Western Graduate&PostdoctoralStudies

Western University
Scholarship@Western

Electronic Thesis and Dissertation Repository

4-18-2011 12:00 AM

Image-Guided Robot-Assisted Techniques with Applications in Minimally Invasive Therapy and Cell Biology

Mahdi Azizian, *The University of Western Ontario*

Supervisor: Dr. Rajni Patel, *The University of Western Ontario*
A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Electrical and Computer Engineering
© Mahdi Azizian 2011

Follow this and additional works at: https://ir.lib.uwo.ca/etd

Part of the Biomedical Commons

Recommended Citation

Azizian, Mahdi, "Image-Guided Robot-Assisted Techniques with Applications in Minimally Invasive Therapy and Cell Biology" (2011). *Electronic Thesis and Dissertation Repository*. 144.
https://ir.lib.uwo.ca/etd/144

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlswadmin@uwo.ca.

Image-Guided Robot-Assisted Techniques with Applications in Minimally Invasive Therapy and Cell Biology

(Spine title: Image-Guided Robotics in Medicine and Biology)

(Thesis format: Integrated-Article)

by

Mahdi Azizian

Faculty of Engineering
Department of Electrical and Computer Engineering

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

© Mahdi Azizian 2011

THE UNIVERSITY OF WESTERN ONTARIO
School of Graduate and Postdoctoral Studies

CERTIFICATE OF EXAMINATION

Supervisor	Examiners
Dr. Rajni V. Patel	Dr. Serguei L. Primak
	Dr. Ilia Polushin
	Dr. Terry M. Peters
	Dr. Yu Sun

The thesis by
Mahdi Azizian
entitled
Image-Guided Robot-Assisted Techniques with Applications in Minimally Invasive Therapy and Cell Biology
is accepted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Date

Chair of the Thesis Examination Board

Abstract

There are several situations where tasks can be performed better robotically rather than manually. Among these are situations (a) where high accuracy and robustness are required, (b) where difficult or hazardous working conditions exist, and (c) where very large or very small motions or forces are involved. Recent advances in technology have resulted in smaller size robots with higher accuracy and reliability. As a result, robotics is finding more and more applications in Biomedical Engineering. ***Medical Robotics*** and ***Cell Micro-Manipulation*** are two of these applications involving interaction with delicate living organs at very different scales.

Availability of a wide range of imaging modalities from ultrasound and X-ray fluoroscopy to high magnification optical microscopes, makes it possible to use imaging as a powerful means to guide and control robot manipulators. This thesis includes three parts focusing on three applications of ***Image-Guided Robotics*** in biomedical engineering, including:

- Vascular Catheterization: a robotic system was developed to insert a catheter through the vasculature and guide it to a desired point via visual servoing. The system provides shared control with the operator to perform a task semi-automatically or through master-slave control. The system provides control of a catheter tip with high accuracy while reducing X-ray exposure to the clinicians and providing a more ergonomic situation for the cardiologists.

- Cardiac Catheterization: a master-slave robotic system was developed to perform accurate control of a steerable catheter to touch and ablate faulty regions on the inner walls of a beating heart in order to treat arrhythmia. The system facilitates touching and making contact with a target point in a beating heart chamber through master-slave control with coordinated visual feedback.

- Live Neuron Micro-Manipulation: a microscope image-guided robotic system was developed to provide shared control over multiple micromanipulators to touch cell membranes in order to perform patch clamp electrophysiology.

Image-guided robot-assisted techniques with master-slave control were implemented for each case to provide shared control between a human operator and a robot. The results show increased accuracy and reduced operation time in all three cases.

Keywords

Biomedical Robotics, Visual Servoing, Image-guided Robotics, Angioplasty, Cardiac Ablation, Patch Clamp Electrophysiology, 3D Visualization, Master-slave Control

Acknowledgements

Various parts of this project would have not been possible without the inspiring supervision of Professor Patel who has helped me throughout the work while being patient and enthusiastic to discuss new ideas and methods and encouraged me to follow them. I would like to thank my former colleagues: Dr. Jayender who has been very helpful during the first part of this project, Professor M. O. Poulter and Cezar Gavrilovici for their collaboration in the third part of the project, Mansoor Alghooneh for his help in the second part of the project, and also all my colleagues at CSTAR and Robarts for their cooperation and kindness.

I also give my special thanks and appreciation to my lovely and loving wife, Mehrnaz for her tolerance and patience during my graduate studies. I thank my loving parents and sisters for all their support during these years. I would like to dedicate this thesis to my best friend *Amin Firouzfar* who is not among us any more; he taught me how to learn and he gave me the most memorable moments of my life in our long hours of discussions.

Contents

CERTIFICATE OF EXAMINATION **ii**

ABSTRACT **iii**

ACKNOWLEDGEMENTS **v**

CONTENTS **vi**

LIST OF TABLES **xii**

LIST OF FIGURES **xiv**

1 Introduction **1**
- 1.1 Image-Guided Robotics 1
- 1.2 Applications to Medicine and Biology 3
  - 1.2.1 Intravascular Interventions: Angioplasty 3
  - 1.2.2 Intracardiac Interventions: Cardiac Ablation 5
  - 1.2.3 Cell Micromanipulation: Patch Clamping 5
- 1.3 Thesis Outline and Organization 6
- 1.4 Contributions and Publications 7
  - 1.4.1 Chapter 2: Angioplasty 7
  - 1.4.2 Chapter 3: Cardiac Ablation 8
  - 1.4.3 Chapter 4: Patch Clamping 8
- Bibliography 8

2 Angioplasty 11

2.1 Introduction . . . 12
2.2 Experimental Setup . . . 16
2.3 Image-based Tracking . . . 19
  2.3.1 Off-line camera calibration and registration . . . 19
    Camera calibration . . . 22
    Registration . . . 23
  2.3.2 Extracting mask image (Roadmap) . . . 26
  2.3.3 Database creation and selection . . . 29
  2.3.4 Tracking the tip of the catheter . . . 32
  2.3.5 Direction Selection . . . 34
2.4 Sensor Fusion . . . 34
  2.4.1 Magnetic tracking system . . . 36
    Offset calibration . . . 36
  2.4.2 Registration of the two sensors . . . 37
  2.4.3 Modeling the catheter tip motion . . . 39
  2.4.4 Data fusion algorithm design . . . 41
2.5 Autonomous Guidance . . . 44
  2.5.1 Path following . . . 44
  2.5.2 Catheter tip position control . . . 46
2.6 Master-Slave Control . . . 48
2.7 Experimental Results . . . 49
  2.7.1 Autonomous catheter insertion . . . 49
  2.7.2 Nonlinear Behavior of a Catheter . . . 50
    Catheter Insertion at Constant Desired Velocity . . . 50
    Catheter Insertion at Variable Desired Velocities . . . 54
    Catheter Insertion at Variable Stroke Lengths . . . 55
  2.7.3 Master-Slave Control Results . . . 55
  2.7.4 Sensor Fusion Results . . . 59
2.8 Remarks . . . 63
2.9 Conclusion . . . 65
Bibliography . . . 67

3 Cardiac Ablation 74
3.1 Introduction 75
3.2 Experimental setup 77
3.2.1 Calibration and Registration 79
3.3 Robotic System Design 82
3.4 Steerable Catheter Modeling and Control 83
3.5 Online 3D volume reconstruction 88
3.5.1 Stereoscopic visualization 89
3.6 Master-slave control 89
3.7 Results 90
3.8 Conclusion 91
Bibliography 91

4 Patch Clamping 98
4.1 Introduction 99
4.2 Experimental Setup 103
4.3 Autofocus (AF) Algorithm 104
4.3.1 Focusing on a micropipette tip 107
4.3.2 Optimization 109
4.4 System Calibration 110
4.4.1 Calibration Procedure 115
Coarse calibration 115
High-accuracy calibration 116
Parameter Estimation 118
4.5 Micropipette Tip Detection and Tracking 119
4.5.1 Detection under a dry objective 120
4.5.2 Detection under a water-immersed objective 123
4.5.3 Image-based Tracking of Micropipettes 124
4.6 Detection of Water Surface Touch 128
4.7 Image-Based Tracking and Control 131
4.7.1 Visual Servoing 134
Trajectory Generation 135

Trajectory Tracking 137
4.8 Collision Avoidance 138
4.9 Haptic Interface 140
4.9.1 Master-Slave Control 142
4.9.2 Studying Performance of the Master-Slave Control . . . 144
4.10 Online 3D Visualization 144
4.10.1 $3D$ Surface Reconstruction 147
Capturing an Image Volume 147
Preprocessing and Segmentation 147
Post-processing 149
Improving the processing speed 154
Representation 159
4.10.2 Master-Slave Control 160
4.10.3 Real-time Visual Feedback 160
4.11 Software Architecture 161
4.12 Experimental Results 163
4.12.1 Autofocusing 164
4.12.2 System Calibration 164
4.12.3 Tip Detection Error 166
4.12.4 Image-Based Tracking 167
4.12.5 Visual Servoing 168
4.12.6 Master-Slave Control 168
4.12.7 3D Reconstruction 174
Accuracy 174
Speed 175
Accuracy vs. Speed 176
4.12.8 Patch Clamping Results 178
4.12.9 Brain slicing procedures and tissue maintenance 180
4.12.10 Electrophysiology 180
4.12.11 Immunohistochemistry and image acquisition 180
4.13 Conclusion 183
Bibliography 186

5 Concluding Remarks and Future Work 193
5.1 Remarks 193
5.1.1 Angioplasty 194
5.1.2 Cardiac Ablation 194
5.1.3 Patch Clamping 195
5.2 Future Research 195
5.2.1 Angioplasty 196
5.2.2 Cardiac Ablation 196
5.2.3 Patch Clamping 197
5.3 Towards supervised automation in biomedical robotics 197
5.3.1 Visual servoing 198
5.3.2 Open access to raw imaging data 199
Bibliography 199

A Mitsubishi PA10 Robot 201
A.1 Jacobian 201
A.2 Gravity Compensation 203

B SMA-Actuated Catheter 205
B.0.1 Closed-loop Control 206
Bibliography 210

C Augmented Hybrid Impedance Control 212
C.1 Augmented Hybrid Impedance Controller 214
C.2 Redundancy Resolution 214
C.3 Joint based controller 215
Bibliography 216

D Micromanipulators in Patch Clamping Setup 218

E Lens Changing Mechanism 219
Bibliography 219

F Steerable Catheter Actuation Mechanism 221

G SensAble Haptic Devices **222**
Bibliography 222

H Magnetic Tracking **224**
Bibliography 224

I Auto Focus Measures **225**

J Ultrasound Imaging **226**
Bibliography 226

K X-ray Fluoroscopy **227**
Bibliography 227

VITA **228**

List of Tables

2.1 Distortion parameters for the top and side cameras 23

2.2 Intrinsic parameters for the top and side cameras 23

2.3 Tracking and data fusion results in normal conditions 60

2.4 Tracking and data fusion results in case of occlusion 63

2.5 Tracking and data fusion results in case of electromagnetic interference 63

3.1 D-H parameters for the continuum robot model of a steerable catheter. 85

3.2 Specifications of the Bard Scorpion$^{\text{TM}}$ 2 ablation catheter. . . 86

4.1 Comparing different focus measures for the dry objective. . . . 105

4.2 Objective and microscope magnification adjustment parameters. 165

4.3 Objective and microscope magnification adjustment parameters for the 20× water immersion objective. 165

4.4 Tip detection errors chart 166

4.5 Accuracy with respect to manual segmentation. The results are averaged for 3 different cases. 175

4.6 Accuracy vs. speed for the reconstruction algorithm, n is the down-sampling factor, $\mathcal{E}$ is the reconstruction error, T_c, T_r are the capturing and reconstruction time respectively. T_t is the overall time required for capturing and reconstruction and E is the optimization criterion defined in Equation. 4.28. The last column reports the results for the case of processing a region of interest. The last row represents the number of points in the reconstructed surface for each case; Number of manual segmentation points was $N_m = 19441$. 177

4.7 Volume sizes for the reconstruction algorithm in different cases reported in Table 4.6 177

B.1 RMS error (%) for H_∞ control 209

D.1 Specifications of micromanipulators used in the patch clamping setup. 218

List of Figures

1.1 Image-guided robotics in biomedical applications 2

1.2 Image-guided robotics with visual servoing in biomedical applications 4

2.1 Autonomous Insertion Algorithm 15

2.2 Experimental setup for catheter insertion 18

2.3 Part of the testbed and catheter under X-ray 20

2.4 X-ray image of a catheter and guide-wire inside a vessel in the thorax area. Image courtesy of Philips Healthcare. 21

2.5 Different views of the testbed from the top and side cameras: (a) the non-planar chessboard pattern used for registration, (b) out of plane junction 24

2.6 Mask image generation block diagram 28

2.7 Mask image generation: step by step - (a) Sample original noisy image, (b) filtered image after temporal (moving average) and spatial (Gaussian) filtering, (c) Edges detected by Canny filtering, (d) Connected contours with extra pieces removed, (e) Flood-filled inside the tube borders, (f) Final mask image (blown-up tubes). (Note that the colors are changed in (c), (d) to increase visibility) 30

2.8 Desired path selected by the user 31

2.9 Block diagram of the tracking algorithm 32

2.10 Registration pattern; The coordinate system of the camera is shown by X_i, Y_i and the coordinate system of the magnetic tracking system is shown by X_m, Y_m, Z_m. The four corner points are marked as $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and $\mathcal{D}$ 37

2.11 The catheter may have a deflection w.r.t the centerline of the tubes because of the curvatures. 40

2.12 Block diagram of the whole system: the internal structure of the Kalman-based data fusion block is shown in Figure 2.13 . 42

2.13 The Kalman-filter based data fusion block diagram 43

2.14 Autonomous Guidance Finite State Machine 45

2.15 Block diagram for the catheter tip position control; a PI controller is applied before the branching and before the target during slow-down, while the high-gain proportional controller is applied after passing a branch during speed-up. 47

2.16 (i) shows the insertion depth (cm) vs. time (sec), (ii) shows the stroke length set-point (digitized from 0 (min.) to 31 (max.)) vs. time, (iii) shows the desired and the actual trajectory of the end effector of the robot (mm) vs. time, (iv) shows the average force of insertion over one stroke (N) vs. time and (v) shows the actual force of insertion (N) vs. time. The tip of the catheter has been shown in the images above. Different phases of insertion are shown as: (a) normal insertion before the first branch, (b) slow-down before the first branch, (c) no insertion, catheter bends at junction (d) one stroke insertion, (e) speed-up insertion after the first branch, (f) normal insertion before the second branch, (g) slow-down before the second branch, (h) no insertion, catheter bends at junction (i) one stroke insertion, (j) speed-up insertion after the first branch, (k) normal insertion before the target, (l) slow-down before the target (m) increase in stroke length due to flexing of catheter. 51

2.17 (a) Ensemble averaged maximum master-slave tracking error (percentage of the peak master reference) vs. insertion depth (b) Ensemble averaged difference between the maximum robot end-effector and catheter tip movements (percentage of the maximum slave movements) vs. insertion depth 52

2.18 Ensemble averaged deadband (s) vs. insertion depth (mm) . . 53

2.19 Ensemble averaged force of insertion (N) vs. insertion depth (mm) 54

2.20 Average deadband (sec) vs. insertion speed (mm/sec) and linear regression of the data. 55

2.21 Average master-slave tracking error (percent) vs. stroke length (mm) and linear regression of the data 56

2.22 Experimental results: (i) shows the master (haptic device), slave (catheter tip) and robot end-effector position vs. time during catheter insertion (ii) shows the force of insertion measured at the robot end-effector along the direction of insertion. Different phases of insertion are shown as: (a) normal insertion before the first branch, (b) slow-down before the first branch, (c) no insertion, catheter bends at junction (d) insertion after bending is complete at the first junction, (e) wait for SMA actuators to cool (f) normal insertion before the second branch, (g) slow-down before the second branch, (h) no insertion, catheter bends at junction (j) insertion after bending is complete at the second junction, (k) wait for SMA actuators to cool (l) normal insertion before the target 57

2.23 Magnified view of normal insertion in region (a) of Figure 2.22 58

2.24 Normal image and magnetic tracking data fusion when there is no occlusion and no EM interference: (a) and (b) show X and Y coordinates of the catheter tip in pixels vs. time (sec) for image-based tracking (red), magnetic tracking (blue) and the estimated value by Kalman filer (black). (c) and (d) represent the estimated catheter tip displacements V_x and V_y in pixels vs. time (sec). (e) shows the estimated depth (mm) of catheter inserted into the vascular model vs. time (sec). 61

2.25 Image and magnetic tracking in case of occlusion: (a) and (b) show X and Y coordinates of the catheter tip in pixels vs. time (sec) for image-based tracking (red), magnetic tracking (blue) and the estimated value by Kalman filer (black). (c) and (d) represent the estimated catheter tip displacements V_x and V_y in pixels vs. time (sec). (e) shows the estimated depth (mm) of catheter inserted into the vascular model vs. time (sec). 62

2.26 Image and magnetic tracking in case of electromagnetic (EM) interference: (a) and (b) show X and Y coordinates of the catheter tip in pixels vs. time (sec) for image-based tracking (red), magnetic tracking (blue) and the estimated value by Kalman filer (black). (c) and (d) represent the estimated catheter tip displacements V_x and V_y in pixels vs. time (sec). (e) shows the estimated depth (mm) of catheter inserted into the vascular model vs. time (sec). 64

3.1 The ex-vivo experimental setup: (a) Beating heart phantom (Shelley Medical Imaging Technologies), (b) 4D ultrasound probe, (c) Ultrasound probe holder; An ablation catheter is inserted to a ventricle while the ultrasound probe captures images from an apical view. 78

3.2 The image-guided robotic system for catheter insertion. 79

3.3 Block diagram of the system. 80

3.4 Three magnetic tracking sensors (M_1, M_2, M_3) mounted on the distal part of a steerable catheter and fixed using heat-shrink tubes. 81

3.5 A schematics of the three magnetic sensors mounted on the distal part of the catheter. The points O_0, O_1, O_2, O_3, O_4 are the origins of coordinate systems shown in Figure 3.8 where O_4 is the distal tip of the catheter. 82

3.6 CAD design of the catheter actuation mechanism. 83

3.7 Steerable ablation catheter, catheter actuation mechanism and the Mitsubishi PA10 robot after assembly. 84

3.8 Kinematic modeling of a steerable catheter as a continuum robot. 86

3.9 Left: Slave graphical user interface, Right: Master graphical user interface and the haptic device. 90

3.10 Ultrasound Imaging and 3D Reconstruction of a Heart Phantom with a 7-Fr Ablation Catheter: (a) a single slice of ultrasound images; (b,c,d) 3D reconstruction represented from different points of view. The different parts shown in this image are: (1) Left ventricle, (2) Interventricular Septum, (3) Ablation catheter, (4) Right Ventricle and (5,6,7) the ablation catheter from different points of view. 92

4.1 The experimental setup: all the equipment is installed on an anti-vibration table and located in a Faraday cage. The high-speed camera is used for voltage-sensitive fluorescent dye imaging and the image frames captured by the CCD camera are used for visualization and processing purposes in patch clamping. The head-stage is a signal conditioner that holds the micropipette. The linear actuators (M4(1..4) in Appendix D) are used to move the 3-DOF micromanipulators (M3(1..4) in Appendix D) *in* and *out* to facilitate replacing micropipettes. . . 103

4.2 Normalized autofocus measure vs. vertical lens position. . . . 106

4.3 Illustration of the coordinate systems on a schematic of the setup: Cartesian coordinate systems $\mathcal{C}^{ref}$ and $\mathcal{C}^{mic}$ and coordinate system $\mathcal{C}^{man}$ for one of the micromanipulators and $\mathcal{C}^{pipette}(X^p, Y^p, Z^p)$ for another manipulator are shown. Background image courtesy of *Sutter Instruments*. 113

4.4 Micropipette tip under (a) dry and (b,c) water immersion objective lenses. In (b) the micropipette is filled with water, but in (c) only part of it is filled. There are tiny air bubbles visible in (b,c) and some undesired particles are visible in (c). 119

4.5 Tip detection algorithm – results: (a) original image, (b) mask image (negated for better visibility), (c) overlapped $I_{n_{y-}}$ and $I_{n_{y+}}$, (d) proximity of the micropipette tip as detected in I_{n_x}, negated to increase visibility, and (e) the detected line segment along the central axis of the micropipette. 123

4.6 Image-based tracking under dry objective: (a) Original Gaussian smoothed ROI, (b) Background subtracted, (c) Adaptive thresholded, (d) Morphological opening and closing, (e) Hough line detection, (f) tip detected. All images except (a) have been negated to increase visibility. 126

4.7 Augmentation of images to improve robustness of the phase correlation: (a) Original previous image frame $I_n(k-1)$ and (b) after augmentation $\mathcal{I}_n(k-1)$; (c) Original current image frame $I_n(k)$ and (d) after augmentation $\mathcal{I}_n(k)$. 129

4.8 Phase correlation results: (a) cross correlation $r_{n,k}$ of the augmented images $\mathcal{I}_n(k)$ and $\mathcal{I}_n(k-1)$ with its peak located at $(20, 10)$; (b) overlapped images with the displacement vector; the augmented images are used to increase visibility. 130

4.9 Water Surface Touch: (a) Before touching the water surface; (b) while touching; and (c) back up with stretched water surface. 130

4.10 The graphs on the left show $\mathcal{F}_z(\mathcal{I})$ vs. z and the graphs on the right show $\bar{\mathcal{I}}$ vs. z for four experiments with different light intensity. 132

4.11 Visual servoing block diagram. 136

4.12 Collision avoidance forces are calculated based on the closest distance to other obstacles. P_k shows the k^{th} micropipette. F_i is the repulsion force acting on P_i and generated by other micropipettes. 139

4.13 Haptic-enabled control: the coordinates of the haptic device are aligned with those of the microscope in such a way that X_h, Y_h are matched with X_i, Y_i, and Z_h is matched with the depth. There are four modes defined for master-slave control: (a) In-depth motion, where the motion of the haptic device is limited along Z_h; (b) planar motion, where the haptic device is limited to move in a plane orthogonal to Z_h and the corresponding micropipette moves almost only in the image plane; (c) 3D motion, where the haptic device can move in 3D and the user has full control over the corresponding micropipette; and (d) coaxial motion where Z_h is mapped on to the X-axis in $\mathcal{C}^{pipette_n}$. 141

4.14 Master-Slave control architecture: J^{-1} is the Jacobian inverse as defined in Section 4.4. Each time the user presses the stylus switch on the haptic device, the integrator is reset; This is done to increase the workspace resolution of the haptic device. The user should keep the switch pressed in order to move the slave, this is a useful safety feature. 143

4.15 A single neuron cell viewed at different depths: (a) bottom slice, (b) middle slice and (c) top slice. All the images have been contrast enhanced by histogram equalization because the original image contrast is very poor (Figure 4.16(a)). The image plane is not parallel with the focal plane and the slice thickness is also not uniform, therefore we observe non-uniform light illumination and out-of-plane focus effects from top to bottom across the image slice. 148

4.16 Results of the preprocessing filter sequence: (a) Original image $\mathcal{I}_z$, (b) Histogram equalized image $\mathcal{I}_z^{eq}$ and (c) High-pass filtered image $\mathcal{I}_z^g$. 149

4.17 The captured image volume $\mathcal{V}(.,.,.)$ is processed to obtain the initial $2D$ shape from focus $\mathcal{S}(.,.)$. Mask images $\mathcal{N}(.,.)$ and $\mathcal{M}(.,.)$ are also calculated. 150

4.18 (a) $\mathcal{S}^a$ and (b) the corresponding warped surface. The cell cannot be recognized due to the overwhelming existence of undesirable parts, further processing is required. 151

4.19 Block diagram of the reconstruction algorithm. $\mathcal{S}(.,.)$, $\mathcal{N}(.,.)$ and $\mathcal{M}(.,.)$ are from Figure 4.17. 152

4.20 First-level mask image $\mathcal{N}$, (a) before and (b) after thresholding ($\mathcal{N}^\tau$). 153

4.21 (a) $\mathcal{S}^a_\mathcal{N}$ and (b) the corresponding warped surface. 153

4.22 Second-level mask image: (a) $\mathcal{M}$, (b) $\mathcal{M}^\tau$, (c) $\mathcal{M}^\tau$ after morphological erosion, (d) $\mathcal{M}^c$ which is $\mathcal{M}^\tau$ after morphological erosion and dilation, (e) $\mathcal{S}^c$ (masked) and (f) $\mathcal{S}^s$ (median filtered). 155

4.23 Down-sampling on a Gaussian Pyramid. 156

4.24 Finding the region of interest around a cell: (a) original image $\mathcal{I}_{z_{max}}$, (b) $\mathcal{I}^{eq}_{z_{max}}$, (c) $\mathcal{I}^g_{z_{max}}$, (d) after thresholding, (e) after a sequence of dilation, erosion, dilation, Gaussian smoothing and (f) circular Hough transform. The region of interest is selected as a 128×128 pixels square co-centered with the detected circle. 158

4.25 Reconstructed surface as a result of warping the $\mathcal{S}^s(x, y)$. . . . 159

4.26 (a) Part of the gray-scale image $\mathcal{S}^s(x, y)$ before warping, colors inverted to increase visibility, and (b) snapshot of the 3D representation of interpolated voxels in $\mathcal{V}^r(x, y, z)$. 160

4.27 Graphical user interface, haptic device, and stereoscopic goggles are shown for a sample experiment. The represented 3D scene is an augmented reality environment consisting of the partial $3D$ cell reconstruction, the live 2D microscope image and a virtual representation of the micropipettes. 162

4.28 Image tracking results showing tracked P_t vs. calculated P_i coordinates: (a) X_t vs. X_i, (b) $E_X = X_t - X_i$, (c) Y_t vs. Y_i and (d) $E_Y = Y_t - Y_i$. 167

4.29 Results of visual servoing in proximity of an obstacle: (a), (b) represent X_t and Y_t which are the image-based tracking results and (c), (d) represent the collision avoidance forces in each direction. As observed in (c,d), the virtual forces are zero during the first 9.0 seconds because the minimum distance between the moving micropipette and the obstacle is beyond the specified margin. 169

4.30 Visual servoing in proximity to an obstacle. The moving micropipette goes from the starting point to the target location under visual servoing control while collision avoidance forces cause the path to bend around the obstacle instead of following a straight line. 170

4.31 The trajectory of an micropipette while passing close to another micropipette is shown. Collision avoidance forces are represented as arrows. The length and orientation of each arrow shows the magnitude and orientation of the repulsion force at each point on the trajectory. The collision force is not applied to the haptic device when the minimum distance is out of a range as described in Section. 4.8. The controlled micropipette is shown at the beginning, in the middle and at the end of the trajectory. The 4× dry objective is used in this experiment. . 171

4.32 Slave coordinates in $\mathcal{C}^{man}$ and $\mathcal{C}^{image}$. 172

4.33 Master-slave tracking results: the left column shows the master (blue) vs. the slave (red) displacements along the X, Y and Z axes, the right column shows the haptic force along the same axes. Master-slave control was performed in planar mode where tracking is only done in the X and Y directions. The master coordinates are in mm while the slave coordinates are in μm. . 173

4.34 Reconstructed surface in the case of ROI processing. 179

4.35 (a) Voltage clamp recording of a patch clamped interneuron (downward traces showing sodium current flow) and (b) Neuronal firing pattern of a patch clamped interneuron (upward traces indicate action potentials) in layer III of rat piriform cortex. 181
4.36 Morphological reconstruction of four interneurons in layer III of piriform cortex as shown in Figure 4.37. Partial cell body and dendritic arborization reconstruction for cell C and D indicate a lighter biocytin staining of these neurons. 182
4.37 Closer view of micropipettes, objective lenses and the substrate when four pipettes are patched. The objective lens is changed in order to increase visibility. 183
4.38 Micropipette tip is pushing against a cell membrane, the cell is ready for patch clamping. 184
4.39 Four neurons patched. 185

B.1 Active catheter 207
B.2 SMA wires laser micro-welded to stainless steel pads 208
B.3 Closed-loop control for the Active Catheter 208
B.4 Closed-loop control: Tracking a reference in (a) the β (yaw) orientation and (b) the γ (pitch) orientation 211

C.1 Augmented Hybrid Impedance Control 213

E.1 Cross-section of the lens changing mechanism 220

Co-Authorship/Collaboration Statement

- Chapter 1: M. Azizian - sole author
- Chapter 2:
  - M. Azizian - designed and developed image-based tracking, visual servoing and autonomous/semi-autonomous guidance algorithms, collaborated in experiments
  - Dr. J. Jayender - developed control algorithms for robot and active catheters, developed SMA actuated catheters, collaborated in experiments
  - Dr. R. V. Patel proposed problem, discussed design and results, reviewed manuscript
- Chapter 3:
  - Mahdi Azizian - developed the whole system including hardware, software and algorithm design, running experiments
  - Dr. R. V. Patel - proposed problem, discussed design and results, reviewed manuscript
- Chapter 4:
  - Mahdi Azizian - developed the whole system including hardware, software and algorithm design, running experiments

  - Dr. R. V. Patel - discussed design and results, reviewed manuscript
  - C. Gavrilovici - prepared brain slices and lab equipment, analysis of electrophysiology data
  - Dr. M. O. Poulter - proposed original problem, provided the experimental setup, reviewed manuscripts

- Chapter 5: M. Azizian - Sole author

Nomenclature

3D - Three Dimensional
4D - Four Dimensional (3D + time)
ABS - Acrylonitrile Butadiene Styrene
AF - Autofocus(ing)
AHIC - Augmented Hybrid Impedance Control
APF - Artificial Potential Field
API - Application Programming Interface
ARCNET - Attached Resource Computer Network
ASVD - Arteriosclerotic Vascular Disease
CAD - Computer-Aided Design
CNR - Contrast to Noise Ratio
CPU - Central Processing Unit
CT - Computed Tomography
DEP - Dielectrophoresis
D-H - Denavit-Hartenberg (Convention)
DOF - Degree of Freedom
ECG - Electrocardiograph
EM - Electromagnetic
EMI - EM Interference
EMTS - EM Tracking System
EPP - Enhanced Parallel Port
FFT - Fast Fourier Transform
FOV - Field of View
FSM - Finite State Machine

$G\Omega$ - Giga Ohms
GPU - Graphics Processing Units
GUI - Graphical User Interface
ICE - Intracardiac Echocardiography
IGR - Image-Guided Robotics
JT - Jacobian Transpose
MRI - Magnetic Resonance Imaging
MTS - Magnetic Tracking System
PI - Proportional-Integral
PID - Proportional-Integral-Derivative
PSF - Point Spread Function
RANSAC - Random Sample Consensus
RCA - Right Coronary Artery
RMS - Root Mean Square
ROI - Region of Interest
RF - Radiofrequency
SDK - Software Development Kit
SIU - Sensor Interface Unit
SMA - Shape Memory Alloy
SNR - Signal to Noise Ratio
SSD - Sum of Squared Differences
TEE - Transesophageal Echocardiography
TTE - Transthoracic Echocardiography
US - Ultrasound
VTK - Visualization Toolkit
Voxel - Volumetric Pixel

Chapter 1

Introduction

There are various problems in biomedical applications where robot-assisted techniques can play an important role in achieving higher accuracy and reliability while requiring less time and providing a more ergonomic interface for manual operation. This includes applications ranging from research in medicine and biology to medical diagnosis and treatment. The availability of various imaging modalities along with the evolving computational power of modern computers, makes it efficient to use real-time or online imaging to guide or control robots to perform a task.

1.1 Image-Guided Robotics

Figure 1.1 represents a block diagram of image-guided robotics in its simplest form where the imaging data is not directly used to control the robot. In this configuration, the tool-organ interaction is controlled by the human operator and the overall control loop is closed via visual feedback provided to the user. This is the common configuration used in robotic surgery in systems such as Intuitive Surgical's *da Vinci*®. Peters [1] has studied various applications of image-guided robotics of this type applied to surgical procedures. The same concept is also used in conventional methods of cell micromanipulation where the user looks at live microscope images while moving a tool installed on a micromanipulator using manual control knobs as in patch clamp electrophysiology [2].

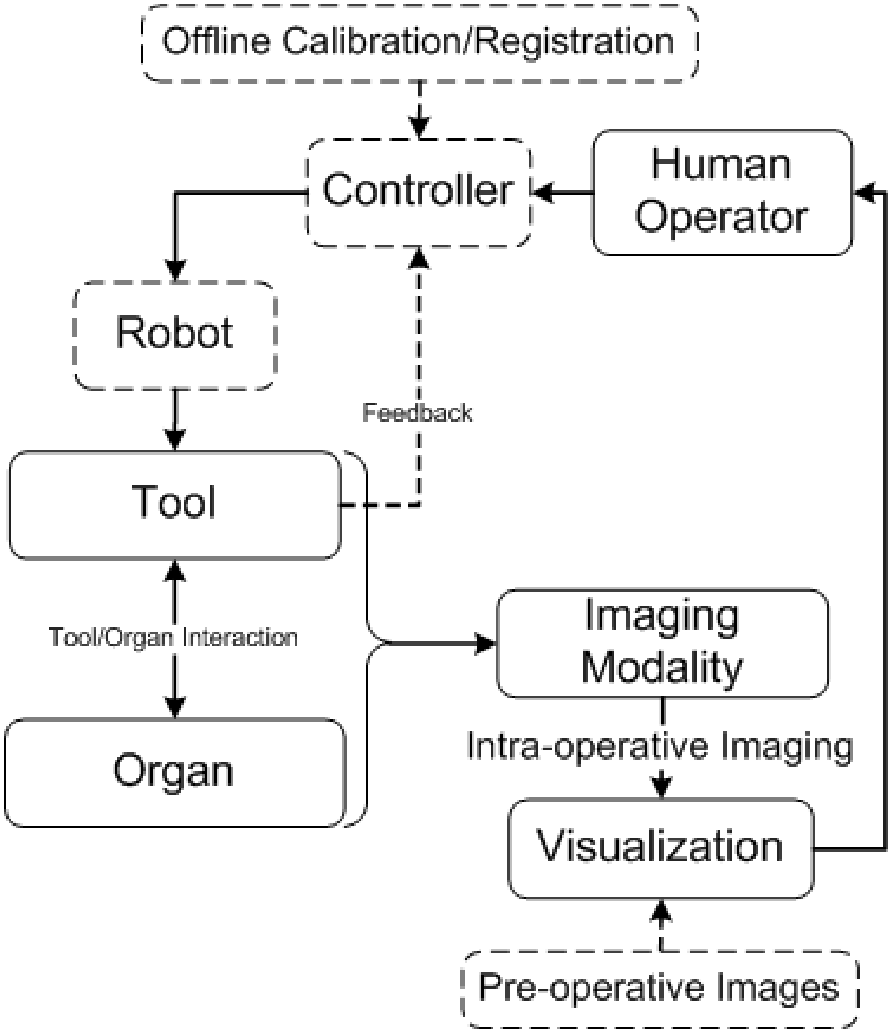

Figure 1.1: Image-guided robotics in biomedical applications

Although visual feedback to the operator can make the procedure more efficient, there is much more information in the images that can be used as feedback to the robot controller. Figure 1.2 represents an alternative configuration where image-based tracking of the tool or the organ is performed on the intra-operative images online or in real time. The user can still get a visual feedback and interact with the controller through a haptic device or a graphical user interface while an internal control loop performs sub-tasks by using feedback provided by image-based tracking algorithms.

