mode to itself was 75% and the transition probability from a mode to another mode was 25%. These numbers were chosen by trial and error and showed excellent results.

**2.5.5.2 Linear Ellipse Model**

The next three models are based on on-going work by Julian Guerrero [17]. The assumption here is that the prostate is elliptical instead of circular.

**State Variables:**

In this model the state variables are the same as in the Circle Model.

**State Space Model:**

We begin with the basic equation of an ellipse in Cartesian coordinates:
\[
\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1 \quad (2.35)
\]

where \(a\) and \(b\) are the semi-major and semi-minor axes of the ellipse respectively. If the above equation is converted to polar coordinates and an angle offset \(\theta\) is introduced, the equation becomes:

\[
\frac{r^2 \cos^2(\theta_i - \theta)}{a^2} + \frac{r^2 \sin^2(\theta_i - \theta)}{b^2} = 1 \quad (2.36)
\]

where \(x = r \cos(\theta_i)\) and \(y = r \sin(\theta_i)\) and \(\theta_i\) is the sweeping angle. Solving for \(r\) we obtain:

\[
r = \frac{ab}{\sqrt{b^2 \cos^2(\theta_i - \theta) + a^2 \sin^2(\theta_i - \theta)}} \quad (2.37)
\]

which is the equation of a rotated ellipse in polar coordinates. Taking the first and second derivatives of the above equation with respect to \(\theta\) we get:

\[
\dot{r} = r \left( \frac{[b^2 - a^2]}{b^2 \cos^2(\theta_i - \theta) + a^2 \sin^2(\theta_i - \theta)} \right) \quad (2.38)
\]

\[
\ddot{r} = r \left( \frac{[b^2 \cos^4(\theta_i - \theta) - [a^2 - b^2] \sin^2(2(\theta_i - \theta))/2^2 \sin^4(\theta_i - \theta))]}{(b^2 \cos^2(\theta_i - \theta) + a^2 \sin^2(\theta_i - \theta))^2} \right) \quad (2.39)
\]

Next, the above equations are discretized using the following equation:

\[
\begin{bmatrix}
\dot{r}(k) \\
\ddot{r}(k)
\end{bmatrix} =
\begin{bmatrix}
\frac{r(k+1)-r(k)}{\Delta \theta} \\
\frac{\dot{r}(k+1)-\dot{r}(k)}{\Delta \theta}
\end{bmatrix} =
\begin{bmatrix}
\frac{r(k+1)-r(k)}{\Delta \theta} \\
\dot{r}(k)D(k)
\end{bmatrix} \quad (2.40)
\]

Re-arranging the above equation and introducing noise, we can write the state space equation of the linear ellipse model as:

\[
\begin{bmatrix}
r(k) \\
\dot{r}(k) \\
x(k)
\end{bmatrix} =
\begin{bmatrix}
1 & \Delta \theta & 0 \\
D(k-1) \Delta \theta & 1 & 0 \\
F(k-1) & 0 & 1
\end{bmatrix}
\begin{bmatrix}
r(k-1) \\
\dot{r}(k-1) \\
x(k-1)
\end{bmatrix} + V(k-1) \quad (2.41)
\]

The output equation for this model is the same as the Constant Velocity model. It should
be noted that the state space equations of the ellipse model are very similar to the state space equations of the Constant Velocity model except that the derivative of the radius is related to the radius in this case. Unlike the next 3 models, the $a, b, \theta$ values are constant here and obtained from the initial ellipse fit (step 3 of the 6 step segmentation algorithm).

Noise Covariances:

The constant velocity and linear model are close enough to assume that they have the same noise covariance.

Interacting Multiple Modes:

This model used the same two modes with the same transition probabilities as the Constant Velocity model.

2.5.5.3 Non-linear Ellipse Model - Derivative Method

In this model we iterate an EKF many times until the states $a, b, \theta$ converge to a constant value. Essentially we want to find the parameters of the ellipse that best describe the prostate shape that we are trying to segment.

State Variables:

The state variables for this model are:

$$x(k) = [r(k) \ a(k) \ b(k) \ \theta(k)]^T$$

$$x_o = [r_o \ a_o \ b_o \ \theta_o]^T$$

and we track their evolution at every iteration of the EKF.

State Space Model:
To find the state-space model we begin with equation 2.38 and add noise to get:

\[ \dot{r} = rC + \eta \]  

(2.43)

Now we discretize the state variables as follows:

\[
\begin{bmatrix}
\frac{r(k) - r(k-1)}{\Delta \theta} \\
\frac{a(k) - a(k-1)}{\Delta \theta} \\
\frac{b(k) - b(k-1)}{\Delta \theta} \\
\frac{\theta(k) - \theta(k-1)}{\Delta \theta}
\end{bmatrix} =
\begin{bmatrix}
r(k)C(k - 1) + \eta_r(k - 1) \\
\eta_a(k - 1) \\
\eta_b(k - 1) \\
\eta_\theta(k - 1)
\end{bmatrix}
\]  

(2.44)

and finally the state space representation becomes:

\[
\begin{bmatrix}
\frac{r(k) - r(k-1)}{\Delta \theta} \\
a(k) \\
b(k) \\
\theta(k)
\end{bmatrix} =
\begin{bmatrix}
r(k)C(k - 1) + \eta_r(k - 1) \\
\frac{\Delta \theta}{1 - C(k - 1) \Delta \theta} & 0 & 0 & 0 \\
0 & \Delta \theta & 0 & 0 \\
0 & 0 & \Delta \theta & 0 \\
0 & 0 & 0 & \Delta \theta
\end{bmatrix}
\begin{bmatrix}
\eta_r(k - 1) \\
\eta_a(k - 1) \\
\eta_b(k - 1) \\
\eta_\theta(k - 1)
\end{bmatrix}
\]  

(2.45)

\[
\begin{bmatrix}
r(k) \\
a(k) \\
b(k) \\
\theta(k)
\end{bmatrix} =
\begin{bmatrix}
[1 0 0 0] x(k-1) + w(k-1) \\
H(k-1)
\end{bmatrix}
\]  

(2.46)

In the state space equation above, the process equation is non-linear (hence use of the EKF) but the measurement equation is linear. In this case, only the process equation has to be linearized as outlined in Appendix F.2.

Noise Covariances:

It is assumed that the noise introduced into each variable, in this system, is independent of the noise in the other variables and is a zero mean, Gaussian white noise. The covariance of the noise of each state (or the process noise covariance) is then:
2.5 IMMPDA Edge Detector

\[
Q(k-1) = \begin{bmatrix}
Q_{\alpha \alpha} & 0 & 0 & 0 \\
0 & Q_{\alpha \alpha} & 0 & 0 \\
0 & 0 & Q_{\alpha \beta} & 0 \\
0 & 0 & 0 & Q_{\alpha \beta}
\end{bmatrix}
\]  

(2.47)

As well, it is assumed that \( w(k-1) \) is a zero mean, Gaussian random variable with variance \( R_o \).

Interacting Multiple Modes:

The EKF iterates until it converges to constant \( a, b, \theta \) values. When we introduce the IMM we mix and change the states constantly making it more difficult and time consuming to reach convergence; therefore, there are no IMMs in the non-linear models.

2.5.5.4 Non-Linear Ellipse Model - Output Method

This model is very similar to the previous model, except here the non-linear portion of the model is in the measurement equation instead of the process equation.

State Variables:

In this model our state variables are:

\[
x(k) = [a(k) \ b(k) \ \theta(k)]^T
\]

\[
x_o = [a_o \ b_o \ \theta_o]^T
\]

(2.48)  

(2.49)

where the radius is omitted as a state variable because it is the output of the measurement equation in the state space model described below.

State Space Model:

As before, to find the state-space model we discretize the state and output variables and add noise.
Discretizing equation 2.37 gives us the output of the system:

\[ r(k) = \frac{a(k)b(k)}{\sqrt{b(k)^2 \cos^2(\theta_i - \theta(k)) + a(k)^2 \sin^2(\theta_i - \theta(k))}} \] (2.50)

Therefore, the final state space model becomes:

\[
\begin{bmatrix}
    a(k) \\
    b(k) \\
    \theta(k) \\
    x(k) \\
    z(k)
\end{bmatrix} =
\begin{bmatrix}
    1 & 0 & 0 & F(k-1) & \Delta \theta & 0 & 0 & \eta_a(k-1) \\
    0 & 1 & 0 & x(k-1) & 0 & \Delta \theta & 0 & \eta_b(k-1) \\
    0 & 0 & 1 & \theta(k-1) & 0 & 0 & \Delta \theta & \eta_\theta(k-1) \\
    r(k-1) & x(k-1) & +w(k-1) & h(k-1, x(k-1))
\end{bmatrix}
\] (2.51)

\[
\begin{bmatrix}
    0 \\
    1 \\
    0 \\
    1 \\
    1
\end{bmatrix}
\]

Note that the measurement equation above is non-linear and must be linearized.

Noise Covariances:

The process and measurement noise covariance for this model are the same as for the previous model. In this case \( Q(k - 1) \) is a 3 \times 3 diagonal matrix with the noise covariances of the state variables on the diagonal entries. The covariance of \( w(k - 1) \) is \( R_o \) as before.

Interacting Multiple Modes:

There are no IMMs for the reasons described in the previous model.

2.5.6 Comparing the Four Models

The 6-step segmentation algorithm consists of the following steps: 1) initialization, 2) forward warp, 3) initial ellipse fit, 4) edge detection along the initial ellipse fit, 5) final ellipse fit and 6) reverse warp. The two linear edge detection techniques are used in the 4th step of the 6-step segmentation algorithm; however, the two non-linear techniques attempt to replace steps 3 to 5. The goal of the EKF is to find the parameters of the ellipse that best fit the shape of the prostate being segmented.
Therefore, if the non-linear techniques are used, the 6-step segmentation algorithm becomes a 4-step segmentation algorithm: 1) initialization, 2) forward warp, 3) non-linear edge detection with EKF and 4) reverse warp. It should also be noted that the non-linear techniques cannot be guided with the initial ellipse fit since the EKF relies on having the same search area at each iteration so that it can converge to a final set of state values. If the EKF is guided, its search area will change at each iteration making it more difficult to converge.

The following question naturally rises: does the EKF with Parameter ID in the 4-step algorithm converge and if it does are the resulting $a, b, \theta$ values close to those obtained from the 6-step algorithm? In order to make this comparison we found the best linear model and the best non-linear model and compared the performance of the two to each other.

2.5.6.1 Comparing the Two Linear Models

We analyzed the two linear models to see which gave the best edge detection results. We also evaluated whether it was necessary to have an IMM with two modes of operation. In Figure 2.12 the thin white contours are the initial ellipse fit and the thick dotted and thick solid lines are the results of the edge detection from the Circle and Ellipse model respectively. As well, this figure illustrates the effect of having multiple modes by comparing the result of the edge detection when there are 2 modes ($Q_1 = 10, R_1 = 20$ and $Q_2 = 10^5, R_2 = 20$) and only one mode ($Q_1 = 10, R_1 = 20$). From this figure, it is clear that the IMMs are necessary for finer, more precise edge detection.

In Figure 2.13, the edge detection results of the Circle model are super-imposed on top of the Ellipse model (both with 2 modes) for four different cases. We can see that both models create very similar results. In conclusion either model would work well so we choose the Circle Model since its simulation time is slightly better than its counterpart.

2.5.6.2 Comparing the Two Non-linear Models

In a previous study, Julian found that the Output method was slightly better than the Derivative method since the derivative method introduces noise into the state variables. Therefore in the analysis for the following section we use the Output method.
2.5 IMMPDA Edge Detector

• Model: Circle • Model: Circle • Model: Ellipse • Model: Ellipse
• IMM: 2 modes • IMM: 1 modes • IMM: 2 modes • IMM: 1 modes
• Time: 2.23 sec • Time: 2.21 sec • Time: 2.65 sec • Time: 2.35 sec

Figure 2.12: Comparison of edge detection results using one mode or two mode IMMs (edge detection is guided along an ellipse (thin solid line) and results are shown for the Circle model (thick dotted line) and Ellipse model (thick solid line))

Figure 2.13: Comparison of edge detection results using the Circle model (thick dotted contour) and the Ellipse model (thick solid contour) for four different prostate contours

2.5.6.3 Comparing Linear vs Non-linear Model

We would like to compare the linear Circle model (with two modes: $Q_1 = 10$, $R_1 = 20$, and $Q_2 = 10^5$, $R_2 = 20$) to the non-linear Ellipse based Output model. Using a complex search algorithm, the $Q$ and $R$ factors for the non-linear model were found to be $Q_1 = \text{diag}[4.5238 \ 0.0069 \ 11.4152]$ and $R_1 = 38.5133$ [17]. Then the 6-step segmentation algorithm using the linear model and the 4-step segmentation algorithm using the non-linear model were applied.

In the non-linear case, since the edge detection could not be guided, and since the images are
extremely noisy, the detected contour wandered heavily. As a result, the $a,b,\theta$ values did not converge even after many iterations. This suggests that due to the noisy nature of the images the edge detection algorithm must be very robust and therefore needs to be guided with the initial ellipse fit. Since the non-linear models cannot be guided (the EKF would not converge since the search area changes at each iteration) we use the Circle model throughout the rest of this thesis with the same $Q$ and $R$ as stated above.
Chapter 3

Study Results and Discussion: 2D Segmentation

3.1 Study Setup

Using the standard equipment (Siemens Sonoline Prima with 5MHz biplane TRUS probe) and the standard image acquisition procedure used at the BC Cancer Agency, we acquired 17 pre-operative TRUS images of the prostate from the midgland region. The image sizes were 480 x 640 with pixel size of 0.18 x 0.18mm. The images had low resolution and contained a superimposed needle grid. Although it is possible to obtain higher resolution images our goal was to work with the images currently used at the Cancer Agency.

A single expert radiation oncologist with vast experience in prostate volume studies manually segmented the images using the Vari-Seed planning software from Varian. We then applied our semi-automatic segmentation algorithm and the same expert chose the initialization points.

The algorithm was written in Matlab 7 (Release 14) and executed on a Pentium 4 PC running at 1.7 GHz.

3.2 Validation Methods

The manual segmentation was used as the 'gold standard' while distance [8] and area [22] based metrics were used to validate the semi-automatic segmentation.
3.2 Validation Methods

3.2.1 Distance Based Metrics

For distance-based metrics we found the radial distance between the manual and semi-automatic contours, with respect to a reference point at the center of the prostate, as a function of theta around the prostate: $r_{diff}(\theta)$. Figure 3.1 shows an example of this radial distance at $\theta = \theta_1$. A value called the Mean Absolute Difference (MAD) was found for each of the 17 cases as follows:

$$MAD = \frac{1}{N} \sum_{i=1}^{N} |r_{diff}(\theta_i)|$$  \hspace{1cm} (3.1)

where on any given prostate, $N$ radial differences were found at $N$ different angles. This value explains how well, on average, the shape of the semi-automatic contour matches the manual contour. A second value, called the Maximum Difference (MAXD) was found for each of the 17 cases as follows:

$$MAXD = max(|r_{diff}(\theta_i)|)$$  \hspace{1cm} (3.2)

where the $max()$ operator simply finds the maximum value in the array $r_{diff}(\theta)$. This value tells us, for any angle $\theta$, what is the greatest radial difference between the two contour shapes.

Figure 3.1: The radial differences (at angles $\theta_i$) between the manual (dotted line) and semi-automatic (solid line) contour, are used in validating the segmentation algorithm.
3.2.2 Area Based Metrics

For area-based metrics we superimposed the manually segmented polygon (gold standard), $\Omega_m$, on top of the semi-automatically segmented polygon, $\Omega_a$ (refer to Figure 3.2). The region common to both $\Omega_m$ and $\Omega_a$ is the True Positive (TP) area. The region in $\Omega_m$ but not in $\Omega_a$ is the False Negative (FN) area and the region in $\Omega_a$ but not in $\Omega_m$ is the False Positive (FP) area. We then found the following area-based metrics for each of the 17 cases [22]:

\[
Sensitivity = \frac{TP}{TP + FN}.
\]  

\[
Accuracy = 1 - \frac{(FP + FN)}{(TP + FN)}.
\]

In medical trials, the Sensitivity of a test determines how sensitive a test is at picking up a disease when the disease is actually present. In our case, Sensitivity tells us how sensitive our algorithm is at finding the pixels that are prostate from the set of pixels that the average manual segmentation assumes are prostate. Ideally the algorithm would be able to identify every pixel that was prostate as prostate (100% Sensitivity). In the Accuracy formula, the second term indicates how incapable the algorithm is at picking the true prostate pixels. Ideally this term would be zero because FP and FN would not exist. Therefore, one minus this term gives us another measure of how accurate the algorithm is when it states that a pixel is prostate.

![Figure 3.2: Pixel by pixel comparison of manual vs semi-automatic contours](image)
### 3.3 Results

The MAD, MAXD, sensitivity and accuracy were found for each image and the overall average values along with standard deviations (STD) are presented in Table 3.1. On average, the mean absolute difference between the manual and semi-automatic contours was $0.68 \pm 0.18\text{mm}$ which is less than half the average variability between human experts ($1.82 \pm 1.44\text{mm}$) [16]. Therefore, the mean absolute distance is on the same order of error as the repeatability between two different expert observers performing manual segmentation. On average, the maximum distance between the manual and semi-automatic contours was $2.25 \pm 0.56\text{mm}$. The average sensitivity and accuracy were over 97% and 93% respectively with std values less than 2%. Refer to Figure 3.3 for sample outputs.