The configuration shown in Figure 1.2 can represent a supervised control (autonomous or semi-autonomous) or a master-slave control scheme. This can provide shared control between the human operator and the controller [3].

1.2 Applications to Medicine and Biology

There are several biomedical applications for image-guided robotics, three of which form the topics of this thesis.

1.2.1 Intravascular Interventions: Angioplasty

Certain diseases can be treated by inserting a long flexible tube (i.e., catheter) through the vasculature and performing a minimally invasive intervention. Arteriosclerotic vascular disease (ASVD) is a result of build-up of fatty materials in arteries which may partially or completely block the blood flow to different organs [4]. Angioplasty is an intravascular intervention where the narrowed or blocked blood vessel is mechanically widened. A collapsed balloon is inserted to the narrowed location, the balloon is then inflated to open up the blood vessel. Sometimes a superelastic *stent* is also deployed to keep the blood vessel open [5]. The major challenges of the conventional methods of angioplasty include: (a) the X-ray exposure to the clinician and (b) the high accuracy and reliability required for positioning the catheter tip [6]. An image-guided robot-assisted technique was proposed to address these challenges. This work is described in more details in Chapter 2. This work was performed in collaboration with Dr. Jayender who developed an active catheter and also the controllers for a robot manipulator used for catheter insertion. Details of his work can be found in his PhD thesis [7] and are not included this thesis.

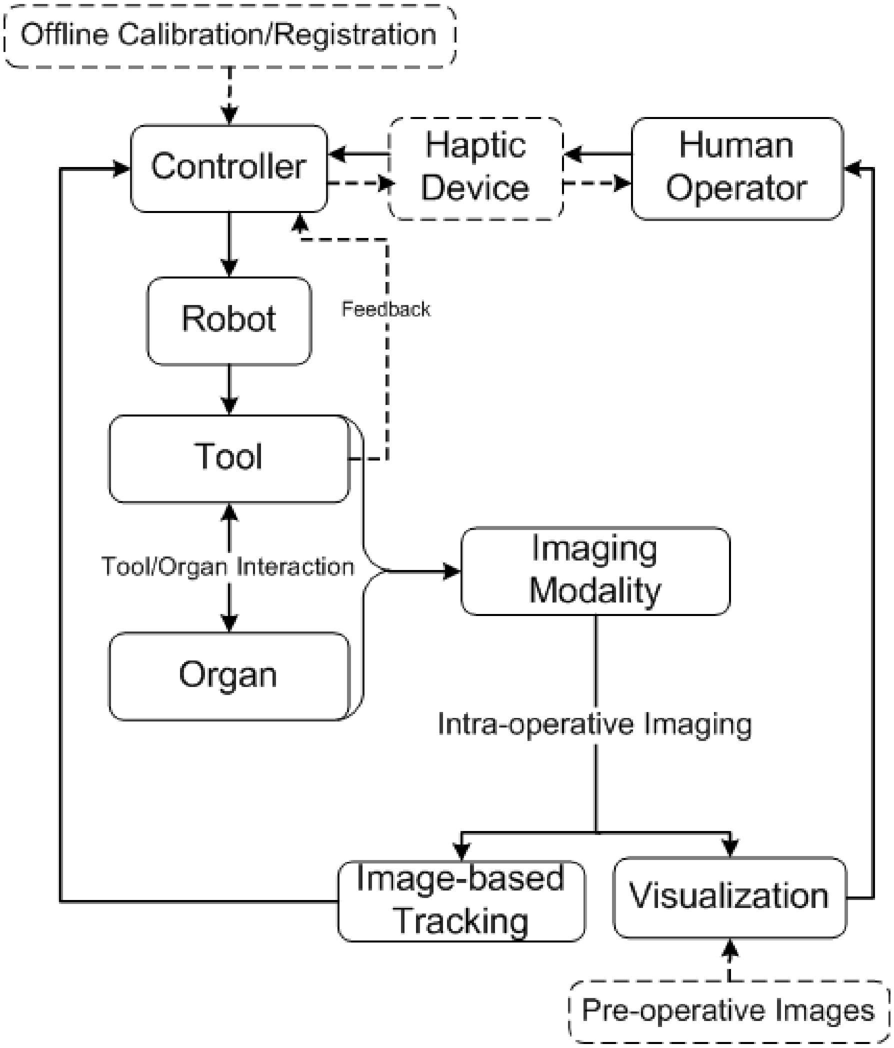

Figure 1.2: Image-guided robotics with visual servoing in biomedical applications

1.2.2 Intracardiac Interventions: Cardiac Ablation

Some of cardiac diseases can be treated by inserting a catheter into the heart chambers through the vasculature. Cardiac arrhythmia is a condition where the heart beat may become too fast or too slow or may become irregular. This can cause different problems ranging from annoying palpitations (conscious awareness of the heart beat) to cardiac arrest. Cardiac arrhythmia is caused by abnormal electrical activity in the heart chambers. One treatment method is to insert a catheter into the heart chambers in order to destroy the faulty regions by burning or freezing part of the endocardium [8]. Major challenges in conventional methods of cardiac ablation include (a) accurate positioning of the catheter tip, and (b) keeping contact with a target point despite the beating heart motion. An image-guided robotic system was proposed to facilitate the positioning of the catheter tip through a master-slave control scheme with coordinated visual feedback. Details of this work are included in Chapter 3.

1.2.3 Cell Micromanipulation: Patch Clamping

There are several applications where micromanipulation of live cells is required. Pronuclear DNA injection, indentation of cell membranes to measure their mechanical properties and several other applications require mechanical micromanipulation of cells. Patch clamp electrophysiology is another application where electrodes are attached to ion channels of excitable cells (e.g. neurons) in order to record their electrophysiological activities [2]. This can be used to study the effect of different drugs for treatment of neurological disorders such as epilepsy. The major challenge rises from the fact that it is very difficult to locate and guide micro-electrodes under microscope images because of the limited field of view. A microscope image-guided micromanipulation technique was developed in order to address these challenges. Detailed description of this work is included in Chapter 4.

1.3 Thesis Outline and Organization

In this thesis we study three applications of image-guided robotics with biomedical engineering applications. In Chapter 2, an image-guided robot-assisted technique for catheter insertion is described where the main application is aimed at intravascular interventions such as angioplasty. In Chapter 3, an image-guided robotic system for intracardiac catheterization is described that focuses on cardiac ablation. In Chapter 4, a microscope image-guided micromanipulation system is presented where the main application is micromanipulation of neurons in live brain tissues for patch clamp electrophysiology. The thesis is prepared in the integrated articles format and each of the Chapters 2, 3 and 4 includes a complete description of the corresponding work as well as review of the related literature for each part. Chapter 5 contains an overall conclusion for the work with suggestions for future research. An outline of the work in each of the three parts of the thesis is as follows:

- Chapter 2 starts with an introduction in Section 2.1 which is followed by a description of the experimental setup in Section 2.2. The image-based tracking catheter tip algorithm is described in Section 2.3 while Section 2.4 discusses the fusion of image-based and magnetic tracking of the catheter tip using Kalman filtering. Section 2.5 focuses on autonomous guidance of a catheter and master-slave insertion of a catheter is described in Section 2.6. Experimental results are discussed in Section 2.7. Some remarks and conclusions are given in Sections 2.8 and 2.9.

- Chapter 3 starts with an introduction to the cardiac ablation problem. The experimental setup used in this part of the work is described in Section 3.2. Section 3.3 describes the robotic system for actuation and control of steerable catheters and Section 3.4 discusses modeling and control of a steerable catheter. Online 3D reconstruction of ultrasound images is described in Section 3.5 while Section 3.6 covers the master-slave control. Results are given in Section 3.7 and Section 3.8 concludes this chapter.

- The experimental setup used in Chapter 4 is described in Section 4.2. Section 4.3 discusses the design and implementation of algorithms for automatic focusing on micropipettes and tissue slices while Section 4.4 describes the calibration and registration algorithms and methods. Image processing algorithms for detection and tracking of micropipette tips using microscope images are described in Section 4.5. Section 4.6 presents the image processing algorithm for detection of water surface contact using microscope images captured through a water immersion objective lens. The visual servoing of micropipette tip positions is described in Section 4.7. Sections 4.8 and 4.9 present the artificial potential field algorithm for calculation of collision avoidance forces and the haptic interface for master-slave control of the micropipettes, respectively. Online 3D reconstruction and visualization of neurons using microscope images is discussed in Section 4.10. Section 4.11 contains the software architecture, and the experimental results for this chapter are presented in Section 4.12.

1.4 Contributions and Publications

This thesis can be divided to three main parts that are covered in Chapters 2, 3 and 4. The major contributions made by the author in each part of the thesis are as follows:

1.4.1 Chapter 2: Angioplasty

The work described in Chapter 2 is the first image-guided robotic system for catheter insertion using visual servoing and supporting different modes of control. This work has been published in a number of peer-reviewed conference proceedings and journal papers including [6, 9–11] and [12].

1.4.2 Chapter 3: Cardiac Ablation

The research described in Chapter 3 is novel in context of a master-slave robotic system which is able to actuate conventional steerable catheters providing online 3D visualization of the beating heart. This work was the latest part of the thesis and part of the results were published in [13]. Continuation of this work could be the subject of another PhD thesis.

1.4.3 Chapter 4: Patch Clamping

The system described in Chapter 4 is the first system reported for performing image-guided robot-assisted multiple electrode patch clamp electrophysiology that can be applied to existing patch clamp setups. Parts of this work have been published or are under review in a number of peer-reviewed conference proceedings and journal papers [14–18].

Bibliography

[1] Terry M Peters. Image-guidance for surgical procedures. *Physics in medicine and biology*, 51(14):R505–40, July 2006.

[2] Areles Molleman. *Patch Clamping: An Introductory Guide to Patch Clamp Electrophysiology.* Wiley, 2003.

[3] W.B. Griffin. *Shared Control for Dexterous Telemanipulation with Haptic Feedback.* PhD thesis, Stanford University, Dept. of Mech. Eng., 2003.

[4] Anthea Maton. *Human Biology and Health.* Englewood Cliffs, New Jersey, USA: Prentice Hall, 1993.

[5] Robert A. Morgan and Eric Walser, editors. *Handbook of Angioplasty and Stenting Procedures.* Techniques in Interventional Radiology. Springer, 1st edition, 2010.

[6] J. Jayender, M. Azizian, and R. V. Patel. Autonomous image-guided robot-assisted active catheter insertion. *IEEE Transactions on Robotics*, 24(4):858–871, 2008.

[7] Jagadeesan Jayender. *Haptics Enabled Robot-Assisted Active Catheter Insertion*. PhD thesis, The University of Western Ontario, 2007.

[8] Shoei K. Stephen Huang and Mark A. Wood, editors. *Catheter Ablation of Cardiac Arrhythmias*. Saunders, 2006.

[9] M. Azizian, J. Jayender, and R.V. Patel. Image processing algorithms for real-time tracking and control of an active catheter. In *European Control Conference (ECC)*, 2007.

[10] J. Jayender, M. Azizian, and R.V. Patel. Autonomous robot-assisted active catheter insertion using image guidance. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 889 –894, 2007.

[11] J. Jayender, M. Azizian, and R.V. Patel. Bilateral telemanipulation of a flexible catheter in a constrained environment. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 649 –654, 2008.

[12] M. Azizian and R. Patel. Data fusion for catheter tracking using kalman filtering: Applications in robot-assisted catheter insertion. In *Proceedings of SPIE*, volume 7964, 796413, 2011.

[13] M. Azizian and R. Patel. Intraoperative 3D stereo visualization for image-guided cardiac ablation. In *Proceedings of SPIE*, volume 7964, 79640F, 2011.

[14] Mahdi Azizian, Rajni Patel, Cezar Gavrilovici, and Michael O. Poulter. Image processing techniques in computer-assisted patch clamping. In *SPIE Imaging, Manipulation, and Analysis of Biomolecules, Cells, and Tissues VIII*, volume 7568, pages 75681E1–12, 2010.

[15] M. Azizian, R. Patel, C. Gavrilovici, and M. Poulter. Computer-assisted patch clamping. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 4131 –4136, 2010.

[16] M. Azizian, R. Patel, C. Gavrilovici, and M. Poulter. Image-guided robot-assisted microscope objective lens positioning: Application in patch clamping. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[17] M. Azizian, R. Patel, C. Gavrilovici, and M. Poulter. Image-guided patch clamp electrophysiology. *under review in Journal of Biomedical Optics*, 2011.

[18] M. Azizian, R. Patel, C. Gavrilovici, and M. Poulter. Semi-automated patch clamp electrophysiology. *under review in IEEE Transactions on Automation Science and Engineering*, 2011.

Chapter 2

Angioplasty

Exposure to X-ray radiation is a great risk for cardiologists during long intravascular interventional procedures. Angioplasty is an intravascular procedure used for treatment of arteriosclerotic vascular diseases wherein the clinician guides a catheter into the femoral artery in the leg (or less commonly through the radial or brachial artery) under X-ray fluoroscopy. The angioplasty procedure often extends to over an hour. A clinician performs several hundred such procedures over his/her lifetime, leading to an accumulation of the total radiation he/she is exposed to. In this part of the thesis, we investigate an image-guided robot-assisted technique for catheter insertion and catheter tip position control. The tip of an active catheter is tracked in real-time to provide information on the location of the catheter which determines the optimal stroke length of insertion for a robot manipulator and the necessary bending angle for an active catheter. The catheter is autonomously guided from the point of entry to the site of plaque buildup, thereby shielding the clinician from harmful radiation due to the X-rays used for imaging and providing a more ergonomic approach for catheter insertion. Experimental results are given to illustrate the robot-assisted catheter insertion procedure using image guidance.

2.1 Introduction

Angioplasty is a minimally invasive procedure that involves the insertion of a catheter into a blood vessel for removal of blockages in blood flow. There are several problems associated with the conventional way of performing angioplasty. The arteries through which the catheter passes are extremely complex and delicate. The repeated insertion of a catheter through several trials could tear a blood vessel at a junction and cause bleeding [1]. Furthermore, the clinician has no feedback on the force applied by the tip of the catheter on the walls of the blood vessel. Excessive force could rupture the blood vessel or dislodge plaque. The clinician could have prolonged exposure to radiation and be subjected to a high-level of fatigue caused by poor ergonomics of the current procedure. These pose danger and/or discomfort to the clinicians who perform the procedure over a prolonged period of time [2], [3]. There is therefore a need to develop technology for a more accurate, safer and more reliable approach for catheter insertion that can reduce the potential for injury to patients and radiation exposure and discomfort to clinicians.

In this part of the thesis, we explore the possibility of employing a robotic system to insert a catheter into the vasculature and control its tip position inside the vasculature based on information received from image processing algorithms. Position/force control of a robot end-effector and design and control of an active catheter were the subject of another PhD thesis that can be found in [4]. A brief description of the active catheter design and control is included in Appendix B. This chapter is focused on the author's work on three main parts including (a) real-time image-based tracking of catheter tip, (b) fusion of image and magnetic tracking information, and (c) shared control of the catheter insertion procedure.

There are a few research groups that have studied the problem of employing a master-slave robotic system to aid in the insertion of a catheter. Fukuda et al. [5], [6], [7] have developed a system with 3 degrees of freedom (DOF) controlled by a joystick. Thakur et al. have developed a master-slave catheter insertion mechanism [8]. Two recent companies, Corindus Inc. [9], [10] and

Hansen Medical [11], [12], [13] have also developed master-slave systems with 2-DOFs (insertion/retraction and twisting) to insert a catheter into the vasculature using a joystick. However, none of these systems uses the information provided by the imaging system in order to control the catheter. In other words, only the position of the proximal end of the catheter is being controlled and the distal tip position is not controlled by a closed-loop system. This can be a major drawback because the catheter is a flexible system operating in a constrained environment (i.e., the vasculature). The catheter shows nonlinear behavior; therefore it is not trivial to control the proximal end and expect that the distal tip follows the same motion. Any abrupt motion of the distal end of the catheter may cause damage to the blood vessels or may dislodge plaque and cause fatal conditions. It is worth mentioning that the robot developed by Hansen Medical is specific for application in cardiac ablation, which requires significantly larger catheters (8Fr) and minimal guidance through the vasculature since the path from the point of insertion in the femoral vein to the heart is quite straightforward. This is not the case with angioplasty wherein a much smaller catheter (typically 5Fr) is used to open blockages in remote arteries. Another company, Stereotaxis [14] has developed a magnetic navigation system that can be used to navigate the distal end of a catheter in 3D using strong magnetic fields. This system requires giant magnets, specialized catheters equipped with permanent magnets and only controls the orientation of the distal end rather than its position while it is being inserted into the vasculature.

Jayender et al. [15] have developed an Augmented Hybrid Impedance Control (AHIC) scheme to perform catheter insertion for angioplasty by controlling the force of insertion, preventing buckling of the catheter by controlling the moments in the orthogonal directions to the direction of insertion and maintaining a remote center of motion. A brief description of the Augmented Hybrid Impedance Control scheme is included in Appendix C. Jayender et al. have also developed an active catheter instrumenting a catheter with Shape Memory Alloy actuators and controlling the strain in the actuators using PI control, the gains of which are obtained by optimizing certain cost functions to ensure robust stability and performance [4]; This is described in Appendix B. We have used this system to develop a visually servoed autonomous catheter insertion algorithm to guide the catheter from the point of entry to the destination using image processing algorithms while performing closed-loop control over its distal tip position (or velocity). A block diagram of this image-guided robot-assisted catheter insertion system is shown in Figure 2.1. The benefits of developing such a system are considerable - reduction in the amount of stress and fatigue to the clinician, reduction in the harmful X-ray exposure to the clinician, higher accuracy in position control of the distal tip of a catheter, and easier navigation of the active catheter within the complex vasculature using image guidance.

For closed-loop control of the catheter distal tip position, it is required to track the end of the catheter in real time. There are different methods for real-time tracking of the tip of a catheter inside the body: (a) one is by using magnetic tracking by RF-coils and MR scanning, which can achieve a tracking rate of 10fps as reported in [16]; (b) another method is using image processing techniques on real-time images captured by X-ray fluoroscopy, MRI, ultrasound or other medical imaging modalities; (c) magnetic tracking can also be used for tracking the catheter tip. MR image-guided catheter tracking, as reported in [17], can achieve a maximum rate of 3.1fps. In [18], an off-line algorithm (based on adaptive spatial-temporal filtering) has been developed to detect the end of a catheter in X-ray fluoroscopic images. A catheter tracking algorithm based on 3-D ultrasound imaging is discussed in [19]. Novotny et al. [20] have developed a real-time tracking method based on 3D ultrasound images. This method was used for visual servoing of a catheter in order to compensate for heart beat motion in mitral valve repair procedures by Kesner et al. [21], [22]. We have developed a novel real-time tracking algorithm to follow the motion of the tip of a catheter inside the body using X-ray fluoroscopy images. The tracking algorithm finds the tip of the catheter in each frame (when a motion is detected) in real time. An image-based visual servoing approach is then used to determine the stroke length of insertion for the robot and desired bending angle for the active catheter in order to control the

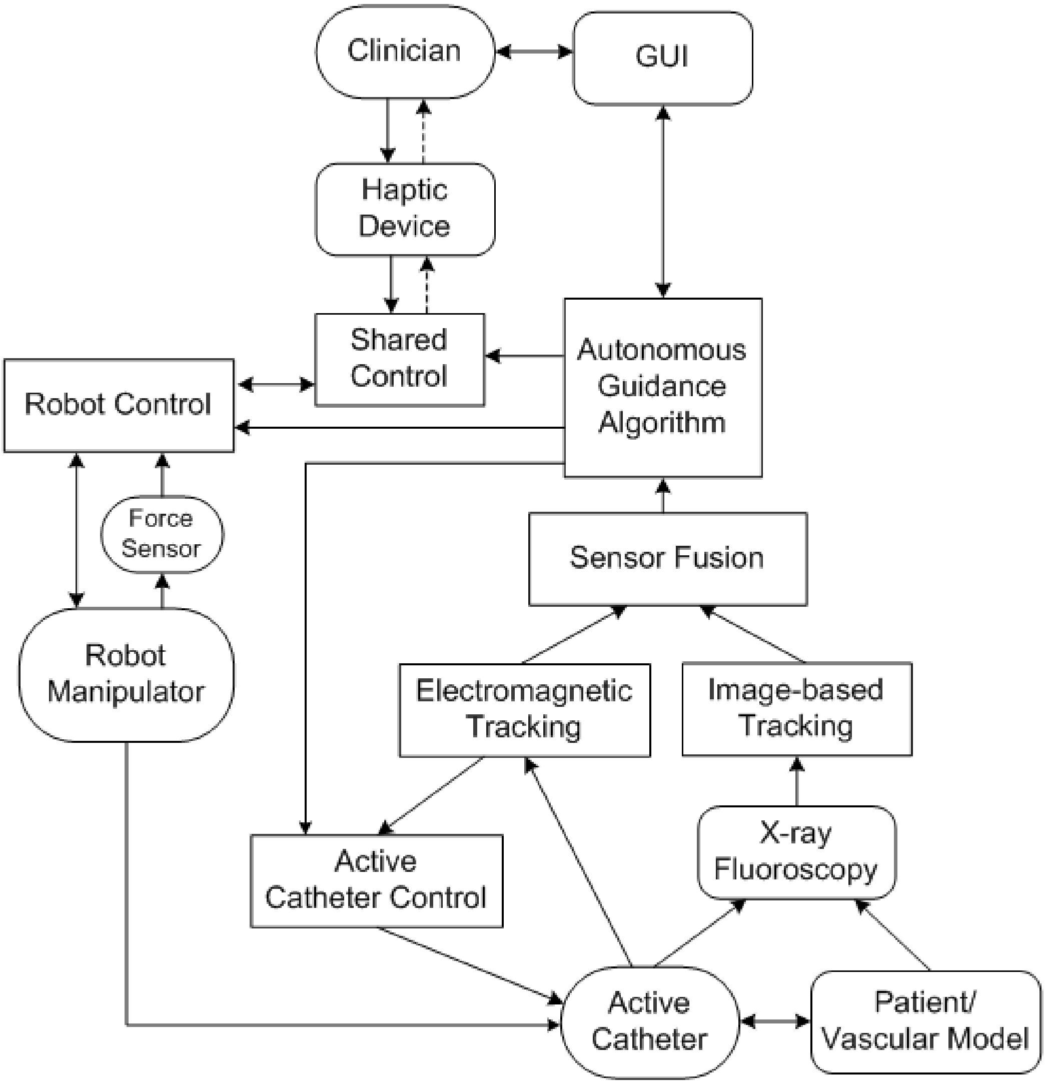

Figure 2.1: Autonomous Insertion Algorithm

distal tip position of the catheter and guide it through the vasculature. We have also developed a data fusion technique to combine image-based tracking and magnetic tracking to achieve higher accuracy and reliability.

The author believes that this is the first work which has developed a visually servoed robotic catheter insertion and guidance system. The active catheter can be guided from the point of insertion to the target under the supervision of a cardiologist autonomously or using a master-slave control scheme. In the autonomous mode, the clinician provides the initial and final destination through a Graphical User Interface (GUI) and maintains supervisory control over the robot while the slave robot inserts the catheter into the body based on the images obtained. Image processing algorithms have been implemented to track the end-point of the catheter in real time. Based on information obtained from the image processing algorithm, the autonomous guidance algorithm provides the stroke of insertion to the *Robot Control* block shown in Figure 2.1 and the desired angle of bending in the catheter to the *Catheter Control*. The robot control block implements an augmented hybrid impedance (AHIC) control scheme in order to control the position of the end-effector as well as the force exerted by the end-effector on the environment [23]. The catheter control block ensures accurate tracking of the desired bending angle. In the master-slave control mode, the catheter distal tip position is controlled using image-based visual servoing in order to track the user's hand motion on the master [24].

2.2 Experimental Setup

The experimental setup is shown in Figure 2.2. A 7-DOF Mitsubishi PA10-7CE robot has been employed to perform robot-assisted catheter insertion. In our laboratory, it is controlled by a host computer via the ARCNET protocol[1]. A two-client/one-server architecture has been implemented to perform the experiments. The clinician interacts with a GUI running on the server,

---

[1] ARCNET(Attached Resource Computer NETwork) is a local area network (LAN) protocol which is popular in the area of embedded systems (http://www.arcnet.com/).

which is a Pentium IV 2.4 GHz, 1 GB RAM computer running Windows XP. The orientation of the catheter tip is measured by means of a 5-DOF magnetic sensor. More details on the magnetic tracking system are included in Appendix H. The magnetic tracking system is interfaced to one of the clients, which is a Pentium IV 3.0 GHz, 1 GB RAM computer running Windows XP. A periodic timer running at 35 ms is responsible for obtaining the magnetic sensor data. The second client computer (Intel Pentium 3.0 GHz, 1.5 GB RAM) controls the robot and sends data packets via the ARCNET protocol to the servo controller. An ATI Gamma 6-DOF force/torque sensor has been used as the wrist force sensor on the robot to measure the force exerted by the robot end-effector while inserting the catheter into the testbed.

The active catheter developed for the experiments uses a typical 5Fr guide catheter from Medtronics Inc. The testbed built to validate the catheter insertion algorithm consists of vinyl tubes of varying internal diameters (3.0mm, 4.5mm and 6.0mm). The diameters of the tubes chosen are smaller than the normal size of the femoral artery, which is 8.75mm ± 2.11mm [25], [26]. The bending angles and radii of the tubes are close to those of arteries in the human body to make the testbed as realistic as possible for a proof-of-concept. The second bifurcation chosen along the desired path was placed at a random angle in 3D space. The image processing algorithms determine the angles of the various branches offline and provide the necessary bending angles to the active catheter when its distal tip reaches a bifurcation. The process is similar to making a roadmap of the vasculature using angiography.

The desired specifications for the catheter insertion problem are as follows: (a) The robot should be programmed to move along a pre-determined path to insert the catheter into the body; (b) the force of insertion should be regulated to ensure that there is no damage to the epithelial cells of the blood vessel; (c) the position of the tip of the catheter should be tracked and the speed of insertion should be controlled to achieve a smooth trajectory near the branches and the target; (d) at a bifurcation point, the active catheter should precisely bend by the amount specified by the image processing algorithm. Items (a) and (b) are addressed in [4] and items (c) and (d) are described in this chapter.

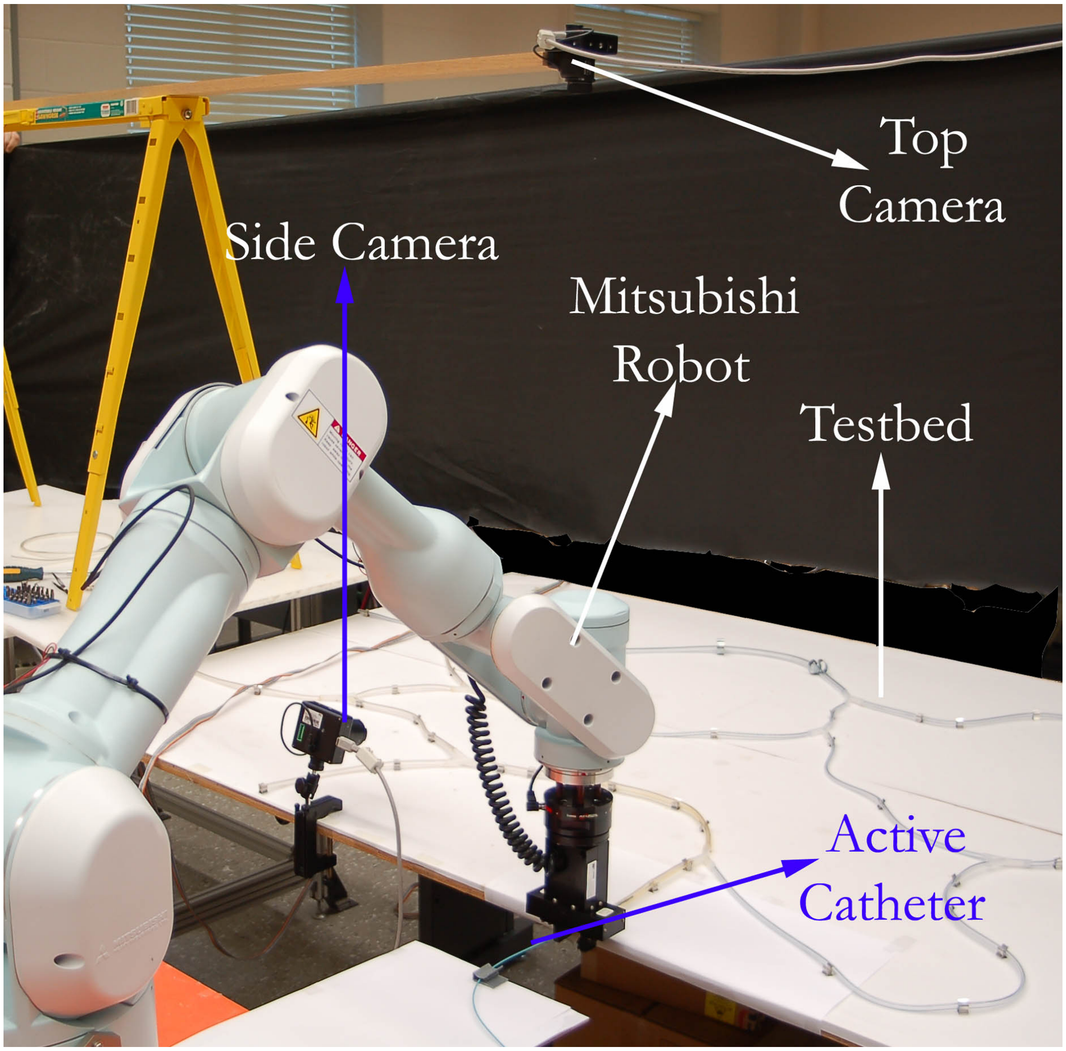

Figure 2.2: Experimental setup for catheter insertion

2.3 Image-based Tracking

In order to develop a visual servoing control scheme with autonomous image guidance, the first step is to develop real-time image-based tracking of the distal end of the catheter. Based on the position of the tip of the catheter, the future course of insertion is decided. In our laboratory tests, we have used a fire-wire digital camera capturing 1024×768 pixel 8-bit gray scale images at 30fps to obtain images as the catheter is inserted into the testbed. The maximum frame rate is limited by two bottlenecks. One is the time resolution of the imaging device (X-ray fluoroscopy) and the other is the time required for processing a captured frame. Most of the modern X-ray fluoroscopy systems have a time resolution of up to 30 frames per second [27]; therefore the selection of a 30fps camera is realistic. The images obtained from this camera are similar to X-ray fluoroscopic images in terms of brightness, contrast and frame capture rate. However, X-ray fluoroscopic images have a higher resolution as compared to the images obtained from the fire-wire camera. These similarities highlight that the algorithm developed should work with X-ray fluoroscopic images and its accuracy can be further improved with higher resolution X-ray images. The spatial resolution of a GE Series 7700 rotational C-arm X-ray fluoroscopy machine is reported by Zhang et al. [28] to be as good as 0.651mm/pixel which is better than the resolution of the firewire camera which has a spatial resolution of around 1.0mm/pixel in our case. A sample X-ray image of a part of our testbed and the active catheter is shown in Figure 2.3. As seen in this image, the SMA and magnetic sensor wires improve the contrast and make the catheter more visible. A clinical fluoroscopic image of a catheter and guide-wire in the human body is shown in Figure 2.4 where the catheter is a standard (non-active) catheter and is clearly visible. An active catheter will result in a higher contrast as seen in Figure 2.3.