The times for manual and semi-automatic segmentation were recorded and are presented in Table 3.2. On average the time for manual segmentation, which includes the time for selecting 16-25 points and then re-adjusting them to obtain approximate symmetry, was $45.35 \pm 4.85$ seconds. The average time for semi-automatic segmentation, which includes the time to choose initialization points and then run the algorithm, was $12.38 \pm 0.98$ seconds. The average time to run the algorithm alone was $1.12 \pm 0.14$ seconds for a single slice; however, we expect this time to decrease by at least five fold when the algorithm is transferred to C++. It should be noted that in the MICCAI paper [4] we reported a time of $14.10 \pm 0.2$ seconds to run the algorithm alone. The improvements to this time are due to Matlab code optimization, especially in the image warping step.

<table>
<thead>
<tr>
<th>Manual</th>
<th>Semi-automatic (Matlab)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Time (sec)</td>
<td>Total Time (sec)</td>
</tr>
<tr>
<td>Mean</td>
<td>Initialization (sec)</td>
</tr>
<tr>
<td>STD</td>
<td>Mean</td>
</tr>
<tr>
<td>45.35</td>
<td>11.26</td>
</tr>
</tbody>
</table>

Table 3.2: Average times for manual and semi-automatic segmentation
3.4 Discussion

The 2D semi-automatic segmentation algorithm does not require training models or time consuming pre-filtering of TRUS images. The result is a simple, computationally inexpensive algorithm with potential for real-time applications and extension to 3D segmentation. Furthermore, the algorithm works well on the low resolution TRUS images currently used at the BC Cancer Agency and can be integrated easily into their setup without any changes to the image acquisition procedure. In preliminary studies, an untrained observer chose the initialization points. The segmented contour resulted in sensitivity and accuracy values over 90% with MAD and MAXD values less than 1 and 3mm respectively. This result shows that unlike most other semi-automatic segmentation techniques the success of the final segmentation does not rely on precise initialization of the prostate boundary. Consequently, during intra-operative procedures, a less experienced observer can choose the initialization points to update the treatment plan allowing the oncologist to continue their work uninterrupted. A more in-depth study of initialization sensitivity may be a worthwhile topic for future students. Due to the nature of this technique, the algorithm works best for prostates that have an elliptical shape, and worst for those that have a diamond shape, after warping.
Chapter 4

Algorithm: 3D Segmentation

4.1 Introduction to 8-step Algorithm

A prostate volume study is composed of a stack of 10-14 transverse, 2D, TRUS images taken at 5mm intervals. In volume study terminology, the base is the most superior slice, the apex is the most inferior slice and the midgland is midway between the base and apex. The boundary of the prostate in and around the midgland slice is usually clearly visible; however, at the base and the apex it is nearly indistinguishable.

In 3D prostate segmentation, the goal is to create a computer algorithm that segments the prostate from the stack of images making up the 3D volume study. In the previous two chapters we showed the success of the 2D segmentation algorithm on the midgland images; however, in 3D we need to segment the base and apex as well. For this reason, in 3D we start off with the information we can easily obtain (prostate boundary in and around the midgland) and work our way outwards, to the more difficult slices, while using the previous slice as a priori data at each step. This technique is exactly what the oncologists use to infer the location of the prostate at the base and the apex. The proposed 3D computer algorithm has 8 steps (Figure 4.1) as follows:

1. Initialize only the midgland slice.

2. Warp every slice in the prostate volume so that the prostate shape becomes ellipsoidal.

3. Segment the midgland slice using the 2D segmentation algorithm (do not perform reverse warp at this stage).
4. Use the segmented midgland contour as initialization for the slice before the midgland (midgland-1) and the slice after the midgland (midgland+1). Segment these slices using the 2D segmentation algorithm (again without reverse warp). Use segmentation of midgland+1 as initialization for midgland+2 and segmentation of midgland-1 as initialization for midgland-2 and segment these slices as well. Continue this iterative segmentation until all slices, except base and apex, have been segmented.

5. Fit an ellipsoid to the stack of 2D elliptical contours from the previous step.

6. Assume circular contours for the base and apex centered at the ellipsoid’s major axis.

7. Fit a final ellipsoid to all 2D contours including base and apex.

8. Slice the final ellipsoid transversally in 5mm increments to get 2D elliptical contours and apply the reverse warping algorithm to this set of 2D contours. The final, segmented 3D prostate volume is a set of 2D contours.

In the initialization step, the user selects 5 initialization points in the same manner as in 2D (refer to section 2.2). In the warping step, we warp each 2D slice according to the 2D warping algorithm described in section 2.3.2. Here a question arises: what $\sigma$ values do we use for each slice? When the TRUS probe is inserted into the rectum, the assumption is that the force exerted by the probe is uniform along the entire length of the prostate and therefore every cross sectional slice experiences the same radial compression. Therefore we use the same $\sigma$ value for each slice and section 2.3.2.1 outlines a technique to find the optimal $\sigma$ based on the initialization points.

In step 4, every slice, except the base and apex, is segmented using the 2D segmentation algorithm with the initialization being the previous slice’s final ellipse fit. It should be noted that as we move away from the midgland, the prostate boundary becomes smaller and so we need to make the 2D segmentation algorithm biased to finding prostate contours that are smaller than the previous contour. This is done by forcing the IMMPDA to find more candidate edge points inside the initial ellipse fit than outside (i.e., search range would be 10 pixels outside the initial ellipse fit but 30 pixels inside).

In TRUS images, the prostate boundary cannot be seen at the base and apex (due to blending with the bladder neck and pelvic floor respectively). A good strategy for segmenting the prostate boundary in these slices is to extrapolate from the contours in the previous slices. This is why
4.1 Introduction to 8-step Algorithm

Perform 2D 6-step segmentation algorithm (only up to 5th step) on midgland slice. Use the result as initialization for the slices immediately before and after the midgland.

Initialize midgland only

Fwd Warp all Slices

Each 'final ellipse fit’ is used as the initialization for the next slice

Base ... Midgland – 2 slices Midgland – 1 slice Midgland + 1 slice Midgland + 2 slices ... Apex - 1 slice

Initial Ellipsoid Fit

Contour Base and Apex

Final Ellipsoid Fit

Reverse Warp

Figure 4.1: Eight steps of the 3D semi-automatic prostate segmentation algorithm
we have chosen ellipsoid fitting so that with a good approximation of the size and shape of the base and apex the ellipsoid will extrapolate and give us the best hypothesis for the actual prostate boundary in these slices. After looking through a large database of manually segmented prostate volumes at the local cancer agency we found that the approximate shape and size of the prostate at the base and apex were circles with average radii of 18mm and 15mm respectively. Therefore, in semi-automatic segmentation, circles of radii 18mm and 15mm were drawn at the base and apex, centered about the center axis of the initial ellipsoid fit. The final ellipsoid fit would extrapolate to give us the actual prostate boundary at the base and apex.

4.2 Ellipsoid Fitting

4.2.1 Equation of an Ellipsoid

As in the 2D case, we would like to derive a general equation for a rotated, translated ellipsoid. The reference frame notations used throughout the rest of this section are outlined in Appendix A. Let us first consider an ellipsoid centered at the origin, with no rotation. There are three ways to represent this ellipsoid. The parametric equation is:

\[
\begin{bmatrix}
  ^5x \\
  ^5y \\
  ^5z
\end{bmatrix} =
\begin{bmatrix}
  a \cos(\theta_i) \sin(\phi_i) \\
  b \sin(\theta_i) \sin(\phi_i) \\
  c \cos(\phi_i)
\end{bmatrix}
\]

where \(^5X = [^5x \ ^5y \ ^5z]\) is a point in the unrotated, untranslated final prostate frame. Here \(a, b\) and \(c\) are the lengths of the axes along the \(^5x\), \(^5y\) and \(^5z\) directions respectively, and \(\theta_i\) and \(\phi_i\) are sweep variables. The Canonical form of the same ellipsoid is:

\[
\frac{^5x^2}{a^2} + \frac{^5y^2}{b^2} + \frac{^5z^2}{c^2} = 1
\]

and the quadratic form is:

\[
P_1^{(\gamma)} x^2 + P_2^{(\gamma)} y^2 + P_3^{(\gamma)} z^2 + P_4^{(\gamma)} = 0
\]

We will now use the rotation/translation convention of equation A.6 to get the above three equations in the \(\{^1\hat{o}, ^1\hat{C}\}\) reference frame. For notational convenience we let \(\cos(\gamma)\) and \(\sin(\gamma)\)
equal $c\gamma$ and $s\gamma$ respectively. The parametric equation of a rotated, translated ellipsoid will be:

$$
\begin{bmatrix}
    x \\
    y \\
    z
\end{bmatrix} =
\begin{bmatrix}
    x_0 \\
    y_0 \\
    z_0
\end{bmatrix} +
\begin{bmatrix}
    c\beta & s\beta & 0 \\
    -s\beta & c\beta & 0 \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    c\phi & 0 & -s\phi \\
    0 & 1 & 0 \\
    s\phi & 0 & c\phi
\end{bmatrix}
\begin{bmatrix}
    1 & 0 & 0 \\
    0 & c\theta & s\theta \\
    0 & -s\theta & c\theta
\end{bmatrix}
\begin{bmatrix}
    a & c\theta_i & s\phi_i \\
    b & s\theta_i & s\phi_i \\
    c & c\phi_i
\end{bmatrix}
$$

(4.4)

where we call the sequence of three $3 \times 3$ rotation matrices $U$. The Canonical equation becomes:

$$
\left( \frac{c\beta c\phi \left( x - x_0 \right) - s\beta c\phi \left( y - y_0 \right) + s\phi \left( z - z_0 \right)}{a} \right)^2 +
\left( \frac{\left( s\beta c\theta + c\beta s\phi s\theta \right) \left( x - x_0 \right) + \left( c\beta c\theta - s\beta s\phi s\theta \right) \left( y - y_0 \right) - c\phi s\theta \left( z - z_0 \right)}{b} \right)^2 +
\left( \frac{\left( s\beta s\theta - c\beta s\phi c\theta \right) \left( x - x_0 \right) + \left( c\beta s\theta + s\beta s\phi c\theta \right) \left( y - y_0 \right) + c\phi c\theta \left( z - z_0 \right)}{c} \right)^2 = 1
$$

(4.5)

In order to get the Quadratic form of a rotated and translated ellipsoid we use the equation of a general conic in 3D:

$$
F(V, P) = V \cdot P
$$

$$
= P_1x^2 + P_2y^2 + P_3z^2 +
2P_4xy + 2P_5xz + 2P_6yz +
2P_7x + 2P_8y + 2P_9z + P_{10}
$$

$$
= [x \ y \ z] A \begin{bmatrix}
    1x \\
    1y \\
    1z
\end{bmatrix} + B \begin{bmatrix}
    1x \\
    1y \\
    1z
\end{bmatrix} + c
$$

(4.6)

where $A = [P_1 \ P_6 \ P_5; \ P_6 \ P_2 \ P_4; \ P_5 \ P_4 \ P_3]$, $B = [P_7 \ P_8 \ P_9]$, $c = P_{10}$, $V = [1 \ x^2 \ y^2 \ z^2 \ 2xy \ 2xz \ 2yz \ 2x \ 2y \ 2z \ 1]$, and $P = [P_1 \ P_2 \ P_3 \ P_4 \ P_5 \ P_6 \ P_7 \ P_8 \ P_9 \ P_{10}]$. As in 2D, if the $A$ matrix is positive definite, the 3D conic represents an ellipsoid (as opposed to a hyperboloid, paraboloid or degenerate case).
Two tests to check the positive definiteness of a symmetric $3 \times 3$ matrix $A$ are:

\[ A \text{ is pos def } \iff \lambda_1, \lambda_2, \lambda_3 > 0 \]
\[ \iff \text{determinant of the principal minors of } A > 0 \]  \hspace{1cm} (4.7)

where $\lambda_i$ is the $i$th eigenvalue, the $1 \times 1$ principal minors are $P_1$, $P_2$ and $P_3$, the $2 \times 2$ principal minors are $[P_1 \ P_6; \ P_6 \ P_2]$, $[P_1 \ P_5; \ P_5 \ P_3]$ and $[P_2 \ P_4; \ P_4 \ P_3]$ and the $3 \times 3$ principal minor is $A$ itself.

In [24], it is shown that $A$ is positive definite if and only if $I, J, K > 0$ where $I, J$ and $K$ are the sum of the $1 \times 1$, $2 \times 2$ and $3 \times 3$ principal minors of $A$ respectively as follows:

\[ I = P_1 + P_2 + P_3 = \lambda_1 + \lambda_2 + \lambda_3 \]  \hspace{1cm} (4.8)
\[ J = \begin{vmatrix} P_1 \ P_6 \\ P_6 \ P_2 \end{vmatrix} + \begin{vmatrix} P_1 \ P_5 \\ P_5 \ P_3 \end{vmatrix} + \begin{vmatrix} P_2 \ P_4 \\ P_4 \ P_3 \end{vmatrix} = \lambda_1 \lambda_2 + \lambda_1 \lambda_3 + \lambda_2 \lambda_3 \]  \hspace{1cm} (4.9)
\[ K = |A| = \lambda_1 \lambda_2 \lambda_3 \]  \hspace{1cm} (4.10)

From equation 4.7 we can see that if $A$ is positive definite then $I, J, K > 0$. To show the inverse implication we find the characteristic equation of $A$ as follows:

\[ \det(A - \lambda I) = -\lambda^3 + \lambda^2 J - \lambda K + K = 0 \]  \hspace{1cm} (4.11)

In this case, if $I, J, K > 0$ then the characteristic polynomial has alternating signs so that the roots, $\lambda_i$, are always positive; therefore, according to equation 4.7, this implies that $A$ is positive definite. We have now proven that for a given $P$ vector, $A$ is positive definite if and only if $I, J, K > 0$ and therefore $P$ describes an ellipsoid. In order to put this into the form of a constraint that is similar to 2D, we say $\alpha J > I^2$ for some $\alpha$; therefore the ellipsoid constraint is $\alpha J - I^2 > 0$. The parameter $\alpha$ will be discussed in more detail in the following section.

Referring to equations 4.4, 4.5, and 4.6, we can see that there are two sets of parameters that uniquely define our ellipsoid. We define the $q$ parameters from equations 4.4, 4.5 as:
4.2 Ellipsoid Fitting

\[ q = \begin{bmatrix} a & b & c & x_o & y_o & z_o & \theta & \phi & \beta \end{bmatrix} \quad (4.12) \]

and the \( P \) parameters from 4.6 as:

\[ P = [P_1 \ P_2 \ P_3 \ P_4 \ P_5 \ P_6 \ P_7 \ P_8 \ P_9 \ P_{10}] \quad (4.13) \]

Note that there is a free scaling factor in \( P \) so that we only need 9 parameters to describe an ellipsoid. As in the 2D case, there is an elegant way to find the \( P \) vector for an ellipsoid that best fits a set of measurement points and this method is outlined in the following section. Again we are faced with the problem of needing the \( q \) parameters to be able to draw the ellipsoid so we must convert the \( P \) parameters to \( q \) parameters. The problem of deriving \( P \) parameters in terms of \( q \) parameters is straight forward; however, deriving \( q \) parameters in terms of \( P \) parameters is quite challenging. Appendix B.2.1 presents a method for finding \( P \) from \( q \) and Appendix B.2.2 presents two methods of finding \( q \) from \( P \).

4.2.2 Fitting an Ellipsoid to Measurement Points

Fitting an ellipsoid to measurement points follows the exact same steps as fitting an ellipse to measurement points and is discussed by Li and Griffiths in [24]. In 3D, our stack of 2D segmented contours create a cloud of points, \( [x_i \ y_i \ z_i]^T, (i = 1...n) \), to which we attempt to fit an ellipsoid. We find the \( P \) vector of the best fit ellipsoid by using the Least Squares Minimization technique (refer to equation 2.18 and Appendix C.1) as follows:

\[ Cost = \sum_{i=1}^{n} (F(V_i, P))^2 = ||SP||^2 \quad (4.14) \]

where, as in 2D, \( S = [V_1^T \ V_2^T \ ... \ V_n^T]^T \), \( P = [P_1 \ P_2 \ P_3 \ P_4 \ P_5 \ P_6 \ P_7 \ P_8 \ P_9 \ P_{10}] \), and \( V_i = [x_i^2 \ y_i^2 \ z_i^2 \ 2^1 x_i y_i z_i \ 2^1 x_i^2 z_i \ 2^1 y_i^2 z_i \ 2^1 x_i y_i^2 \ 2^1 x_i^2 y_i \ 2^1 y_i^2 x_i \ 2^1 z_i^2 \ 1] \). As explained in the previous section, the general conic represents an ellipsoid if the \( P \) vector satisfies the constraint [24]:

\[ \alpha J - I^2 > 0. \quad (4.15) \]

In our application we are dealing with prostate shapes whose small radii are greater than half
4.2 Ellipsoid Fitting

their large radii. For this subset of ellipsoids Li and Griffiths [24] state:

*Given a P vector, the constraint \( 4J - I^2 > 0 \) is satisfied if and only if the corresponding quadratic surface is an ellipsoid with \( r_{\text{small}} \geq \frac{1}{2}r_{\text{large}} \).*

Therefore, we use \( 4J - I^2 = 1 \) as our constraint, which can be written in matrix form as \( P^TNP = 1 \).

\[
N_1 = \begin{bmatrix}
-1 & 1 & 1 & 0 & 0 & 0 \\
1 & -1 & 1 & 0 & 0 & 0 \\
1 & 1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & -4 & 0 & 0 \\
0 & 0 & 0 & 0 & -4 & 0 \\
0 & 0 & 0 & 0 & 0 & -4
\end{bmatrix}
\]

\[
N = \begin{bmatrix}
N_1 & 0_{6 \times 4} \\
0_{4 \times 6} & 0_{4 \times 4}
\end{bmatrix}
\]

(4.16)

We can now re-state our ellipsoid fitting problem as follows. Given a set of measurement points in 3D, we would like to find the \( P \) vector that minimizes \( Cost = ||SP||^2 \) subject to the constraint \( P^TNP = 1 \). Using Lagrangian Multipliers (explained in Appendix D) we get the following set of equations that we must solve for \( \lambda \).

\[
S^TSP = \lambda NP
\]
\[
P^TNP = 1
\]

(4.17)

The above equation can be re-written as:

\[
(N - \mu S^T S)P = 0
\]

(4.18)

The \( P \) vector for the best fit ellipsoid is the eigenvector corresponding to the unique positive eigenvalue, \( \mu \), in the above GEP. The reasoning for this follows directly from the 2D case.
Chapter 5

Study Results and Discussion: 3D Segmentation

In this study, we evaluated the 3D semi-automatic prostate segmentation algorithm outlined in the previous chapter. In section 5.1 we give an overview of the study setup. Section 5.2 outlines the distance and volume based validation techniques and results along with speed performance. Section 5.3 gives a discussion of these results.

5.1 Study Outline

Ten TRUS prostate volumes (also referred to as cases), each containing between 10 to 14 transverse slices from base to apex, were chosen from a database at the BCCA. All image volumes were gathered using the standard image acquisition procedure at BCCA and standard equipment (Siemens Sonoline Prima with 5MHz biplane TRUS probe). As in 2D, the images were 480x640 with pixel size of 0.18 x 0.18mm. All images had low resolution and contained a superimposed needle grid; however, we attempted to choose volumes with clearly visible midglands so as to give the computer algorithm a fair chance and a good place to start. The prostate near the base and apex was not visible in any of the volumes - this is the essence of the difficulty in 3D segmentation. It is possible to obtain better images using more advanced TRUS machines (ie, 3D GE Volusion machine with a real-time 4D transducer); however, we wanted to see how well the 3D segmentation algorithm would perform, compared to manual segmentation, with BCCA’s current image acquisition procedure. Although 3D imaging technology is not yet mainstream, our algorithm can easily be adjusted
to work with 3D images and will have even better performance due to the finer inter-slice spacing.

5.1 Study Outline

5.1.1 Study Setup

This study was comprised of three observers; two non-expert observers (observers 1 and 2) who were trained for manual segmentation by radiation oncologists and radiologists, and one expert observer (observer 3) who is the head of prostate brachytherapy at BCCA. Before starting the study the non-expert observers practiced manual segmentation on many volumes (other than the 10 volumes used here) in order to gain efficiency and repeatability. The study was carried out as follows (refer to Figure 5.1). All three observers manually segmented the prostate contour in each slice of every case. For each case they also choose 5 initialization points to be used for the segmentation algorithm. Two weeks later, the same observers manually segmented and initialized 5 of the original 10 cases for the purpose of evaluating intra-observer variability. It should be noted that each case had \( n \) slices (where \( n \) was different for each case) and each slice was associated with \( m \) manually selected points (where \( m \) was different for each slice and for each observer). The semi-automatic segmentation algorithm produced 120 points on each slice of every case.

The segmentation algorithm was written in Matlab 7 (Release 14) and executed on an Intel Pentium M PC running at 2.0 GHz with 512MB of RAM.

**Initial 10 cases**

![Manual Segm Initialization](image)

**Two weeks later: Repeat of 5 cases**

![Manual Segm Initialization](image)

Figure 5.1: Each of the three observers manually segmented and initialized the initial 10 cases. Two weeks later, each observer manually segmented and initialized 5 of the original 10 cases.
5.1 Study Outline

5.1.2 Removing Bias

In this study, the following steps were taken to minimize subjective bias:

1. All identifying information was removed on all slices.

2. Observers carried out segmentation and initialization independently from each other.

3. The order in which the cases were given to observer 1 was different than the order given to observer 2 and so on.

4. For a given observer, the volumes to be manually segmented and the volumes to be initialized were not presented in the same order.

5. For a given observer, the order of the initial 10 volumes was different from the order of the repeat 5 volumes.

5.1.3 Segmentation Guidelines for Observers

There were no restrictions on the number of manually selected points, the location of the points, the distance between points or the starting point. The reason for this was because we wanted the manual segmentation to resemble as closely as possible the manual contours created at BCCA on a daily basis. Since the prostate boundary is not distinguishable at the base and apex, BCCA oncologists are required to abide by the following segmentation rules:

**Base:** Draw a circle, centered at the center of the midgland slice, of minimum 1.5 cm radius.

**Apex:** Draw a circle, centered at the center of the midgland slice, of minimum 1.0 cm radius.

For semi-automatic segmentation, the observers were required to choose 5 points on a slice close to the midgland where the prostate contour was visible (for consistency all observers initialized slice Base+2.5 for every case). The observers chose one initialization point at the center of the TRUS axis, one at the center of the prostate and three around the boundary of the prostate in the manner shown in Figure 5.2. These initialization points were then fed into the segmentation algorithm to find the computer generated 3D contour.
5.2 Validation Methods and Results

In the 2D study, we used distance and area based metrics to validate our results - here we extend these validation techniques to 3D. In Section 5.2.1, we use distance based metrics to see: i) if the computer generated contours differ from the gold standard as much as different observers' manual segmentations differ from each other, and ii) if the computer algorithm is more repeatable than the observers. In Section 5.2.2, we find the sensitivity and accuracy of the segmentation algorithm by comparing the computer segmented volume to the average manually segmented volume. Section 5.2.3 compares the speed of the 3D segmentation algorithm to manual segmentation.

5.2.1 Distance Based Metrics

In the following sections we compare many curves whose number of points, location of points and order of points are different. We need to find corresponding points on two different curves so that we can use distance based metrics to compare the two curves. In Figure 5.3, the less densely populated curve (o-marks) is the reference curve and the more densely populated curve (x-marks) is the secondary curve. For every point on the reference curve we find the corresponding point on the secondary curve by finding the intersection of two lines: i) the line passing from the center of the prostate through the reference point, and ii) the line passing through two consecutive points on the reference curve.
the secondary curve that fall on either side of the reference point. The image to the right in Figure 5.3 shows the final result. For all the distance based analysis we will assume that all curves being compared have gone through this process and therefore have corresponding points. Note that we always choose the reference curve to be the less densely populated curve.

Figures 5.4 and 5.5 show the result of the semi-automatic segmentation algorithm on a sample case where the first subfigure is BASE+0.5, the subfigure to its right is BASE+1.0 and so on until the last subfigure which is the apex (BASE+4.5). Appendix G shows the segmentation results for every volume in this 3D study. It would be desirable to assess the algorithm's performance on the base, midgland and apex separately to be able to localize and fix problems in the future. For this reason we have split up all the proceeding analysis into 'Base slices' (which are the initial 2 slices), 'middle slices' and 'Apex slices' (which are the final two slices). To assess the overall performance we also show the results for 'all slices' analyzed together.

5.2.1.1 Comparing Difference Between Semi-automatic Segmentation and the Gold Standard to Inter-observer Variability

We would like to compare computer generated contours to a gold standard to assess the performance of the algorithm. Then we would like to put this comparison in perspective by looking at the variability in manual segmentation by different observers (inter-observer variability). Our goal is to show that the semi-automatic segmentation differs from a gold standard no more than manual
5.2 Validation Methods and Results

Figure 5.4: A 3D comparison of semi-automatic segmentation (*astrix marks*) to manual segmentation (*circle marks*).

segmentation by one observer differs from manual segmentation by another observer. For every slice in every case we carry out the following three steps:

1. Find the average of the three manual contours in order to have one gold standard for comparison.

2. Compare the three computer generated contours, each initialized by a different observer, to the gold standard from step 1.

3. Compare the difference between semi-automatic segmentation and the gold standard to inter-observer variability.

The above steps are explained in more detail below.

**Step 1:** Finding the gold standard by averaging three manual contours
5.2 Validation Methods and Results

For any of the $n$ slices in any of the 10 cases we would like to find one average manual contour (gold standard) from three manual contours by carrying out the following steps:

1. Find corresponding points on all three contours as explained previously and shown in Figure 5.3.

2. For each set of corresponding points find the average of the $x$ and $y$ coordinates (refer to Figure 5.6). The resulting curve is the average manual contour (gold standard).

**Step 2: Comparing three computer generated contours to the gold standard**

For each slice in any of the 10 cases we have three computer generated contours which were initialized by three different observers. For notational convenience, let us call the gold standard $GS$.
5.2 Validation Methods and Results

Figure 5.6: Finding the average manual contour (solid line) from three manual contours by three different observers (square marks, x-marks and o-marks).

and each of the computer generated contours Comp1, Comp2 and Comp3 relating to initialization by observer 1, 2 and 3 respectively. For a given slice, we would like to compare Comp1 to GS, Comp2 to GS and Comp3 to GS. Since there are a large number of variables to keep track of we have created a stacking system, shown in Figure 5.7, in which we organize the data for Comp1, Comp2 Comp3 and GS in 4 different stacks (3D matrices). Each stack contains 10 cards and each card corresponds to one of the 10 cases. Each card contains 13 rows and each row corresponds to a slice in the case. Each row contains 40 entries and each entry corresponds to the radial distance from the center of the prostate to a point on the slice. It should be noted that some cases have \( n < 13 \) slices and some slices have \( m < 40 \) points (where 13 and 40 were the upper bounds). In this case the extra entries are zero padded.

As an example, card 1, row 2, column 5 contains the radial distance from the center of the prostate to the 5th point in the 2nd slice of case 1. Figure 5.7 shows how to compare comp1 to GS, on a point-by-point basis, by comparing the radial distance for the first point in case 1, slice 2 of GS (which is 56.2) to the first point in case 1, slice 2 of comp1 (which is 47.0). Every point in the GS stack and the comp1 stack can be compared in this manner. It should be noted that due to the technique of finding corresponding points on comp1 with respect to GS, row s, card c in the comp1 stack will have the same number of entries as row s, card c in the GS stack; therefore point-by-point comparison is valid and possible.
To compare \( \text{comp}_1 \) to \( GS \) we find the Pearson correlation, which indicates the strength of a linear relationship between two random variables, at different regions of the prostate. In the first subfigure of Figure 5.8 we've plotted the first two rows of the \( GS \) stack, for every case, on the horizontal axis and the first two rows of the \( \text{comp}_1 \) stack, for every case, on the vertical axis. The second and third subfigures contain the correlation for the midgland and the final two slices respectively. Note the smaller radii in the initial two and final two slices and larger radii in the midgland. Also note that there are far fewer points in the initial two and final two slices than in the midgland slices. The final subfigure shows the overall correlation between \( \text{comp}_1 \) and \( GS \). We make the following conclusion from this point-by-point comparison:

- Correlation is strongest in the middle slices (correlation coefficient is 0.8807); therefore the segmentation algorithm is closest to the \( GS \) in this region.

- Correlation is weakest in the initial two slices (correlation coefficient is 0.7456); however, the correlation coefficient is still large enough to indicate a linear relationship.

The Pearson correlation coefficients for different observers and for different regions of the prostate are presented in Table 5.1. We can see from this table that there is a strong correlation between semi-automatic segmentation and \( GS \) for all the observers; however, no one observer has consistently better results than any other observer. Since observers 1 and 2 are non-expert and observer 3 is expert, this could imply that the segmentation algorithm is not sensitive to the initialization points or to the experience of the observer making the initialization points.

### Step 3: Comparing difference between semi-automatic segmentation and \( GS \) to inter-observer variability

<table>
<thead>
<tr>
<th></th>
<th>Observer 1</th>
<th>Observer 2</th>
<th>Observer 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial 2 Slices</td>
<td>0.7456</td>
<td>0.7093</td>
<td>0.7373</td>
</tr>
<tr>
<td>Middle Slices</td>
<td>0.8807</td>
<td>0.8856</td>
<td>0.8541</td>
</tr>
<tr>
<td>Final 2 Slices</td>
<td>0.8251</td>
<td>0.8276</td>
<td>0.7818</td>
</tr>
<tr>
<td>All Slices</td>
<td>0.9053</td>
<td>0.9057</td>
<td>0.8877</td>
</tr>
</tbody>
</table>

Table 5.1: Pearson correlation coefficients (correlating computer to manual segmentation) for different regions of the prostate and for different observers.
Figure 5.7: Comparing each computer generated contour to the gold standard on a point-by-point basis to evaluate the segmentation algorithm (all units are millimeters).
5.2 Validation Methods and Results

The difference between semi-automatic segmentation and \( GS (\Delta_{SA-GS}) \) for case \( c (c = 1...10) \), slice \( s (s = 1...n) \), point \( p (p = 1..m) \) is found according to:

\[
\Delta_{SA-GS}(c, s, p) = \frac{|comp_1(c, s, p) - GS(c, s, p)|}{3} + |comp_2(c, s, p) - GS(c, s, p)| + |comp_3(c, s, p) - GS(c, s, p)|/3 \tag{5.1}
\]

Next, we need to find the average proportion of disagreement, in manual segmentation, between the three observers (inter-observer variability). As in step 2, we organize the manual segmentation by observers 1, 2 and 3 into three stacks called \( Manual_1 \), \( Manual_2 \) and \( Manual_3 \). Referring to Figure 5.9, we find the inter-observer variability (IOV) for case \( c \), slice \( s \), point \( p \) as follows:
5.2 Validation Methods and Results

\[
IOV(c, s, p) = +\left|\text{Manual}_2(c, s, p) - \text{Manual}_3(c, s, p)\right| + \frac{\left|\text{Manual}_3(c, s, p) - \text{Manual}_1(c, s, p)\right|}{3}
\] (5.2)

It should be noted that \(\text{Manual}_1\), \(\text{Manual}_2\), \(\text{Manual}_3\), \(\text{comp}_1\), \(\text{comp}_2\), \(\text{comp}_3\) and \(GS\) all have corresponding points so that for card \(c\), row \(s\), they all have the same number of entries; therefore, we can make valid comparisons. At this point we have \(10 \times n \times m\) measures of inter-observer variability and \(10 \times n \times m\) measures for the difference between semi-automatic segmentation and \(GS\). The average \(IOV\) and \(\Delta_{SA-GS}\) for the initial two slices, middle slices, final two slices and all slices are presented in Table 5.2 along with standard deviations and maximum values. From this table we make the following observations:

- **Computer algorithm and observers have more difficulty segmenting the base and apex than the midgland.** The \(IOV\) at the base (2.27±1.34) is quite poor since observers are free to choose any radius greater than 15mm. Refer to Figure 5.10 for three different manual segmentations of the same base slice. The \(IOV\) results at the midgland (1.31±0.86) are better than at the base which is due to the clear prostate boundary in these slices.

![Figure 5.9](image)

Figure 5.9: Finding the inter-observer variability for case \(c\), slice \(s\), point \(p\) where \(rm_i\) is \(\text{Manual}_i(c, s, p)\). Manual segmentation by observers 1, 2 and 3 are represented by (square marks), (o-marks) and (x-marks) respectively.
5.2 Validation Methods and Results

<table>
<thead>
<tr>
<th></th>
<th>$\Delta_{SA-GS}$ (MAD) (mm)</th>
<th>$IOV$ (MAXD (mm))</th>
<th>$IOV - \Delta_{SA-GS}$ p-value</th>
<th>CI (95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>1.34±0.89</td>
<td>2.27±1.34</td>
<td>0.93±1.6</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Slices</td>
<td>5.09</td>
<td>7.28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle</td>
<td>1.21±0.78</td>
<td>1.31±0.86</td>
<td>0.10±1.13</td>
<td>0.005</td>
</tr>
<tr>
<td>Slices</td>
<td>4.27</td>
<td>5.30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apex</td>
<td>1.54±0.83</td>
<td>1.64±1.21</td>
<td>0.11±1.43</td>
<td>0.174</td>
</tr>
<tr>
<td>Slices</td>
<td>4.29</td>
<td>6.44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>1.28±0.82</td>
<td>1.56±1.11</td>
<td>0.29±1.34</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Slices</td>
<td>5.09</td>
<td>7.28</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2: Comparing the average difference between semi-automatic segmentation and the gold standard ($\Delta_{SA-GS}$) to inter-observer variability ($IOV$) for different regions of the prostate.

Figure 5.10: Degree of inter-observer variability at the base slice of case 1. Manual segmentations by observer 1 (solid line), observer 2 (dashed line) and observer 3 (dotted line).

- The difference between computer segmentation and the gold standard is in the range of inter-observer variability. At BCCA, treatment plans would be designed for observer 1, observer 2 and observer 3's manually segmented prostate volumes. The error in semi-automatic segmentation is actually lower, for all regions of the prostate, than the $IOV$ meaning that if a treatment plan was designed for the semi-automatically segmented volume no one could tell whether the original segmentation was done by the computer or the observer.

To check the statistical soundness of our analysis we used a two sample t-test on paired data ($IOV$ and $\Delta_{SA-GS}$) to find p-values and 95% Confidence Intervals (CI). Note that the difference between $IOV$ and $\Delta_{SA-GS}$ is always positive indicating that the error in semi-automatic segmentation is actually lower than the $IOV$. The data is statistically significant in the initial two slices, middle
5.2 Validation Methods and Results

slices and when looking at all the slices together. In the final two slices there is insufficient data to conclude statistical significance. We had sufficient data so that all values for $IOV - \Delta_{SA-GS}$ were within the 95% CI.

Figure 5.11 further illustrates the performance of the segmentation algorithm, relative to $IOV$, using notch plots. In each sub-figure, the lower and upper lines in the notch plot represent the inter-quartile range (the data between the 25th and the 75th percentile range in a histogram). The line in the middle is the median of the data range and an off-center median line would indicate a skew in the data. The whiskers show the entire range of the data with the exception of the outliers. By default, in MATLAB, outliers are data points that are more than 1.5 times the inter-quartile range away from the top or bottom of the box and are represented by plus signs above or below the whiskers.

In each notch plot we compare $A_{comp1-cs}$ (the difference between $comp_1$ and GS), $A_{comp2-GS}$ and $A_{comp3-GS}$ to $IOV$. We can see that in all regions of the prostate the median difference between semi-automatic segmentation and GS is lower than or equal to $IOV$. The only exception is $A_{COMP3-GS}$ in the final two slices which explains why there is a high p-value (0.174) in that region. We should take note of the scale in these plots and put them in perspective. All medians are less than 2mm, which is a very small value compared to the finest accuracy available in planning (5mm spacing of the needle guide template).