2.3.1 Off-line camera calibration and registration

Two cameras, one on top of the testbed and another at one side of the testbed, were used for building a 3D roadmap of the tubes. The two cameras are non-

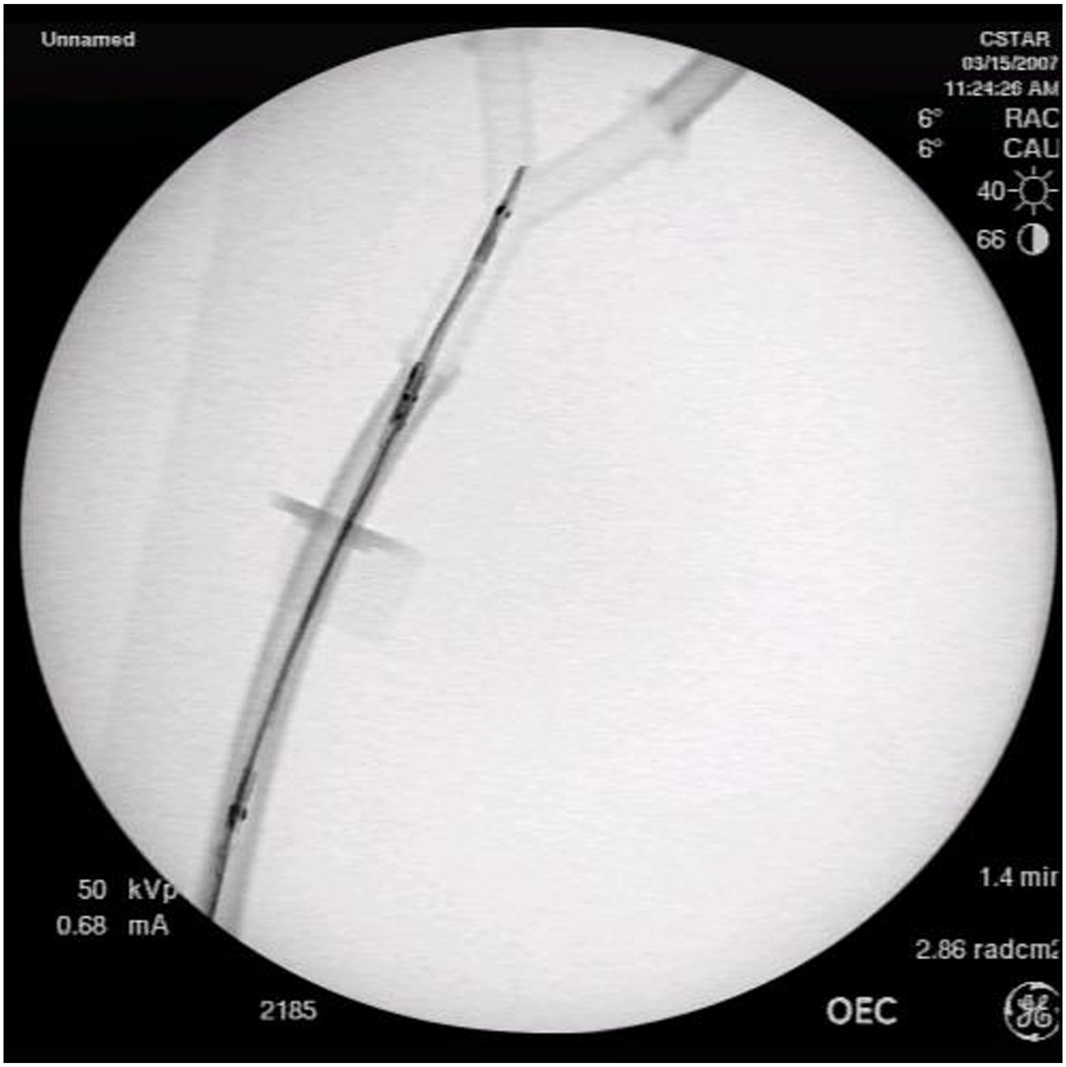

Figure 2.3: Part of the testbed and catheter under X-ray

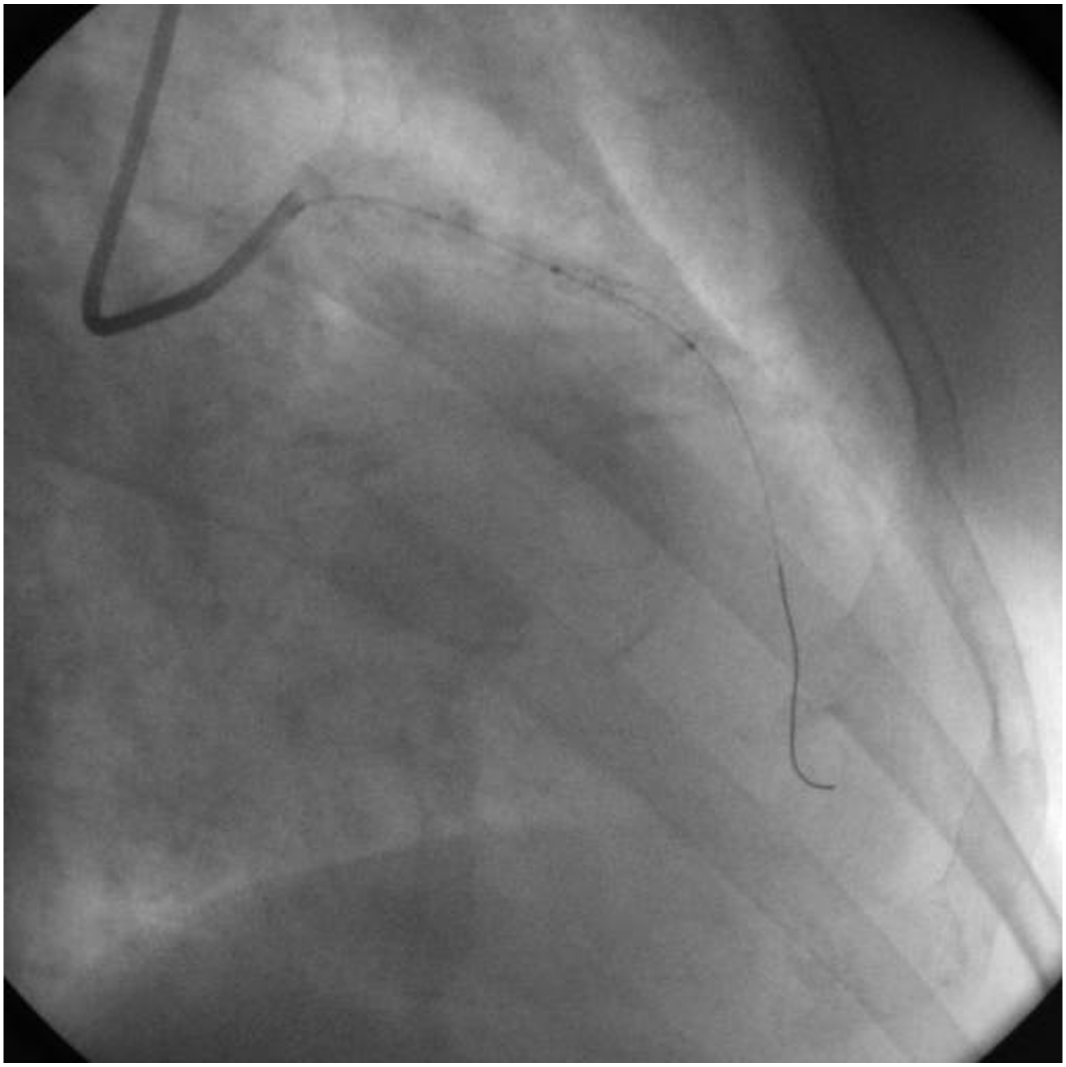

Figure 2.4: X-ray image of a catheter and guide-wire inside a vessel in the thorax area. Image courtesy of Philips Healthcare.

parallel. The side camera emulates an alternate angle of the C-Arm around the patient.

Camera calibration

To remove the barrel effect (due to lens distortion) and to find the perspective parameters, each of the cameras was calibrated using Zhang's method [29], [30]. This method is simple and is mainly based on different non-parallel views of a planar pattern (e.g. a chessboard).

We have used a quadratic radial distortion model with four parameters. If the coordinates of a point in the undistorted image are (ξ, η), then the coordinates of the same point in the distorted image, $(\hat{\xi}, \hat{\eta})$ are obtained as follows:

$$\begin{aligned}\hat{\xi} &= \xi(1 + k_1 r^2 + k_2 r^4 + 2p_1\eta + 2p_2\xi) + p_2 r^2 \\ \hat{\eta} &= \eta(1 + k_1 r^2 + k_2 r^4 + 2p_1\xi + 2p_2\eta) + p_2 r^2\end{aligned} \tag{2.1}$$

where $r = \sqrt{\xi^2 + \eta^2}$ and k_1, k_2, p_1 and p_2 are the coefficients of radial distortion. The perspective transformation matrix is assumed as:

$$A = \begin{bmatrix} f_\xi & \alpha & c_\xi \\ 0 & f_\eta & c_\eta \\ 0 & 0 & 1 \end{bmatrix}$$

where (c_ξ, c_η) is the principal point of the image (near the center of the image), α is the parameter describing the skewness of the two image axes and f_ξ, f_η are the focal lengths along the horizontal and vertical axes of the image, respectively. The algorithm that is used for computing intrinsic and distortion parameters is as follows: a) Capture different non-parallel views of a 5 × 7 black and white planar chessboard pattern; b) detect the corners of the pattern in each of the views; c) find a homography for all points in the set of images, where a homography is a matrix of perspective transforms between the calibration pattern plane and the camera view plane; d) initialize intrinsic parameters, and set the distortion parameters to zero; e) find extrinsic parameters for each image of the pattern; f) minimize the error of the projection points with all the parameters using a maximum likelihood algorithm. For more details, please refer to [29], [30].

The distortion parameters k_1, k_2, p_1, p_2 and the intrinsic parameters f_ξ, f_η, α, c_ξ, c_η, resulting from this algorithm for each of the cameras are listed in Tables 2.1 and 2.2, respectively.

	k_1	k_2	p_1	p_2
Side	-0.257098	0.281629	0.0	0.0
Top	-0.250099	0.200001	0.0	0.0

Table 2.1: Distortion parameters for the top and side cameras

	f_ξ	f_η	α	c_ξ	c_η
Side	585.933	560.612	6.959	335.826	151.475
Top	1033.32	1031.90	-20.145	641.685	346.308

Table 2.2: Intrinsic parameters for the top and side cameras

The accuracy of this calibration method has been studied by Sun et al. [31] and has been reported to be 1.0028 pixel.

Registration

A non-planar chessboard pattern, as shown in Figure 2.5, is used to register the two cameras to the universal coordinates. The registration process is as follows:

- The coordinates of the corners of the non-planar pattern are obtained by means of a magnetic sensor and are logged into the image processing server. The set of the coordinates obtained from this process is denoted by M.

- Images of the non-planar pattern are obtained from each of the cameras (side and top cameras). We denote these images as I_s and I_t respectively.

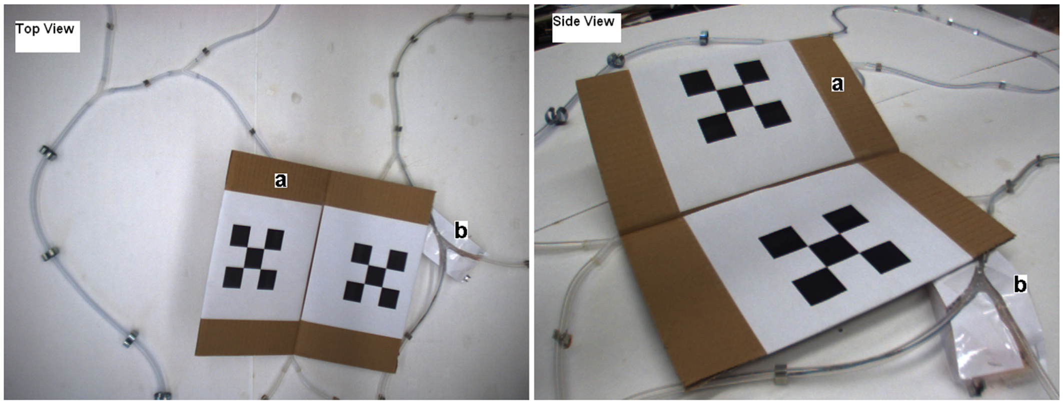

Figure 2.5: Different views of the testbed from the top and side cameras: (a) the non-planar chessboard pattern used for registration, (b) out of plane junction

- The images I_s and I_t are undistorted using the distortion parameters obtained in the offline camera calibration for each camera. We call the undistorted images I'_s and I'_t respectively.
- A pattern recognition algorithm based on Harris corner detection [32] is used to find the coordinates of all the corner points in the images I'_s and I'_t separately. We denote the set of corner points obtained from I'_s and I'_t as P_s and P_t respectively.
- The recognized points from both cameras are arranged in the same order as the points in M. The set of the points P_s and P_t arranged in the new order are denoted by P'_s and P'_t respectively.
- A least-squares method is applied to find a transformation that maps the coordinates of points in both cameras to the coordinates of those points in the universal frame (i.e., the coordinate frame of the magnetic tracking system).

We take $P'_s = \{(\xi_{s_k}, \eta_{s_k}) | k = 1 \cdots n\}$ and $P'_t = \{(\xi_{t_k}, \eta_{t_k}) | k = 1 \cdots n\}$ and $M = \{(x_k, y_k, z_k) | k = 1 \cdots n\}$ where n is the number of points used for registration. We also define Q and P matrices as follows:

$$Q = \begin{bmatrix} \xi_{t_1} & \cdots & \xi_{t_n} \\ \eta_{t_1} & \cdots & \eta_{t_n} \\ \xi_{s_1} & \cdots & \xi_{s_n} \\ \eta_{s_1} & \cdots & \eta_{s_n} \end{bmatrix}, \; P = \begin{bmatrix} x_1 & \cdots & x_n \\ y_1 & \cdots & y_n \\ z_1 & \cdots & z_n \end{bmatrix}$$

Now we solve the following optimization problem for the 3 × 4 matrix T:

$$\min_{T} ||TQ - P||$$

This optimization problem will have a unique solution for an appropriate number of points (at least 4 points) which are not in the same plane. As our pattern (shown in Figure 2.5) has 8 points located in two non-parallel planes, it should result in a unique optimal solution. We call this solution, T_o. The matrix T_o is then used to reconstruct the 3D coordinates, based on the information from the two images. If a point p is detected in both cameras with the coordinates $p_s = [\xi_s \; \eta_s]^T$ and $p_t = [\xi_t \; \eta_t]^T$, then its 3D coordinates in the universal coordinate frame ($p = [x \; y \; z]^T$) will be:

$$p = T_o \begin{bmatrix} p_t \\ \cdots \\ p_s \end{bmatrix}$$

This registration method is an efficient algorithm developed for this application based on similar methods that can be found in the computer vision literature [33]. In a clinical application, the registration process would be modified by using a non-planar pattern that can be recognized in the X-ray images. Images of this pattern from different angles (by rotating the C-arm), can be used to find the transformation. Krueger et al. [34] have developed a 2D-3D registration scheme by attaching an magnetic tracking system to the C-arm.

The magnetic sensor that we use is the Aurora 5-DOF sensor from Northern Digital Inc. The RMS error for this sensor is in the range of 0.9mm to 1.3mm depending on the location of the sensor coil with respect to the field generator. The error has around the same value as the spatial resolution of the camera (1.0mm) and the calibration algorithm error (1.0028 pixel [31]). The quality of the registration can be improved only if the accuracy of both the magnetic sensor and the camera is improved. For X-ray fluoroscopic images which have higher spatial resolution, we can use a wire grid with pre-known geometry. This wire grid can also be used for calibration [28].

2.3.2 Extracting mask image (Roadmap)

An off-line algorithm has been developed to extract a mask image, which represents the pattern of the tubes in the testbed (representing the pattern of blood vessels in the body). Although it is only shown for the top camera, a mask image is generated for each of the cameras. Using the mask images, the image processing is limited to areas of the image which contain the blood vessels, thereby speeding up the algorithm and making it robust to illumination and contrast variations.

As there is no movement of the testbed during the procedure, the mask image is a still image. However, an online calibration technique can be used to modify the mask image according to changes in the position of the vessels due to involuntary motions of the patient.

To generate the mask, a number of frames are captured and time filtering is applied on these frames (assuming there is no movement in the testbed during capturing of these frames). After time filtering, a single image will be available. Then a spatial Gaussian filter is applied to this image to filter out noise, the result is shown in Figure 2.7(b). A Canny filter [35], [36] is then applied to detect edges in the image.

The resulting image (Figure 2.7(c)), which mainly consists of contours showing the edges in the main image, may have some single points or extra pieces of contours which should be removed. It may also have some discontinuities in the main contours which should be connected. A heuristic algorithm is applied to this image to do these tasks. This algorithm searches for tube-like objects (i.e., two parallel curves) which are connected. It connects the disconnected parts of the curves and removes single points and curves which are not detected to be part of the tubes. The main parameters used for closing discontinuities are l_{max}, l_{min}, d_{max} and d_{min}, all measured in pixels. The parameter l_{max} is the radius of the quarter circle slice centered at the endpoint of a contour and in the same direction as the contour is expected to be continued. The contour connection algorithm searches this area for the continuation of the contour. If no continuation is found at both two ends of a piece of contour, that piece will be omitted. The continuation should have a length of at least l_{min} pixels. The parameters d_{min} and d_{max} are the minimum and maximum distances across the contour that the algorithm searches for the other edge of the tube. If no other contour is found in this area, the contour is considered as a single linear object and omitted. The values of d_{min} and d_{max} are set by the user based on the diameter of the tubes. The user also sets an initial point on the detected contours which determines the starting point for the contour connection algorithm. The user can also connect contours or remove parts of the image by clicking on the image. This is required when the algorithm fails to connect the contours or remove the extra parts. The resulting image consists of connected contours which show the borders of the tubes/vessels. The algorithm works well on the existing testbed as shown in Figure 2.7(d), but it may need to be slightly modified to fit the clinical application with X-ray images.

Then a flood-filling algorithm is applied to this image which fills the areas between the contours covering the tubes/vessels in the image (Figure 2.7(e)). Since there could be some small motions in the tubes during catheter insertion (due to the flexibility of the tubes and the force applied by the catheter on the walls of the tubes), we apply a "blow-up" algorithm to make the flood-filled areas larger. This increases the search area to maintain tracking in case the tubes deform slightly. Mask blow-up is done using a heuristic algorithm which is basically an iterative sequence of spatial Gaussian filters and block-based thresholding that acts in the same way as a dilation operation around the centerline of the tubes (vessels). The spatial Gaussian filtering of the black and white image (Figure 2.7(e)) creates a gray scale image with shades around the tubes. Using a block-based thresholding algorithm creates a new black and white image with blown-up tubes. This process is repeated until we get the final mask image as shown in Figure 2.7(f). The amount of dilation can be adjusted by four parameters: the standard deviation of the Gaussian filter, the block size and the threshold level of the block-based thresholding, and the number of iterations.

The resulting image is called the *Mask Image*. The images obtained at each time instant are masked by the mask image (using the logical operation AND). The resulting image includes the areas around the tubes/vessels neglecting the parts outside these areas. This increases the processing speed as well as makes the algorithm more robust. The robustness results from the fact that the algorithm ignores any movement or illumination variations outside of the tubes.

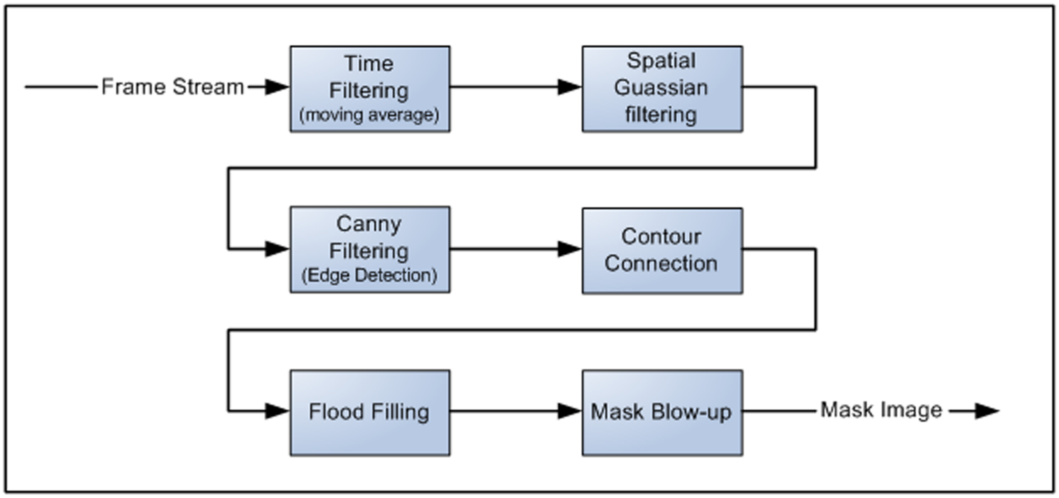
Frame Stream
Time Filtering (moving average)
Spatial Guassian filtering
Canny Filtering (Edge Detection)
Contour Connection
Flood Filling
Mask Blow-up
Mask Image

Figure 2.6: Mask image generation block diagram

A block diagram of the mask generation algorithm is shown in Figure 2.6. The results of the mask generation algorithm applied to our testbed are shown in Figure 2.7 which shows the step-by-step result of the algorithm.

The mask image acts as the *roadmap* in the clinical case. There are several issues that need to be considered for extending the algorithms for clinical applications.

- The assumption of a still roadmap is not valid for the clinical case as cardiac contraction (heart beat) and respiration cause movement and deformation of the blood vessels in the thorax. As an example the right coronary artery (RCA) moves up to 3.0mm due to cardiac motion and the RCA root moves up to 10.5mm due to respiratory motion [37]. Timinger et al. [38], [39] have developed an elastic motion compensation algorithm which uses ECG data as well as the data from a magnetic tracking sensor to compensate for the motion of the roadmap due to cardiac contraction and respiratory motion.
- In the clinical case, a contrast agent is injected into the vascular system to make the blood vessels visible under X-ray images. An important factor for detecting the tubular structures is the CNR (contrast to noise ratio) of the vessels with respect to the background. The contrast media will help to improve the CNR to a relatively higher level than what it is in our testbed. This will help in segmentation of the vascular tree [40].

2.3.3 Database creation and selection

A pattern recognition algorithm based on the images from both cameras is used to determine the junctions, and the transformation obtained from the registration algorithm is used to find the 3D positions and orientations with reference to the universal coordinate frame. To detect the bifurcations, we have used block search and correlation analysis with a set of 3×3 block patterns representing possible Y-shaped bifurcations and the $\frac{\pi}{4}$ rotations of these patterns [41]. The block size is adjusted according to the width of the tubes in the mask image and the latter is searched to find the matches. Once

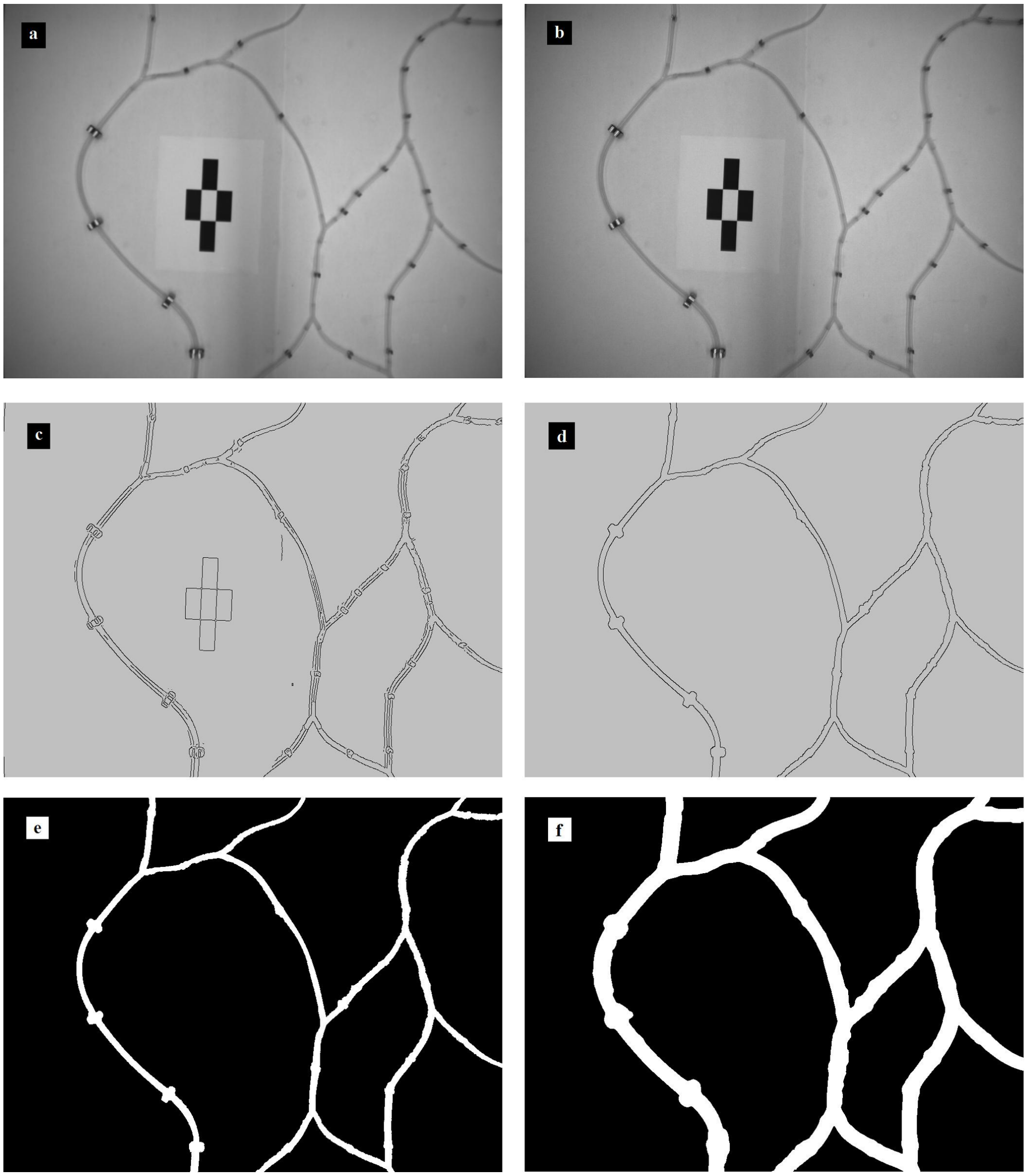

Figure 2.7: Mask image generation: step by step - (a) Sample original noisy image, (b) filtered image after temporal (moving average) and spatial (Gaussian) filtering, (c) Edges detected by Canny filtering, (d) Connected contours with extra pieces removed, (e) Flood-filled inside the tube borders, (f) Final mask image (blown-up tubes). (Note that the colors are changed in (c), (d) to increase visibility)

a match is found, the algorithm tries fine tuning to find the orientation of the bifurcation with higher accuracy. This algorithm is applied to the mask images of both of the cameras. The bifurcations found in the two images are matched using the transformation found by registration and the user is asked to confirm each bifurcation. Zifan et al. have used a similar approach to detect the bifurcations in a vascular tree using X-ray angiogram images [42].

The starting point, the location of bifurcation points, and the 3D orientation of branches are stored in a database. The database is loaded each time the program starts. The user chooses the desired path and sets the target point on the image, as shown in Figure 2.8.

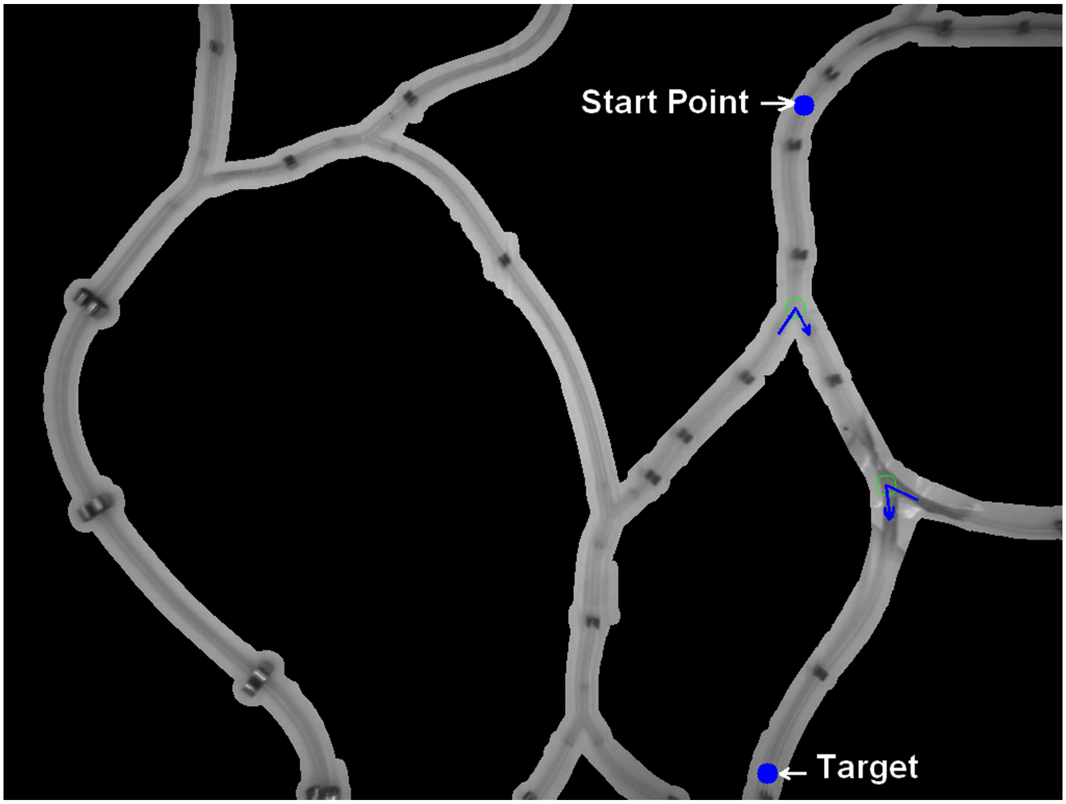

Figure 2.8: Desired path selected by the user

2.3.4 Tracking the tip of the catheter

The images are processed using a novel real-time algorithm to track the tip of the catheter [43]. The block diagram for real-time tracking is shown in Figure 2.9. Some of the parameters of this algorithm are set by the autonomous guidance algorithm. For example, the search area is different for normal and slow insertion, as determined by the autonomous guidance algorithm. The algorithm mainly consists of the following steps:

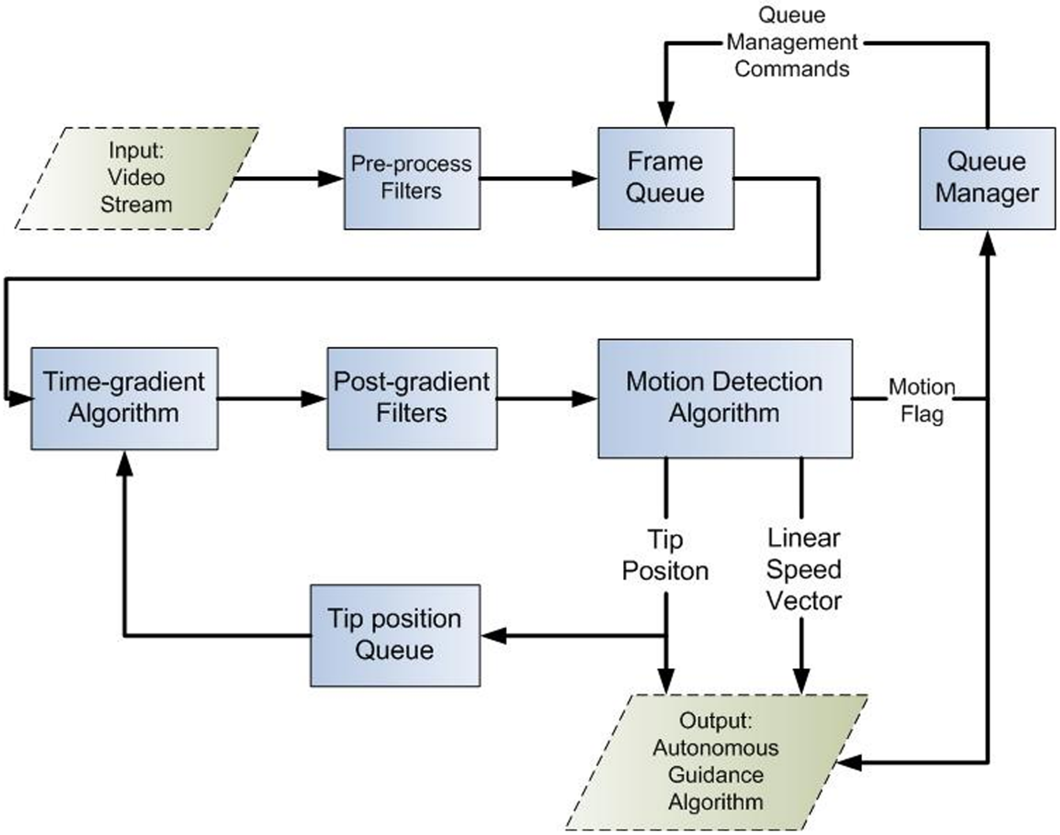

Figure 2.9: Block diagram of the tracking algorithm

- The video stream consists of a stream of frames grabbed by the top camera at 30fps.

- The pre-processing filter block consists of a set of filters for suppressing noise, masking the image and improving the contrast of the image.
- The frame queue is an image buffer structure managed by the queue manager block. This queue provides the appropriate input for the time gradient algorithm.
- The time gradient algorithm takes the time gradient of the images in the frame queue. It concentrates on an area around the previously known catheter tip position to limit the search area and decrease the processing time.
- The post-gradient filters suppress the noise in the gradient image and make it useful for the motion detection algorithm.
- The motion detection algorithm searches for the catheter tip in a neighborhood around the previously known position of the catheter tip. The neighborhood is a rectangle which extends more to the previously known direction of motion and its size is determined by the autonomous guidance algorithm.
- When the new position of the catheter tip is detected, the algorithm calculates the speed vector of the catheter by subtracting the previous catheter tip position from the new detected tip position. The calculated speed vector is then filtered using a moving average filter.
- The algorithm also generates a true/false flag called motion flag by thresholding the linear speed (norm of the speed vector). This flag is true when a motion is detected.

The signals generated by the motion detection algorithm are fed back to the autonomous guidance algorithm. The image processing algorithm is not synchronous and has a varying frame-rate from 14fps (as the worst case) to 30fps. The algorithm was tested on an Intel P4 2.4GHz CPU, with 1GB of RAM running under Windows XP. The image processing algorithm can be further improved by optimizing the algorithm and implementing it on a platform with higher processing power and a hard real-time operating system.

2.3.5 Direction Selection

When the autonomous guidance algorithm detects the tip of the catheter at a bifurcation point, the direction selection algorithm is called to choose the direction according to the path selected by the user. The algorithm determines the desired angles of bending (in the horizontal and vertical planes) for the active catheter, depending on the bifurcation.