5.2.1.2 Comparing Repeatability of Semi-automatic Segmentation to Intra-observer Variability

If we recall, in this study 10 cases were initially manually segmented and initialized by three observers and then 5 of the original 10 cases were re-segmented and initialized two weeks later. In this section we only deal with those cases that were re-segmented and we would like to compare the intra-observer variability (average proportion of disagreement in one observer’s multiple manual segmentations of the same case) to the computer algorithm’s repeatability (ability to re-generate the same contour given two different sets of initialization points by the same observer). Basically, we would like to see whether this computer algorithm is better at reproducing results than humans.

We will organize the data into stacks again for convenience; however, this time our stacks only have 5 cards because we are dealing with 5 cases. Referring to Table 5.3, for each observer we have the the initial manual contour (IMC), repeated manual contour (RMC), initial computer contour

...
5.2 Validation Methods and Results

Looking at Base Slices for all Cases

Looking at Middle Slices for all Cases

Looking at Apex Slices for all Cases

Looking at All Slices for all Cases

Figure 5.11: Notched box plots comparing the difference between \( \text{comp}_1 \) and \( GS \), \( \text{comp}_2 \) and \( GS \), \( \text{comp}_3 \) and \( GS \) to IOV for different regions of the prostate.

(IICC) and repeated computer (RCC) contour for each of the 5 cases. The intra-observer variability (IAOV) is calculated as follows:

\[
\text{IAOV}(c, s, p) = \text{IMC}(c, s, p) - \text{RMC}(c, s, p)
\] (5.3)

The computer repeatability (Comp_repeat) is found as follows:

\[
\text{Comp}_\text{repeat}(c, s, p) = \text{ICC}(c, s, p) - \text{RCC}(c, s, p)
\] (5.4)
5.2 Validation Methods and Results

If the observer or the computer algorithm is good at reproducing results then these subtractions will result in very small values. At this point, for each observer, we have $5 \times n \times m$ measures for $IAOV$ and $5 \times n \times m$ measures for $Comp_{repeat}$. The average and standard deviations of $IAOV$ and $Comp_{repeat}$ are presented in Table 5.4 for different observers and for regions of the prostate. We carry out a 2 sample, paired ($IOV$ and $Comp_{repeat}$) t-test with the null hypothesis that the difference between the two sample means is zero. As we can see from $IAOV - Comp_{repeat}$, the difference between the means is not zero and the p-values (all < 0.001) suggest rejecting the null hypothesis. From this table we can make the following observations:

- **Computer algorithm is always more repeatable.** The difference between the means of $IAOV$ and $Comp_{repeat}$ is always positive and the 95% CI shows that 95% of the time this difference will be positive for all regions of the prostate and for all observers. This implies that computer segmentation will always be more reproducible than manual segmentation. As well, note the consistently smaller variance in $Comp_{repeat}$ as compared to $IAOV$.

- **Computer algorithm is consistent in repeatability at different regions of the prostate.** Looking at $IAOV$ at different regions of the prostate we can see that the observers can reproduce the same contours at the midgland more consistently than at the base and apex. In contrast, the computer algorithm is as repeatable at the base and apex as it is at midgland.

- **Computer algorithm is more repeatable than the most experienced observer.** The $IAOV$ for observers 1 and 2 (both non-expert) is higher than the $IAOV$ for observer 3 (expert). This

<table>
<thead>
<tr>
<th></th>
<th>Obs1</th>
<th>Obs2</th>
<th>Obs3</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manual Segmentation</td>
<td>RMC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAOV</td>
<td>=</td>
<td>=</td>
<td>=</td>
</tr>
<tr>
<td>ICC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer Segmentation</td>
<td>RCC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Comp_{repeat}$</td>
<td>=</td>
<td>=</td>
<td>=</td>
</tr>
</tbody>
</table>

Table 5.3: Comparing the repeatability of the computer algorithm to intra-observer variability in manual segmentation.


5.2 Validation Methods and Results

<table>
<thead>
<tr>
<th>Slices</th>
<th>( IAOV )</th>
<th>( CR )</th>
<th>( IAOV - CR )</th>
<th>( p)-value</th>
<th>CI (95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Observer 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Base</td>
<td>1.87±1.40</td>
<td>0.48±0.36</td>
<td>1.39±1.38</td>
<td>&lt;0.001</td>
<td>(1.16,1.62)</td>
</tr>
<tr>
<td>Middle</td>
<td>1.09±0.86</td>
<td>0.54±0.50</td>
<td>0.55±0.91</td>
<td>&lt;0.001</td>
<td>(0.46,0.63)</td>
</tr>
<tr>
<td>Apex</td>
<td>1.32±1.24</td>
<td>0.61±0.66</td>
<td>0.70±1.30</td>
<td>&lt;0.001</td>
<td>(0.49,0.91)</td>
</tr>
<tr>
<td>All</td>
<td>1.28±1.10</td>
<td>0.55±0.52</td>
<td>0.74±1.14</td>
<td>&lt;0.001</td>
<td>(0.65,0.82)</td>
</tr>
<tr>
<td></td>
<td>Observer 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Base</td>
<td>1.70±1.66</td>
<td>0.80±0.65</td>
<td>0.90±2.02</td>
<td>&lt;0.001</td>
<td>(0.73,1.21)</td>
</tr>
<tr>
<td>Middle</td>
<td>1.30±1.15</td>
<td>1.02±1.07</td>
<td>0.28±1.86</td>
<td>&lt;0.001</td>
<td>(0.17,0.41)</td>
</tr>
<tr>
<td>Apex</td>
<td>1.43±1.59</td>
<td>1.03±0.97</td>
<td>0.40±2.12</td>
<td>&lt;0.001</td>
<td>(0.26,0.68)</td>
</tr>
<tr>
<td>All</td>
<td>1.40±1.36</td>
<td>1.01±1.00</td>
<td>0.39±1.97</td>
<td>&lt;0.001</td>
<td>(0.22,0.41)</td>
</tr>
<tr>
<td></td>
<td>Observer 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Base</td>
<td>0.95±1.07</td>
<td>0.43±0.31</td>
<td>0.52±1.44</td>
<td>&lt;0.001</td>
<td>(0.23,0.71)</td>
</tr>
<tr>
<td>Middle</td>
<td>0.72±0.73</td>
<td>0.40±0.28</td>
<td>0.32±1.33</td>
<td>&lt;0.001</td>
<td>(0.29,0.50)</td>
</tr>
<tr>
<td>Apex</td>
<td>1.27±1.06</td>
<td>0.44±0.35</td>
<td>0.83±1.52</td>
<td>&lt;0.001</td>
<td>(0.36,0.86)</td>
</tr>
<tr>
<td>All</td>
<td>0.89±0.91</td>
<td>0.41±0.30</td>
<td>0.48±1.29</td>
<td>&lt;0.001</td>
<td>(0.37,0.56)</td>
</tr>
</tbody>
</table>

Table 5.4: Comparing intra-observer variability to computer repeatability in different regions of the prostate and for different observers.

makes sense because we assume that with experience an oncologist can more consistently reproduce their results; however, Comp\textsubscript{repeat} is less than \( IAOV \) even for this most experienced observer.

5.2.2 Volume Based Metrics

We can find the volume of the prostate from a set of 2D contours using a method called Planimetry. The equation for Planimetry is:

\[
Volume = \left\{ \sum_{i=1}^{n} A(i) \right\} \times d \tag{5.5}
\]

where there are \( n \) slices in a volume, each with area \( A \) and uniform inter-slice distance \( d \). This equation essentially says that we can find the volume of an ellipsoid by approximating it as a series of cylinders.

In 3D, we superimpose the manually segmented prostate volume, \( \Upsilon_m \), on top of the computer generated prostate volume, \( \Upsilon_a \), and call the volume common to both the True Positive volume
5.2 Validation Methods and Results

\( TP_{vol} \). As in 2D, the volume common to \( \Gamma_m \) but not to \( \Gamma_a \) is the False Negative (\( FN_{vol} \)) volume and the volume common to \( \Gamma_a \) but not \( \Gamma_m \) is the False Positive (\( FP_{vol} \)) volume. Then the sensitivity of the segmentation algorithm for a given prostate volume becomes:

\[
Sensitivity = \frac{TP_{vol}}{\text{manually segmented volume}} = \frac{\left\{ \sum_{i=1}^{n} TP(i) \right\} * d}{\left\{ \sum_{i=1}^{n} A_m(i) \right\} * d} = \frac{\sum_{i=1}^{n} TP(i)}{\sum_{i=1}^{n} A_m(i)} \tag{5.6}
\]

where \( TP(i) \) is the True Positive area and \( A_m(i) \) is the area of the manual contour for each of the \( n \) slices in the volume. We can see from the second line in the above equation that the inter-slice distance cancels out and plays no role in the final answer. Following the same approach, the accuracy of the segmentation algorithm for a given volume is:

\[
Accuracy = 1 - \frac{\sum_{i=1}^{n} (FP(i) + FN(i))}{\sum_{i=1}^{n} A_m(i)} \tag{5.7}
\]

Table 5.5 shows the sensitivity and accuracy results when comparing the computer generated volumes to the gold standard volume for each case and for each observer. Figures 5.12, 5.14 and 5.13 show the semi-automatically segmented contours (solid line) superimposed on top of the gold standard (dashed line) for every slice in some sample volumes. Figure 5.12 shows observer 3’s segmentation of case 1 in which the sensitivity and accuracy of the algorithm were 99.6 and 75.6 percent respectively. In this figure, the semi-automatic contour is encompassing almost every gold standard contour resulting in the very high sensitivity value; however, there are a lot of false positive areas which contribute to the lower accuracy value. In Figure 5.14, observer 2’s segmentation of case 4 resulted in sensitivity and accuracy results of 86.9 and 84.4 percent respectively. In this figure, the semi-automatic contour is encompassed by almost every gold standard contour resulting in the lower sensitivity value; however, there are a lot of false negative areas which contribute to the lower accuracy value. Figure 5.13 shows the segmentation results of case 5 by observer 3 in which sensitivity and accuracy results are 95.7 and 90.3 percent respectively. In this figure, the semi-automatic contours line up almost exactly with the gold standard resulting in the high
5.2 Validation Methods and Results

sensitivity value and there are very small false positive and false negative areas contributing to the high accuracy values.

The sensitivity and accuracy of the 3D segmentation algorithm, averaged over all cases and all observers, are 96.5 and 84.7 percent respectively. These numbers indicate:

- The algorithm is highly sensitive when picking up pixels that the gold standard specifies belong to the prostate.

- The algorithm tends to over-compensate and cover too much so that there are lots of false positives; therefore the accuracy is lower than the sensitivity.

5.2.3 3D Segmentation Algorithm Speed

The times for manual and semi-automatic segmentation of an entire prostate volume were recorded and are presented in Table 5.6. The time for manually segmenting an entire volume was approximately 15 minutes for the non-expert observers and approximately 5 minutes for the expert observer; therefore the average time for manual segmentation was approximately 11.67 minutes or 700 seconds. It took no more than approximately 10 seconds to choose 5 initialization points for the semi-automatic algorithm. The computer algorithm itself took, on average, 8.16 ± 0.43 seconds to run with more than 80% of that time being tied up in the warping step. This means that on average

<table>
<thead>
<tr>
<th>Case</th>
<th>Sens</th>
<th>Accu</th>
<th>Sens</th>
<th>Accu</th>
<th>Sens</th>
<th>Accu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>97.07</td>
<td>73.69</td>
<td>98.67</td>
<td>80.05</td>
<td>99.61</td>
<td>75.59</td>
</tr>
<tr>
<td>Case 2</td>
<td>97.46</td>
<td>84.56</td>
<td>98.36</td>
<td>84.09</td>
<td>99.04</td>
<td>79.57</td>
</tr>
<tr>
<td>Case 3</td>
<td>97.32</td>
<td>89.27</td>
<td>94.81</td>
<td>87.42</td>
<td>95.59</td>
<td>78.13</td>
</tr>
<tr>
<td>Case 4</td>
<td>97.43</td>
<td>84.37</td>
<td>86.95</td>
<td>84.04</td>
<td>94.85</td>
<td>78.89</td>
</tr>
<tr>
<td>Case 5</td>
<td>94.97</td>
<td>90.51</td>
<td>96.14</td>
<td>90.58</td>
<td>95.70</td>
<td>90.28</td>
</tr>
<tr>
<td>Case 6</td>
<td>95.82</td>
<td>88.29</td>
<td>97.41</td>
<td>89.21</td>
<td>97.63</td>
<td>88.01</td>
</tr>
<tr>
<td>Case 7</td>
<td>96.37</td>
<td>89.77</td>
<td>97.95</td>
<td>93.48</td>
<td>98.51</td>
<td>88.53</td>
</tr>
<tr>
<td>Case 8</td>
<td>95.22</td>
<td>79.43</td>
<td>96.57</td>
<td>83.85</td>
<td>98.18</td>
<td>74.38</td>
</tr>
<tr>
<td>Case 9</td>
<td>93.25</td>
<td>87.71</td>
<td>93.28</td>
<td>87.86</td>
<td>93.78</td>
<td>88.21</td>
</tr>
<tr>
<td>Case 10</td>
<td>99.56</td>
<td>81.68</td>
<td>98.06</td>
<td>87.26</td>
<td>99.19</td>
<td>83.53</td>
</tr>
</tbody>
</table>

| Avg Case | 96.45 | 84.93 | 95.82 | 86.78 | 97.21 | 82.51 |

Table 5.5: Sensitivity and accuracy of 3D semi-automatic segmentation in comparison to the gold standard.
5.2 Validation Methods and Results

Figure 5.12: Gold Standard (dashed line) vs semi-automatic (solid line) segmentation results for observer 3, case 1. (Sensitivity=99.6%, Accuracy=75.6%)

the semi-automatic segmentation algorithm, which takes approximately $18.16 \pm 0.43$ seconds from beginning to end is 37 times faster than manual segmentation. In the future the following two steps could be carried out to further improve the simulation time: i) convert the program code to C++ (making it at least 5 times faster), and ii) optimize the warping algorithm to perform faster.

<table>
<thead>
<tr>
<th></th>
<th>Manual</th>
<th>Semi-automatic (Matlab)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Time (sec)</td>
<td>Initialization (sec)</td>
</tr>
<tr>
<td>Mean</td>
<td>N/A</td>
<td>Mean</td>
</tr>
<tr>
<td>~ 700</td>
<td>N/A</td>
<td>8.16</td>
</tr>
</tbody>
</table>

Table 5.6: Average times for manual and semi-automatic segmentation of an entire prostate volume composed of approximately 10-13 slices.
5.2 Validation Methods and Results

Figure 5.13: Gold Standard (*dashed line*) vs semi-automatic (*solid line*) segmentation results for observer 3, case 5. In this case sensitivity=95.7% and Accuracy=90.28%.

Figure 5.14: Gold Standard (*dashed line*) vs semi-automatic (*solid line*) segmentation results for observer 2, case 4. In this case sensitivity=86.95% and Accuracy=84.04%.
5.3 Discussion

5.3.1 Performance

The 3D semi-automatic segmentation algorithm is accurate, repeatable, real-time, robust and requires only 5 initialization points to segment the entire prostate volume. The error in the accuracy of the segmentation algorithm is on the order of the error of manual segmentation by two different observers (inter-observer variability). As well, the segmentation algorithm is more repeatable when re-segmenting the same volume (and more consistently repeatable in different regions of the prostate) than intra-observer variability in manual segmentation by the most experienced observer. Although the algorithm is currently written in Matlab, it is approximately 37 times faster than manual segmentation and upon transfer to C++ we expect the algorithm to be almost 200 times faster than manual segmentation (performance on the order of 1 second).

5.3.1.1 Potential for Integration at BCCA

This algorithm works well with low resolution images currently used at BCCA and can be integrated easily into their setup without any changes to their image acquisition procedure. Currently most cancer centers use prostate volumes that are composed of 2D slices with 5mm inter-slice spacing. When 3D TRUS technology, with finer inter-slice spacing, becomes mainstreams our algorithm can be easily adjusted to work with such volumes for even better performance.

The potential for intra-operative integration is high because the algorithm: 1) can be easily optimized to obtain real-time speed, 2) requires only 5 initialization points to segment the entire volume, and 3) is insensitive to initialization points so that an inexperienced observer can initialize the volume while the oncologist continues un-interrupted with the implant procedure.

5.3.1.2 Algorithm Limitations

This algorithm will perform poorly on cases where the prostate boundary is not visible at the midgland; however, the inter and intra-observer variability in manual segmentation is significantly larger in such cases as well. The shape and radius of the base and apex are very simplified and more complex statistical analysis could be applied there at the sake of computational complexity.