2.4 Sensor Fusion

To perform autonomous catheter insertion, closed-loop position control of the distal tip of the catheter is required during insertion. Therefore accurate real-time position feedback is needed for this purpose. We have developed a real-time image-processing algorithm for catheter tip position tracking as described in Section 2.3. The performance of the image-based tracking algorithm has been shown to be acceptable, however it has a major drawback: It is sensitive to X-ray image artifacts caused by bones and dense tissues. A magnetic tracking system (MTS) is another modality that has also been used for catheter tip position tracking [44]. The performance of MTS is good within certain limits; however it has two major deficiencies. First, the existing magnetic tracking systems have a limited range which cannot cover the whole catheter insertion path for angioplasty - from the groin to the thorax, and second they are sensitive to external interferences produced by ferrous metals and oscillating magnetic fields. Combining the measurement data provided by both imaging and magnetic sensors can compensate for the deficiencies of each and can also improve the robustness of catheter tip position tracking.

Multisensor data fusion has a wide range of applications in different areas. Sensory information from different sources can be applied to extract the maximum amount of information possible about the sensed object(s) (e.g. catheter) under various conditions (e.g. in case of an occlusion), in real time. This will result in higher performance and reliability in the presence of uncertain sensory data. It may also help in achieving higher data rates as a result of sampling at different time instants. The purpose of data fusion is to produce a model or representation of a system from a set of independent data sources, from which a single view or perception of the system is detected; Normally, a state-space model of the process is used for this purpose. The combination of information from different sensors and the subsequent state estimation should result in reduced uncertainty. The data fusion algorithm should be able to handle multiple observations and multiple sensors, and at the same time it should be able to consider faults in sensory data. A common application of data fusion techniques is the estimation of target position/velocity information from multiple measurements from a single or multiple sensors. Two essential processes are involved in the derivation of position/velocity: (a) data association and (b) state estimation. Data association is done to distinguish between the true information and clutter. State estimation refers to optimal estimation of the variables, e.g., the position and velocity of the target, using the observation data after the data association process has been performed [45].

Kalman filtering is the best known and most widely applied state estimator algorithm [46]. The Kalman filter gives a linear, unbiased, and minimum error variance recursive algorithm to optimally estimate the unknown state of a linear dynamic system from Gaussian distributed noisy observations. The Kalman filtering process can be considered as a predict-estimate-update formulation. The algorithm uses a predefined (linear) model of the system to predict the state and its covariance at the next time step. Then the measurement data is used to estimate the optimal Kalman gain and this gain is then used to update the estimate for the system state and its covariance. The Kalman gain is calculated to minimize the mean-square error of the state estimate.

Kalman filtering has found widespread application in data fusion problems [45]. Kalman filtering based data fusion models can be classified into two categories [47]: (a) measurement level fusion and (b) state level fusion.

Measurement level fusion methods are further divided into two subcategories: (1) augmented observation and (2) minimum mean-squared measurement estimate fusion. State level fusion techniques are basically different variations of Bar-Shalom's track to track fusion method [47], [48].

To combine the data from the two different sources, the coordinates of the MTS and the imaging system need to be registered. An off-line registration algorithm has been developed that finds the transformation between the coordinates of the two sensors. To compensate for the nonlinearities of the imaging system (e.g. radial distortion), a camera calibration method has been used that finds the intrinsic parameters of the imaging system. A Kalman filter has been developed which uses the measurements of both sensors to provide accurate and reliable position feedback. The results show robust and accurate tracking regardless of occlusions in the images or interferences in the magnetic field. It also provides a higher sampling rate by interpolating the estimated data. This method provides accurate and reliable position feedback that can be used by a robotic system to control the catheter tip position in real-time.

2.4.1 Magnetic tracking system

We have used a 5-DOF magnetic tracking sensor, implemented at the tip of the catheter to detect the tip position. More details on the magnetic tracking system is included in Appendix H. The RMS error for this sensor is in the range of $0.9mm$ to $1.3mm$ depending on the location of the sensor coil with respect to the field generator. This sensor is able to measure 3-DOF position and 2-DOF orientation of the catheter tip.

Offset calibration

The magnetic sensor is usually not located at the tip of the catheter but at a fixed distance from the tip. An off-line calibration needs to be performed to measure this offset. The value obtained for the offset is then used to calculate the actual position of the catheter tip in real-time. Zhang et al. [49] have designed a simple pivot calibration algorithm. We have used this pivot calibration algorithm with three different points in the workspace and obtained 1000 samples for each point, by fixing the tip of the catheter at that point and changing the orientation. The results show an offset of $17.7323mm$ with an RMS error of $0.9973mm$.

2.4.2 Registration of the two sensors

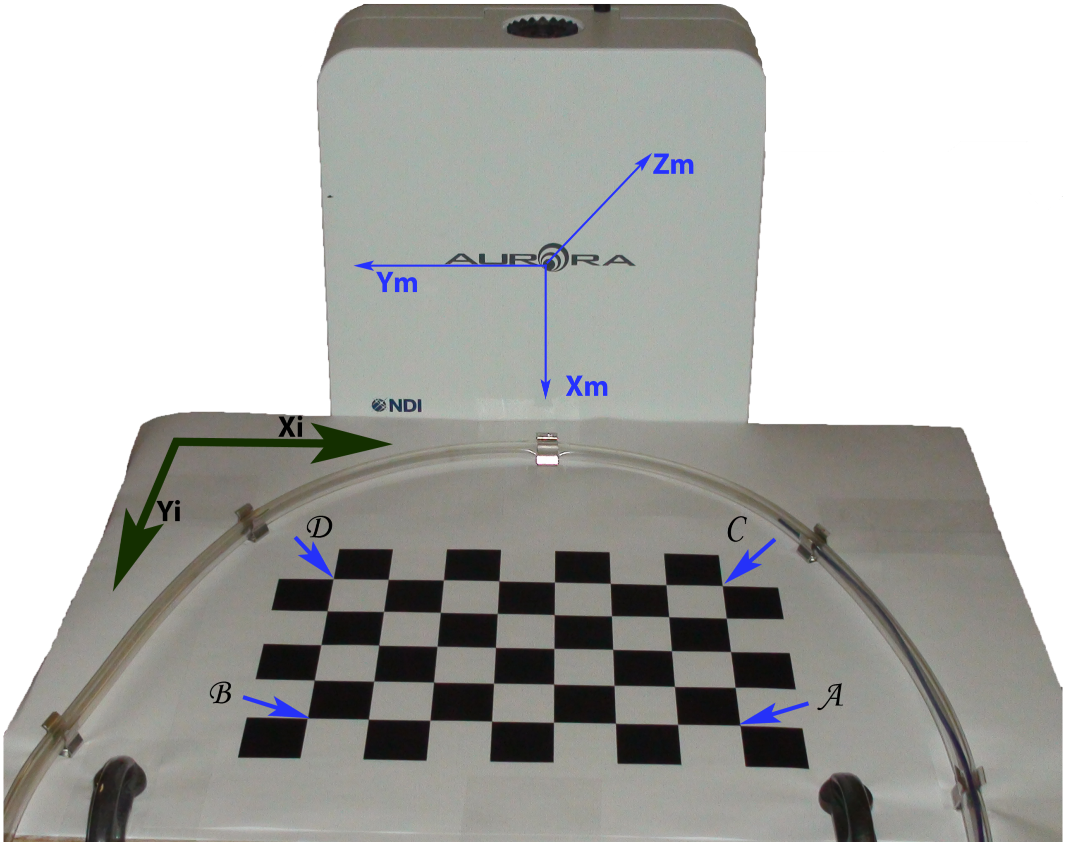

Figure 2.10: Registration pattern; The coordinate system of the camera is shown by X_i, Y_i and the coordinate system of the magnetic tracking system is shown by X_m, Y_m, Z_m. The four corner points are marked as $\mathcal{A},\mathcal{B},\mathcal{C}$ and $\mathcal{D}$

A planar chessboard pattern with known size, as shown in Figure 2.10, is used to register the image tracking and the magnetic tracking coordinate systems. The registration process is as follows:

- The coordinates of the four outer corners of the non-planar pattern (as marked by $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and $\mathcal{D}$ in Figure 2.10) are obtained by means of the magnetic sensor and are logged into the computer. Three of these points are enough to obtain the geometry of the pattern and the fourth point is only used to reduce the errors.

- The registration algorithm calculates the 3D coordinates of the inner corners of the pattern using the coordinates of outer corners and the pre-known geometry of the pattern. The set of the coordinates obtained from this process is denoted by M.

- An image of the pattern (I) is obtained from the camera and undistorted using the distortion parameters obtained in the off-line camera calibration procedure. We call the undistorted image as I'.

- A pattern recognition algorithm is used to find the coordinates of all the corner points in the undistorted image I'. We denote the set of corner points obtained from I' as P'.

- The obtained set of points P' is rearranged corresponding to the order of the points in M; The rearranged matrix is called P.

- A least-squares method is applied to find a 2D-3D transformation that maps the 2D coordinates of points in the image plane to the 3D coordinates of the points in the magnetic tracking system.

We take $P = \{(\xi_k, \eta_k) | k = 1 \cdots n\}$ and $M = \{(x_k, y_k, z_k) | k = 1 \cdots n\}$ where n is the number of points used for registration. We also define matrices Q and R as follows:

$$Q = \begin{bmatrix} \xi_1 & \cdots & \xi_n \\ \eta_1 & \cdots & \eta_n \end{bmatrix}, \; R = \begin{bmatrix} x_1 & \cdots & x_n \\ y_1 & \cdots & y_n \\ z_1 & \cdots & z_n \\ 1 & \cdots & 1 \end{bmatrix}$$

Now we solve the following optimization problem for the 2 × 4 matrix T, $\min_T ||TR - Q||$. This optimization problem will have a unique solution for an appropriate number of points (at least four points). As our pattern (shown in Figure 2.10) has 5 × 8 points located on the same plane, it should result in a unique optimal solution. We call this solution, T_o. The matrix T_o is then used for registration of the undistorted image plane coordinates to the coordinates of the magnetic tracking system. A point $p = [x\ y\ z]^T$ detected by the magnetic tracking system, will be corresponding to the point $q = [\xi\ \eta]^T$ in the undistorted image plane, where $q = T_o \left[p^T\ 1\right]^T$. This registration method is similar to what we used to register the top and side cameras in Section 2.3. We have measured the RMS registration error and it has an average value of $1.1269mm$. For X-ray fluoroscopic images, we can use a wire grid with pre-known geometry.

2.4.3 Modeling the catheter tip motion

The catheter is usually moved slowly and without sudden velocity changes to avoid rupturing the blood vessels; Therefore we can ignore the acceleration and develop the motion model at velocity level. The catheter tip motion model for real-time tracking can be described as a state-space equation [46]:

$$\begin{cases} P_t(k+1) = P_t(k) + V_t(k) + \omega_P(k) \\ V_t(k+1) = V_t(k) + \omega_V(k) \end{cases} \tag{2.2}$$

and the augmented output equations for our application can be modeled as:

$$\begin{cases} P_I(k) = P_t(k) + \nu_I(k) \\ P_M(k) = P_t(k) + \nu_M(k) \end{cases} \tag{2.3}$$

where $P_t(k)$ and $V_t(k)$ are the states of the system at time instant k, $P_t(k) = [x_t(k)\ y_t(k)]^T$ is the position of the distal tip of the catheter in the two dimensional undistorted image plane and $V_t(k) = [v_{x_t}(k)\ v_{y_t}(k)]^T$ is the 2D displacement vector, at the time instant k; $\omega_P(.)$, $\omega_V(.)$ and $\nu_I(.)$, $\nu_M(.)$ are independent zero mean white Gaussian two dimensional random processes which characterize the process noise and measurement noise respectively.

We also need to calculate the actual length of the catheter inserted into the vascular model. To calculate this variable, we integrate the catheter tip displacements along the centerline of the tube/vessel over the time samples. The depth of the catheter inserted into the vasculature is then used in catheter tip position control by the robot client. $V_t(k)$ is the displacement vector for the catheter tip at the time instant k, but it may have a deflection around the centerline of the vessel, specially when it is near the curvatures. This is shown in Figure 2.11. The centerline is extracted from the mask image by dilation of mask image and fitting polynomials to each segment.

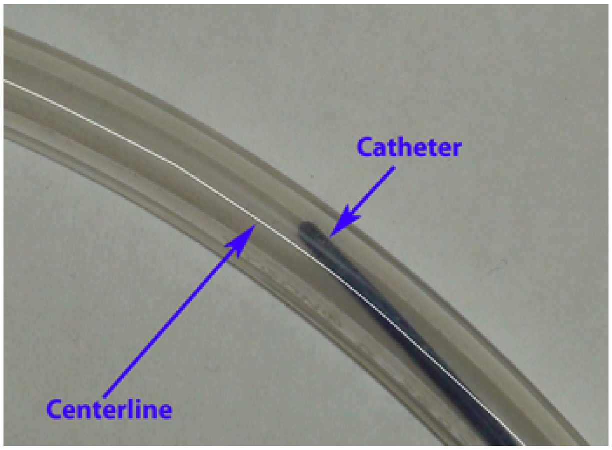

Figure 2.11: The catheter may have a deflection w.r.t the centerline of the tubes because of the curvatures.

To obtain the amount of displacement along the centerline of the tubes/vessels, we use the projection of the displacement vector $V_t(k)$ on the unit vector along the centerline at that point. The direction of the unit vector is chosen to point forward to the direction of insertion, this helps to distinguish whether the catheter is being pushed in or being pulled out. The unit vector $D(.)$ is a function of the current position of the catheter tip, $P_t(k)$, and depends on the a priori knowledge of the geometry of the vasculature.

$$s(k+1) = s(k) + < V_t(k), D(P_t(k)) > \tag{2.4}$$

where $< . , . >$ represents the inner product of two vectors. The variable $s(.)$ is used by the robot client to control the catheter tip position.

2.4.4 Data fusion algorithm design

We have used Kalman-filter based data fusion at the measurement level which provides optimal tracking compared to the sub-optimal tracking performance of Bar-Shalom's method [47]. The fusion is performed at the observation level and as we have similar sensors (i.e., the sensors measure the same variables), the performance will be equivalent to the performance provided by the minimum mean-squared measurement estimate fusion [47]. The overall structure of the data fusion scheme is shown in Figure 2.12 and the structure of the Kalman-filter is shown in Figure 2.13.

The augmented state-space equations for the system can be described as:

$$\begin{cases} X_k = \mathcal{F} X_{k-1} + \omega_{k-1} \\ Y_k = \mathcal{C} X_k + \nu_k \end{cases} \tag{2.5}$$

where $X_k = \left[P_t^T(k),\ V_t^T(k)\right]^T$ is the system state and $Y_k = \left[P_I^T(k),\ P_M^T(k)\right]^T$ is the augmented system output at time instant k. Using Equations (2.2) and (2.3), we obtain:

$$\mathcal{F} = \begin{bmatrix} I_{2\times2} & I_{2\times2} \\ 0_{2\times2} & I_{2\times2} \end{bmatrix},\ \mathcal{C} = \begin{bmatrix} I_{2\times2} & 0_{2\times2} \\ I_{2\times2} & 0_{2\times2} \end{bmatrix} \tag{2.6}$$

where $I_{2\times2}$ is the 2×2 identity matrix. The process noise is $\omega(k) = \left[\omega_P^T(k),\ \omega_V^T(k)\right]^T$, the measurement noise is $\nu(k) = \left[\nu_P^T(k),\ \nu_V^T(k)\right]^T$ and the process and measurement noise covariances are Q_k and R_k respectively.

The Kalman filter process performs data fusion using the augmented measurements as shown in Figure 2.13. The three steps of Kalman filtering, including prediction, estimation and updating are formulated as follows [46]:

- Predict the state and its covariance:
  - $\hat{X}_{k|k-1} = \mathcal{F}\hat{X}_{k-1|k-1}$

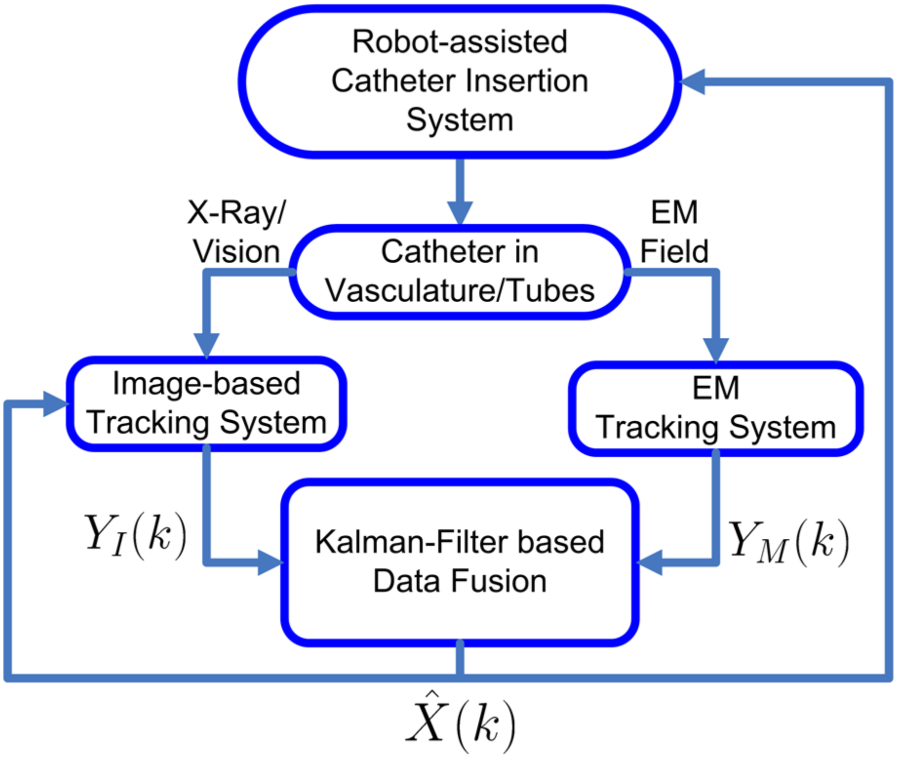
Robot-assisted Catheter Insertion System
X-Ray/ Vision
Catheter in Vasculature/Tubes
EM Field
Image-based Tracking System
EM Tracking System
$Y_I(k)$
Kalman-Filter based Data Fusion
$Y_M(k)$
$\hat{X}(k)$

Figure 2.12: Block diagram of the whole system: the internal structure of the Kalman-based data fusion block is shown in Figure 2.13

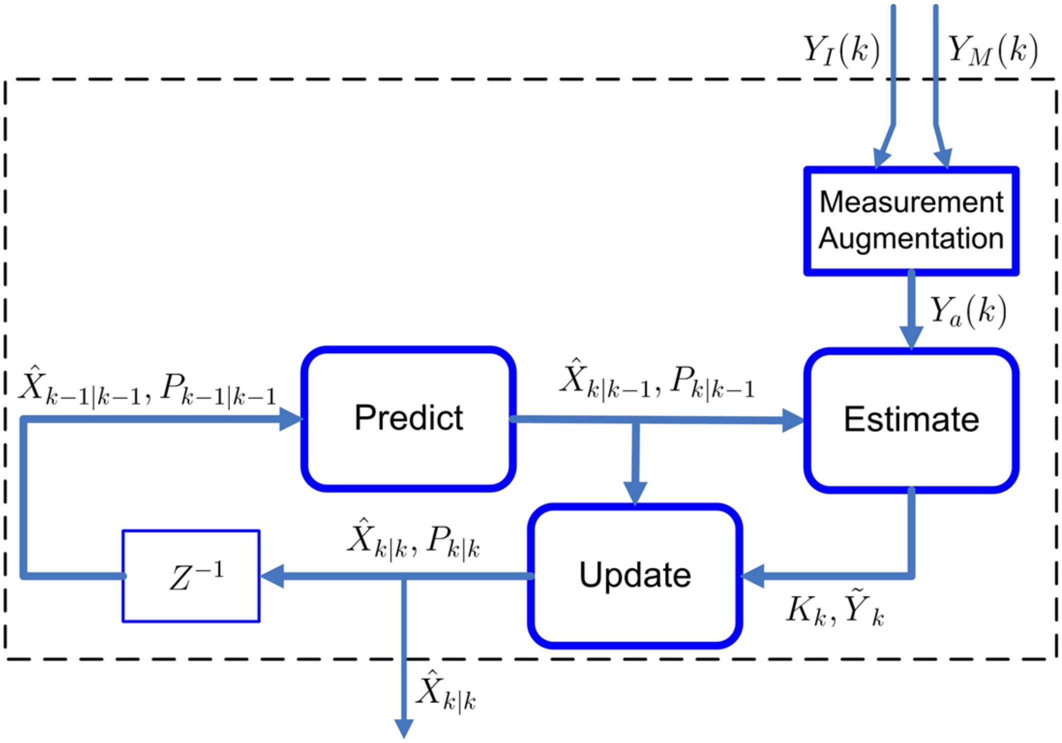

Figure 2.13: The Kalman-filter based data fusion block diagram

  - $P_{k|k-1} = \mathcal{F}P_{k-1|k-1}\mathcal{F}^T + \mathcal{Q}_{k-1}$

- Estimate the innovation, its covariance and the optimal Kalman gain, based on the prediction:
  - $\tilde{Y}_k = Y_a(k) - \mathcal{C}\hat{X}_{k|k-1}$
  - $S_k = \mathcal{C}P_{k|k-1}\mathcal{C}^T + \mathcal{R}_k$
  - $K_k = P_{k|k-1}\mathcal{C}^T S_k^{-1}$
- Update the state and its covariance:
  - $\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k\tilde{Y}_k$
  - $P_{k|k} = (I - K_k\mathcal{C})P_{k|k-1}$

where $P_{k|k} = \text{cov}(X_k - \hat{X}_{k|k})$, $P_{k|k-1} = \text{cov}(X_k - \hat{X}_{k|k-1})$ and $S_k = \text{cov}(\tilde{Y}_k)$. Results are included in Section 2.7.

2.5 Autonomous Guidance

A finite-state machine (FSM) structure is designed and implemented to control the insertion process. The structure is shown in Figure 2.14.

2.5.1 Path following

The algorithm chooses a path connecting the starting point to the target point as chosen by the user, based on the information stored in the database. The algorithm can also allow the clinician to choose the desired path for catheter insertion. A simple graphical user interface (*GUI*) has been developed to interact with the user/clinician. The user selects the starting point and the target point by clicking on the image and verifying his/her selection. The user can click on the bifurcations/joints on the path between the starting point and the target point to specify the selected path using the interactive GUI. The selected path can be saved in a profile in the database and the profile can be loaded using the assigned profile number in case of repeated

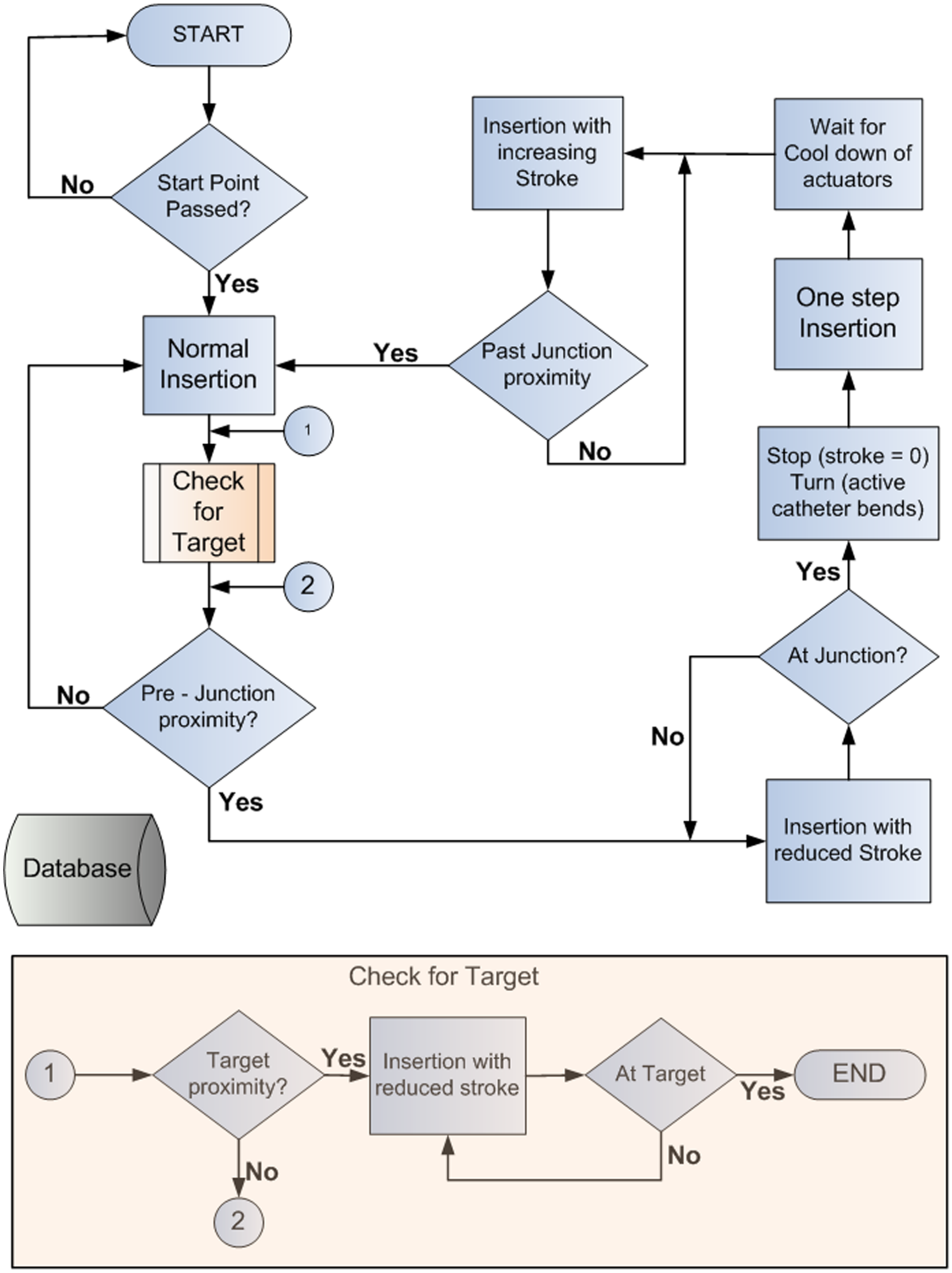

Figure 2.14: Autonomous Guidance Finite State Machine

insertions. Direction selection is done based on the selected path. At each bifurcation/junction, the direction is set towards the next bifurcation/junction in the path using the database information.

2.5.2 Catheter tip position control

The tip position of the catheter is controlled in the proximity of the branches and near the target to provide smooth movement and prevent any damage to the epithelial cells of the blood vessel or dislodging of plaque. Near a branch, the catheter insertion slows down gradually until the tip of the catheter reaches the bifurcation. At the junction, the robot stops insertion of the catheter and the active catheter bends in the desired direction. When the bending is complete, the robot inserts the catheter using a single stroke to insert the catheter into the desired path. After the actuators have regained their original shape, the robot resumes the insertion process by increasing the stroke length gradually until it reaches normal speed. This process is repeated at each junction in the path. Near the target, the robot slows down the insertion to gradually reach the target. The tip position follows a smooth trajectory to prevent any damage to the inner lining of the arteries and to avoid dislodging plaque.

The tip position control also takes care of the difference between the actual stroke length and the tip position movement via image-based visual servoing. It is observed that the catheter tip does not necessarily follow the motion of the robot's gripper because of the significant amount of flexibility in the catheter and frictional forces acting along the length of the catheter. The catheter may flex inside the tubes and not advance as the robot pushes the catheter. This is more predominant close to junctions and the target point when the stroke length of insertion is small. In such cases the controller increases the stroke length to compensate for this effect.

We have implemented a PI controller with saturation to control the tip position of the catheter while approaching the branches and the target since we need a smooth motion without any overshoot. The integral term in the controller provides zero steady-state error and compensates for the flexing of the catheter due to frictional forces between the catheter and the walls of the tubes (blood vessels). The controller gains are set to provide a smooth motion. A block diagram of the controller is shown in Figure 2.15. We have used a high gain proportional controller for the speed-up phase beyond the branch.

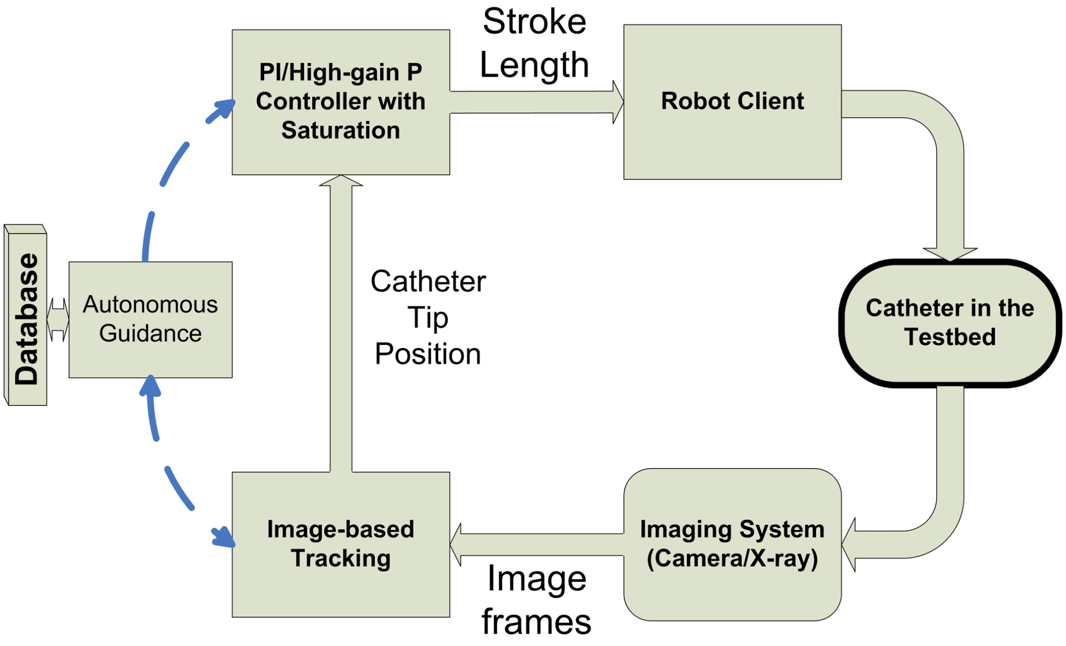

Figure 2.15: Block diagram for the catheter tip position control; a PI controller is applied before the branching and before the target during slow-down, while the high-gain proportional controller is applied after passing a branch during speed-up.

The implemented PI controller is a model-free visual servoing controller which is only applied for catheter tip position control and its performance is good since the catheter motion is very slow. Robot manipulator control is done using Augmented Hybrid Impedance Control (AHIC) which is also shown to have good performance. The catheter insertion problem has some similarities with the problem of needle steering, e.g., see [50], [51], [52]. However, the constraints imposed by the complexity of the vasculature, the significantly higher flexibility and the longer length of the catheter make the two problems substantially different. Modeling and model-based control of the catheter inside the vascular system is a challenging problem as the catheter is a constrained under-actuated hyper-redundant dynamic system.

2.6 Master-Slave Control

Although autonomous catheter insertion provides high accuracy for catheter tip position control and significantly reduces X-ray exposure to clinicians, the latter may need more control over the procedure. A master-slave control scheme was developed in order to involve a cardiologist directly in the control loop via a haptic device without being exposed to harmful X-ray radiations. Visual servoing is used as an internal control loop in order to make sure that the catheter distal tip position follows the user's hand motion on the haptic device.

It should be noted that the dynamics of the slave, i.e, the catheter is highly complex, and depends on factors such as the material of the catheter, geometry of the path, static and dynamic friction coefficients, force of insertion, diameter of the vessel and catheter, initial orientation of insertion, fluid flow, etc. In addition, the catheter is indirectly manipulated using a robot at the remote site of insertion. Due to the high flexing of the catheter inside the arteries, the robot end-effector velocity is not necessarily equal to the velocity of the tip of the catheter. The desired value for the robot end-effector velocity is generated based on the catheter tip position and velocity measured by the image-based tracking algorithm, as given by the following equation:

$$\dot{\mathbf{x}}_{rd} = K_p(\mathbf{x}_{sd} - \mathbf{x}_s) + K_d(\dot{\mathbf{x}}_{sd} - \dot{\mathbf{x}}_s) + K_i \int (\mathbf{x}_{sd} - \mathbf{x}_s)dt \tag{2.7}$$

where $\dot{\mathbf{x}}_{rd}$ is the desired robot end-effector velocity, $\mathbf{x}_{sd}$ and $\dot{\mathbf{x}}_{sd}$ are the desired catheter tip position and velocity provided by the master-slave teleoperation and $\mathbf{x}_s$ and $\dot{\mathbf{x}}_s$ are the actual catheter tip position and velocity measured by the image-processing algorithm or provided by the data-fusion algorithm. The parameters K_p, K_i and K_d are the proportional, integral and derivative controller gains. The desired velocity, $\dot{\mathbf{x}}_{rd}$ is provided to the robot end-effector controller as described in [4]. The desired catheter tip position and velocity, $\mathbf{x}_{sd}$ and $\dot{\mathbf{x}}_{sd}$ can be directly set by the master position and velocity ($\mathbf{x}_m$ and $\dot{\mathbf{x}}_m$) or can be applied through a teleoperation scheme. We have studied the application of wave-variables for this purpose [24].

2.7 Experimental Results

In this section, we have included some of the experimental results for autonomous and master-slave catheter insertion, sensor fusion and also the nonlinear behavior of the catheter inside the vascular model.

2.7.1 Autonomous catheter insertion

A desired path on the testbed is chosen by the user, as shown in Figure 2.8. The path consists of two active branches - at the first branch the catheter bends to the left (-24.7^0 in the horizontal plane and -3.85^0 in the vertical plane) and at the second, the catheter bends right and down (20.46^0 in the horizontal plane and 22.17^0 in the vertical plane). The robot inserts the catheter into the testbed while controlling the force of insertion. The image processing algorithm tracks the distal end of the catheter, as it is inserted into the testbed. A proportional-integral controller (PI-controller during slow-down and a high gain P-controller during speed-up) with saturation controls the position of the tip of the catheter near the branches and the target. Figure 2.16, shows twelve images as the image processing algorithm tracks the end of the catheter. The results for the entire experiment are shown in Figure 2.16. As the catheter enters a region in the proximity of the junction, the stroke of insertion is reduced gradually, as shown in Figure 2.16 - region b (hereafter denoted as Figure 2.16:b). At the junction (shown in Figure 2.16:c,h), the stroke is zero and a command is given to the catheter to bend in the appropriate direction. The one stroke insertion after the catheter is bent ensures that the catheter enters the desired branch, as shown in Figure 2.16:d,i. The robot waits until the actuators have cooled down and have regained the original shape before resuming further insertion of the catheter. After the actuators have cooled down, the stroke length is increased gradually until it reaches the normal stroke-length, as shown in the speedup phase in Figure 2.16:e,f and Figure 2.16:j,k. This is controlled using the speedup proportional controller. Regions (l) and (m) of Figure 2.16 show the slowing down of the insertion near the target. It can be observed that in region (l) the stroke length is reduced as the catheter approaches the target point. However, due to flexing in the catheter and frictional forces, the end of the catheter does not advance. The integral term, in order to minimize the steady state error, increases the stroke length, as can be seen in region (m) in Figure 2.16(ii). This controls the position of the catheter and advances the catheter towards the target point.

The position of the end-effector is shown in Figure 2.16(iii). The stroke length is gradually varied as the catheter reaches and leaves a junction to enable the catheter to enter the appropriate branch. The average force of insertion increases as the catheter is inserted into the tube mounted in the testbed due to an increase in frictional forces, as shown in Figure 2.16(iv). The actual force measured in the direction of insertion shown in Figure 2.16(v) indicates peak insertion force of 0.13N. The peak insertion force is less than the maximum force of insertion of 0.63N [26], which is the amount of force that could damage the epithelial cells of the artery.

2.7.2 Nonlinear Behavior of a Catheter

As mentioned before, a catheter shows nonlinear behavior inside the vascular model. A number of experiments were performed in order to study these nonlinearities.

Catheter Insertion at Constant Desired Velocity

In this set of experiments, the flexing of the catheter was studied while it is being inserted into the testbed. The stroke length of the robots end-effector was kept constant throughout the experiment and a constant desired velocity

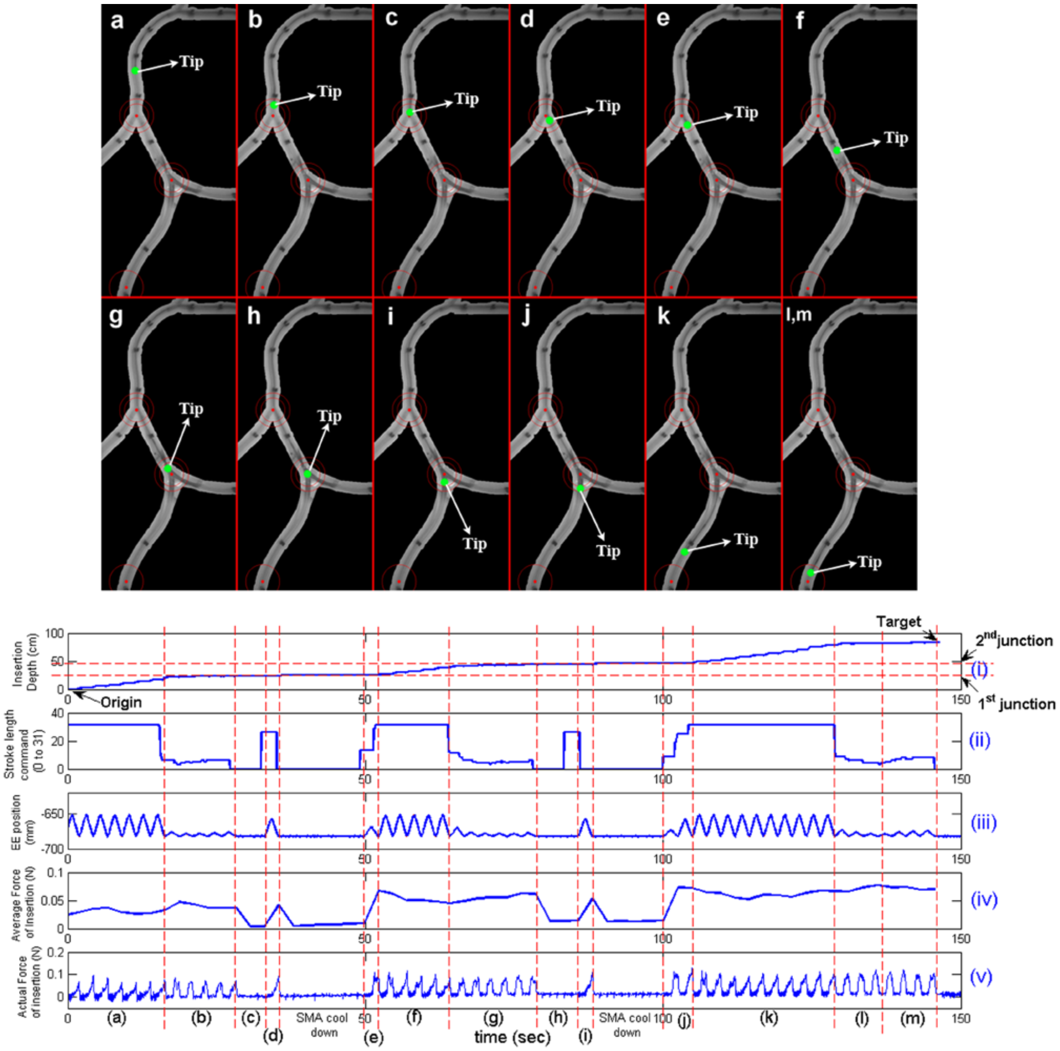

Figure 2.16: (i) shows the insertion depth (cm) vs. time (sec), (ii) shows the stroke length set-point (digitized from 0 (min.) to 31 (max.)) vs. time, (iii) shows the desired and the actual trajectory of the end effector of the robot (mm) vs. time, (iv) shows the average force of insertion over one stroke (N) vs. time and (v) shows the actual force of insertion (N) vs. time. The tip of the catheter has been shown in the images above. Different phases of insertion are shown as: (a) normal insertion before the first branch, (b) slow-down before the first branch, (c) no insertion, catheter bends at junction (d) one stroke insertion, (e) speed-up insertion after the first branch, (f) normal insertion before the second branch, (g) slow-down before the second branch, (h) no insertion, catheter bends at junction (i) one stroke insertion, (j) speed-up insertion after the first branch, (k) normal insertion before the target, (l) slow-down before the target (m) increase in stroke length due to flexing of catheter.

of $7mm/sec$ was applied. The experiments were repeated 10 times, keeping the parameters constant, to obtain reliable predictions by looking at the ensemble averages of the results. Figure 2.17(a) shows the ensemble averaged maximum tracking error between the master and slave as a percentage of the master reference. The difference between the maximum robot end-effector and the catheter tip movements as a percentage of the slave movement is shown in Figure 2.17(b). As can be seen in Figure 2.17(a), the tracking error varies from 4 to 13% of the reference which proves that the visual servoing controller provides good tracking despite the nonlinearities of the catheter. The difference between the maximum robot end-effector and the catheter tip movements as shown in Figure 2(b) varies between 57 and 126%. This difference is mainly caused by the flexing of the catheter and by frictional forces acting along the catheter.

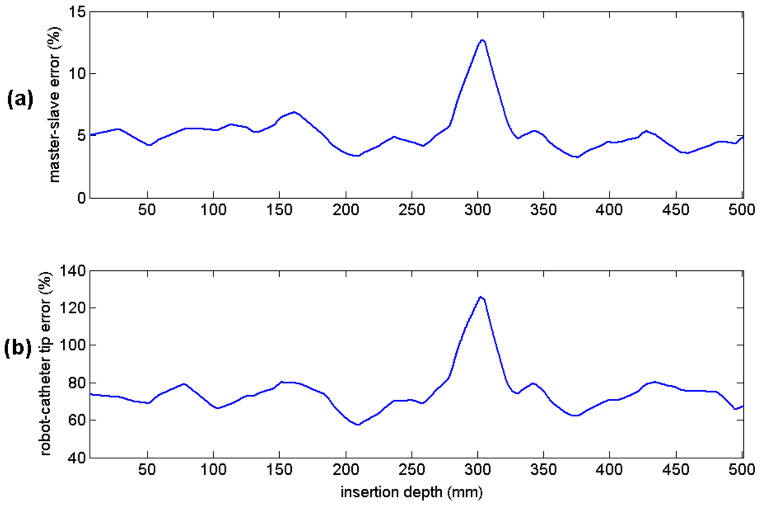

Figure 2.17: (a) Ensemble averaged maximum master-slave tracking error (percentage of the peak master reference) vs. insertion depth (b) Ensemble averaged difference between the maximum robot end-effector and catheter tip movements (percentage of the maximum slave movements) vs. insertion depth

Figure 2.18 shows the ensemble averaged deadband vs. insertion depth. The deadband is the time it takes for the tip of the catheter to start moving after the master starts moving. There are four main factors that affect the deadband:

- vision tracking delay
- network delay (negligible)
- stiction along the contact points between the catheter and the tube
- flexing of the catheter

As the delay caused by vision tracking is constant and is a small percentage of the total deadband (less than 15%) and the network delay is negligible, we can conclude that the main factors causing the deadband delay are stiction and flexing. Stiction depends on the number of contact points between the catheter and the tube, the curvatures of both the catheter and the tube at the points of contact, physical properties of the tube and the catheter and the friction coefficients. Stiction can be modeled as a semi-static effect. The deadband delay varies between 0.66 and 0.87 seconds with a standard deviation between 0.01 and 0.065 seconds over 10 ensembles, proving the reliability of the results.

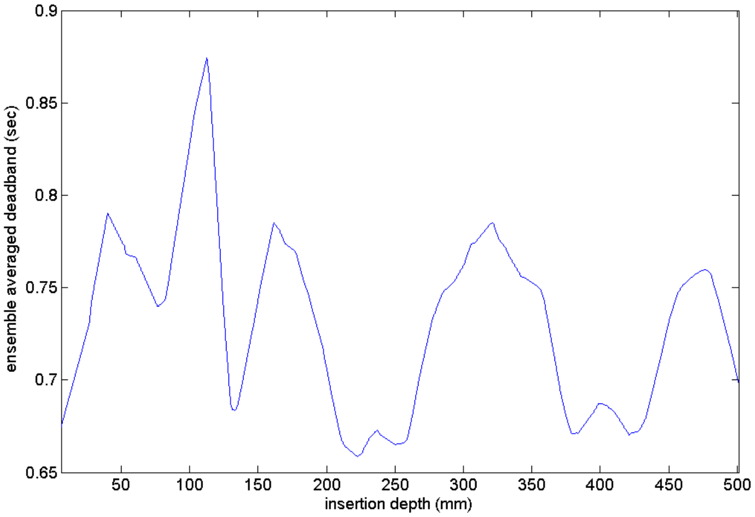

Figure 2.18: Ensemble averaged deadband (s) vs. insertion depth (mm)

In Figure 2.19 the average force of insertion is shown versus the insertion depth. The average force of insertion increases as the catheter advances into the testbed due to an increase in friction acting along the length of the catheter.

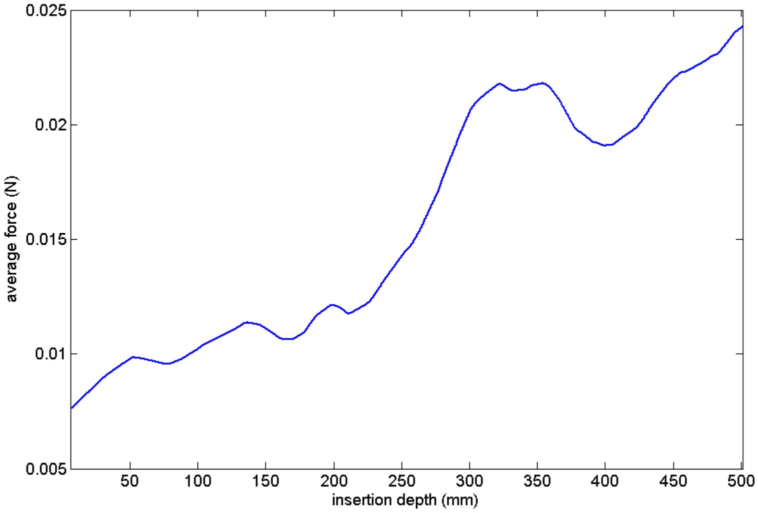

Figure 2.19: Ensemble averaged force of insertion (N) vs. insertion depth (mm)

Catheter Insertion at Variable Desired Velocities

In the next set of experiments, the flexing of the catheter was observed as a function of the master velocity. For each insertion performed, the desired velocity was kept constant. Several such experiments were performed by varying the desired velocity from $3mm/sec$ to $7mm/sec$. The mean deadband delay was measured to be 0.73 seconds with a standard deviation of $38ms$. The average tracking error over the entire insertion is within ±10% of the reference for different desired velocities. As shown in Figure. 2.20, for high insertion speeds, the deadband delay decreases. This is generally expected since the catheter overcomes stiction forces sooner due to higher insertion speeds.

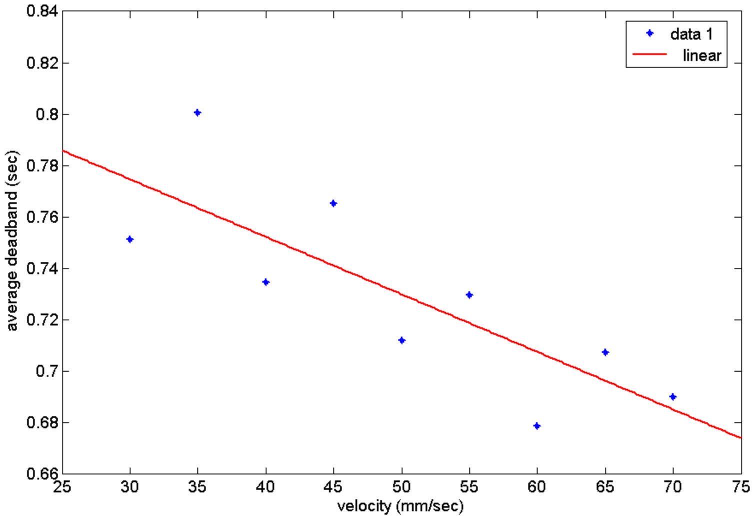

Figure 2.20: Average deadband (sec) vs. insertion speed (mm/sec) and linear regression of the data.

Catheter Insertion at Variable Stroke Lengths

In this set of experiments, the unactuated catheter was inserted into the testbed with varying stroke lengths of the robots end-effector. It was observed that the mean deadband delay is 0.76 second and the standard deviation is $63ms$ for a variation in stroke length from $12mm$ to $30mm$. Figure 2.21 shows the average tracking error over the entire insertion plotted against the stroke length. The graph shows that the average tracking error is lower for higher stroke lengths. This is quite expected since the flexing of the catheter is more pronounced for lower stroke lengths, resulting in greater tracking error.

2.7.3 Master-Slave Control Results

In the master-slave control mode, a catheter is inserted into the testbed using a haptic device. The catheter was instrumented with SMA actuators to control the orientation of the distal end of the catheter. The user inserts the active catheter into the testbed using a haptic device. The overhead camera (simulating X-ray fluoroscopy) captures images of the testbed in real time. The

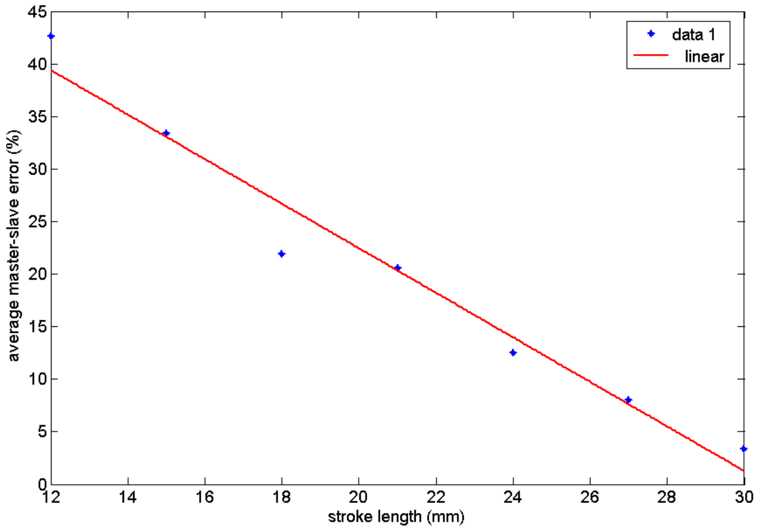

Figure 2.21: Average master-slave tracking error (percent) vs. stroke length (mm) and linear regression of the data

position of the distal tip of the catheter detected by the image-based tracking algorithm, is overlayed with the captured images to provide a visual feedback to the user. In addition, the image processing algorithm also provides the user with indications to control the stroke length provided from the haptic device - away from the bifurcations, the algorithm provides normal stroke length command; close to the junctions the algorithm recommends to the user to reduce the stroke length; at the junction, it provides the user with the necessary the information to stop insertion and bend the catheter in the desired direction. Images obtained from a secondary camera also provide the user with visual information about the position of the end-effector of the robot with respect to the port of entry on the testbed.

In the experiments performed here, the user remotely commanded the robot to insert the catheter into the testbed. During the test, the user was sufficiently isolated from the robot and the testbed and operated the robot and the catheter using only the images provided to him/her, thereby mimicking an actual remotely controlled master-slave catheter insertion. The path consists of two junctions, at the first the user has to bend the catheter to the right and at the second the user has to bend it to the left. The overall path length from the point of entry to the target is roughly about $1.7m$. The results of the experiment are shown in Figure 2.22.

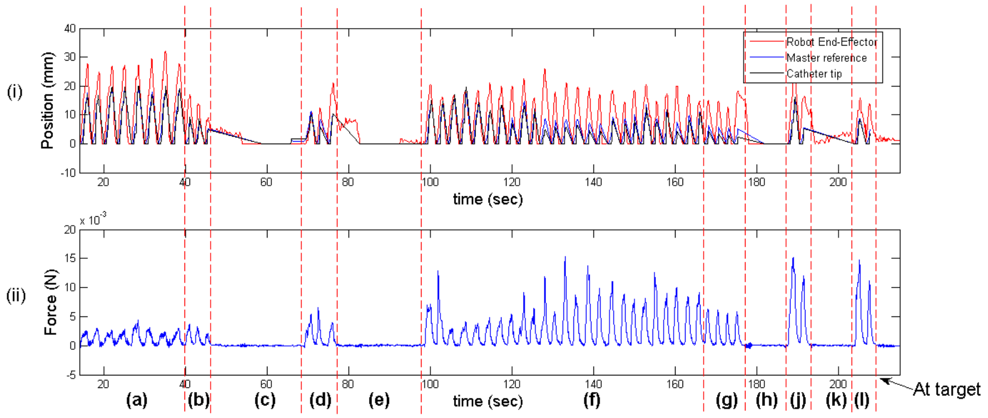

Figure 2.22: Experimental results: (i) shows the master (haptic device), slave (catheter tip) and robot end-effector position vs. time during catheter insertion (ii) shows the force of insertion measured at the robot end-effector along the direction of insertion. Different phases of insertion are shown as: (a) normal insertion before the first branch, (b) slow-down before the first branch, (c) no insertion, catheter bends at junction (d) insertion after bending is complete at the first junction, (e) wait for SMA actuators to cool (f) normal insertion before the second branch, (g) slow-down before the second branch, (h) no insertion, catheter bends at junction (j) insertion after bending is complete at the second junction, (k) wait for SMA actuators to cool (l) normal insertion before the target

As seen in Figure 2.22(a), the user inserted the catheter into the testbed with the maximum stroke. The visual servoing control algorithm ensured that the tip of the catheter followed the reference generated by the haptic device. Figure 2.22(i) shows the position of the end-effector along with the position of the master and the tip of the catheter.

Figure 2.23 shows a magnified view of the positions of the master, the robot end-effector and the catheter tip in region (a) of Figure 2.22. It can be seen that the catheter tip followed the reference generated by the haptic device quite accurately. The robot end-effector motion is nearly 50% higher

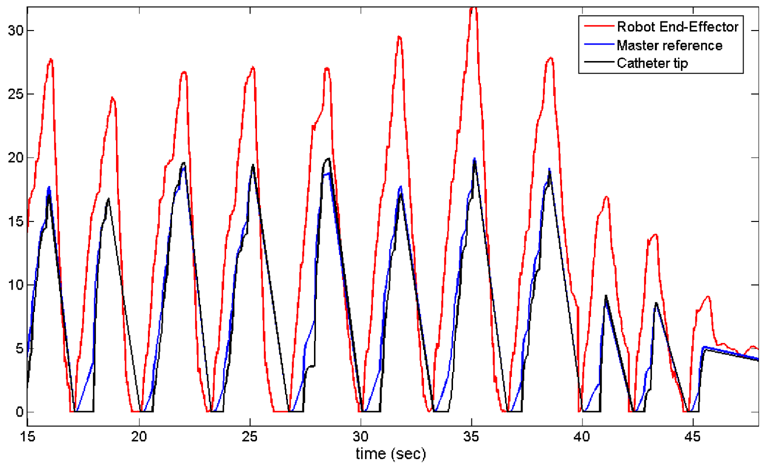

Figure 2.23: Magnified view of normal insertion in region (a) of Figure 2.22

than the catheter tip position in order to compensate for the flexing in the catheter. The figure also shows a significant delay in advancing the distal end of the catheter due to the frictional forces and high flexibility of the catheter.

Close to the junctions, the user reduced the stroke length using the haptic device. The image processing algorithm tracked the tip of the catheter and provided the desired suggestions to the user based on the proximity of the catheter from the junction. At the junction (Figure 2.22:(c)&(h)), the user was informed by the image processing algorithm to stop inserting the catheter. On pressing a key, the robot was disabled and the haptic device now controlled the active catheter. The user commanded the catheter to bend in the desired direction while the force exerted on the tip of the catheter was reflected to the stylus of the haptic device. Once the catheter reached the desired orientation, the user re-activated the robot to continue insertion of the catheter into the desired branch, as shown in Figure 2.22:(d)&(j). The user then waited for the actuators on the catheter to cool and regain their original shape before resuming normal insertion, as shown in Figure 2.22:(e)&(k). The force of insertion for the entire procedure is shown in Figure 2.22(ii).

The mean master-slave error was 9.72% of the master reference with a standard deviation of 0.1475%. It can be observed from the results that both overshoot and undershoot occur while the catheter tip tracks the master reference, thereby proving the non-linear nature of the dynamics of catheter insertion. The mean difference between the robot end effector motion and the catheter tip motion is 106.11% of the catheter tip motion, reaching a maximum around 600% near the second junction. The relatively large motion of the end-effector compared to the catheter tip is to compensate for the flexing of the catheter within the tubes. The deadband delay is variable and lies between 0.4 and 0.9 seconds.

2.7.4 Sensor Fusion Results

We have used a Firewire digital camera which captures images at rates up to $30Hz$ and a 5-DOF magnetic tracking sensor (Appendix H) which can capture the position and orientation at a sampling rate of up to $40Hz$. A multi-threaded application has been developed to perform the image-based tracking and capture the magnetic sensor data as well as the Kalman filtering, in real time. The image processing thread is performed asynchronously whenever a new frame is grabbed; The magnetic sensor measurement is updated at the same time instant. The Kalman filtering is run synchronously every 3 ms, the robot client is controlled at the same rate. As the catheter tip position measurement is done in a lower rate than the Kalman filter update rate, we have used a zero-order hold to match the lower rate with the higher rate, i.e., the last valid measurement is used in each iteration of the Kalman filter update. The initial value for the system state X_0 is set to $[x_0,\ y_0,\ 0.0,\ 0.0]^T$ where $(x_0,\ y_0)$ is the position of the start point (in the undistorted image place coordinates), which is the point where image-based tracking and data fusion start working. The initial state covariance P_0 is set to diag(1, 1, .01, .01). The process noise covariance matrix is assumed to be constant but the measurement noise covariance is varied during the process; The noise covariance for the magnetic sensor is increased whenever it is unable to provide valid measurement data and generates a data missing error.

Three sets of experiments were performed to evaluate the performance of the data fusion scheme:

1. In the first experiment set, all the conditions were normal, i.e., there was no major occlusion in the camera's view and there was no major electromagnetic interference within the working area of the MTS. The results are shown in Table 2.3. and Figure 2.24. As seen in Table 2.3, the standard deviation of the difference between the estimated value and each of the measured values is less than the standard deviation of the difference between the data from two sensors; This means that the uncertainty has decreased due to data fusion.

x-coordinate	std$(x_I - x_M)$	std$(x_M - \hat{x})$	std$(x_I - \hat{x})$
Values	5.0523	3.7553	2.3331
Percentage	100	74.3285	46.1792
y-coordinate	std$(y_I - y_M)$	std$(y_M - \hat{y})$	std$(y_I - \hat{y})$
Values	5.0251	3.8876	2.4971
Percentage	100	77.3638	49.6935

Table 2.3: Tracking and data fusion results in normal conditions

2. In the second experiment set, all the conditions were normal except that there were significant occlusions at two different regions (as shown in Figure 2.25(f)), i.e., the view was totally blocked when the catheter was within these regions. As a result, the image-based tracking fails to detect the catheter tip position in these areas. The results are shown in Table 2.4 and Figure 2.25. As seen in Table 2.4, the standard deviation of the difference between the estimated value and each of the measured values is less than the standard deviation of the difference between the data from the two sensors; This means that the uncertainty has decreased due to data fusion although the image-based tracking had a major failure twice during the experiment. As seen in Figure 2.25(e), the estimated catheter insertion depth is not affected by the failure of the image-based tracking sensor.

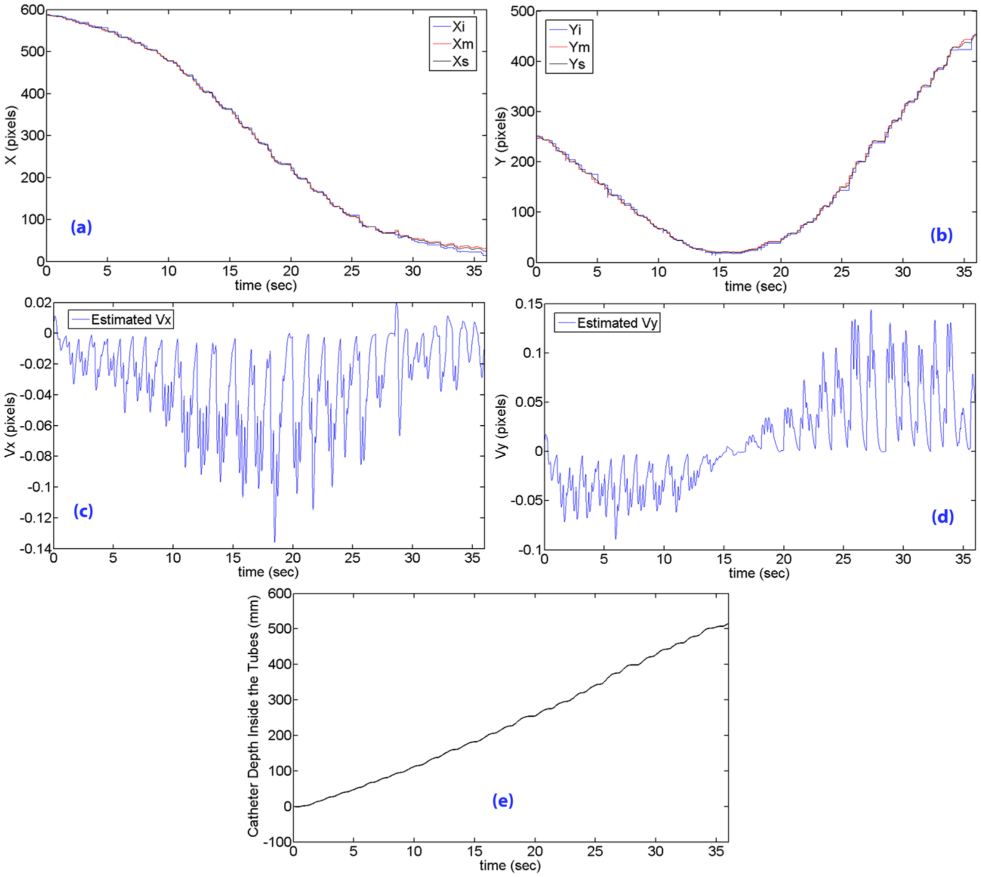

Figure 2.24: Normal image and magnetic tracking data fusion when there is no occlusion and no EM interference: (a) and (b) show X and Y coordinates of the catheter tip in pixels vs. time (sec) for image-based tracking (red), magnetic tracking (blue) and the estimated value by Kalman filer (black). (c) and (d) represent the estimated catheter tip displacements V_x and V_y in pixels vs. time (sec). (e) shows the estimated depth (mm) of catheter inserted into the vascular model vs. time (sec).

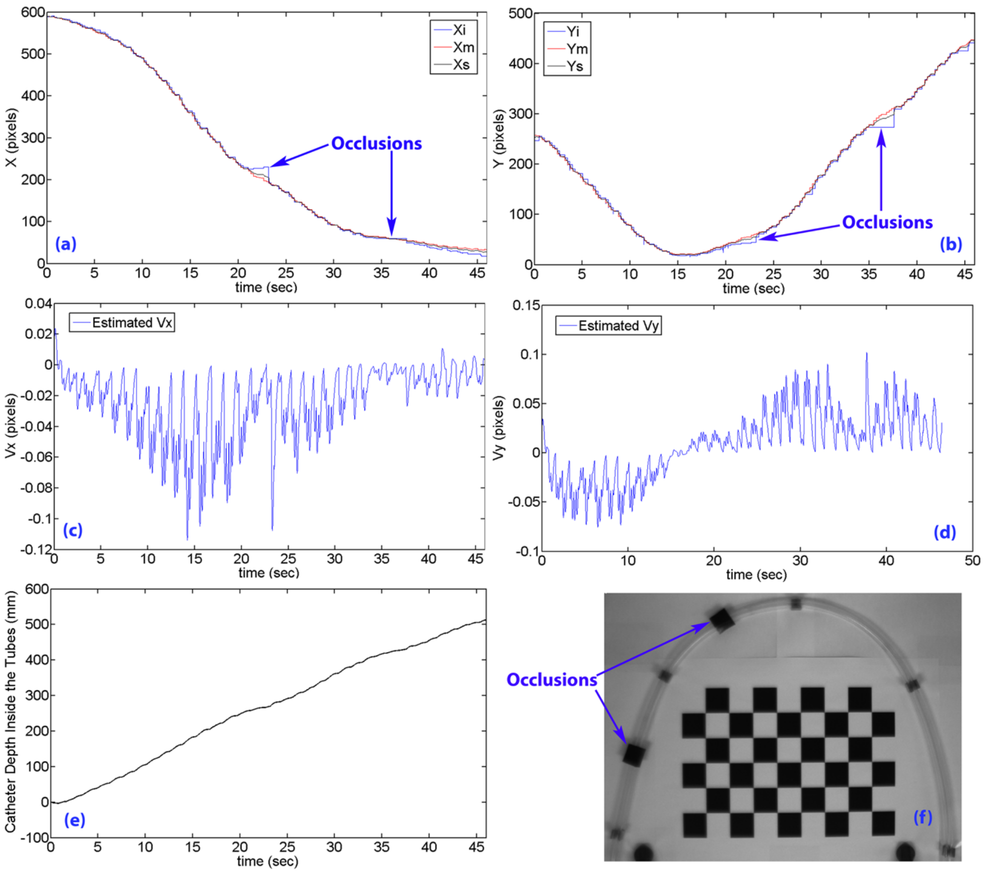

Figure 2.25: Image and magnetic tracking in case of occlusion: (a) and (b) show X and Y coordinates of the catheter tip in pixels vs. time (sec) for image-based tracking (red), magnetic tracking (blue) and the estimated value by Kalman filer (black). (c) and (d) represent the estimated catheter tip displacements V_x and V_y in pixels vs. time (sec). (e) shows the estimated depth (mm) of catheter inserted into the vascular model vs. time (sec).

x-coordinate	std$(x_I - x_M)$	std$(x_M - \hat{x})$	std$(x_I - \hat{x})$
Values	7.0261	4.8755	2.6704
Percentage	100	69.3907	38.0063

y-coordinate	std$(y_I - y_M)$	std$(y_M - \hat{y})$	std$(y_I - \hat{y})$
Values	6.8302	4.7697	2.6443
Percentage	100	69.8324	38.7149

Table 2.4: Tracking and data fusion results in case of occlusion

3. In the third experiment set, all the conditions were normal except that there was a considerable amount of electromagnetic interference (EMI). EMI was intentionally created by putting a wire carrying AC current, near the testbed as shown in Figure 2.26(f). The MTS became very noisy when it approached this area and lost tracking within this region. The results are shown in Table 2.5 and Figure 2.26. As seen in Table 2.5, the uncertainty has decreased due to data fusion although the magnetic tracking system had a major failure during the experiment. As seen in Figure 2.26(e), the estimated catheter insertion depth is not affected by the failure of the magnetic tracking sensor.

x-coordinate	std$(x_I - x_M)$	std$(x_M - \hat{x})$	std$(x_I - \hat{x})$
Values	28.4498	10.0257	23.4819
Percentage	100	35.2400	82.5379

y-coordinate	std$(y_I - y_M)$	std$(y_M - \hat{y})$	std$(y_I - \hat{y})$
Values	22.2869	5.3636	20.1194
Percentage	100	24.0662	90.2747

Table 2.5: Tracking and data fusion results in case of electromagnetic interference

2.8 Remarks

The experimental results show the autonomous and master-slave insertion of a catheter instrumented with SMA actuators (active catheter) into a testbed

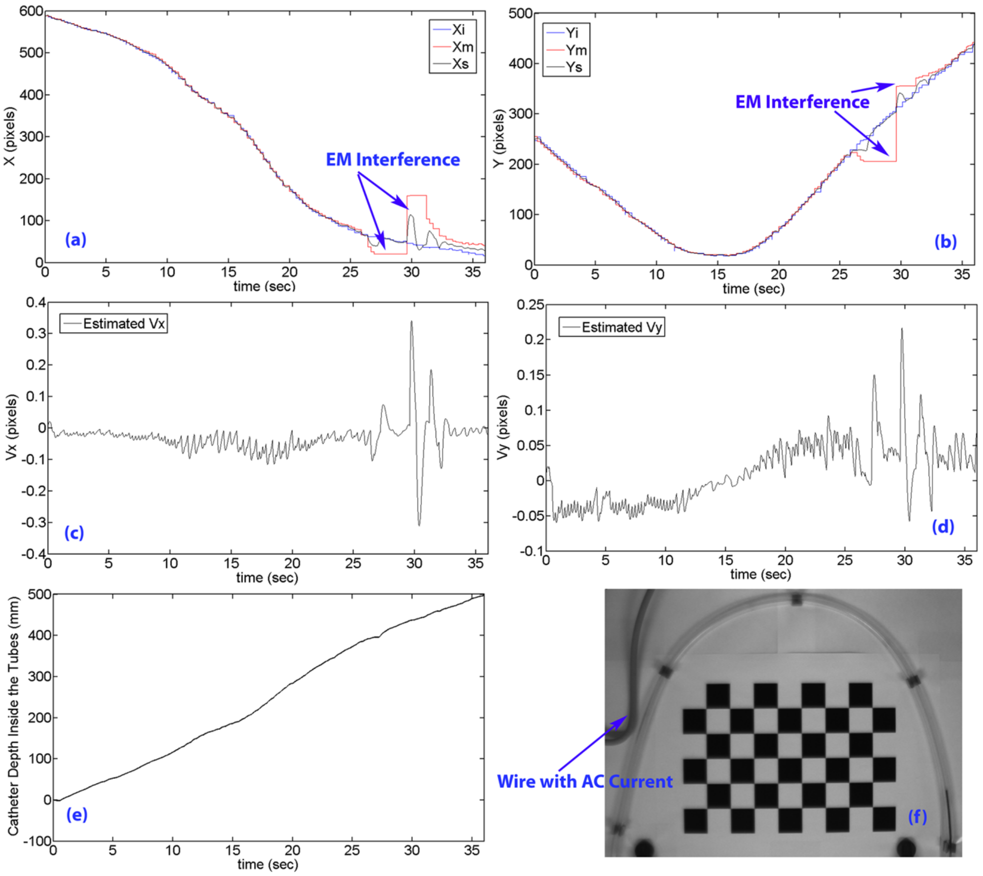

Figure 2.26: Image and magnetic tracking in case of electromagnetic (EM) interference: (a) and (b) show X and Y coordinates of the catheter tip in pixels vs. time (sec) for image-based tracking (red), magnetic tracking (blue) and the estimated value by Kalman filer (black). (c) and (d) represent the estimated catheter tip displacements V_x and V_y in pixels vs. time (sec). (e) shows the estimated depth (mm) of catheter inserted into the vascular model vs. time (sec).

depicting a layout of arteries in the body. In these experiments, we have used a Mitsubishi PA10 7-DOF robot to insert the catheter into the testbed using a gripper, while controlling the force of insertion to prevent buckling at the proximal end of the catheter. The force sensor on the wrist of the robot measures the force of insertion and its readings are fed to the AHIC scheme to regulate the force along the direction of insertion. It should be noted that this force reading is not representative of the actual force acting at the tip of the catheter since the catheter is highly flexible. The force measured is the sum of the frictional and other forces acting along the catheter and the force acting at the tip of the catheter. We have also developed a master-slave system for performing semi-autonomous catheter insertion. In this work, all 7 degrees of freedom of the PA10 robot are required to control the position and orientation of the end-effector of the robot while regulating the force of insertion. The image processing algorithms need modifications for clinical applications. The image-based tracking algorithm may fail in case of an occlusion. We have also developed a data fusion scheme to combine both magnetic tracking and image-based tracking data as well as the roadmap information to make catheter tip tracking more reliable and robust, specifically with regard to occlusions.