Though it is not common, some prostates have a diamond or triangular shaped cross section and in those cases this algorithm would simply return the greatest ellipse that would encompass
the prostate. In this case the prostate would be over-dosed which is always more preferable than being under-dosed.
Chapter 6

Conclusion and Future Work

In this thesis we proposed a 2D semi-automatic segmentation algorithm for TRUS images of the prostate. Then we carried out a study to evaluate the success of the 2D segmentation algorithm on midgland, transverse TRUS images. After successful results we extended the 2D algorithm to 3D. In order to assess the 3D algorithm a large clinical study was set up using full 3D prostate volumes and multiple observers. This thesis has resulted in the following contributions:

1. **2D semi-automatic segmentation algorithm**

   (a) Algorithm steps:

   i. Choose 5 initialization points
   ii. Warp image and initialization points so that prostate boundary becomes elliptical
   iii. Fit an ellipse to warped initialization points
   iv. Perform edge detection (IMMPDA [2]) on the prostate boundary along the ellipse fit
   v. Fit a second ellipse to edge detected contour
   vi. Apply a reverse warp to the second ellipse to obtain the final prostate contour

   (b) Clinical Study:

   i. Algorithm was applied to 17 pre-operative, midgland TRUS images of the prostate
   ii. A single expert observer manually segmented all 17 images and the results were used as the gold standard
(c) Contributions and benefits:

i. Algorithm is insensitive to initialization points and can be initialized by an inexperienced observer

ii. Works well on TRUS images that have low resolution

iii. Distance based metrics between manual and semi-automatic contours showed a mean absolute difference of 0.67±0.18mm which is lower than inter-observer variability [16]

iv. Area based metrics show sensitivity and accuracy results greater than 97% and 93% respectively

v. The algorithm was two times faster than manual segmentation

2. **3D semi-automatic segmentation algorithm**

(a) Algorithm steps:

i. Choose 5 initialization points on midgland only

ii. Warp all 2D slices and initialization points so that prostate shape becomes ellipsoidal

iii. Perform 2D segmentation on midgland slice

iv. Propagate the midgland contour as initialization to adjacent slices and iteratively segmented every 2D slice up to, but not including, the base and apex

v. Fit an initial ellipsoid to all 2D contours from the previous step

vi. Segment base and apex using circles (of 18mm and 15mm radii respectively) centered about the major axis of the ellipsoid from the previous step

vii. Fit a second ellipsoid to all 2D contours including base and apex

viii. Slice the final ellipsoid at 5mm increments and apply a reverse warp to all resulting 2D contours to get the segmented 3D contour

(b) Clinical Study:

i. Algorithm was applied to 10 pre-operative TRUS prostate volumes, each with 10-14 slices

ii. Three observers (two non-expert and one expert) were used in the study

iii. In the first week all observers manually segmented and initialized (for the purpose of semi-automatic segmentation) all 10 volumes
iv. In the second week all observers manually re-segmented and re-initialized 5 of the original 10 volumes (for the purpose of evaluating intra-observer variability)

(c) Contributions and benefits:

i. Requires only 5 initialization points to segment entire prostate volume

ii. Can be initialized by an inexperienced observer since algorithm is insensitive to initialization points

iii. Robust algorithm works well on low resolution images

iv. Has potential for intra-operative use and integration into BCCA's current brachytherapy procedure

v. Pearson correlation between manual segmentation and semi-automatic segmentation shows a linear relationship (Pearson coefficient of at least 0.7)

vi. Segmentation results are on the order of inter-observer variability

vii. Segmentation algorithm’s repeatability is better than intra-observer variability

viii. Sensitivity and accuracy of the algorithm are approx 97% and 85% with respect to manual segmentation

ix. Although the program is written in Matlab and not optimized it is currently one of the fastest 3D prostate segmentation algorithms and upon transfer to C++ the algorithm will perform on the order of milliseconds (almost 200 times faster than manual segmentation)

3. Novel warping algorithm. A technique was introduced to undo the compression effect of the TRUS probe on the prostate images resulting in elliptical prostate cross sections. A polar coordinate system was used with origin at the center of the rectum in the TRUS image and angle of 0° along the horizontal. In this technique, the pixels of an image were warped radially according to a Gaussian function where pixels close to the rectum were stretched maximally and pixels far away from the rectum were left alone. In the angular direction, the pixels were warped according to a \( \sin(\theta) \) function with maximum stretch along \( \theta = 90^\circ \) and minimum stretch along \( \theta = 0^\circ \). The purpose of having elliptical prostate cross sections was to be able to fit ellipses to the prostate shape.

4. Ellipse/ellipsoid fitting applied to the prostate contour. We used a priori knowledge of the
prostate shape and the extensive literature on ellipse/ellipsoid fitting to a set of scattered points to fit ellipses/ellipsoids to the prostate contour resulting in symmetrical, smooth and convex shapes. The ellipse/ellipsoid fitting problem is convex (always returns an elliptical/ellipsoidal solution), robust, computationally inexpensive and rotational and translational invariant.

(a) \( P \) to \( q \) parameter conversions and vice versa: In both 2D and 3D we derived the \( P \) parameters from the \( q \) parameters and vice versa where the \( P \) and \( q \) parameters were used in quadratic and parametric representation of an ellipse/ellipsoid respectively.

6.1 Future Work

Possible future work and improvements to some of the algorithm’s drawbacks are suggested below:

1. The warping algorithm is computationally expensive:

   (a) Transfer the algorithm to C++ and optimize the code.

   (b) Explore to see if there are built-in image warping functions in the Visualization ToolKit\(^{TM}\) (VTK) or other programming languages which make extensive use of graphics.

2. The warping algorithm can be made more accurate:

   (a) One can use motion tracking and elasticity (the basics of elastography [31]) to track the prostate motion due to different compressions by the probe. A statistical model of this deformation can be created and used to design the warping function more accurately.

3. Overall segmentation (especially at the base and apex) could be improved:

   (a) Create a statistical shape atlas for the prostate using fiducial markers and MRI images of the prostate.

   (b) In parallel transverse TRUS images, the boundary of the midgland is easily visible yet the base is blended with the bladder neck and the apex is blended with the pelvic floor. We could obtain 3D TRUS volumes with finer inter-slice spacing and slice the volume at 5° intervals in a star pattern. In such images the contour of all slices are equally visible
resulting is equal segmentation performance in all regions of the prostate. Alternatively, we could incorporate sagittal images to be used along with transverse images in order to infer the location of the base and apex more easily.

(c) Choose more initialization points with some points at base and apex.

4. Other improvements:

(a) It would be informative to design a study with prostate volumes where even the midgland slice contains poorly visible boundaries to further test the robustness of this algorithm.

(b) This algorithm, combined with seed localization [34], real-time dosimetry [14] and elastography [31] technology (all developed in our group), can be incorporated into a research workstation to be used for clinical procedures. This research station can also be used as a visualization tool for training residents since the learning curve for prostate brachytherapy is very steep.

(c) We are currently running a clinical study at the BCCA to test the feasibility of replacing manual segmentation with semi-automatic segmentation. To do this, radiation oncologists are designing dosimetric plans for the semi-automatically segmented volumes presented in this paper. These treatment plans are then transferred to their corresponding manually segmented volumes to see whether the resulting dosimetric parameters ($V_{100}$, $D_{90}$, conformality index etc [30]) are satisfactory. If they are, then semi-automatic segmentation can indeed replace manual segmentation. The results have been promising so far and will be presented in a paper when the study is complete.
Bibliography


Appendix A

Ellipse/Ellipsoid Reference Frame Conventions

In this Appendix, we define this thesis' reference frame conventions for representing ellipses and ellipsoids in two and three dimensional space.

A.1 Reference Frame Convention in 2D

The base frame will be \{^1\alpha, ^1\mathcal{C}\} where \(^1\alpha\) represents the center and the columns of \(^1\mathcal{C}\) represent the coordinate axes of frame 1. With respect to this frame the prostate is rotated and translated as can be seen in Figure A.1b. The prostate center frame will be \{^2\alpha, ^2\mathcal{C}\} and with respect to this frame the prostate is rotated but not translated. This frame is translated with respect to the base frame. The final prostate frame, \{^5\alpha, ^5\mathcal{C}\}, is rotated with respect to the center frame and is rotated and translated with respect to the base frame. With respect to this frame the prostate is neither rotated nor translated. We define a point in reference frame \(i\) as \(^iX = [x \ y]^T\).

If we wanted to rotate (but not translate) every point \([^5x \ ^5y]^T\) in frame 5, we would use the following convention:

\[
\begin{bmatrix}
^2x \\
^2y
\end{bmatrix}
= \begin{bmatrix}
\cos(\theta) & -\sin(\theta) \\
\sin(\theta) & \cos(\theta)
\end{bmatrix}
\begin{bmatrix}
^5x \\
^5y
\end{bmatrix}
\]  

(A.1)

In this convention, if we take all the points in the frame \{^5\alpha, ^5\mathcal{C}\} and rotate them counter-clockwise by an angle \(\theta\), the new untranslated, but rotated coordinate system will be \{^2\alpha, ^2\mathcal{C}\}. Refer to Figure A.1a for an illustration.
Next, we take the rotated ellipse in frame 2 and translate it with the following convention:

\[
\begin{bmatrix}
  1x \\ 1y
\end{bmatrix} = \begin{bmatrix}
  x_o \\ y_o
\end{bmatrix} + \begin{bmatrix}
  2x \\ 2y
\end{bmatrix}
\]  \hspace{1cm} (A.2)

where \( X_o = [x_o \ y_o]^T \) is an offset that is added to every point in frame 2 to get every point in frame 1. Alternatively, we can say that \( X_o \) is the center of the ellipse in the \( \{1\hat{o}, 1\hat{C}\} \) reference frame. Refer to Figure A.1b for an illustration. The final result is a rotated and translated ellipse with respect to reference frame 1 as follows:

\[
\begin{bmatrix}
  1x \\ 1y
\end{bmatrix} = \begin{bmatrix}
  x_o \\ y_o
\end{bmatrix} + \begin{bmatrix}
  \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta)
\end{bmatrix} \begin{bmatrix}
  5x \\ 5y
\end{bmatrix}
\]  \hspace{1cm} (A.3)

A.2 Reference Frame Convention in 3D

In 3D, we will use the same reference frames as we used in two-dimensions. Here the \( \{1\hat{o}, 1\hat{C}\}, \{2\hat{o}, 2\hat{C}\} \) and \( \{5\hat{o}, 5\hat{C}\} \) frames represent the rotated/translated, rotated/untranslated and unrotated/untranslated reference frames respectively. Here again we define a point in reference frame \( i \) as \( ^iX = [^ix \ ^iy \ ^iz] \).

Referring to Figures A.2 and A.3, we define the convention to get from \( \{5\hat{o}, 5\hat{C}\} \) to \( \{1\hat{o}, 1\hat{C}\} \) as follows:

1. Rotate \( 5\hat{C} \) about \( 5x \) by \( \theta \) → get frame \( \{4\hat{o}, 4\hat{C}\} \)
2. Rotate \( 4\hat{C} \) about \( 4y \) by \( \phi \) → get frame \( \{3\hat{o}, 3\hat{C}\} \)
3. Rotate \( 3\hat{C} \) about \( 3z \) by \( \beta \) → get frame \( \{2\hat{o}, 2\hat{C}\} \)
4. Add \( X_o \) to any point in frame 2, \( ^2X \), to get that same point in frame 1, \( ^1X = X_o + ^2X \)

Combining steps 1-3 gets us from frame 5 to frame 2 as follows:

\[
^2X = U^5X
\]  \hspace{1cm} (A.4)

where \( U \) is the combination of the three rotation matrices. Taking step 1 as an example, each rotation matrix can be written in the following form:

\[
\exp(\theta^5x_x)
\]  \hspace{1cm} (A.5)
Finally, if we combine all 4 steps together we get:

\[ 1^X = X_0 + \big[ \exp \left( \beta^3 x \right) \exp \left( \phi^4 y \right) \exp \left( \theta^5 x \right) \big] \ 5^X \quad (A.6) \]

where \( X_0 = [x_0 \ y_0 \ z_0] \) is the ellipsoid center with respect to frame 1. According to this equation, given any point in frame 5, we can rotate it by \( U \) and translate it by \( X_0 \) to get that same point in frame 1.
A.2 Reference Frame Convention in 3D

Figure A.2: Convention for reference frames in 3D

Figure A.3: Moving between reference frames in 3D
Appendix B

Ellipse/Ellipsoid Parameter Conversions

For visualization purposes, it is important to be able to express the $q$ parameters of an ellipse/ellipsoid in terms of the $P$ parameters and vice versa. Sections B.1 and B.2 show the parameter conversion techniques for ellipses and ellipsoids respectively.

B.1 Fitting and Ellipse to Measurement Points in 2D

B.1.1 Deriving $P$ Parameters from $q$ parameters ($q$ to $P$)

Given the $q$ parameters, it is easy to find the $P$ parameters. This can be done by expanding equation 2.13, collecting like terms and letting the coefficients equal the $P$ parameters in equation 2.14. Below are the set of equations that convert the given $q$ parameters to $P$ parameters.

$$
\begin{align*}
\Phi &= -\cos(\theta)x_o - y_o\sin(\theta) \\
\Theta &= \sin(\theta)x_o - y_o\cos(\theta)
\end{align*}
$$

$$
\begin{bmatrix}
P_1 \\
P_2 \\
P_3 \\
P_4 \\
P_5 \\
P_6
\end{bmatrix} =
\begin{bmatrix}
\left(\frac{\cos(\theta)}{a}\right)^2 + \left(\frac{\sin(\theta)}{b}\right)^2 \\
2\left(\frac{\sin(\theta)\cos(\theta)}{a^2}\right) - 2\left(\frac{\sin(\theta)\cos(\theta)}{b^2}\right) \\
\left(\frac{\sin(\theta)}{a}\right)^2 + \left(\frac{\cos(\theta)}{b}\right)^2 \\
2\left(\frac{\Phi\cos(\theta)}{a^2}\right)^2 - 2\left(\frac{\Theta\sin(\theta)}{b^2}\right) \\
2\left(\frac{\Phi\sin(\theta)}{a^2}\right)^2 + 2\left(\frac{\Theta\cos(\theta)}{b^2}\right) \\
\left(\frac{\Phi}{a}\right)^2 + \left(\frac{\Theta}{b}\right)^2 - 1
\end{bmatrix}
$$

(2.2)
B.1.2 Deriving $q$ Parameters from $P$ parameters ($P$ to $q$)

The Least Squares Minimization (LSM) method from Section 2.4.2 returns the $P$ parameters of the ellipse that best fits a set of measurement points around the boundary of the prostate. In order to plot this ellipse in Matlab, we need to be able to convert the $P$ parameters to $q$ parameters. Converting from $P$ to $q$ parameters is not as straightforward as the $q$ to $P$ conversion in the previous section. Two methods for converting from $P$ to $q$ are presented here.

B.1.2.1 Method 1

Step 1: Finding the ellipse origin $[x_0, y_0]^T$

We start with the general conic representation of an ellipse in equation 2.14 and translate all the points in frame 1, using equation A.2, to get the reference frame $\{\bar{\xi}, \bar{\eta}\}$ as follows:

\[
\begin{bmatrix}
(2x + x_0)^2 & (2y + y_0)^2 & (2x + x_0) & (2y + y_0) & 1
\end{bmatrix}
\begin{bmatrix}
P_1 & P_2 & P_3 & P_4 & P_5 & P_6
\end{bmatrix} = 0
\]  

(B.3)

After expanding this equation and collecting like terms we see that there is an $2x$ and $2y$ term:

\[
P_1 2x^2 + P_2 2x_0^2 + P_3 2y_0^2 + (2x_0 P_1 + P_2 y_0 + P_4) 2x
\]
\[
+ (2y_0 P_3 + P_2 x_0 + P_5) 2y + (P_2 x_0 y_0 + P_4 x_o + P_5 y_0 + P_6) = 0
\]  

(B.4)

We know that in the $\{\bar{\xi}, \bar{\eta}\}$ frame the ellipse has no translation so we can set the coefficients of these $2x$ and $2y$ terms to zero. We then solve these equations for $x_0$ and $y_0$ in terms of the $P$ parameters to get:

\[
\begin{bmatrix}
x_0 \\
y_0
\end{bmatrix} = \frac{1}{4P_1 P_3 - P_2^2}
\begin{bmatrix}
-2P_3 P_4 + P_2 P_5 \\
P_2 P_4 - 2P_1 P_5
\end{bmatrix}
\]  

(B.5)

Note that the above equation would be undefined if $4P_1 P_3 - P_2^2 = 0$, which is in agreement with our ellipse constraint $4P_1 P_3 - P_2^2 > 0$. The equation of our ellipse in the rotated, but untranslated reference frame then becomes:

\[
P_1 2x^2 + P_2 2x_0^2 + P_3 2y_0^2 + (2x_0 y_0 + P_4 x_o + P_5 y_0 + P_6) = 0
\]  

(B.6)
Step 2: Finding the ellipse angle $\theta$

We start with equation B.6 and rotate all the points in the $\{^2\mathbf{G}, ^2\mathbf{C}\}$ frame, using equation A.1, to get the $\{^5\mathbf{G}, ^5\mathbf{C}\}$ frame. After expanding and collecting like terms we get:

\[
\begin{align*}
5x^2(P_1\cos^2(\theta) + P_2\cos(\theta)\sin(\theta) + P_3\sin^2(\theta)) &+ 5xy(2\cos(\theta)\sin(\theta)(P_3 - P_1) + P_2(\cos^2(\theta) - \sin^2(\theta))) \\
+ 5y^2(P_1\sin^2(\theta) - P_2\cos(\theta)\sin(\theta) + P_3\cos^2(\theta)) &+ (P_2x_0y_0 + P_4x_o + P_5y_o + P_6) = 0
\end{align*}
\]

(B.7)

Using trigonometric identities, setting the coefficients of the $5x5y$ term to zero (since in frame 5 the ellipse has no rotation) and making use of equation A.1, it can be shown that:

\[
\theta = \arctan\left(\frac{(P_3 - P_1) \pm \sqrt{(P_3 - P_1)^2 + P_2^2}}{P_2}\right)
\]

(B.8)

The question is now which sign do we use in the above equation? If we let $\theta_1$ be the angle in which the positive sign is used and $\theta_2$ be the angle with the negative sign, then using trigonometric identities we can prove that $\tan(\theta_1 - \theta_2) = \infty$, which implies that the two angles are 90 degrees apart. Referring to Figure B.1, we can see that any ellipse can be represented with two different sets of $(\theta, a, b)$ values where the two angles are 90 degrees apart. In conclusion, it does not matter which sign is used because the $a$ and $b$ values in step 3 will be adjusted accordingly depending on whichever angle we choose.

The equation of our ellipse in the unrotated and untranslated reference frame then becomes:

\[
\begin{align*}
5x^2(P_1\cos^2(\theta) + P_2\cos(\theta)\sin(\theta) + P_3\sin^2(\theta)) &+ 5y^2(P_1\sin^2(\theta) - P_2\cos(\theta)\sin(\theta) + P_3\cos^2(\theta)) \\
+ (P_2x_0y_o + P_4x_o + P_5y_o + P_6) & = 0
\end{align*}
\]

(B.9)

Step 3: Finding the ellipse size $(a, b)$

At this point we have the $P, x_o, y_o$, and $\theta$ for this best fit ellipse and the only remaining variables are $a$ and $b$. If we substitute $x_o, y_o$, and $\theta$ into equation 2.13 we get one equation with four unknowns: $x, y, a$ and $b$. If we can find two points, $(x_1, y_1)$ and $(x_2, y_2)$, that are on this ellipse then we can sub-
B.1 Fitting and Ellipse to Measurement Points in 2D

(0, a, b) → (θ, a, β)

\[ \begin{align*}
(0, a, b) &= (\theta, \alpha, \beta) \\
&= (\theta_1, \alpha, \beta) \\
&= (\theta_2, \beta, \alpha)
\end{align*} \]

Figure B.1: Two ellipses with different parameters (left) can give the same final ellipse shape (right)

Substitute each point into equation 2.13 and get two equations with two unknowns, as shown below, and we can solve for \( a \) and \( b \). First, however, we need to find the two points on the ellipse as follows. We know the approximate location of the ellipse in the TRUS images. We choose \( x_1 \) to be the middle of the TRUS image's horizontal axis and \( x_2 \) to be 10 pixels away from \( x_1 \). These points are chosen arbitrarily but we are sure they are on the ellipse. Then we substitute each \( x \) value into equation 2.14 and knowing \( P \), we use the quadratic equation to find the corresponding \( y \) value. Finally, the \( a \) and \( b \) values are found as follows:

\[
\left( \frac{x_1 - x_o}{a} \cos(\theta) + \frac{y_1 - y_o}{b} \sin(\theta) \right)^2 + \left( \frac{-(x_1 - x_o) \sin(\theta) + (y_1 - y_o) \cos(\theta)}{b} \right)^2 = 1
\]

\[
\Rightarrow \left( \frac{A_1}{a} \right)^2 + \left( \frac{B_1}{b} \right)^2 = 1
\]

\[
\left( \frac{x_2 - x_o}{a} \cos(\theta) + \frac{y_2 - y_o}{b} \sin(\theta) \right)^2 + \left( \frac{-(x_2 - x_o) \sin(\theta) + (y_2 - y_o) \cos(\theta)}{b} \right)^2 = 1
\]

\[
\Rightarrow \left( \frac{A_2}{a} \right)^2 + \left( \frac{B_2}{b} \right)^2 = 1 \quad \text{(B.10)}
\]

\[
a = \frac{\sqrt{A_2^2 B_1^2 - A_1^2 B_2^2}}{B_1^2 - B_2^2}
\]

\[
b = \frac{\sqrt{A_2^2 B_1^2 - A_1^2 B_2^2}}{A_2^2 - A_1^2} \quad \text{(B.11)}
\]
B.1.2.2 Method 2

Step 1: Finding the ellipse origin \([x_o \ y_o]^T\)

We begin by normalizing and re-writing equation 2.14 as follows:

\[
\begin{bmatrix}
x & y \\
\end{bmatrix} \frac{1}{P_6} \begin{bmatrix}
P_1 & P_2 & P_3 \\
P_2 & P_3 & P_4 \\
\end{bmatrix} \begin{bmatrix}
x \\
y \\
\end{bmatrix} + \frac{1}{P_6} \begin{bmatrix}
P_4 & P_5 \\
\end{bmatrix} \begin{bmatrix}
x \\
y \\
\end{bmatrix} + 1 = 0
\]  
(B.12)

\[1^X C 1^X + B^T 1^X + 1 = 0 \]  
(B.13)

where \(1^X = [x \ y]^T\), \(C = \frac{1}{P_6} [P_1 \ P_2 \ P_3]\) and \(B^T = \frac{1}{P_6} [P_4 \ P_5]\). Now we translate all the points in the \(\{1^o, 1^C\}\) frame, using equation A.2, to get the \(\{2^o, 2^C\}\) reference frame.

\[((2^X - X_o)^T C (2^X - X_o) + B^T (2^X - X_o)) + 1 = 0\]  
(B.14)

If we expand the above equation we get:

\[2^X C 2^X + (2^X C + B^T) 2^X + (X_o^T C X_o + B^T X_o + 1) = 0\]  
(B.15)

However, in the \(\{2^o, 2^C\}\) frame there is no translation and therefore the coefficients of the \(2^X\) term should be zero. Solving \((2^X_o C + B^T)\) for \(X_o\) we get:

\[X_o = \begin{bmatrix}
x_o \\
y_o \\
\end{bmatrix} = \frac{1}{2} (C^{-1})^T B = \frac{1}{4P_1 P_4 - P_2^2} \begin{bmatrix}
-2P_3 P_4 + P_2 P_5 \\
2P_4 - 2P_1 P_5 \\
\end{bmatrix}\]  
(B.16)

Note that \([x_o \ y_o]^T\) was found using the same technique as in Method 1 except using matrix form because we will need this form for steps 2 and 3.

Step 2: Finding the ellipse size \((a, b)\)

From equation B.15 we see that the equation of the ellipse in the rotated, but untranslated reference frame is now:

\[2^X C 2^X + (X_o^T C X_o + B^T X_o + 1) = 0\]  
(B.17)

Since \(C\) is a square, positive definite matrix, it can be broken down into the product of 3 matrices using an Eigenvalue Decomposition (EVD) as follows:
where $U$ is a square matrix with orthogonal columns and $D_c$ is a diagonal matrix. Substituting the above equation into equation B.17 we get:

$$2X^T U^T D_c U^2 X + (X_o^T C X_o + B^T X_o + 1) = 0$$  \hspace{1cm} (B.19)

From equation A.1 we can see that $5X = U^2 X$ and with this the above equation becomes:

$$5X^T D_c 5X + (X_o^T C X_o + B^T X_o + 1) = 0$$  \hspace{1cm} (B.20)

which can be re-arranged as:

$$5X^T \begin{pmatrix} -D_c & \frac{X_o^T C X_o + B^T X_o + 1}{D_c (1,1)} \end{pmatrix} 5X = 1$$  \hspace{1cm} (B.21)

Now if we re-write equation 2.10 in matrix form, the general equation of an unrotated, untranslated ellipse is:

$$5X^T \begin{pmatrix} \frac{1}{a^2} & 0 \\ 0 & \frac{1}{b^2} \end{pmatrix} 5X = 1$$  \hspace{1cm} (B.22)

Using equations B.21 and B.22 we can find $a$ and $b$ as follows:

$$a = \sqrt{- \frac{X_o^T C X_o + B^T X_o + 1}{D_c (1,1)}}$$

$$b = \sqrt{- \frac{X_o^T C X_o + B^T X_o + 1}{D_c (2,2)}}$$  \hspace{1cm} (B.23)

**Step 3: Finding the ellipse angle $\theta$**

When we carry out the EVD of $C (C = U^T D_c U)$, what we are looking for are the diagonal entries in $D_c$ (which will give us the $a, b$ information) and the columns of the rotation matrix $U$ (which will give us the $\theta$ information). Ideally the EVD of $C$ would be:

$$C = [v_2 \ v_1] \text{diag} \{ f (a) \ f (b) \} [v_2^T \ v_1^T]^T$$  \hspace{1cm} (B.24)
B.2 Fitting an Ellipsoid to Measurement Points in 3D

So how do we know if the first entry in $D_c$ is actually $a$ or $b$? We make the accurate assumption that the angle of rotation of the prostate is never greater than 45°. We then take the dot product of the basis vectors $v_1$ and $v_2$ with the coordinate axes of the unrotated, untranslated reference frame (frame 5). For example, if the dot product of $v_1$ with $\mathbf{5x}$ is less than the dot product of $v_2$ with $\mathbf{5x}$ then the first entry in $D_c$ is actually $f(b)$.

Now that we have corrected the order of the columns in $D_c$ and $U$ we have:

$$C = U^T D_c U = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} \begin{bmatrix} f(a) & 0 \\ 0 & f(b) \end{bmatrix} \begin{bmatrix} v_{11} & v_{21} \\ v_{12} & v_{22} \end{bmatrix}$$ (B.25)

Recalling from equation A.1, the relationship between $U$ and $\theta$ is:

$$\mathbf{5x} = \begin{bmatrix} 5x \\ 5y \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \mathbf{2x} \\ \mathbf{2y} \end{bmatrix} = U^2 \mathbf{x}$$ (B.26)

Using the above two sets of equation we can find $\theta$ as follows:

$$\theta = \arccos(v_{11})$$ (B.27)

B.2 Fitting an Ellipsoid to Measurement Points in 3D

B.2.1 Deriving $P$ Parameters from $q$ parameters ($q$ to $P$)

We can find the $P$ parameters by expanding equation 4.5 and letting the coefficients equal the $P$ parameters in 4.6 to get:

$$\begin{align*}
\Theta_1 &= c\beta c\theta - s\beta s\phi s\theta \\
\Theta_2 &= s\beta c\theta + c\beta s\phi s\theta \\
\Theta_3 &= -\Theta_2 x_{\phi} - \Theta_1 y_{\phi} + c\phi s\theta z_{\phi} \\
\Theta_4 &= c\beta s\theta + s\beta s\phi c\theta \\
\Theta_5 &= s\beta s\theta - c\beta s\phi c\theta \\
\Theta_6 &= -\Theta_5 x_{\phi} - \Theta_4 y_{\phi} - c\phi c\theta z_{\phi} \\
\Theta_7 &= -c\beta c\phi x_{\phi} + s\beta c\phi y_{\phi} - s\phi z_{\phi}
\end{align*}$$ (B.28)
B.2 Fitting an Ellipsoid to Measurement Points in 3D

B.2.2 Deriving \( q \) Parameters from \( P \) parameters (\( P \) to \( q \))

We would like to fit an ellipsoid to a cloud of points in 3D. Li and Griffiths [24] show an algorithm to find the \( P \) parameters of this 'best fit' ellipsoid using a least squares approach and an ellipsoidal constraint on the \( P \) parameters. The problem has since been to convert this \( P \) vector to the \( q \) parameter representation so that we can draw the ellipsoid. We will present two methods for finding the \((x_0, y_0, z_0)\) and \((a, b, c)\) values. At the end we will show one method of finding the \((\theta, \phi, \beta)\) values.

Before we proceed, we divide both sides of equation 4.6 by \( P_{10} \) to get:

\[
F(V, P) = VP
\]

\[
= PP_1 x^2 + PP_2 y^2 + PP_3 z^2
\]

\[
+ 2PP_4 x^1 y^1 z + 2PP_5 x^1 z + 2PP_6 y^1
\]

\[
+ 2PP_7 x + 2PP_8 y + 2PP_9 z + 1
\]

(B.30)

where \( PP_i = P_i / P_{10} \). In doing this we have eliminated the free scaling factor and this is the equation that we will use in the proceeding sections.

### B.2.2.1 Method 1

**Step 1: Finding the ellipsoid origin \([x_0, y_0, z_0]^T\)**
In this method we start in frame 5 and work our way to frame 1. An unrotated, untranslated ellipsoid in frame 5 is described by:

\[ ^5X^T D_A ^5X = 1 \] (B.31)

where \( D_A = \text{diag} \{1/a^2, 1/b^2, 1/c^2\} \). Referring to Appendix A, we can describe frame 1 with respect to frame 5 as follows:

\[ ^1X = X_o + U ^5X \]
\[ ^5X = U^T (^1X - X_o) \] (B.32)

Substituting equation B.32 into B.31 we get the ellipsoid equation in frame 1:

\[ (^1X - X_o)^T U D_A U^T (^1X - X_o) = 1 \] (B.33)

Expanding the above equation and letting \( A = U D_A U^T \) we get:

\[ ^1X^T A^1X - 2 (X_o^T A) ^1X + (X_o^T AX_o - 1) = 0 \] (B.34)

Now we divide both sides by \( (X_o^T AX_o - 1) \) to get:

\[ ^1X^T \left( \frac{A}{X_o^T AX_o - 1} \right) ^1X + \left( \frac{-2 (X_o^T A)}{X_o^T AX_o - 1} \right) ^1X + 1 = 0 \] (B.35)

and we define:

\[ C = \frac{A}{X_o^T AX_o - 1} \] (B.36)
\[ B^T = \frac{-2 (X_o^T A)}{X_o^T AX_o - 1} \] (B.37)

If we re-write equation B.30 in matrix form we get:
B.2 Fitting an Ellipsoid to Measurement Points in 3D

Combining equations B.35, B.36, B.37 and B.38 it can be shown that:

\[ \begin{bmatrix} 1_x & 1_y & 1_z \end{bmatrix} \begin{bmatrix} PP_1 & PP_6 & PP_5 \\ PP_6 & PP_2 & PP_4 \\ PP_5 & PP_4 & PP_3 \end{bmatrix} \begin{bmatrix} 1_x \\ 1_y \\ 1_z \end{bmatrix} + \begin{bmatrix} 2PP_7 & 2PP_8 & 2PP_9 \end{bmatrix} \begin{bmatrix} 1_y \\ 1_z \end{bmatrix} + 1 = 0 \]  \hfill (B.38)

\[ B^T = -21_x^T C \]

\[ \therefore X_o = -\frac{1}{2} C^{-1} B \]  \hfill (B.39)

**Step 2: Finding the ellipse size \((a, b, c)\)**

At this point we have the \(P\) vector and \(X_o\) and we would like to find \((a, b, c)\). For the time being we make an assumption that we will prove shortly. We assume:

\[ A = -\frac{C}{X_o^T C X_o + B^T X_o + 1} \]  \hfill (B.40)

Now that we have \(A\) in terms of the known variables, we can find \((a, b, c)\) by carrying out an EVD on \(A\) as follows:

\[ A = U D_A U^T \]

\[ = U \begin{bmatrix} 1/a^2 & 0 & 0 \\ 0 & 1/b^2 & 0 \\ 0 & 0 & 1/c^2 \end{bmatrix} U^T \]  \hfill (B.41)

From this we can find \((a, b, c)\) values easily:
B.2 Fitting an Ellipsoid to Measurement Points in 3D

\[
a = \sqrt{1/D_A(1, 1)} \\
b = \sqrt{1/D_A(2, 2)} \\
c = \sqrt{1/D_A(3, 3)}
\]  
(B.42)

B.2.2.2 Method 2

Step 1: Finding the ellipse origin \([x_o, y_o, z_o]^T\)

In this method we start in frame 1 and work our way to frame 5. A rotated, translated ellipsoid in frame 1 is described by:

\[
^1X^T C^1X + B^T^1X + 1 = 0
\]  
(B.43)

If we translate every point on this ellipse according to \(^2X = ^1X - x_o\) we get the equation of the ellipsoid in frame 2:

\[
(^2X + X_o)^T C (^2X + X_o) + B^T (^2X + X_o) + 1 = 0
\]  
(B.44)

Expanding the above equation gives us:

\[
^2X^T C^2X + (2X_o^T C + B^T) ^2X + (X_o^T C X_o + B^T X_o + 1) = 0
\]  
(B.45)

But in frame 2 there is no translation so we set the coefficient of \(^2X\) to zero to find \(X_o\):

\[
(2X_o^T C + B^T) = 0 \\
\therefore X_o = -\frac{1}{2} C^{-1} B
\]  
(B.46)

This result agrees with our previous finding for \(X_o\). The equation of the ellipsoid in frame 2 becomes:

\[
^2X^T C^2X + (X_o^T C X_o + B^T X_o + 1) = 0
\]  
(B.47)
B.2 Fitting an Ellipsoid to Measurement Points in 3D

Step 2: Finding the ellipse size \((a, b, c)\)

Once again we use EVD to break down \(C\) into a sequence of orthogonal, diagonal and orthogonal matrices \((C = V^T D_c V)\). Equation B.47 then becomes:

\[
2X^T V^T D_c V X + (X_o^T C X_o + B^T X_o + 1) = 0
\]

where \(V = U^T\). If we recall from equation B.32 that \(X = U^T 2X\) then we can re-write the above equation as:

\[
5X^T \left( \frac{D_c}{X_o^T C X_o + B^T X_o + 1} \right)^5 X = 0
\]  \hspace{1cm} (B.48)

Comparing the above equation to equation B.31 we can find \((a, b, c)\):

\[
a = \sqrt{\frac{X_o^T C X_o + B^T X_o + 1}{D_c(1,1)}}
\]

\[
b = \sqrt{\frac{X_o^T C X_o + B^T X_o + 1}{D_c(2,2)}}
\]

\[
c = \sqrt{\frac{X_o^T C X_o + B^T X_o + 1}{D_c(3,3)}}
\]  \hspace{1cm} (B.49)

B.2.2.3 Comparing Method 1 and Method 2

The following check proves equation B.40 and shows that the two methods are in agreement. From Method 2 we saw that:

\[
D_A = \frac{-D_c}{X_o^T C X_o + B^T X_o + 1}
\]

\[
= \frac{-V C V^T}{X_o^T C X_o + B^T X_o + 1}
\]  \hspace{1cm} (B.50)

but, from Method 1 we saw that:
B.2 Fitting an Ellipsoid to Measurement Points in 3D

\[ A = UD_AU^T \]  \hspace{1cm} (B.51)

Substituting equation B.50 into B.51 we get:

\[ A = U \left( \frac{\mathbf{VCVT}}{X_o^T C X_o + B^T X_o + 1} \right) U^T \]  \hspace{1cm} (B.52)

But recall that \( U = V^T \). Therefore the above equation becomes:

\[ A = \frac{C}{X_o^T C X_o + B^T X_o + 1} \]  \hspace{1cm} (B.53)

which is the proof for equation B.40. Re-arranging the above equation we get:

\[ C = \frac{A}{X_o^T A X_o - 1} \]  \hspace{1cm} (B.54)

which agrees with equation B.36. Now we have proven that the above two Methods are in agreement.