2.9 Conclusion

The objective of this work was to develop a safe and reliable way to perform robot-assisted catheter insertion and guidance through a blood vessel. A novel method was proposed for performing catheter insertion based on image-based real-time tracking of the tip of the catheter. In our experimental testbed, we have used a Mitsubishi 7-DOF PA 10-7C robot to perform catheter insertion using a catheter instrumented with SMA actuators. We have demonstrated the concept of autonomous guidance of a catheter to the site of plaque buildup using visual servoing. We believe that the use of this system can reduce the amount of stress and fatigue to the clinician. In addition, implementation of the approach using a master-slave control scheme would also reduce the exposure of the clinician to the X-ray radiation present during conventional angioplasty while providing direct control for the clinician.

We have developed a teleoperation framework to perform robot-assisted active catheter insertion. The tip of the catheter very closely follows the motion of the haptic device. In addition, we have shown that the robot end-effector moves up to nearly 600% more than the tip of the catheter in order to compensate for flexing in the catheter. By controlling the tip of the catheter (instead of the robot end-effector) as the slave, the clinician is provided with a more transparent procedure, wherein the tip of the catheter very closely follows the motion of the master. We have also studied various factors influencing flexing in a catheter inside a vasculature. These results should help in developing a more accurate model for flexing of the catheter and aid in improving the accuracy of image-guided robot-assisted catheter insertion for different applications.

We have also introduced a Kalman filter for data fusion of image-based and magnetic tracking systems in a catheter insertion application. The results indicate reduced uncertainty in the estimated data. The standard deviation of the difference between the image-based and magnetic sensor tracking data, represents a measure of uncertainty of the measured variables (x and y). The standard deviations of the differences between the measurements of each sensor and the corresponding estimated variable, show decreased uncertainty after data fusion. The percentage of uncertainty after data fusion to before data fusion is shown in the third and sixth rows of Tables. 2.3, 2.4 and 2.5 for different experiments with different conditions. The data fusion scheme provides better estimation with more accuracy and less uncertainty compared to each of the sensors individually. It also provides an estimation of the displacement in each direction (x and y) which is then used to estimate the catheter insertion depth. The estimate of the insertion depth (the length of the catheter inserted into the vasculature) is then used to control the catheter tip position in the vasculature. The estimator can also provide higher-frequency feedback compared to each of the sensors, by interpolating the data. This would be very useful for efficient catheter tip position control using robot-assisted insertion.

Bibliography

[1] Alexandre R. Abreu, Michael A. Campos, and Bruce P. Krieger. Pulmonary artery rupture induced by a pulmonary artery catheter: A case report and review of the literature. *Journal of Intensive Care Medicine*, 19, 2004.

[2] J. Karppinen, T. Parviaine, A. Servomaa, and T. Komppa. Radiation risk and exposure of radiologists and patients during coronary angiography and percutaneous transluminal coronary angioplasty (PCTA). *Nuclear Technology Publishing*, 57:481–485, 1995.

[3] Richard D. Nawfel, Philip F. Judy, Stuart G. Silverman, Stuart Hooton, RT Kemal Tuncali, and Douglass F. Adams. Patient and personnel exposure during CT fluoroscopy-guided interventional procedures. *Radiology*, 216:180–184, 2000.

[4] Jagadeesan Jayender. *Haptics Enabled Robot-Assisted Active Catheter Insertion*. PhD thesis, The University of Western Ontario, 2007.

[5] Fumihito Araf, Ryo Fuji, Toshio Fukuda, and Makoto Negoro. New catheter driving method using linear stepping mechanism for intravascular neurosurgery. In *IEEE International Conference on Robotics and Automation Washington, DC*, May 2002.

[6] M. Tanimoto, F. Arai, T. Fukuda, K. Itoigawa, M. Hashimoto, L. Takahashi, and M. Negoro. Telesurgy system for intravascular neurosurgery. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2000.

[7] F. Arai, M. Tanimoto, T. Fukuda, I.C. Shimojma, H. Matsuurq, and M. Negoro. Multimedia tele-surgery using high speed optical fiber network and its application to intravascular neurosurgery. In *IEEE International Conference on Robotics and Automation*, 1996.

[8] Y. Thakur, J.S. Bax, D.W. Holdsworth, and M. Drangova. Design and performance evaluation of a remote catheter navigation system. *IEEE Transactions on Biomedical Engineering*, 56(7):1901 –1908, 2009.

[9] Corindus vascular robotics. `http://www.corindus.com/products/CorPath200.aspx`.

[10] R. Beyar, T. Wenderow, D. Lindner, G. Kumar, and R. Shofti. Concept, design and pre-clinical studies for remote control percutaneous coronary interventions. *Eurointervention*, 1:340–345, 2005.

[11] Hansen medical's sensei robotic catheter system. `http://www.hansenmedical.com/sensei`.

[12] A. Amin, J. Grossman, and P. Wang. Early experience with a computerized robotically controlled catheter system. *Journal of Interventional Cardiac Electrophysiology*, 12:199–202, 2005.

[13] N. Marrouche, J. Brachmann, O. Wazni, V. Shibgilla, S. Beheiri, J. Guenther, and A. Natale. Preliminary human experience using a novel robotic catheter remote control. *Heart Rhythm*, 2:S63, 2005.

[14] Stereotaxis's niobe, magnetic catheter navigation system. `http://www.stereotaxis.com/niobe.html`.

[15] J. Jayender, R.V. Patel, and S. Nikumb. Robot-Assisted Active Catheter Insertion: Algorithms and Experiments. *The International Journal of Robotics Research (submitted)*, 28(9):1101–1117, 2009.

[16] M. Bock, S. Muller, S. Zuehlsdorff, P. Speier, C. Fink, P. Hallscheidt, R. Umathum, and W. Semmler. Active catheter tracking using parallel MRI and real-time image reconstruction. *Magnetic Resonance in Medicine, Wiley InterScience*, 55(5):14541459, May 2006.

[17] R. A. Omary, O. Unal, D. S. Koscielski, R. Frayne, F. R. Korosec, C. A. Mistretta, C. M. Strother, and T. M. Grist. Real-time MR imaging-guided

passive catheter tracking with use of gadolinium-filled catheters. *Journal of Vascular and Interventional Radiology*, 11(8):1079–1085, Sep 2000.

[18] G. Schoonenberg, M. Schrijver, Qi Duan, R. Kemkers, and A. Laine. Adaptive spatial-temporal filtering applied to x-ray fluoroscopy angiography. *Proceedings of SPIE*, 5744(8):870–878, Apr 2005.

[19] C. L. Merdes and P. D. Wol. Locating a catheter transducer in a three-dimensional ultrasound imaging field. *IEEE Transactions on Biomedical Engineering*, 48(12):1444–1452, Dec 2005.

[20] Paul M Novotny, Jeff a Stoll, Nikolay V Vasilyev, Pedro J del Nido, Pierre E Dupont, Todd E Zickler, and Robert D Howe. GPU based real-time instrument tracking with three-dimensional ultrasound. *Medical image analysis*, 11(5):458–64, October 2007.

[21] S Kesner, S Yuen, and R Howe. Ultrasound Servoing of Catheters for Beating Heart Valve Repair. *Information Processing in Computer-Assisted Interventions*, 2010.

[22] S Kesner and RD Howe. Design and control of motion compensation cardiac catheters. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1059–1065, 2010.

[23] J. Jayender, R. V. Patel, and S. Nikumb. Robot-assisted catheter insertion using hybrid impedance control. In *2006 IEEE International Conference on Robotics and Automation (ICRA)*, May 2006.

[24] J. Jayender, M. Azizian, and R.V. Patel. Bilateral telemanipulation of a flexible catheter in a constrained environment. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 649 –654, 2008.

[25] Lewis A. Eisen, Taro Minami, Hiroshi Sekiguchi, Jeffrey S. Berger, Paul Mayo, and Mangala Narasimhan. Ultrasound demonstration of asymmetry between the left and right femoral and radial arteries. *Chest*, 103:201S–a, 2006.

[26] Yuji Ikari, Masakazu Nagaoka, Jae-Young Kim, Yoshihiro Morino, and Teruhisa Tanabe. The physics of guiding catheters for the left coronary artery in transfemoral and transradial interventions. *The Journal of Invasive Cardiology*, 17:636 – 641, December 2005.

[27] Jordin D. Green, Reed A. Omary, Brian E. Schirf, Richard Tang, Biao Lu, James A. Gehl, J. Jenny Huang, James C. Carr, F. Scott Pereles, and Debiao Li. Comparison of x-ray fluoroscopy and interventional magnetic resonance imaging for the assessment of coronary artery stenoses in swine. *Magnetic Resonance in Medicine*, 54(5):1094–1099, 2005.

[28] W. Q. Zhang, K. R. Dai, and C. T. Wang. Novel method for correction of x-ray fluoroscopic image. *27th Annual International Conference of the Engineering in Medicine and Biology Society, IEEE-EMBS*, pages 6340 – 6343, 2005.

[29] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1330–1334, 2000.

[30] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. *International Conference on Computer Vision*, pages 666–673, 1999.

[31] Wei Sun and J. R. Cooperstock. Requirements for camera calibration: Must accuracy come with a high price? *Seventh IEEE Workshops on Application of Computer Vision*, 1:356–361, 2005.

[32] C. Harris and M. Stephens. A Combined Corner and Edge Detection. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.

[33] E. Trucco and A. Verri. *Introductory techniques for 3-D computer vision.* Englewood Cliffs, NJ: Prentice-Hall, 1998.

[34] Sascha Krueger, Holger Timinger, Ruediger Grewer, and Joern Borgert. Modality-integrated magnetic catheter tracking for x-ray vascular interventions. *Physics In Medicine And Biology*, 50:581–597, 2005.

[35] J. F. Canny. Computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, Nov. 1986.

[36] R. C. Gonzalez and R. E. Woods. *Digital Image processing.* Prentice Hall International, second edition edition, 2002.

[37] A. J. Duerinckx. *Coronary Magnetic Resonance Angiography*, chapter 2, Anatomy and Physiology of Coronary Arteries. Springer-Verlag, 2002.

[38] Holger Timinger, Sascha Krueger, Klaus Dietmayer, and Jrn Borgert. Motion compensation for interventional navigation on 3D static roadmaps based on a dynamic motion model. *Proceedings of the 18th International Congress and Exhibition on Computer Assisted Radiology and Surgery, CARS*, 1268:1055–1060, 2004.

[39] Holger Timinger, Sascha Krueger, Joern Borgert, and Ruediger Grewer. Motion compensation for interventional navigation on 3D static roadmaps based on an affine model and gating. *Physics In Medicine And Biology*, 49:719732, 2004.

[40] M. Grassa, R. Koppea, E. Klotza, R. Proksaa, M.H. Kuhna, H. Aertsb, J. Op de Beekb, and R. Kemkers. Three-dimensional reconstruction of high contrast objects using c-arm image intensifier projection data. *Computerized Medical Imaging and Graphics*, 23:311321, 1999.

[41] Hildegard Koehler, Michel Couprie, Sahla Bouattour, and Dietrich Paulus. Extraction and analysis of coronary-tree from single x-ray angiographies. *Proceedings of the SPIE*, 5367:810–819, 2004.

[42] Ali Zifan, Panos Liatsis, Panagiotis Kantartzis, Manolis Gavaises, Nicos Karcanias, and Demosthenes Katritsis. Automatic 3d reconstruction of coronary artery centerlines from monoplane x-ray angiogram images. *International Journal Of Biological And Medical Sciences*, 1:44–49, 2008.

[43] M. Azizian, J. Jayender, and R.V. Patel. Image processing algorithms for real-time tracking and control of an active catheter. In *European Control Conference (ECC)*, 2007.

[44] C. Tercero, S. Ikeda, T. Uchiyama, T. Fukuda, F. Arai, Y. Okada, Y. Ono, R. Hattori, T. Yamamoto, M. Negoro, and I. Takahashi. Autonomous catheter insertion system using magnetic motion capture sensor for endovascular surgery. *International Journal of Medical Robotics and Computer Assisted Surgery*, 3(1):52–58, 2006.

[45] H. B. Mitchell. *Multi-Sensor Data Fusion: An Introduction.* Springer, 2007.

[46] Charles K. Chui and Guanrong Chen. *Kalman Filtering with Real-Time Applications.* Springer-Verlag, third edition, 1998.

[47] J. B. Gao and C. J. Harris. Some remarks on kalman filters for the multisensor fusion. *Information Fusion*, 3(3):191–201, September 2002.

[48] Y. Bar-Shalom and L. Campo. The effect of the common process noise on the two-sensor fused-track covariance. *IEEE Transactions on Aerospace Electronic Systems*, 22:803–805, November 1986.

[49] Hui Zhang, Kevin Cleary, Filip Banovac, and Amy White. Freehand 3D ultrasound calibration using an electromagnetically tracked needle. In *Medical Imaging 2006: Visualization, Image-Guided Procedures, and Display*, volume 6141. SPIE, 2006.

[50] S. Okazawa, R. Ebrahimi, J. Chuang, S. E. Salcudean, and R. Rohling. Hand-held steerable needle device. *IEEE/ASME Transactions on Mechatronics*, 10:285 – 296, 2005.

[51] R. J. Webster, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura. Nonholonomic modeling of needle steering. *International Journal of Robotics Research*, 25:509 – 525, 2006.

[52] Niki Abolhassani, Rajni Patel, and Mehrdad Moallem. Needle insertion into soft tissue: a survey. *Medical Engineering and Physics*, 29(4):413–431, 2007.

Chapter 3

Cardiac Ablation

There are commercial products which provide 3D rendered volumes, reconstructed from pre-operative CT or MRI images of patient's heart with tools for highlighting target locations for cardiac ablation applications. However, it is not possible to update the 3D volume intra-operatively to provide the interventional cardiologist with more up-to-date feedback at each instant of time. There are also commercial products for robot-assisted cardiac ablation which require specialized catheters or guide catheters. In this part of the thesis, we describe the system we have developed for master-slave robotic actuation and control of conventional steerable catheters as well as real-time three dimensional visualization for cardiac ablation. A 4D ultrasound probe is used to acquire and update a 3D image volume. A magnetic tracking device is used to track the distal part of the ablation catheter in real time. 3D ultrasound image volumes go through some processing to make the heart tissue and the catheter more visible. The ultrasound probe is also equipped with a magnetic tracker which is used for online registration of the ultrasound image volumes. We have used transthoracic echocardiography (TTE) instead of the conventional transesophageal (TEE) or intracardiac (ICE) echocardiogram. A beating heart model has been used to perform the experiments. This method can be used both for diagnostic and therapeutic applications as well as training interventional cardiologists. A master-slave control system was developed for catheter tip position control.

3.1 Introduction

Cardiovascular diseases are among the major causes of death around the world. Among these is cardiac arrhythmia, which is caused by abnormal electrical activity in the heart. Cardiac arrhythmia may cause fast, slow or irregular heart beats. It may become life threatening and cause cardiac arrest or it may only become annoying as a result of the conscious awareness of the heart beat (palpitations). Cardiac ablation is an invasive procedure for destroying faulty electrical pathways which cause arrhythmia. One method of cardiac ablation involves inserting a catheter into the heart chambers in order to destroy the faulty electrical activity by burning or freezing parts of the endocardium [1].

Keane [2] has studied and compared different catheter-based techniques for cardiac ablation. Radiofrequency (RF) ablation is the most commonly used cardiac ablation technique [2]. The major challenges in RF ablation include (a) accurate positioning of the catheter tip, and (b) keeping contact with a target point in the presence of the beating heart motion. The success rate of cardiac ablation is highly dependent on the ability to exactly ablate the specified faulty region. Therefore accurate positioning of the catheter tip is required. On the other hand, the size and depth of the lesions depend on the integral of the force at the catheter tip during the ablation [3]. Applying extra force during ablation may cause perforation of the heart wall and insufficient contact force may result in incomplete therapy and recurrence of arrhythmia.

In catheter-based cardiac ablation procedures, usually an electro-anatomical mapping of the patient's heart chambers is constructed and visualized, and the target locations are marked on the 3D image. *CARTO* (Biosense Webster, USA) and *EnSite NavX* (StJude Medical, St Paul, MN, USA) systems can also combine a pre-operative 3D CT or MRI scan of the patient's heart with a detailed electro-anatomical mapping of the heart [4, 5]. Use of pre-operative 3D imaging improves the understanding of the locations of ablation lesions with respect to structures such as the esophagus but studies show that it does not provide significant improvement in terms of reducing the X-ray fluoroscopy time [5] which means that intraoperative imaging feedback is still required for localizing the catheter within the anatomical structure. Therefore, intraoperative 3D imaging can be very useful. The ablation catheter is also visualized in some of the existing systems. However, there are some drawbacks which limit the capabilities of these systems, including: (a) the lack of depth perception, (b) the 3D image of the heart is not updated during the procedure, and (c) specific arrangements are required to capture preoperative images and align them with the intraoperative system. We have used 4D ultrasound imaging which can provide updated 3D images of the heart during the procedure through a transthoracic apical view of the heart. The same approach can also be used with transesophageal ultrasound imaging. We have also used a magnetic tracking system to localize the distal end of the catheter and to register the catheter to the 3D ultrasound volume, intraoperatively. It is expected that these will improve the overall perception for the interventional cardiologists, of the catheter tip position in a beating heart.

On the other hand, ablation catheters are usually guided manually using a pull-wire mechanism by manual actuation of knobs on the catheter handle. This makes it very difficult to coordinate steering of the catheter tip with the 3D visual feedback. Robotic manipulation of the catheter can be used to solve this problem and provide a more ergonomic interface for the clinicians to perform cardiac ablation. Hansen Medical (Mountain View, CA) has developed a catheter control system (Sensei^{TM} Robotic Catheter System) that allows a remotely-located operator to control catheters [6]. Stereotaxis (St. Louis, MO) has developed a magnetic catheter navigation system (NIOBE® Magnetic Navigation System [7]). In this system, a soft catheter with a magnetic tip is manipulated using two large external permanent magnets. However, the Sensei^{TM} system uses a specialized steerable guide catheter [8–10] and the NIOBE® system uses a special magnetic catheter and also requires two large permanent magnets around the patient [11]. It would be ideal to develop a robotic system that can work with existing steerable catheters.

In this part of the thesis, an image-guided robotic system is proposed to facilitate the positioning of the catheter tip using a master-slave control scheme with coordinated visual feedback. It has been shown that force feedback can improve the results of cardiac ablation [12, 13]. Several projects have been aimed at mounting miniature force sensors at the tip of a catheter [14–17]. The developed system is capable of implementing haptic feedback and bilateral teleoperation provided that the catheter is equipped with a force sensor. The developed method was tested on a beating heart phantom. In this part of the thesis, general aspects of the developed system, including the catheter actuation mechanism and online 3D visualization are described. The experimental setup is shown in Section 3.2 where various parts of the system are described along with a discussion about the system calibration and registration. The design of the robotic system for actuation of a steerable catheter is described in Section 3.3 while modeling and control of a steerable catheter is discussed in Section 3.4. The 3D reconstruction and visualization algorithms are discussed in Section 3.5. Section 3.6 describes the master-slave control. Results are given in Section 3.7, and Section 3.8 concludes this part of the thesis.

3.2 Experimental setup

The experimental setup is shown in Figure 3.1. A beating heart phantom with a realistic model of the left and right ventricles has been used. A 4D ultrasound probe (4DC7-3/40) was used with the SonixTouch ultrasound system (Ultrasonix, Vancouver, BC, Canada). A magnetic tracking system, Aurora (Northern Digital Inc., Waterloo, ON, Canada) was also used to track the distal part of the catheter and to register the ultrasound probe coordinates.

Figure 3.2 shows an overall view of the experimental setup. A custom designed catheter actuation mechanism with two degrees of freedom (DOF) was used for actuation of steerable catheters. The catheter actuation mechanism is mounted on a Mitsubishi PA10-7C robot manipulator. The actuation mechanism is used to move the knobs on a steerable catheter handle in order to actuate the distal part of the catheter using pull wires. The PA10 robot manipulator is used to insert/retract and twist the catheter as well as position the actuation mechanism close to the port of entry (where the catheter is in-

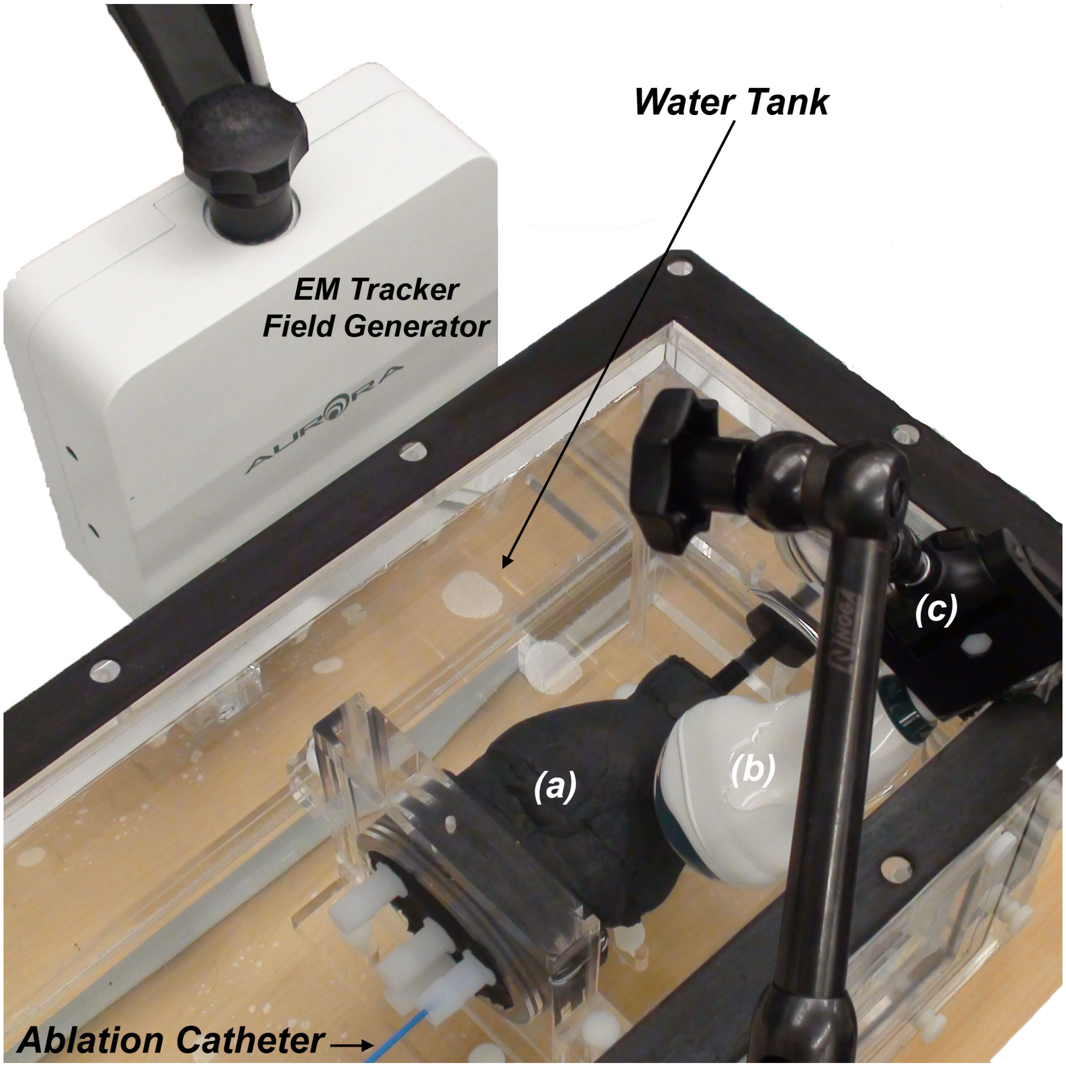

Figure 3.1: The ex-vivo experimental setup: (a) Beating heart phantom (Shelley Medical Imaging Technologies), (b) 4D ultrasound probe, (c) Ultrasound probe holder; An ablation catheter is inserted to a ventricle while the ultrasound probe captures images from an apical view.

serted into the body) while maintaining a remote center of motion. Figure 3.3 represents a block diagram of the system showing different elements of the overall system with their interconnections.

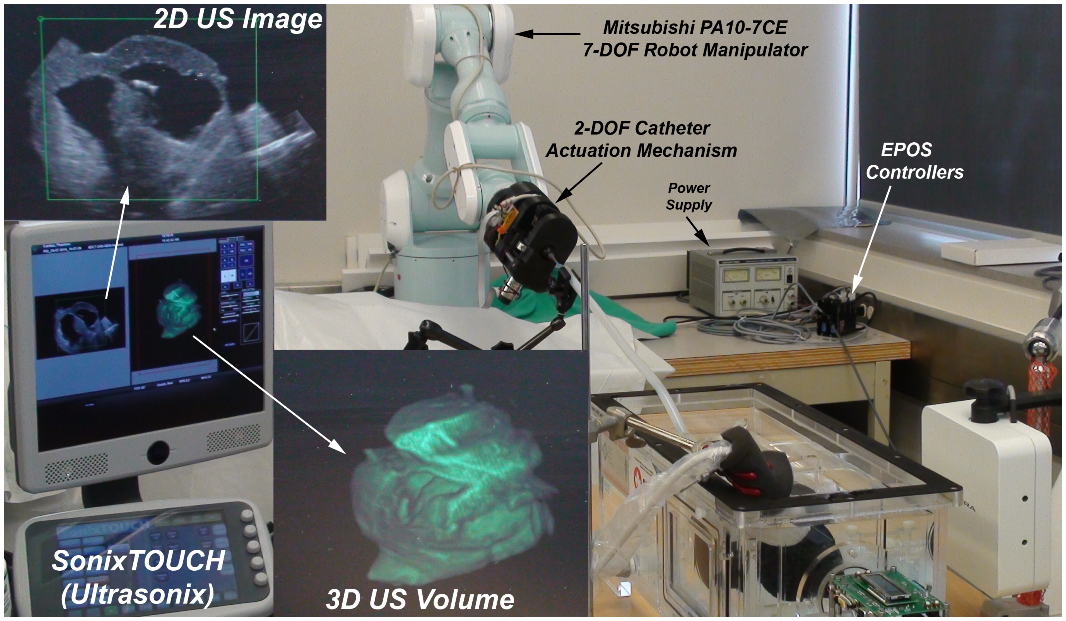

Figure 3.2: The image-guided robotic system for catheter insertion.

3.2.1 Calibration and Registration

Figure 3.4 shows the distal part of a steerable catheter with the magnetic trackers installed on it. Offset calibration should be performed to estimate the position/orientation offset of magnetic trackers mounted on the catheter.

Figure 3.5 shows a schematics of the magnetic tracking sensors mounted on the catheter; $\delta_1, \delta_2, \delta_3$ are the linear offsets of M_1, M_2, M_3 with reference to O_0, O_1, O_3, respectively. We assume that the central axis of M_1, M_2, M_3 is parallel to the central axis of the catheter, Z with angular offsets of ϕ_1, ϕ_2, ϕ_3 with respect to the X axis. The lateral offsets of the magnetic sensors (r_1, r_2, r_3) are assumed to be equal to the addition of the catheter and magnetic sensor diameters which is a known value. In order to calibrate for the offsets, we put the distal part of the catheter in a nonmagnetic straight tube to make sure

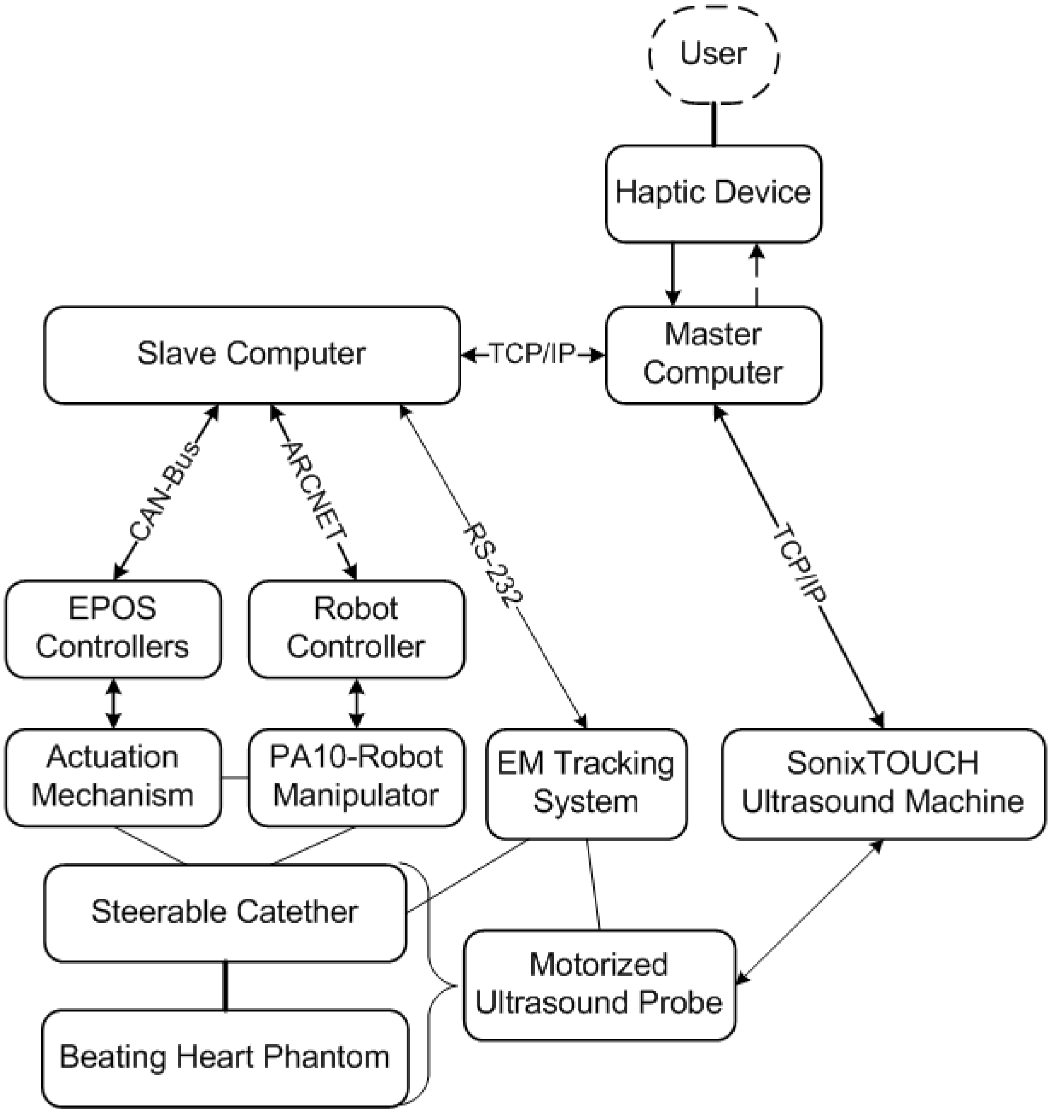

Figure 3.3: Block diagram of the system.

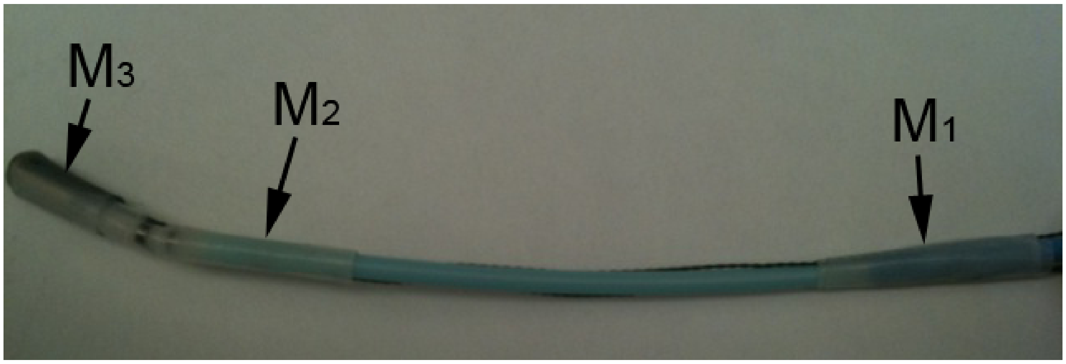

Figure 3.4: Three magnetic tracking sensors (M_1, M_2, M_3) mounted on the distal part of a steerable catheter and fixed using heat-shrink tubes.

the distal part of the catheter stays straight during calibration. The linear and angular offsets are then estimated by pivoting the catheter tip at a point and then using the method developed by Zhang et al. [18]. The problem is well-posed because (a) the offset calibration is independent of the roll angle for each 5-DOF magnetic tracker, and (b) the number of unknowns is limited to 6 while we can collect data in many different poses by pivoting the catheter around a point in a limited time.

Freehand ultrasound image calibration is performed offline using a catheter instrumented with magnetic trackers [18, 19]. Online registration of each ultrasound image volume to the reference coordinates is performed based on the probe location/orientation measured by a 6-DOF magnetic tracker mounted on the probe. The magnetic tracker is evaluated at the moments when the motorized probe is turned off to avoid magnetic interference. This is used to register the coordinates of the catheter with respect to the 3D ultrasound volume. This registration information is then used to coordinate the motion of the master haptic device with the actual motion of the catheter as seen in the 3D visual feedback. This is very important to provide an intuitive interface to control the distal tip of the catheter. It can also be used to visualize the catheter tip as a virtual tool in the reconstructed 3D ultrasound image in a virtual reality environment.

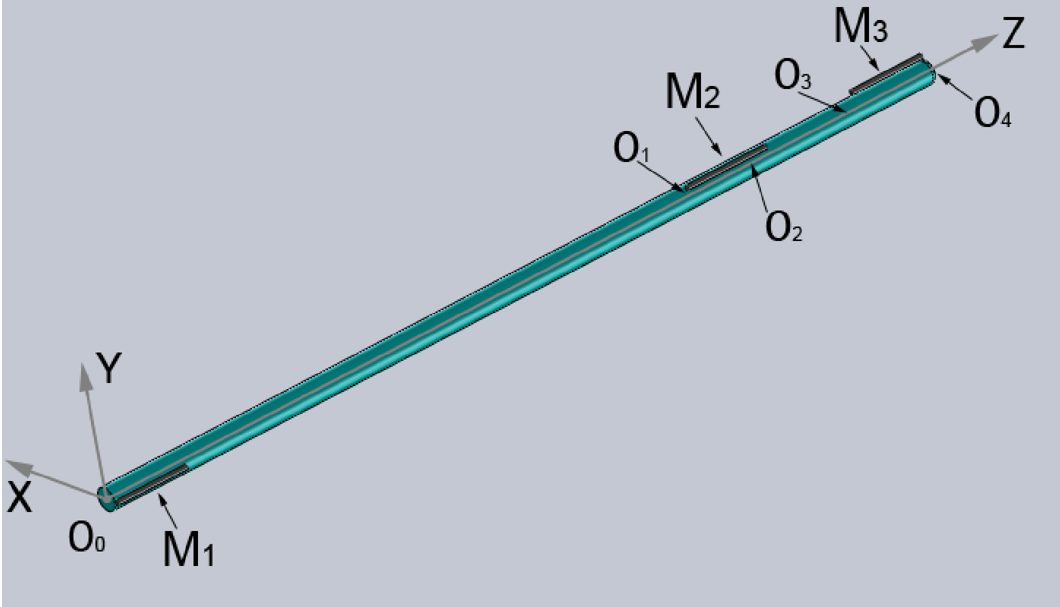

Figure 3.5: A schematics of the three magnetic sensors mounted on the distal part of the catheter. The points O_0, O_1, O_2, O_3, O_4 are the origins of coordinate systems shown in Figure 3.8 where O_4 is the distal tip of the catheter.

3.3 Robotic System Design

Conventional steerable catheters are usually actuated manually using knobs on the handles of each catheter. Rotary or prismatic knobs are most common and steerable ablation catheters use a pull-wire mechanism to transfer the motion from the knobs to the distal section of the catheter. Steerable catheters usually have one or two bending sections. The idea was to develop a catheter actuation mechanism which can manipulate a conventional steerable ablation catheter without making any modifications to the catheter. A 2-DOF actuation mechanism was designed that holds a steerable catheter handle and actuates the manual knobs using servo motors through small adapters. The CAD design of this actuation mechanism is shown in Figure 3.6.

Although the catheter actuation mechanism deploys the pull-wires of the catheter to bend its distal section, extra degrees of freedom are required to insert/retract and twist the catheter as well as to locate and orient the proximal

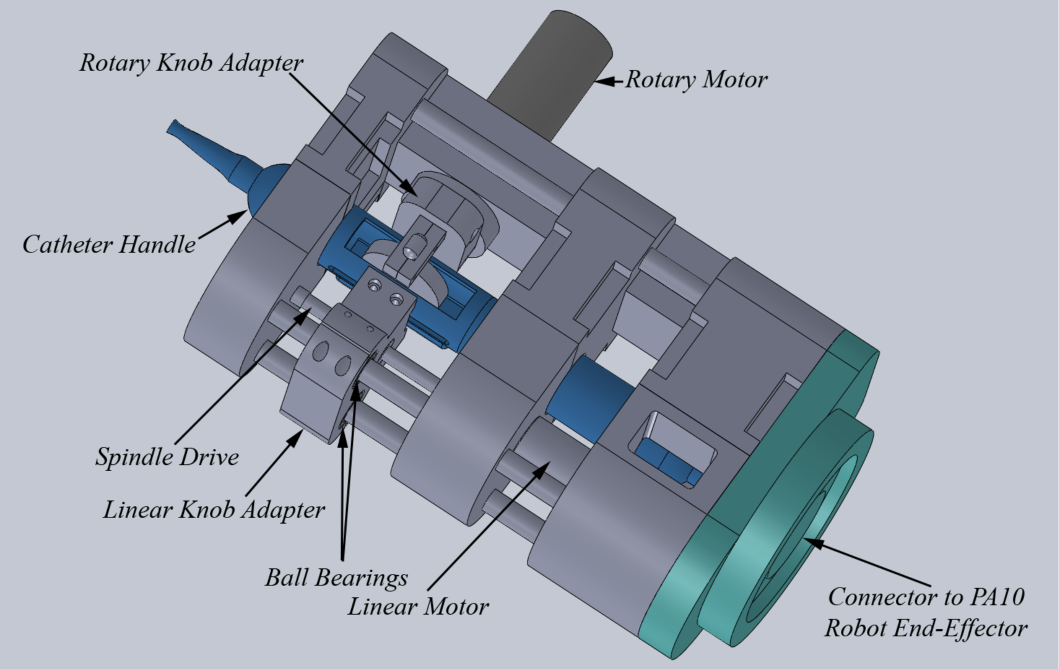

Figure 3.6: CAD design of the catheter actuation mechanism.

part of the catheter with respect to the port of insertion in the patient's body (i.e., usually through the femoral vein in the groin). Therefore, we have used a commercial robot manipulator, Mitsubishi PA10-7C to provide these extra degrees of freedom. This 7-DOF robot holds and moves the 2-DOF catheter actuation mechanism, where 3 DOFs are used to adjust the position with respect to the patient's body, 2 DOFs to align the orientation of the catheter at the proximal end, 1 DOF for insertion/retraction of the catheter and the last remaining DOF for twisting the catheter. Figure 3.7 shows the whole robotic system.

3.4 Steerable Catheter Modeling and Control

A continuum robot model [20], [21] has been developed for the steerable part of the catheter. Continuum robot models have been used to describe kinematics of single section steerable catheters by Jayender et al. [22] and Ganji et al. [23],

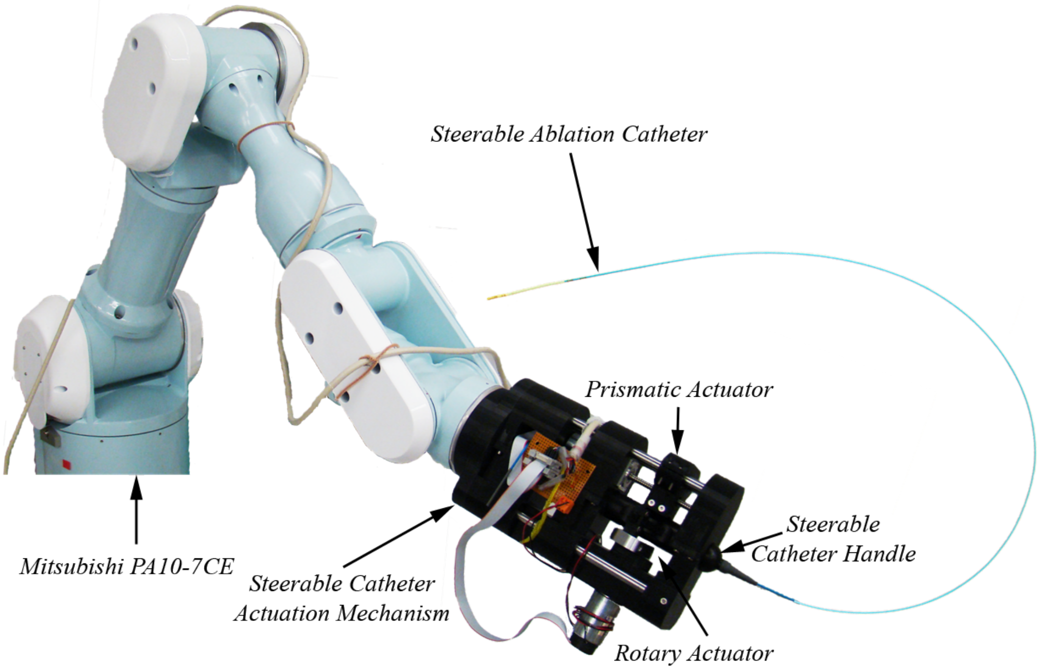

Figure 3.7: Steerable ablation catheter, catheter actuation mechanism and the Mitsubishi PA10 robot after assembly.

[24], [25]). The coordinates of the virtual joints used for modeling of a two-section steerable catheter are shown in Figure 3.8. The associated Denavit-Hartenberg (D-H) parameters [26], are shown in Table 3.1.

	1	2	3	4	5	6	7	8	9	10	11	12	13
α_{i-1}	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	0
a_{i-1}	0	0	0	0	0	0	0	0	0	0	0	0	0
d_i	δd_1	0	0	d_4	0	0	d_7	0	0	d_{10}	0	0	d_{13}
θ_i	0	θ_2	$\frac{\pi}{2}+\theta_3$	0	$\pi+\theta_5$	θ_6	0	θ_8	$\frac{\pi}{2}+\theta_9$	0	θ_{11}	$\pi+\theta_{12}$	0

Table 3.1: D-H parameters for the continuum robot model of a steerable catheter.

The independent joint variables are: δd_1, θ_2, θ_3 and θ_9. The continuum robot model poses constraints on some of the joint variables. The revolute joints should satisfy the following constraints:

$$\begin{cases} \theta_2 + \theta_6 = \frac{\pi}{2} \\ \theta_3 + \theta_5 = \frac{\pi}{2} \\ \theta_6 = \theta_8 \\ \theta_8 + \theta_{12} = \frac{\pi}{2} \\ \theta_9 + \theta_{11} = \frac{\pi}{2} \end{cases} \tag{3.1}$$

On the other hand, the prismatic joints d_4 and d_7 are coupled with θ_3 and θ_9, respectively. The coupling equations are described as follows:

$$\begin{cases} d_4 = \frac{\mathcal{L}_1}{\frac{\pi}{2}-\theta_3}\cos(\theta_3) \\ d_{10} = \frac{\mathcal{L}_2}{\frac{\pi}{2}-\theta_9}\cos(\theta_9) \end{cases} \tag{3.2}$$

where $\mathcal{L}_1$ and $\mathcal{L}_2$ are the lengths of the first and second bending sections, respectively. We assume there is no elongation in the catheter when each section bends; d_7 and d_{13} are constant lengths.

For the experiments in this chapter, we have used an ablation catheter, Bard Scorpion$^{\text{TM}}$ 2 (Bard Electrophysiology Division, C. R. Bard, Inc., Lowell, MA, US). The parameters of this catheter are reported in Table 3.2

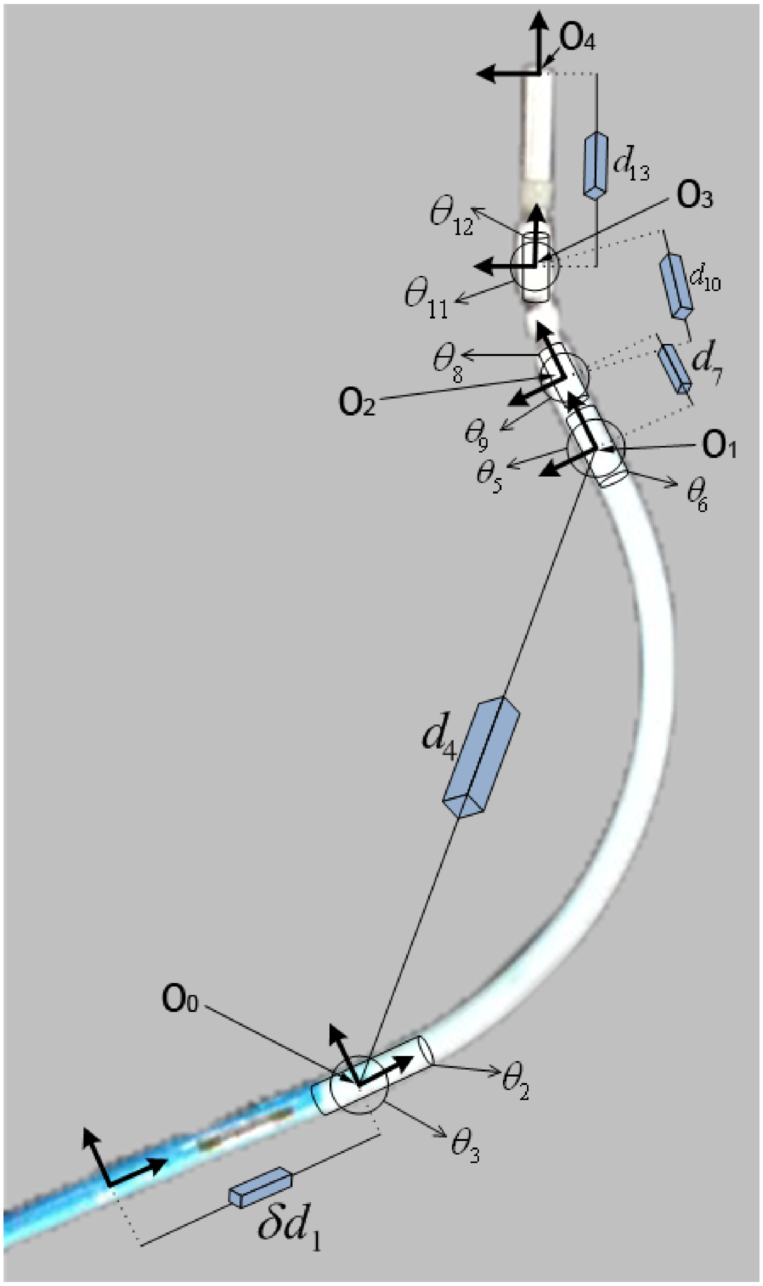

Figure 3.8: Kinematic modeling of a steerable catheter as a continuum robot.

Parameter	$\mathcal{L}_1$	$\mathcal{L}_2$	d_7	d_{13}
Value	$56.03mm$	$11.19mm$	$6.81mm$	$11.16mm$

Table 3.2: Specifications of the Bard Scorpion^{TM} 2 ablation catheter.

Although the D-H parameters are used to obtain the forward kinematics and the Jacobian matrix for the steerable catheter, the actuated variables are not the same as the independent variables δd_1, θ_2, θ_3 and θ_9. The catheter is actuated at its proximal end and the actuated variables are:

- δd which is the displacement along the catheter axis at its proximal end.
- θ_{twist} which determines the rotation of the catheter around its axis at its proximal end.
- α_{knob} which is the angle of the rotary knob.
- γ_{knob} which is the position of the prismatic knob.

The relationship between the actuated variables $(\delta d, \theta_{twist}, \alpha_{knob}, \gamma_{knob})$ and the controlled variables $(\delta d_1, \theta_2, \theta_3, \theta_9)$ is not linear due to several parameters including: (a) flexing of the catheter in the vasculature, (b) interaction with the walls of the vessels (e.g., friction), (c) backlash and hysteresis in the catheter pull-wire mechanism. We have shown the nonlinearities due to flexing of the catheter which determine the mapping from δd to δd_1 in Chapter 2. Mapping from θ_{twist} to θ_2 shows similar nonlinearities due to torsion of the catheter inside the vasculature. Ganji et al. [25] developed a simple model for the mapping between a steerable catheter knob angle and the bending angle of its distal end. However, we do not use models for the dead zone, backlash, hysteresis and other nonlinearities in these mappings. We use a roughly estimated linear model between the actuated variables and the controlled variables where the coefficients of this linear model are determined experimentally.

To adjust the position and orientation of the proximal part of the catheter with respect to the patient's body (or vascular model), the robot is put on gravity compensation mode. Appendix A describes the gravity compensation torque required for the PA10 robot. A real-time controller calculates gravity at each instant of time based on the current configuration of the robot and applies the calculated torques to the joint motors. Once the user adjusts the position and orientation of the robot with respect to the vasculature (this requires 5 degrees of freedom), a Jacobian transpose controller is used to control the remaining 2 DOFs, i.e., insertion/retraction of the catheter along the adjusted orientation and also twisting the catheter.

3.5 Online 3D volume reconstruction

3D ultrasound images are captured in real time. The capturing rate is 1.5 volumes per second with 31 frames per volume. The volume covers a fan-shaped scan with a field of view of 45.4^o with 1.46^o for each frame. The motorized probe is controlled and images are captured using a customized application developed based on *Porta SDK* [27] (Ultrasonix, Vancouver, BC, Canada). The pre-scan converted B-mode ultrasound frames are transferred over a TCP/IP network to the master computer using the OpenIGTLink [28] protocol. The *Pando SDK* [29] (Ultrasonix, Vancouver, BC, Canada) is used for scan conversion, i.e., converting from linear B-scan data into geometrically correct images. This method reduces network traffic, by sending the linear data and performing the conversion (and interpolation) on the destination computer. The linear B-scan image size is 192×584 pixels and the B-mode image size is 640×480 pixels after conversion. We use an 8-bit data depth and therefore the image size is $109.5KB$ before conversion vs. $300KB$ after conversion. Therefore, this method saves 63.5% in network traffic. The net data rate is approximately $4.97MBps$ (over a $1Gbps$ network) without considering the *OpenIGTLink* protocol overhead.

It is shown that 3D volumetric processing after 3D reconstruction, results in better quality 3D ultrasound images compared to 2D image processing techniques applied before reconstruction [30]. However, 3D volumetric processing is very time consuming and computationally intensive. Fast 2D image processing is performed on each captured frame to reduce noise and speckles by replacing each pixel by the maximum of mean and median in a region around that pixel [31]. The processing also includes a simple segmentation algorithm consisting of adaptive thresholding. Adding the information of each image slice to the volume is performed based on the angle of the probe at each step. The compounding of 2D pixels into 3D voxels is then performed using the weighted accumulation compounding technique developed by Gobbi et al. [32]. The 3D ultrasound image capturing is performed at a rate of 1.5 volumes per second which is higher than the beating rate of the dynamic heart phantom (i.e., $0.5 Hz$ or less).

3.5.1 Stereoscopic visualization

The 3D scene (augmented with the representation of a catheter as a virtual tool) can be visualized using VTK in order to provide better depth perception and 3D understanding of the scene to the user. The user can activate the stereoscopic view in order to see the scene using stereoscopic goggles while he/she manipulates the catheter through a haptic device. The coordination of the haptic and visual interface provides an intuitive way of controlling the distal tip of the catheter.

3.6 Master-slave control

A unilateral master-slave control scheme has been developed. The *OpenIGTLink* [28] protocol [33] is used for real-time communication over TCP/IP. Figure. 3.9 shows the master and slave user interfaces. The master interface includes a graphical user interface which embeds the 3D visualization as well. It also includes a haptic interface through which the user controls the motion of the catheter tip. The graphical user interface on the slave side includes tools to initialize and control the robot manipulator and catheter actuation mechanism. The graphical user interfaces have been developed using *KWWidgets*.

The slave robot is controlled asynchronously at an average rate of $1.5KHz$. The real-time communication between the master and slave computers is performed at a rate of $1KHz$ synchronously. A PHANTOM® Desktop^{TM} haptic device is used as the master device, connected to the master computer through an Enhanced Parallel Port (EPP) interface. The haptic device is controlled asynchronously at an average rate of $1KHz$ using the OpenHaptics [34]

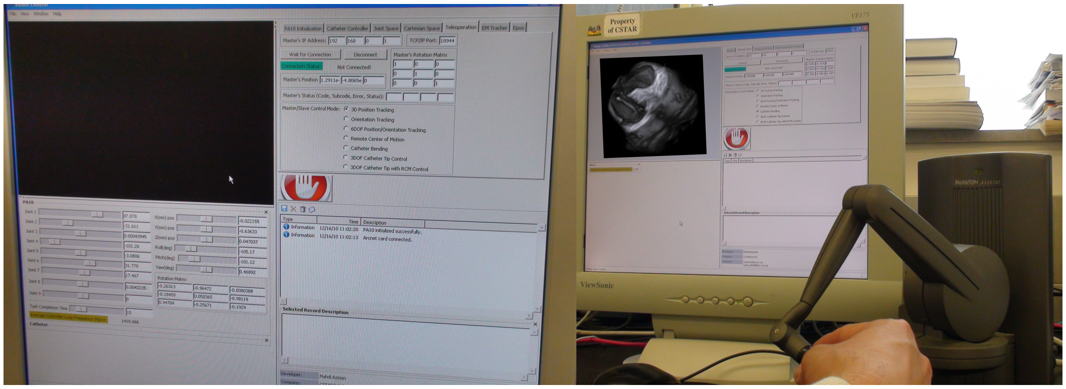

Figure 3.9: Left: Slave graphical user interface, Right: Master graphical user interface and the haptic device.

API (SensAble Technologies, Inc., Wilmington, MA). The interconnections between various parts of the system are shown in Figure 3.3.

3.7 Results

We have tested the developed system on a realistic beating heart phantom. The results are illustrated in Figure 3.10 where an ultrasound image slice as well as different views of the reconstructed volume are shown. The ablation catheter is clearly visible in the reconstructed volume. The catheter can also be visualized as a virtual tool using its coordinates captured through the magnetic trackers mounted on the catheter; This can be performed at a higher rate (up to 30Hz).

The volume capturing is performed at $1.5Hz$ which is more than three times the beating rate of the dynamic heart phantom (i.e., $0.5Hz$ or less). Although the sampling rate is enough based on the Nyquist criterion, there will be some motion artifacts in the 3D reconstructed volumes. There are two methods to reduce these artifacts: (a) using ECG gating [35] to overcome the artifacts caused by a beating heart, and (b) using 4D ultrasound probes with a higher capturing rate. Using ECG gating reduces the motion artifacts significantly but the overall reconstruction rate will be much slower (approximately

1-2 volumes per minute) when ECG gating is involved. This method is also sensitive to cardiac arrhythmia. Higher capturing rates (close to 30 volumes per second) can be provided using non-mechanical 4D ultrasound probes [36].

3.8 Conclusion

We have applied intraoperative transthoracic echocardiography for cardiac ablation applications. We have developed 4D ultrasound image reconstruction that can be used to visualize the heart chambers as well as ablation catheters. Magnetic trackers have been used to register the coordinates of the ultrasound probe as well as track the distal end of the catheter. We also developed a master-slave control scheme with robot-assisted actuation of a steerable catheter. We expect this method to improve the cardiac ablation procedure by providing intraoperative 3D ultrasound-based visual feedback coordinated with a haptic device at a master console. Changes in the condition and motion of the patient will have minimal effect whereas in the case of using preoperative images, registration should be updated if the patient moves. The accuracy of catheter steering will also be improved by providing a stereoscopic (3D) view of the catheter with respect to the heart along with master-slave control of the ablation catheter. The contributions of this work include: (a) application of transthoracic intraoperative ultrasound for catheter guidance in cardiac ablation; (b) use of intraoperative 3D ultrasound instead of preoperative CT/MR images for (stereoscopic) visualization of the heart; and (c) robot-assisted actuation of steerable ablation catheters using a master-slave control scheme.

Bibliography

[1] Shoei K. Stephen Huang and Mark A. Wood, editors. *Catheter Ablation of Cardiac Arrhythmias.* Saunders, 2006.

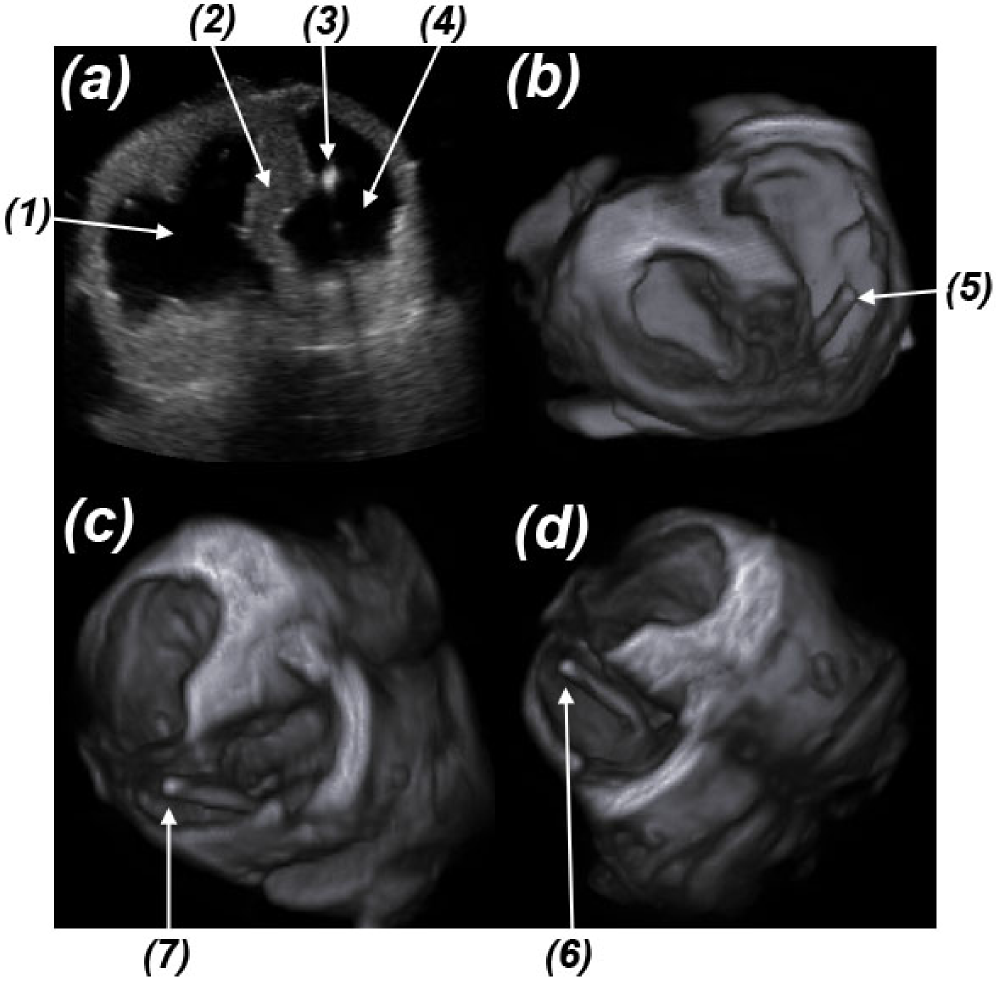

Figure 3.10: Ultrasound Imaging and 3D Reconstruction of a Heart Phantom with a 7-Fr Ablation Catheter: (a) a single slice of ultrasound images; (b,c,d) 3D reconstruction represented from different points of view. The different parts shown in this image are: (1) Left ventricle, (2) Interventricular Septum, (3) Ablation catheter, (4) Right Ventricle and (5,6,7) the ablation catheter from different points of view.

[2] David Keane. New catheter ablation techniques for the treatment of cardiac arrhythmias. *Cardiac Electrophysiology Review*, 6:341–348, 2002.

[3] Dipen C Shah, Hendrik Lambert, Hiroshi Nakagawa, Arne Langenkamp, Nicolas Aeby, and Giovanni Leo. Area Under the Real-Time Contact Force Curve (Force-Time Integral) Predicts Radiofrequency Lesion Size in an In Vitro Contractile Model. *Journal of cardiovascular electrophysiology*, pages 1038–1043, March 2010.

[4] Peter M Kistler, Kim Rajappan, Mohammed Jahngir, Mark J Earley, Stuart Harris, Dominic Abrams, Dhiraj Gupta, Reginald Liew, Stephen Ellis, Simon C Sporton, and Richard J Schilling. The impact of CT image integration into an electroanatomic mapping system on clinical outcomes of catheter ablation of atrial fibrillation. *Journal of cardiovascular electrophysiology*, 17(10):1093–101, October 2006.

[5] Brian D Powell and Douglas L Packer. Does image integration improve atrial fibrillation ablation outcomes, or are other aspects of the ablation the key to success? *Europace : European pacing, arrhythmias, and cardiac electrophysiology*, 11(8):973–4, August 2009.

[6] Hansen medical's sensei robotic catheter system. `http://www.hansenmedical.com/sensei`.

[7] Stereotaxis's Niobe, Magnetic Catheter Navigation System. `http://www.stereotaxis.com/niobe.html`.

[8] A. Amin, J. Grossman, and P. Wang. Early experience with a computerized robotically controlled catheter system. *Journal of Interventional Cardiac Electrophysiology*, 12:199–202, 2005.

[9] N. Marrouche, J. Brachmann, O. Wazni, V. Shibgilla, S. Beheiri, J. Guenther, and A. Natale. Preliminary human experience using a novel robotic catheter remote control. *Heart Rhythm*, 2:S63, 2005.

[10] Prapa Kanagaratnam, Michael Koa-Wing, Daniel T Wallace, Alex S Goldenberg, Nicholas S Peters, and D Wyn Davies. Experience of robotic catheter ablation in humans using a novel remotely steerable catheter sheath. *Journal of Interventional Cardiac Electrophysiology*, 21(1):19–26, January 2008.

[11] Luigi Di Biase, Tamer S Fahmy, Dimpi Patel, Rong Bai, Kenneth Civello, Oussama M Wazni, Mohamed Kanj, Claude S Elayi, Chi Keong Ching, Mohamed Khan, Lucie Popova, Robert a Schweikert, Jennifer E Cummings, J David Burkhardt, David O Martin, Mandeep Bhargava, Thomas Dresing, Walid Saliba, Mauricio Arruda, and Andrea Natale. Remote magnetic navigation: human experience in pulmonary vein ablation. *Journal of the American College of Cardiology*, 50(9):868–74, August 2007.

[12] Aravinda Thiagalingam, Andre D'Avila, Lori Foley, J. Luis Guerrero, Hendrik Lambert, Giovanni Leo, Jeremy N. Ruskin, and Vivek Y. Reddy. Importance of catheter contact force during irrigated radiofrequency ablation: Evaluation in a porcine ex vivo model using a force-sensing catheter. *Journal of Cardiovascular Electrophysiology*, (4):1–6, 2009.

[13] C. Weiss, M. Antz, O. Eick, K. Eshagzaiy, T. Meinertz, and S. Willems. Radiofrequency catheter ablation using cooled electrodes: Impact of irrigation flow rate and catheter contact pressure on lesion dimensions. *Pacing and Clinical*, 25(4 Pt 1):463–469, April 2002.