**Step 3: Finding the ellipsoid angles \((\theta, \phi, \beta)\)**

Finding the ellipsoid angles is the same for both methods. Before we find the angles we must make sure the order of the eigenvalues and eigenvectors from the EVD are correct. When we carry out the eigenvalue decomposition of \( A = U^T D_A U \), what we are looking for are the diagonal entries in \( D_A \), which contain the \((a, b, c)\) information, and the columns of \( U \), which contain the basis vectors (and therefore the \((\theta, \phi, \beta)\) information) for \( S^C \). So ideally we have:

\[ A = [v_1 \ v_2 \ v_3] \text{diag} \{f(a) \ f(b) \ f(c)\} [u_1^T \ u_2^T \ u_3^T]^T \]  \hspace{1cm} (B.55)

However, the eigenvalue decomposition does not necessarily give the \((a, b, c)\) and basis columns in the right order. This means that Matlab could return any of the following eigenvalue decompositions, all of which give \( A \):
B.2 Fitting an Ellipsoid to Measurement Points in 3D

\[
A = \begin{bmatrix} v_2 & v_1 & v_3 \end{bmatrix} \text{diag} \{ f(b), f(a), f(c) \} \begin{bmatrix} v_2^T & v_1^T & v_3^T \end{bmatrix}^T
\]

\[
= \begin{bmatrix} v_3 & v_2 & v_1 \end{bmatrix} \text{diag} \{ f(c), f(b), f(a) \} \begin{bmatrix} v_3^T & v_2^T & v_1^T \end{bmatrix}^T
\]

\[
= \begin{bmatrix} v_1 & v_3 & v_2 \end{bmatrix} \text{diag} \{ f(a), f(c), f(b) \} \begin{bmatrix} v_1^T & v_3^T & v_2^T \end{bmatrix}^T
\]

So how do we know if the first entry in \( D \) is actually \( a \) or \( b \) or \( c \)? We make the assumption that the angles of rotation of the prostate, \((\theta, \phi, \beta)\), are never greater than 45°. This is a practical assumption since the prostate would never rotate more than 45° about any of the 3 axes. Next, we take the dot product of each vector \( v_i \) with \([1 \ 0 \ 0]^T\), \([0 \ 1 \ 0]^T\) and \([0 \ 0 \ 1]^T\) to see which axis it is aligned with. (Refer to Figure B.2 for an illustration) Also, sometimes the \( v_i \) are the negative of what they should be. In that case if the largest \( v_i \) entry in is negative we multiply the whole vector by \(-1\).

Once we have corrected the order of the basis vectors in \( U \), the correct EVD of \( A \) becomes:

\[
A = U D_A U^T
\]

\[
= \begin{bmatrix} v_{11} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32} \\ v_{13} & v_{23} & v_{33} \end{bmatrix} \begin{bmatrix} f(a) & 0 & 0 \\ 0 & f(b) & 0 \\ 0 & 0 & f(c) \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ v_{31} & v_{32} & v_{33} \end{bmatrix}
\]

(B.56)

Referring back to equation A.6 we recall that:

\[
\begin{array}{c}
\text{Figure B.2: Correcting the order of the basis vectors in 3D}
\end{array}
\]
\[ U = \exp(\beta^3 z_x) \exp(\phi^4 y_z) \exp(\theta^5 x_z) \] (B.57)

and when we expand \( U \) we get:

\[
U = \begin{bmatrix}
c\beta c\phi & s\beta c\theta + c\beta s\phi s\theta & s\beta s\theta - c\beta s\phi c\theta \\
-s\beta c\phi & c\beta c\theta - s\beta s\phi s\theta & c\beta s\theta + s\beta s\phi c\theta \\
s\phi & -c\phi s\theta & c\phi c\theta
\end{bmatrix}
\] (B.58)

At this point we can use equations B.58 and B.56 to find the angles as follows:

\[
\phi = \arcsin(v_{13})
\]
\[
= \arcsin(U(3,1))
\] (B.59)

\[
\theta = \arcsin\left(-\frac{v_{23}}{c\phi}\right)
\]
\[
= \arcsin\left(-\frac{U(3,2)}{c\phi}\right)
\] (B.60)

\[
\beta = \arcsin\left(-\frac{v_{12}}{c\phi}\right)
\]
\[
= \arcsin\left(-\frac{U(2,1)}{c\phi}\right)
\] (B.61)
Appendix C

Ellipse/Ellipsoid Fitting - Mathematical Proofs

C.1 LSM Cost Function

Our purpose is to prove that the cost function for LSM, $\sum_{i=1}^{n} (F_i (V_i, P))^2$ is equal to $\|SP\|^2$. Given a set of $n$ points in 2D space, $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, where we have omitted the reference frame notation for convenience, the corresponding $V_i$ vectors are:

\[
V_1 = \begin{bmatrix} x_1 & x_1y_1 & y_1 & x_1 & y_1 & 1 \end{bmatrix} \\
V_2 = \begin{bmatrix} x_2 & x_2y_2 & y_2 & x_2 & y_2 & 1 \end{bmatrix} \\
\vdots \\
V_n = \begin{bmatrix} x_n & x_ny_n & y_n & x_n & y_n & 1 \end{bmatrix}
\] (C.1)

The distance from each point to an ellipse described by $P = [P_1 \ P_2 \ P_3 \ P_4 \ P_5 \ P_6]$ is:
C.1 LSM Cost Function

\[ F_1(V_1, P) = P_1x_1^2 + P_2x_1y_1 + P_3y_1^2 + P_4x_1 + P_5y_1 + P_6 \]
\[ = [x_1 \ x_1y_1 \ y_1 \ x_1 \ y_1 \ 1] \cdot P \]

\[ F_2(V_2, P) = P_1x_2^2 + P_2x_2y_2 + P_3y_2^2 + P_4x_2 + P_5y_2 + P_6 \]
\[ = [x_2 \ x_2y_2 \ y_2 \ x_2 \ y_2 \ 1] \cdot P \]

\[ \vdots \]

\[ F_n(V_n, P) = P_1x_n^2 + P_2x_ny_n + P_3y_n^2 + P_4x_n + P_5y_n + P_6 \]
\[ = [x_n \ x_ny_n \ y_n \ x_n \ y_n \ 1] \cdot P \] (C.2)

We would like to find the \( P \) vector that minimizes the sum of the square of these algebraic distances as follows:

\[
\text{Cost} = \sum_{i=1}^{n} (F_i(V_i, P))^2
\]
\[ = (F_1(V_1, P))^2 + (F_2(V_2, P))^2 + \ldots + (F_n(V_n, P))^2 \]
\[ = (V_1 \cdot P)^2 + (V_2 \cdot P)^2 + \ldots + (V_n \cdot P)^2 \] (C.3)

We will now prove that the above equation equals \( ||SP||^2 \), where \( S = [V_1^T \ V_2^T \ \ldots \ V_n^T]^T \) by expanding:
C.2 Minimizing Cost Function using GEP

The mathematical proofs in this section were obtained from Dr. Jim Varah. Our purpose is to show that the ellipse which best fits a set of measurement points is represented by the vector (eigenvector) corresponding to the unique positive eigenvalue in equation 2.23. This will be proven in the following two steps.

Step 1: Proof of the Existence of a Single Positive Eigenvalue

We know that $N$ is a singular matrix and has eigenvalues $[-2 \quad -1 \quad 0 \quad 0 \quad 0 \quad 2]$. The inertia of a matrix corresponds to the number of negative, zero and positive eigenvalues that the matrix has. For example, the inertia of $N$ is $\{2, 3, 1\}$. We also know that $S^T S$ is positive semi-definite because for all $P \in \mathbb{R}^n$, $P^T S^T S P = ||SP||^2 > 0$ if $S$ is full rank and $P^T S^T S P = ||SP||^2 \geq 0$ if $S$ is not full rank. We let $S^T S$ equal $M$ and use the Cholesky Decomposition to re-write $M$ as follows:

$$M = L^T L$$

(C.5)

where $L$ is a lower triangular matrix. We now substitute equation 2.23 into (C.5) and re-arrange:
C.2 Minimizing Cost Function using GEP

\[(L^{-T}N - \mu L^T) P = 0\]  \hspace{1cm} (C.6)

If we let \( P = L^{-1}q \) and we substitute this into the above equation we get:

\[(L^{-T}NL^{-1} - \mu I) q = 0\]  \hspace{1cm} (C.7)

If we let \( B = L^{-T}NL^{-1} \) then we can say that the eigenvalues of \( B \) are \( \mu \) from the above equation. Now we let \( Q = L^{-1} \) to get \( B = Q^T N Q \) which is a congruency transformation. By Sylvester’s Law, a congruency transformation maintains the inertia of a matrix. This means that \( B \) has the same number of negative, zero and positive eigenvalues as \( N \). This implies that \( \mu \) from equation 2.23 can only have one unique positive eigenvalue because \( N \) has a single positive eigenvalue.

**Step 2: Proof That the Cost Function is Minimized by the P Vector Corresponding to the Unique Positive Eigenvalue**

We use the ‘Simultaneous Diagonalization of two Quadratic Forms’ to simultaneously diagonalize \( M \) and \( N \), using the same eigenvector matrix \( z \), as follows:

\[
z^T M z = D_\alpha \]
\[
z^T N z = D_\beta \]  \hspace{1cm} (C.8)

where \( D_\alpha \) and \( D_\beta \) are diagonal matrices containing \( \alpha_i \) (eigenvalues of \( M \)) and \( \beta_i \) (eigenvalues of \( N \)) on the diagonal entries. Starting with equation 2.23 and plugging in the above equations it can be shown that \( \mu_i = \alpha_i/\beta_i \). Since \( M \) is positive definite all its eigenvalues are positive and therefore \( \alpha_i > 0 \) for \( i = 1...6 \). We recall from step 1 that \( N \) has only one positive eigenvalue; therefore, \( D_\beta \) has only one positive entry and so we let \( \beta_1 > 0 \) and \( \beta_i < 0 \) for \( i = 2...6 \). We now revisit the problem statement:

\[
\min_P \text{Cost} = P^T M P \text{ subject to } P^T N P = 1 \]  \hspace{1cm} (C.9)

Using a simple transformation, \( P = zy \) we change the problem statement to:

\[
\min_y \text{Cost} = y^T (z^T M z) y \text{ subject to } y^T (z^T N z) y = 1 \]  \hspace{1cm} (C.10)

Substituting equation C.8 into the above equation we get:

\[
\min_y \text{Cost} = y^T D_\alpha y \text{ subject to } y^T D_\beta y = 1 \]  \hspace{1cm} (C.11)
which essentially simplifies to finding the $y$ value that minimizes $\text{Cost} = \sum_{i=1}^{6} \alpha_i y_i^2$ subject to $\sum_{i=1}^{6} \beta_i y_i^2 = 1$. Solving the constraint for $y_1$ we get:

$$y_1^2 = \frac{1 - \sum_{i=2}^{6} \beta_i y_i^2}{\beta_1} \quad \text{(C.12)}$$

Then substituting this into the cost function and re-arranging gives us:

$$\text{Cost} = \frac{\alpha_1}{\beta_1} + \sum_{i=2}^{6} \left( \frac{\alpha_i - \beta_i}{\beta_1} \right) y_i^2 \quad \text{(C.13)}$$

Since $\alpha_i$ is always positive and $\beta_i$ is always negative for $i > 2$, the contents of the bracket, in the equation above, will always be positive. This means that the minimum of this cost function is $\alpha_1/\beta_1$ and this occurs when $y_i$, for all $i > 2$, are zero. Therefore the cost function is minimized when:

$$y = \left[ \frac{1}{\sqrt{\beta_1}} \quad 0 \quad 0 \quad 0 \quad 0 \right]^T \quad \text{(C.14)}$$

We still need to find the $P$ vector that minimizes the cost function of equation C.9. If $z = [z_1 \quad z_2 \quad z_3 \quad z_4 \quad z_5 \quad z_6]$ where $z_i$ is the eigenvector corresponding to $\alpha_i$ and $\beta_i$ then we can use the transformation $P = zy$ to find $P$.

$$P = \frac{z_1}{\sqrt{\beta_1}} \quad \text{(C.15)}$$

This shows that the $P$ vector we are looking for is the one that corresponds to the first eigenvector in $z$, which corresponds to the first eigenvalues $\alpha_1$ and $\beta_1$- both of which are positive. Note that if $\mu_i = \alpha_i/\beta_i$ then $\mu_1$ is the only positive eigenvalue and the eigenvector that minimizes our cost function is the one corresponding to this eigenvalue.

This ellipse fitting technique always returns a $P$ vector that represents an ellipse regardless of initial $(x_i, y_i)$ points. Also, the fit has rotational and translational invariance (meaning if you translate/rotate the data, the resulting ellipse fit will translate/rotate by the same amount), is robust, easy to implement and computationally cheap.
Appendix D

Lagrangian Multipliers

Lagrangian multipliers are used in mathematical optimization problems that attempt to find the local extrema of a function of one or several variables subject to one or more constraints. Using this method we can reduce a problem with $n$ equations, $n$ variables and $k$ constraints to a problem with $n + k$ equations, $n + k$ variables and no constraints by introducing a new scalar multiplier, called the Lagrangian multiplier, for each constraint.

Let's consider the two dimensional case with one constraint so that $n = 2, k = 1$. Let the function to be optimized be $f(x,y)$ and let the constraint be $h(x,y) = g(x,y) - c$. Figure D.1 shows that as we traverse the constraint $h(x,y)$ we are moving up and down along the level curves of $f(x,y)$ and at some point we will reach a maximum value on $f(x,y)$ along the $h(x,y)$ path. This maximum occurs when the $g(x,y) = c$ curve is tangential to a level curve, $f(x,y) = d_n$, and is no longer crossing the level curves. At this point the gradients of $f(x,y)$ and $h(x,y)$ are parallel.

Figure D.1: The dashed lines represent the level curves of the function we are trying to optimize. The solid line represents the constraint. The arrows point in the direction of maximum gradient.
\[ \nabla f(x, y) = \lambda \nabla (g(x, y) - c) \quad (D.1) \]

which gives us:

\[
\begin{align*}
\frac{\partial f}{\partial x} &= \lambda \frac{\partial h}{\partial x} \\
\frac{\partial f}{\partial y} &= \lambda \frac{\partial h}{\partial y}
\end{align*}
\quad (D.2)
\]

Using the above two equations and \( h(x, y) \) we have 3 equations and 3 unknowns (\( \lambda, x \) and \( y \)) that we can find easily.

In the general case, assuming \( f \) has \( n \) variables and there are \( k \) constraints, we get the following equation:

\[ \nabla f - \lambda_1 \nabla (h_1) - \lambda_2 \nabla (h_2) \cdots = 0 \quad (D.3) \]

If we take the gradient of this equation we get \( n \) equations with \( n + k \) variables. Using the \( k \) constraint equations we will have \( n + k \) equations with \( n + k \) variables and we will be able to solve the system of equations to find the \( n \) variables that optimize \( f \) subject to the \( k \) constraints.
Appendix E

Newton-Raphson Method

Figure E.1 shows that the warping function, \( r_{\text{new}} \), as a function of \( r \) (\( \theta \) and \( \sigma \) are constant), is a monotonically increasing function. This means that given any warped radius, \( r_{\text{new}}(i) \), we can find its single corresponding unwarped radius, \( r(i) \), using the Newton-Raphson method. In this method we start with an arbitrary initial \( r \) value, \( r_0 \), and we check to see if \( r_{\text{new}}(r_0) \) is sufficiently close to \( r_{\text{new}}(i) \) by checking to see whether the difference between \( r_{\text{new}}(r_0) \) and \( r_{\text{new}}(i) \) is less than some threshold \( (\text{diff} < \text{thresh}) \). If the difference is less than a pre-defined threshold then \( r_0 \) is the single \( r \) value corresponding to \( r_{\text{new}}(i) \) and we have found the unwarped radius value. If the difference is not less than the threshold then we find the intersection of the line tangent to the warping function at \( (r_0 , r_{\text{new}}(r_0)) \) and the x-axis. This x-intercept becomes our new \( r_0 \) and we find its corresponding \( r_{\text{new}}(r_0) \) and again check to see if it is close to \( r_{\text{new}}(i) \). These steps are iterated until we find the \( r_0 = r(i) \) that corresponds to \( r_{\text{new}}(i) \). Then we move on to \( r_{\text{new}}(i + 1) \) until all warped radii are unwarped.
Figure E.1: Warping function as a function of $r$ only with $\theta = 90^\circ$ and $\sigma = 40$. Newton’s method is used to find the $r(i)$ value corresponding to $r_{new}(i)$.
Appendix F

Tracking theory and (Extended) Kalman Filters

F.1 Kalman Filter

The Kalman Filter (KF) is a recursive numerical technique that estimates and tracks the internal states of a time-varying system which has noise superimposed both on top of its state variables (process noise) and its output signal (measurement noise). Figure F.1 was obtained from [5] and is an excellent diagram for describing the steps of the KF. Equation F.1 shows the state space representation of a linear stochastic process (or plant):

\[
\begin{align*}
    x(k+1) &= F(k)x(k) + G(k)u(k) + \Gamma(k)V(k) \\
    z(k+1) &= H(k+1)x(k+1) + w(k+1)
\end{align*}
\]  

(F.1)

where \(F(k), G(k)\) and \(H(k)\) describe our model of the plant and \(u(k), x(k)\) and \(z(k)\) are the input, true state and output variables, respectively. The variables \(V(k)\) and \(w(k)\) are the zero mean white Gaussian process and measurement noise with variance \(Q(k)\) and \(R(k)\) respectively [10]. In general, the discrete state-space model of a system is obtained by first finding its continuous state space model and then discretizing it. If the continuous model does not perfectly model the real system, then a noise term is added to the continuous model, prior to discretization, which results in the existence of \(\Gamma(k)\) after discretization. If the continuous model perfectly models the real system then the process noise, \(V(k)\), exists solely due to the discretization and \(\Gamma(k)\) is the identity matrix.

In Figure F.1, the first column (left most column) shows the evolution of the true system state from
Kalman Filter

\( x(k) \) to \( x(k+1) \) and the true system output (or measurement) \( z(k+1) \). However, we do not have access to the system state, but we do have access to the system input and output and using these we would like to estimate the system state. Referring to the third column in Figure F.1 we can see that there are 2 steps in updating the state estimate from time \( t_k \) to time \( t_{k+1} \). Assuming we know the state at \( t_k \), first we predict the state and measurement at \( t_{k+1} \) using the following equations:

\[
\begin{align*}
\hat{x}(k+1|k) &= F(k)\hat{x}(k|k) + G(k)u(k) \\
\hat{z}(k+1|k) &= H(k+1)\hat{x}(k+1|k)
\end{align*}
\]  

\( \text{(F.2)} \)

where \( \hat{x}(k|k) \) is the state estimate at time \( t_k \), \( \hat{x}(k+1|k) \) is the state prediction given the state estimate at \( t_k \) and \( \hat{z}(k+1|k) \) is the measurement prediction given the state prediction. Note that the above equations are very similar to equation F.1 except there are no noise terms (the noise is taken into account in the prediction covariance). Next we find the updated state estimate at \( t_{k+1} \) using the state prediction as follows:

\[
\begin{align*}
\hat{x}(k+1|k+1) &= \hat{x}(k+1|k) + W(k+1) \{ z(k+1) - \hat{z}(k+1|k) \} \\
W(k+1) &= \frac{P(k+1|k)H^T(k+1)}{H(k+1)P(k+1|k)H^T(k+1) + R(k+1)}
\end{align*}
\]  

\( \text{(F.3)} \)

where the KF Gain, \( W(k+1) \), corrects the predicted state by amplifying the incremental difference between the true measurement \( (z(k+1)) \) and the measurement prediction \( (\hat{z}(k+1|k)) \). From the fourth column in Figure F.1, we see that the state prediction covariance, \( P(k+1|k) \), is found from and updated by the following equations:

\[
\begin{align*}
P(k+1|k) &= F(k)P(k|k)F^T(k) + \Gamma(k)Q(k)\Gamma(k) \\
P(k+1|k+1) &= P(k+1|k) \\
&\quad - W(k+1) \{ H(k+1)P(k+1|k)H^T(k+1) + R(k+1) \} W^T(k+1)
\end{align*}
\]  

\( \text{(F.4)} \)

where \( P(k+1|k) \) is the covariance of the state at \( t_k \). At this point we have the updated estimates for the state, output (measurement) and state covariance at time \( t_{k+1} \) and we can re-iterate the same steps to get these values for time \( t_{k+2} \). The KF continues this iteration with the goal of making the difference between the true measurement and the measurement prediction zero which would imply a perfect estimate of the system states.

In Kalman filtering the initial conditions, \( \hat{x}(0|0) \) and \( P(0|0) \) are important in determining the speed with
which the state estimate tracks the true estimate. If the initial conditions are far from the true values it will take a long time for the difference between \( x(k) \) and \( \hat{x}(k|k) \) (or \( z(k+1) \) and \( \hat{z}(k+1|k) \)) to go to zero \[6\].

### F.2 Extended Kalman Filter

The Extended Kalman Filter (EKF), shown in Figure F.2 \[6\], has the same purpose and function as the KF except for systems that are not linear. Non-linear systems, in this case, are ones where the state transition and measurement models are a (differentiable) function of the state variables as follows:

\[
\begin{align*}
  x(k+1) &= f(k, x(k), u(k)) + \Gamma(k)V(k) \\
  z(k+1) &= h(k+1, x(k+1)) + w(k+1)
\end{align*}
\]  

(F.5)

Therefore, the State Prediction and Measurement Prediction become:

\[
\begin{align*}
  \hat{x}(k+1) &= f(k, \hat{x}(k), u(k)) \\
  \hat{z}(k+1) &= h(k+1, \hat{x}(k+1))
\end{align*}
\]  

(F.6)

Since \( f \) and \( h \) are non-linear functions of the state variables they cannot be used, as they are, to find the State Prediction Covariance and Innovation Covariance. The solution is to linearize the system by finding \( F \) and \( H \), which are the Jacobians of \( f \) and \( h \) respectively, and then to use \( F \) and \( H \) in finding \( P(k+1|k) \) and \( S(k+1) \). The Jacobians are calculated as follows:

\[
\begin{align*}
  F(k) &= \left. \frac{\partial f(k)}{\partial x} \right|_{x=\hat{x}(k|k)} \\
  H(k+1) &= \left. \frac{\partial h(k+1)}{\partial x} \right|_{x=\hat{x}(k+1|k)}
\end{align*}
\]  

(F.7)

The rest of the EKF is the same as the KF.

### F.3 Interacting Multiple Modes

If a system is complex, it is possible to represent the system with multiple state-space models (modes). In the Interacting Multiple Modes (IMM) \[6\] approach \( n \) models run in parallel and the inputs and outputs to and from each model are mixed to give the overall system's state variables and covariances.
<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>Transition to $t_{k+1}$: $x(k+1) = F(k)x(k) + G(k)u(k) + \Gamma(k)V(k)$</td>
</tr>
<tr>
<td>(2)</td>
<td>Measurement at $t_{k+1}$: $z(k+1) = H(k+1)x(k+1) + w(k+1)$</td>
</tr>
<tr>
<td>(3)</td>
<td>State Prediction: $\hat{x}(k+1</td>
</tr>
<tr>
<td>(4)</td>
<td>Measurement Prediction: $\hat{z}(k+1</td>
</tr>
<tr>
<td>(5)</td>
<td>Innovation: $v(k+1) = z(k+1) - \hat{z}(k+1</td>
</tr>
<tr>
<td>(6)</td>
<td>Updated State estimate: $\tilde{x}(k+1</td>
</tr>
<tr>
<td>(7)</td>
<td>State Prediction Covariance: $P(k+1</td>
</tr>
<tr>
<td>(8)</td>
<td>Innovation Covariance: $S(k+1) = H(k+1)P(k+1</td>
</tr>
<tr>
<td>(9)</td>
<td>Filter Gain: $W(k+1) = P(k+1</td>
</tr>
<tr>
<td>(10)</td>
<td>Updated State Covariance: $P(k+1</td>
</tr>
</tbody>
</table>

Figure F.1: One iteration of the Kalman Filter [5]
Evolution of the true system state $x$

Input values at time $t_k$

- State at $t_k$ $\chi(k)$

Transition to $t_{k+1}$

- $x(k+1) = f(k, x(k), u(k)) + \Gamma(k) V(k)$

Measurement at $t_{k+1}$

- $z(k+1) = h(k+1, x(k+1)) + w(k+1)$

Output values at time $t_{k+1}$

Evolution of the state covariance $P$

State error covariance at $t_k$ $P(k|k)$

Evolution of the estimated system state $\hat{x}$

Controller

- Control input at $t_k$ $u(k)$

- State estimate at $t_k$ $\hat{x}(k|k)$

State Prediction Covariance $P(k+1|k) = F(k) P(k|k) F'(k) + \Gamma(k) Q(k) \Gamma'(k)$

Measurement Prediction $\hat{z}(k+1|k) = h(k+1, \hat{x}(k+1|k))$

Innovation $v(k+1) = z(k+1) - \hat{z}(k+1|k)$

Filter Gain $W(k+1) = P(k+1|k) H'(k+1) S^I(k+1)$

Updated State covariance $P(k+1|k+1) = P(k+1|k) H'(k+1) S^I(k+1)$

Updated State estimate $\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1) v(k+1)$

Next Cycle starting with $t_{k+1}$ as input and creating data for $t_{k+2}$
F.3 Interacting Multiple Modes

Let’s consider a system that can be represented by \( r \) modes. The Mode Probability, \( \mu_i(k-1) \) is the probability that the system is behaving according to mode \( i (i = 1...r) \) at time \( k - 1 \). The Mode Transition Probability, \( p_{ij} \), is the probability that the system will transition and behave according to mode \( i \) given that it is currently behaving according to mode \( j (j = 1...r) \). Let’s assume that we know \( p_{ij}, \mu_i(k-1) \) and the states and state error covariances for each mode at time \( k - 1 \). In order to find the states, state error covariances and Mode Probability at time \( k \) we use the following IMM estimator algorithm (refer to Figure F.3):

1. **Calculation of Mixing Probabilities.** The Mixing Probabilities for mode \( i \) given that the system is in mode \( j \) at time \( k - 1 \) are found by:

\[
\mu_{ij}(k-1|k-1) = \frac{p_{ij}\mu_j(k-1)}{\sum_{i=1}^{r} p_{ij}\mu_i(k-1)} \tag{F.8}
\]

2. **Mixing.** The states and state error covariances of each mode, at time \( k - 1 \), are mixed together using the Mixing Probabilities founds in the previous step. The result is the input state and input state error covariance into each mode.

\[
\begin{align*}
\hat{x}^{oij}(k-1|k-1) &= \sum_{i=1}^{r} \hat{x}^{ij}(k-1|k-1) \mu_{ij}(k-1|k-1) \\

P^{oij}(k-1|k-1) &= \sum_{i=1}^{r} \mu_{ij}(k-1|k-1) \begin{pmatrix} P^i(k-1|k-1) + \\
[\hat{x}^i(k-1|k-1) - \hat{x}^{oij}(k-1|k-1)] \\
[\hat{x}^i(k-1|k-1) - \hat{x}^{oij}(k-1|k-1)]^T 
\end{pmatrix}
\end{align*}
\tag{F.9}
\]

3. **Mode-Matched Filtering.** The states and state error covariance for each mode are processed through each mode and updated to get \( \hat{x}^i(k|k) \), \( P^1(k|k) \) and \( \hat{x}^2(k|k) \), \( P^2(k|k) \). The likelihood function, shown below, is used to find the mode probability update which is then used to find the updated states and state error covariances.

\[
\Lambda_j(k) = \frac{\exp \left\{ -\frac{(z(k) - \hat{x}^j(k|k-1))^2}{2S^j(k)} \right\}}{\sqrt{2\pi S^j(k)}} \tag{F.10}
\]

4. **Mode Probability Update.** The Mode Probability is updated for time \( k \) according to the following equation:

\[
\mu_j(k) = \frac{\Lambda_j(k) \sum_{i=1}^{r} p_{ij}\mu_i(k-1)}{\sum_{j=1}^{r} (\Lambda_j(k) \sum_{i=1}^{r} p_{ij}\mu_i(k-1))} \tag{F.11}
\]

5. **Estimate and covariance combination.** Finally the state and state error covariance for the overall
system are found by mixing the state and state error covariance from each mode as follows:

$$
\hat{x}(k|k) = \sum_{j=1}^{r} \hat{x}^j(k|k) \mu_j(k)
$$

$$
P(k|k) = \sum_{j=1}^{r} \mu_j(k) \left\{ P^j(k|k) + [\hat{x}^j(k|k) - \hat{x}(k|k)] \cdot [\hat{x}^j(k|k) - \hat{x}(k|k)]^T \right\}
$$

(F.12)

### F.4 Probabilistic Data Association Filter

Let's consider the problem of using a radar to track the trajectory of an aircraft moving along in space (Figure F.4). If we have an estimate of the aircraft's radial position at time $k-1$, $\hat{x}(k-1|k-1)$, we can use the KF to find the position at time $k$, $\hat{x}(k|k)$ according to:

$$
\hat{x}(k|k) = \hat{x}(k|k-1) + W(k) (z(k) - z(k|k-1))
$$

$$
\hat{x}(k|k-1) = F(k-1)\hat{x}(k-1|k-1) + G(k-1)u(k-1)
$$

$$
\hat{x}(k|k-1) = H(k)\hat{x}(k|k-1)
$$

(F.13)

In order to find $\hat{x}(k|k)$ we need a measurement value, $z(k)$, from the radar. Assume the plane is moving in an environment that is immersed in white Gaussian noise so that at any time $k$ the radar receives several signals - some of which are noise and some of which are echoes off of the plane with superimposed noise. This problem, where at one instant in time there are numerous measurement points and we are looking for one measurement point, $z(k)$, can be solved using the Probabilistic Data Association (PDA) filter [5,7]. At time $k$ there will be $E$ measurement values (alternatively called candidate edge points) and we will represent each point as $r_i(k) (i = 1...E)$. Then each candidate edge point is assigned a Gaussian weight based on the magnitude of the signal (higher magnitude receives higher weight) and its distance away from the prediction of the aircraft’s previous position $\hat{x}(k|k-1)$ . (All distances are radial distances from the radar).

$$
p_i(k) = \frac{Mag_i^2(k)}{\sqrt{2\pi S(k)}} \exp \left( \frac{-(r_i(k) - \hat{x}(k|k-1))^2}{2S(k)} \right)
$$

(F.14)

where $p_i(k)$ is the Gaussian weight, $Mag_i^2(k)$ is the magnitude of the $i$th signal and $S(k)$ is the innovation covariance. Note that in this equation, the candidate edge points closer to the state prediction, $\hat{x}(k|k-1)$, receive a higher weight. Finally, $z(k)$ is found by performing a weighted average of the candidate boundary points.
F.4 Probabilistic Data Association Filter

Figure F.3: One cycle of the Interacting Multiple Mode Estimator
\[ z(k) = \sum_{i=1}^{E} \left( \frac{p_i(k)}{\sum_{i=1}^{E} p_i(k)} r_i(k) \right) \quad i = 1 \ldots E, \; k = 1 \ldots N_r \] (F.15)

Figure F.4: Tracking aircraft motion using a radar and PDA theory
Appendix G

Semi-automatic Segmentation Results

In the 3D segmentation study presented in Chapter 5, three observers manually segmented and initialized 10 cases in the first week and manually re-segmented and re-initialized 5 of the original 10 cases two weeks later. In this appendix we present the semi-automatic segmentation results, superimposed on top of the gold standard, for every case in this study. Due to space limitations we have shown only nine slices from each case. If there were more slices in the volume we omitted slices 7 and/or 8. It should be noted that the first subfigure is always the base and the last subfigure is always the apex.
Figure G.1: Comparing semi-automatic (*solid line*) to manual segmentation (*dotted line*) for Observer 1, Case 1, initial 10 cases.

Figure G.2: Comparing semi-automatic (*solid line*) to manual segmentation (*dotted line*) for Observer 1, Case 2, initial 10 cases.
Figure G.3: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 1, Case 3, initial 10 cases.

Figure G.4: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 1, Case 4, initial 10 cases.
Figure G.5: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 1, Case 5, initial 10 cases.

Figure G.6: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 1, Case 6, initial 10 cases.
Figure G.7: Comparing semi-automatic (*solid line*) to manual segmentation (*dotted line*) for Observer 1, Case 7, initial 10 cases.

Figure G.8: Comparing semi-automatic (*solid line*) to manual segmentation (*dotted line*) for Observer 1, Case 8, initial 10 cases.
Figure G.9: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 1, Case 9, initial 10 cases.

Figure G.10: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 1, Case 10, initial 10 cases.
Figure G.11: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 1, initial 10 cases.

Figure G.12: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 2, initial 10 cases.
Figure G.13: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 3, initial 10 cases.

Figure G.14: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 4, initial 10 cases.
Figure G.15: Comparing semi-automatic \textit{(solid line)} to manual segmentation \textit{(dotted line)} for Observer 2, Case 5, initial 10 cases.

Figure G.16: Comparing semi-automatic \textit{(solid line)} to manual segmentation \textit{(dotted line)} for Observer 2, Case 6, initial 10 cases.
Figure G.17: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 7, initial 10 cases.

Figure G.18: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 8, initial 10 cases.
Figure G.20: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 10, initial 10 cases.
Figure G.21: Comparing semi-automatic (*solid* line) to manual segmentation (*dotted* line) for Observer 3, Case 1, initial 10 cases.

Figure G.22: Comparing semi-automatic (*solid* line) to manual segmentation (*dotted* line) for Observer 3, Case 2, initial 10 cases.
Figure G.23: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 3, initial 10 cases.

Figure G.24: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 4, initial 10 cases.
Figure G.25: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 5, initial 10 cases.

Figure G.26: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 6, initial 10 cases.
Figure G.28: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 8, initial 10 cases.
Figure G.30: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 10, initial 10 cases.

Figure G.30: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 10, initial 10 cases.
Figure G.31: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 1, Case 2, repeat of 5 cases.

Figure G.32: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 1, Case 3, repeat of 5 cases.
Figure G.33: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 1, Case 6, repeat of 5 cases.

Figure G.34: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 1, Case 8, repeat of 5 cases.
Figure G.35: Comparing semi-automatic (*solid line*) to manual segmentation (*dotted line*) for Observer 1, Case 10, repeat of 5 cases.

Figure G.36: Comparing semi-automatic (*solid line*) to manual segmentation (*dotted line*) for Observer 2, Case 2, repeat of 5 cases.
Figure G.38: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 6, repeat of 5 cases.

Figure G.37: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 3, repeat of 5 cases.
Figure G.39: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 8, repeat of 5 cases.

Figure G.40: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 2, Case 10, repeat of 5 cases.
Figure G.42: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 3, repeat of 5 cases.
Figure G.43: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 6, repeat of 5 cases.

Figure G.44: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 8, repeat of 5 cases.
Figure G.45: Comparing semi-automatic (solid line) to manual segmentation (dotted line) for Observer 3, Case 10, repeat of 5 cases.