[14] M Tanimotol, F Arail, T Fukuda, H Iwata, K Itoigawa, Y Gotohx M Hashimoto, and M Negoro. Micro force sensor for intravascular neurosurgery and in vivo experiment. *The Eleventh Annual International Workshop on Micro Electro Mechanical Systems*, pages 504–509, 1998.

[15] Tobias Schaeffter, Lakmal Seneviratne, and Kaspar Althoefer. A fibreoptic catheter-tip force sensor with MRI compatibility: A feasibility study. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1501–054, 2009.

[16] Pinyo Puangmali, Tobias Schaeffter, and Reza Razavi. Novel miniature MRI-compatible fiber-optic force sensor for cardiac catheterization procedures. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2598–2603, 2010.

[17] Katsuaki Yokoyama, Hiroshi Nakagawa, Dipen C Shah, Hendrik Lambert, Giovanni Leo, Nicolas Aeby, Atsushi Ikeda, Jan V Pitha, Tushar Sharma, Ralph Lazzara, and Warren M Jackman. Novel contact force sensor incorporated in irrigated radiofrequency ablation catheter predicts lesion size and incidence of steam pop and thrombus. *Circulation. Arrhythmia and electrophysiology*, 1(5):354–362, December 2008.

[18] Hui Zhang, Kevin Cleary, Filip Banovac, and Amy White. Freehand 3D ultrasound calibration using an electromagnetically tracked needle. *Proceedings of SPIE*, 6141:61412M:1–12, 2006.

[19] P.W. Hsu, R.W. Prager, A.H. Gee, and G.M. Treece. Freehand 3D ultrasound calibration: a review. *Advanced Imaging in Biology and Medicine*, (December):47–84, 2007.

[20] B.A. Jones and I.D. Walker. Kinematics for multisection continuum robots. *IEEE Transactions on Robotics*, 22(1):43 – 55, 2006.

[21] E. Tatlicioglu, I.D. Walker, and D.M. Dawson. New dynamic models for planar extensible continuum robot manipulators. *Magnetic Resonance in Medicine, Wiley InterScience*, 55(5):1454–1459, May 2007.

[22] Jagadeesan Jayender, Rajni Patel, Gregory Michaud, and Nobuhiko Hata. Optimal transseptal puncture location for robot-assisted left atrial catheter ablation. In *Medical Image Computing and Computer-Assisted Intervention MICCAI 2009*, volume 5761 of *Lecture Notes in Computer Science*, pages 1–8. Springer Berlin / Heidelberg, 2009.

[23] Yusof Ganji, Farrokh Janabi-Sharifi, and Asim Cheema. Robot-assisted catheter manipulation for intracardiac navigation. *International Journal of Computer Assisted Radiology and Surgery*, 4:307–315, 2009.

[24] Y. Ganji and F. Janabi-Sharifi. Kinematic characterization of a cardiac ablation catheter. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1876 –1881, 2007.

[25] Y. Ganji and F. Janabi-Sharifi. Catheter kinematics for intracardiac navigation. *IEEE Transactions on Biomedical Engineering*, 56(3):621 –632, 2009.

[26] John J. Craig. *Introduction to Robotics: Mechanics and Control*. Pearson Education, 3 edition, 2005.

[27] Porta SDK. `http://www.ultrasonix.com/wikisonix/index.php?title=Porta`.

[28] OpenIGT Link Protocol. `http://www.na-mic.org/Wiki/index.php/OpenIGTLink/Protocol`.

[29] Pando SDK. `http://www.ultrasonix.com/wikisonix/index.php?title=Pando`.

[30] Flemming Forsberg, Vincenzo Berghella, Daniel A. Merton, Keith Rychlak, Joann Meiers, and Barry B. Goldberg. Comparing image processing techniques for improved 3-dimensional ultrasound imaging. *Journal of Ultrasound in Medicine*, 29(4):615–619, 2010.

[31] K. Thangavel, R. Manavalan, and I. Laurence Aroquiaraj. Removal of speckle noise from ultrasound medical image based on special filters: Comparative study. *ICGST International Journal on Graphics, Vision and Image Processing*, 9:25–32, 2009.

[32] David Gobbi and Terry Peters. Interactive intra-operative 3D ultrasound reconstruction and visualization. In Takeyoshi Dohi and Ron Kikinis, editors, *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 2489 of *Lecture Notes in Computer Science*, pages 156–163. Springer Berlin / Heidelberg, 2002.

[33] Junichi Tokuda, G.S. Fischer, Xenophon Papademetris, Ziv Yaniv, L. Ibanez, Patrick Cheng, Haiying Liu, Jack Blevins, Jumpei Arata, A.J. Golby, and Others. OpenIGTLink: an open network protocol for image-guided therapy environment. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 5(4):423–434, 2009.

[34] OpenHaptics Toolkit, Sensable Technologies. `http://www.sensable.com/products-openhaptics-toolkit.htm`.

[35] D. Pace, D. Gobbi, C. Wedlake, J. Gumprecht, J. Boisvert, J. Tokuda, N. Hata, and T. Peters. An open-source real-time ultrasound reconstruction system for four-dimensional imaging of moving organs. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2009.

[36] Shelten G. Yuen, Nikolay V. Vasilyev, Pedro J. del Nido, and Robert D. Howe. Robotic Tissue Tracking for Beating Heart Mitral Valve Surgery. *Medical Image Analysis*, 5528:97–103, July 2010.

Chapter 4

Patch Clamping

An image-guided technique is presented that facilitates multiple electrode patch clamping. In conventional methods of patch clamp electrophysiology, placement of electrodes is a time-consuming and complicated task due to the lack of depth perception, small depth of field, limited field of view and the possibility of collision between micropipettes. Microscope image processing is used to partially avoid these limitations. In this chapter, image processing algorithms are applied for system calibration and to perform autofocusing, relative depth estimation, distance estimation, tracking of the micropipettes in the images without making any major changes in the existing patch clamp equipment, and detection of lens/water contact. An autofocusing algorithm was developed and the relative depth estimation was performed based on autofocusing. A micropipette tip detection algorithm was also developed which could be used to initialize or reset a tracking algorithm. An image-based tracking algorithm was developed to track a micropipette tip in real time. A combination of these image processing techniques is used to assist the user to focus and find target cells in a tissue and locate multiple micropipettes close to the target neurons. Once this is done, the user can move the micropipettes through a master-slave control scheme in order to perform patch clamping.

4.1 Introduction

Patch clamp electrophysiology is a technique that permits the study of single or multiple ion channels in cells. The technique can be applied to a wide variety of cells, but is especially useful in the study of excitable cells such as neurons and cardiomyocytes [1]. To do patch clamp recordings glass micropipettes having an open tip diameter of about $0.2\mu m$, and outside tip diameter of about $0.4\mu m$ are used to electrically isolate a membrane surface area or patch which may contain as few as one or two ion channel molecules. The micropipette is pressed against a cell membrane and suction is applied to assist in the formation of a high resistance seal between the glass and the cell membrane (a *GigaOhm* ($G\Omega$) seal or *Gigaseal* since the electrical resistance of that seal should be more than a $G\Omega$). The high resistance of this seal makes it possible to electronically isolate the currents measured across the membrane patch with little competing noise thereby resulting in a high signal to noise ratio. It also provides some mechanical stability to the recording. The interior of the micropipette is filled with a solution matching the ionic composition of the interior of the cell. A differential amplifier is used along with a bath electrode, which allows a researcher to keep the voltage constant while observing changes in current. Alternatively, the cell can be current clamped in whole-cell mode, keeping the current constant while observing changes in the membrane voltage.

Neher and Sakmann first introduced the patch clamping technique in 1976 [2]. Their work received the *Nobel Prize in Physiology or Medicine* in 1991. While this technique has been employed for more than twenty years it is a difficult and time consuming process, requiring the use of micromanipulators having high precision ($< 0.2\mu m$), no vibration and no drift over long ($>$ 2hours) periods of time. It is also difficult to train researchers to perform patch clamping experiments and hence the data throughput is extremely low [3]. Even for someone with patch clamping experience, patch clamping is a very time consuming and tedious task. These difficulties become significant when more than one simultaneous recording is attempted. Only a handful of laboratories routinely do more than one recording at a time. While recordings of three or more at a time are done by only one or two laboratories in the world. Some of these difficulties include the following: (a) Each micromanipulator is moved individually and there is no coordination between the micromanipulators; (b) the microscope is moved independently of the micromanipulators; on the other hand a high level of magnification is required and therefore the depth of field and field of view are very limited. This makes it very difficult for the user to follow the micromanipulator movement when it is out of the field of view; (c) focusing needs to be done manually and it is like a blind search for an object before it comes close to the field of view; (d) there is a high possibility of collision between micro-pipettes or a micro-pipette and the objective or the substrate; (e) it is very difficult to relocate a micro-pipette tip at its previous location after it is changed. It is usually required to change a micropipette frequently during the procedure.

There are several automated patch clamp tools in the market today but almost all of them use a different approach than conventional patch clamping. Farre et al. [3] have listed most of the automated patch clamping systems available in the market. Most can only record from cells that approach the micropipettes instead of the conventional way of approaching a cell by a micropipette. Therefore, this approach cannot be used to monitor the electrophysiological activity of cells located in tissue (such as a brain slice). Instead, this approach is generally used by the pharmaceutical industry to screen for drugs that affect ion channel activity [3], [4]. To the best of our knowledge, there is no system available for automated/semi-automated multichannel patch clamping which could be used on living cells in tissue, i.e., approaching cells by multiple micropipettes.

The problem under study can be viewed as *cell micromanipulation* with special considerations. Sun and Nelson [5] have studied different techniques of cell manipulation including (a) optical trapping, (b) dielectrophoresis(DEP), and (c) mechanical micromanipulation. The first two methods can only be applied to isolated cells. The third method has the potential to be applied for cells in tissue. Mechanical micromanipulation has several applications including measuring mechanical properties of cells [6], [7] and microinjection [8], [9].

Unless otherwise specified, hereafter in this chapter, cell micromanipulation refers to mechanical micromanipulation of cells.

Sun and Nelson [10], [5] have developed an automated system for cell injection. They have used a 3-DOF micromanipulator to approach a mouse embryo cell (average $55\mu m$ diameter) and inject DNA into the cell while another micropipette is used to hold the embryo cell in place. They have used a generalized *Hough* transform to detect the nuclei of cells and a sum of squared differences for tracking the micropipette. An image-based visual servoing controller is used to bring the micropipette tip close to the nucleus and a hybrid position control is used to perform the automatic injection. An automatic focusing algorithm based on template matching has been used. Wang et al. [9, 11] have extended these techniques for automated pronuclear injection of zebrafish embryo ($> 400\mu m$ diameter) at a rate of 15 embryos per second.

Almost all of the techniques for cell micromanipulation have been developed and tested on isolated cells [5, 11, 12]. However, micromanipulation of live cells located in a tissue is far more challenging due to factors such as: (a) light/tissue interaction which causes image artifacts and images with low contrast and poor quality compared to isolated cell images; (b) tool/tissue interaction which causes motion of the cell even before the tool tip reaches the cell membrane; and (c) mechanical limitations posed by the fact that the tissue should be floating in a specially prepared oxygenated liquid to keep the cells alive. Therefore, simple image processing techniques used for detection and tracking of micropipettes, segmentation of cells or nuclei and automation using image-based visual servoing cannot be applied to the case of cell micromanipulation in tissue.

In contrast to techniques used for automating cell micromanipulation, it would be ideal to develop techniques to help a user perform multi-electrode patch clamping more easily and quickly while making minimal changes to the existing equipment. We have developed an approach that provides microscope image processing tools and techniques in order to make patch clamping faster and easier. This approach can be applied to existing commercially available equipment used for patch clamping. We have developed a software platform which is able to control the position of the objective lenses as well as the tip positions of the micropipettes individually. This software platform makes it possible to automatically focus on different objects, move the manipulators or the microscope lens to desired positions selected by the user and avoid collisions between the micropipettes. It also provides a graphical user interface (GUI) with different control tools and the capability of visualization of the process. The software also makes it possible to use a haptic device to move any of the micromanipulators or the microscope lens through master-slave control. The haptic device exerts forces on the user's hand, using the concept of artificial potential fields to generate forces for the collision avoidance algorithm and act when the micropipette comes close to an obstacle (which can be another micropipette).

In this part of the thesis, we describe the image processing techniques developed for this system which include the autofocusing algorithm and micropipette tip detection and tracking methods. This is part of the overall project which also includes microrobot control, collision avoidance, master-slave control and haptic feedback. The outline of the rest of this chapter is as follows: The experimental setup is described in Section 4.2. In Section 4.3, the autofocusing algorithm is discussed. Calibration of the system is briefly presented in Section 4.4, while Section 4.5 contains algorithms for detection and tracking of micropipette tips. Section 4.6 presents an algorithm for detection of lens-water contact. Section 4.7 describes the visual servoing control of micropipette tip positions. The collision avoidance technique is presented in Section 4.8. Section 4.9 describes the haptic-enabled master-slave control and Section 4.10 presents an online algorithm for partial 3D reconstruction and visualization. The architecture of the developed software is described in Section 4.11. Experimental results are given in Section 4.12 and finally, Section 4.13 concludes the chapter.

4.2 Experimental Setup

Figure 4.1 shows the experimental setup including the microscope, lenses, camera, micromanipulators and other equipment. Appendix D lists the equipment used in the system with various associated features. An objective lens changer was designed for automatic changing of the lenses [13], more details can be found in Appendix E.

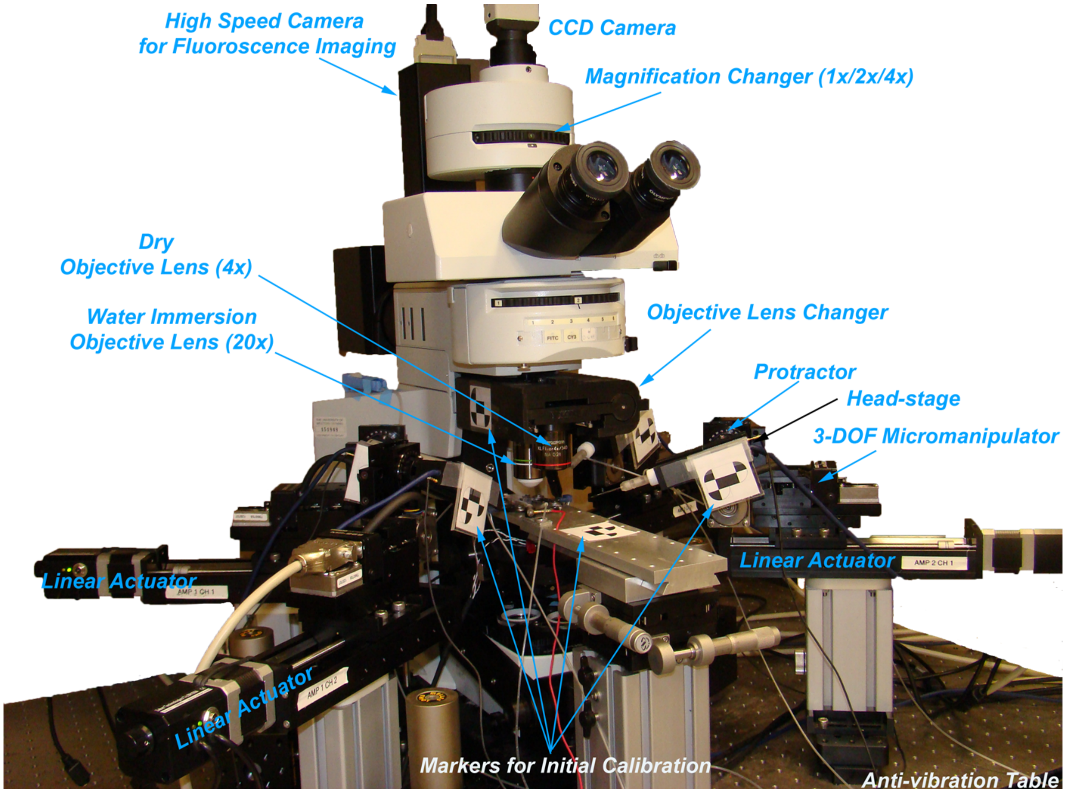

Figure 4.1: The experimental setup: all the equipment is installed on an anti-vibration table and located in a Faraday cage. The high-speed camera is used for voltage-sensitive fluorescent dye imaging and the image frames captured by the CCD camera are used for visualization and processing purposes in patch clamping. The head-stage is a signal conditioner that holds the micropipette. The linear actuators (M4(1..4) in Appendix D) are used to move the 3-DOF micromanipulators (M3(1..4) in Appendix D) *in* and *out* to facilitate replacing micropipettes.

4.3 Autofocus (AF) Algorithm

Autofocus (AF) is the process of automatically adjusting the camera lens position to obtain maximum focus of a scene. Focus is defined in terms of sharpness, the sharpness of the image is changed by changing the focal length. The goal of autofocusing is to find a focal length where the obtained image has maximum sharpness. Image processing techniques are applied to estimate a focus measure of the acquired image which is maximum when the image is in focus.

There are different focus measures which can be used to evaluate an image or a region of interest (ROI) in an image in terms of being focused or being out of focus. A good focus measure should have its maximum value when the image is in focus, it should also be computationally inexpensive and robust to noise and illumination changes. It is desired to have a unimodal focus measure which has a single global maximum rather than multiple local maxima. Yap et al. [14] have developed a method based on discrete orthogonal Chebyshev moments of the image, which is robust to both additive Gaussian noise and changes in illumination, but it is computationally intensive. Riaz et al. [15] have used generalized Laplacian as a focus measure for depth estimation and 3D shape recovery. Shih has compared several different methods for autofocusing by their unimodality and accuracy [16]. Sun et al. [17] have studied 18 different AF measures with different observation methods. Our goal is not to find an optimal AF measure with the best accuracy; we are looking for a computationally efficient AF measure with reasonable precision (repeatability) and unimodality (factors not studied by Sun et al. [17]). Therefore, we have studied different AF measures in order to find the optimal solution for our application.

Two kind of focus measures should be defined for this application: (a) for the whole image or a region of interest (ROI) in the image of a tissue slice and (b) for the tip of a linear object (i.e., micropipette). The reason for choosing a special focus measure for a micropipette tip is that applying the general focus measure to a region around the micropipette tip may not be a good option when the micropipette tip is close to another object. In such cases, the maximum focus measure may occur when the other object is in focus and the software may fail to focus on the micropipette tip. Among several possible focus measures, we study those which are computationally more efficient (i.e., we choose methods based on filters with simple kernels like Sobel). For a $M \times N$ image I, different focus measures are defined in Appendix. I.

We evaluate different focus indices based on the following criteria: unimodality, precision, sensitivity to illumination changes, and computational efficiency of the algorithm. In most of the cases, we are not dealing with flat objects parallel to the focal plane, so it might not be possible to determine the accuracy by comparing the results to what a human user specifies. Therefore the precision (repeatability) of each focus measure is studied. The focus index is measured for an object with the objective lens position varying in a range of $\pm 100\mu m$. The results are shown in Table 4.1. The peak detected by the *Laplacians* $\mathcal{L}_1, \mathcal{L}_3, \mathcal{L}_5$ is slightly (around $4\mu m$) different from that detected by the *Gradients*[1] and *Laplacian* $\mathcal{L}_7$.

Method	$T_d(ms)$	$\mathcal{U}$	$\sigma(\mu m)$	$\mathcal{R}(\mu m)$	$\mathcal{S}_I(\mu m)$
$\mathcal{L}_1$	3.4430	33	1.540	5.4	2.936
$\mathcal{L}_3$	3.4483	12	1.587	4.8	0.404
$\mathcal{L}_5$	4.2155	8	1.184	3.5	1.353
$\mathcal{L}_7$	13.1246	4	1.318	4.0	0.723
$\mathcal{C}$	4.2234	4	2.019	6.3	1.447
$\mathcal{S}_3$	4.1962	4	2.018	6.3	1.447
$\mathcal{S}_5$	6.3077	4	1.445	4.4	0.7095
$\mathcal{S}_7$	7.9062	4	1.852	5.5	1.922
$\mathcal{M}$	4.7983	4	1.663	4.7	1.531

Table 4.1: Comparing different focus measures for the dry objective.

In Table 4.1, the left part shows the normalized AF measure vs. lens position within $\pm 100\mu m$ and the right part represents information to compare different AF measures based on their properties; The first column represents the

---

[1] Scharr, Sobel and Morphological

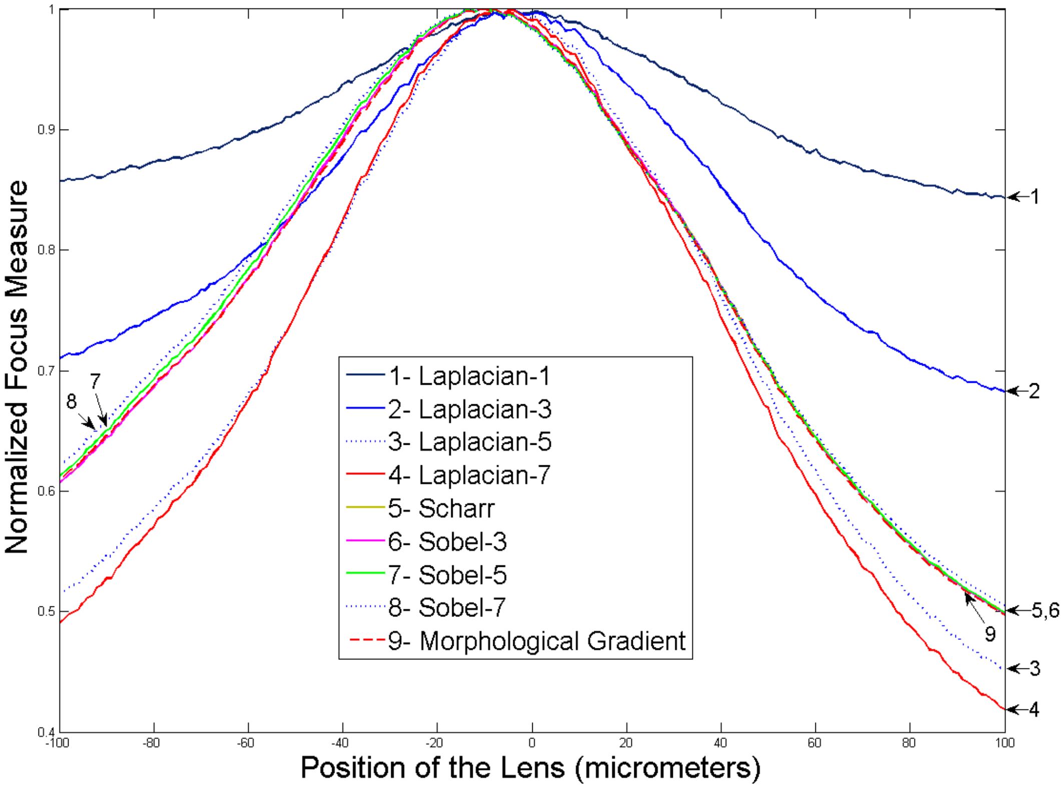

Figure 4.2: Normalized autofocus measure vs. vertical lens position.

autofocusing method, the second column shows $T_d(ms)$ which is the average evaluation time measured by running each method for 1000 times and taking the average time it takes to run[2]. The third column represents the unimodality index $\mathcal{U}$ which is the number of local maxima for each autofocus measure [16]. Less $\mathcal{U}$ means better unimodality. The fourth and fifth columns show $\sigma(\mu m)$ and range(μm); Each autofocus measure is being evaluated 100-times and the location of maximum autofocus is detected each time. The standard deviation ($\sigma(\mu m)$) of these locations shows how precise the method could be and how repeatable the results are. We have also recorded the *range*(μm) in which the maximum focus is being found. Each method is being evaluated under three different illumination intensities (low, medium and high), the location of maximum sharpness is detected under each condition and standard deviation of these locations are evaluated as $\mathcal{S}_I$ and shown in the last column.

To choose the best AF measure based on the data in Table 4.1, we first sort them based on $\mathcal{U}$ because unimodality is very important in achieving the global maximum of the AF; $\mathcal{L}_1$, $\mathcal{L}_3$ and $\mathcal{L}_5$ are not good choices based on this factor. Among the rest of the AF measures, we choose those with smaller σ and $\mathcal{S}_I$ to have better precision and less sensitivity to illumination changes: i.e., $\mathcal{L}_7$ and $\mathcal{S}_5$. Although $\mathcal{L}_7$ has slightly better precision, but it is more computationally intensive and more sensitive to illumination changes compared to $\mathcal{S}_5$. We choose $\mathcal{S}_5$ based on this comparison, it takes less than half computation time while its precision is only slightly less compared to $\mathcal{L}_7$.

These results have been obtained under a $4X$ dry objective and the precision of each AF index will be much better (i.e., in sub-micron range) using a $20X$ water-immersion objective lens which provides a smaller pixel size. The same method is applied to the water immersion objective.

4.3.1 Focusing on a micropipette tip

Autofocusing on a micropipette tip is different from autofocusing on an image or a rectangular region of interest. It can be performed in two different modes

---

[2] Although absolute values of T_d might be different being run on a different computer or a different image size, but the relative values remain the same.

either by moving the microscope lens to focus on a micropipette tip or by moving a micropipette to bring it to the focal plane. The autofocus algorithm is the same in both modes, the only difference is the object which is moved to achieve the focusing objective. Subbarao et al. [18] have shown that the optimal focusing measure might be different for different objects. We have modified the AF measure to obtain a better performance in autofocusing on a micropipette tip. First, the general AF algorithm is performed on a ROI around the micropipette tip; Then the micropipette tip position and its central axis direction is being estimated as described in Section. 4.5; Then the AF algorithm is applied again, but with a different AF measure which uses the directional gradient of the image in a ROI around the micropipette tip and along the direction orthogonal to the micropipette central axis.

If $\begin{bmatrix} u_x & u_y \end{bmatrix}^T$ is a unit vector along the central axis of a micropipette, then $\begin{bmatrix} -u_y & u_x \end{bmatrix}^T$ points in the orthogonal direction. Then the new AF measure will be defined as:

$$\Phi_D(I) = \frac{1}{M_w N_w} \sum_{(i,j)\in\mathcal{W}} \left| \left\langle \begin{bmatrix} (S_x * I)_{i,j} \\ (S_y * I)_{i,j} \end{bmatrix}, \begin{bmatrix} -u_y \\ u_x \end{bmatrix} \right\rangle \right| \tag{4.1}$$

where M_w and N_w are the ROI window width and height in pixels where we take $M_w = N_w = 64$ by default but can be modified by the user; $\mathcal{W}$ is a $M_w \times N_w$ window centered at (p_x, p_y) which is the approximate position of the tip; $\langle \cdot, \cdot \rangle$ is the inner product of two vectors and $|\cdot|$ is the absolute value operator. S_x and S_y are defined in Appendix. I, for various focus measures.

We have evaluated this focus measure for the micropipette tip 100-times and the standard deviation of this method is $2.680\mu m$ while $\mathcal{S}_5$ applied to the same ROI around the micropipette tip results in a standard deviation of $2.799\mu m$. Therefore, the directional gradient method has improved the AF measure for micropipette tips slightly.

The reason that the standard deviation of autofocusing on a micropipette tip is higher compared to the whole image is that we are processing a smaller area of the image but the pixel size is the same, this means that less information is being collected which results in increased uncertainty and less precision. If we apply a whole image AF to focus on a micropipette, the results will not be acceptable because the micropipette always makes an angle with the focal plane, so the AF will not focus on the tip but on an average of the whole length of the micropipette visible in the image.

4.3.2 Optimization

We define the focus measure as $\Phi_I(z)$ where z is the position of the lens in the direction orthogonal to the image plane or the position of a micropipette in the same direction when the lens is fixed. The index I is used to show the dependence of Φ on the image. The optimization problem is stated as:

$$\max_{z \in [z_l, z_u]} \Phi_I(z) \tag{4.2}$$

The autofocus (AF) approach is based on moving the lens or a micropipette up and down (orthogonal to the image plane) to find a place where the AF index is maximum. However it is very important to find the maximal point with the minimum amount of movement to keep the total time required for each autofocus as small as possible. To quantify this minimum amount of movement and incorporate it in the optimization procedure, consider the sequence of the evaluated points represented by: $z_0,\ z_1, \cdots, z_n$, where z_0 is the starting point and $z_n = z_{opt}$ is the final optimal point. The objective function is defined:

$$J(z_0, \cdots, z_n) = \sum_{k=1}^{n} |z_k - z_{k-1}| \tag{4.3}$$

The problem is then solved via the optimization problem in Equation. 4.2 while keeping J as low as possible. J is proportional to the total time it takes for the AF optimization algorithm to converge. In an ideal case, starting from z_0 and stopping at z_{opt} without changing the direction of motion, results in $J_{min} = |z_{opt} - z_0| + J_0$, where J_0 is a small distance for initial gradient estimation. The worst case would be a blind search, scanning the whole range of $[z_l, z_u]$ with the smallest possible step size, resulting in $J_{max} = \min(|z_0 - z_l|, |z_0 - z_u|) + |z_u - z_l|$. As we normally do not know which direction ends in the autofocus, usually a symmetric range of search is chosen, i.e., $z_0 = \frac{1}{2}(z_l + z_u)$. So we can simplify $J_{max} = \frac{3}{2}|z_u - z_l|$.

The steepest descent method could be the best option, but as shown in Table 4.1, each of the focus measures might have multiple local maxima and the steepest descent might become trapped in one of these local optima. Based on our experiments, all of the local maxima for the selected AF measure are located within $\pm 10\mu m$ of the global maximum for the selected AF measure.

We have used a mixed optimization method to keep J as low as possible while avoiding local maxima traps. This method is a combination of the steepest descent method and blind search. A steepest descent approach is used to find the maximum within a $\pm 10\mu m$ range and then a blind search is performed around this point. The steepest descent method provides a fast convergence while we avoid its zig-zag property and the local maxima traps in a neighborhood of the global maximum, by switching to blind search. Using this method, we obtain $J \leq (J_{min} + 30\mu m)$. Although we cannot guarantee that this results in the minimum possible value of J, it results in a method which is fast enough considering a black box optimization problem without *a priori* information about the objective function gradient.

4.4 System Calibration

System calibration includes estimation of critical system parameters including microscope lens and camera parameters, registration of relative coordinates of the micromanipulators and micropipettes with respect to a reference coordinate system and relative coordinates of external obstacles with respect to the reference coordinate system if applicable. Image processing algorithms are used for calibration and registration. A brief description of the calibration procedure as follows:

- Microscope camera calibration: The microscope is moved around on a random pattern and a micropipette tip is detected at each position. This information is used to find the pixel size and skewness of microscope images.

- Coarse registration using stereo vision: A stereo vision tracking system is used for detection and tracking of markers on the substrate, micromanipulators and objective lenses with a sub-millimeter accuracy. This is used to achieve initial coarse registration of different coordinate systems in order to bring the micropipettes into the field of view of the microscope.

- High accuracy registration using microscope images: The micropipette tips are moved around on a random pattern. A combination of autofocusing and tip detection algorithms is used to focus and detect the micropipette tip at each step. RANSAC (RANdom SAmple Consensus) and least-squares error optimization algorithms are used for rigid registration of each micropipette to the microscope coordinate frame [19].

- Calibration procedure: The system calibration can be performed fully autonomously or semi-autonomously. In the autonomous mode, there is no user interaction but in semi-autonomous mode, the user is responsible for bringing each micropipette tip into the field of view, moving the microrobots to different locations and clicking on specified micropipette tips.

System calibration involves estimation of critical system parameters including microscope lens and camera parameters, relative coordinates of the micromanipulators and micropipettes with respect to a reference coordinate system, and relative coordinates of external obstacles with respect to the reference coordinate system if applicable. Figure 4.3 illustrates different coordinate systems on a schematic of the setup.

- $\mathcal{C}^{ref}$ is the reference Cartesian coordinate system which specifies the task space.

- $\mathcal{C}^{mic}$ is the coordinate system attached to the microscope's micromanipulator and moves with it.
- $\mathcal{C}^{man_n}$ is the coordinate system attached to n^{th} micromanipulator and is fixed with respect to $\mathcal{C}^{ref}$. This coordinate system specifies the micromanipulator's joint space, i.e., its axes are along the micromanipulator's prismatic joints and the micropipette tip is at the origin when the robot is in home position.
- $\mathcal{C}^{pipette_n}$ is the coordinate system attached to the tool on the n^{th} micromanipulator, its origin is at the same point as $\mathcal{C}^{man_n}$ but its X-axis is along the micropipette's central axis.
- $\mathcal{C}^{image}$ is the coordinate system attached to the center of the image as shown in Figure 4.13. The focal plane is orthogonal to Z^{mic}.
- $\mathcal{C}^{haptic}$ is attached to the haptic device as shown in Figure 4.13.

In order to register different coordinate systems to the reference coordinate system, we define a number of homogenous transformation matrices. The transformation matrices are all 4 × 4 matrices. T^{ref}_{mic} is the transformation that maps $\mathcal{C}^{mic}$ to $\mathcal{C}^{ref}$. Therefore:

$$T^{ref}_{mic} = \begin{bmatrix} 1 & 0 & 0 & x_{mic} \\ 0 & 1 & 0 & y_{mic} \\ 0 & 0 & 1 & z_{mic} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

where $(x_{mic}, y_{mic}, z_{mic})$ is the position of the microscope's micromanipulator with respect to the reference. $\mathcal{C}^{mic}$ is identical to $\mathcal{C}^{ref}$ when the microscope's robot is in its initial position.

It could be assumed that all micromanipulators have parallel Z coordinates, i.e., $\forall n,\ Z^{man}_n || Z^{mic}$ [19]; however this may become a source of error because (a) large vertical displacements are involved, and (b) there may be machining imperfections. Without making such assumption, we use $T^{ref}_{man_n}$ as the transformation matrix which maps $\mathcal{C}^{man_n}$ to the $\mathcal{C}^{ref}$.

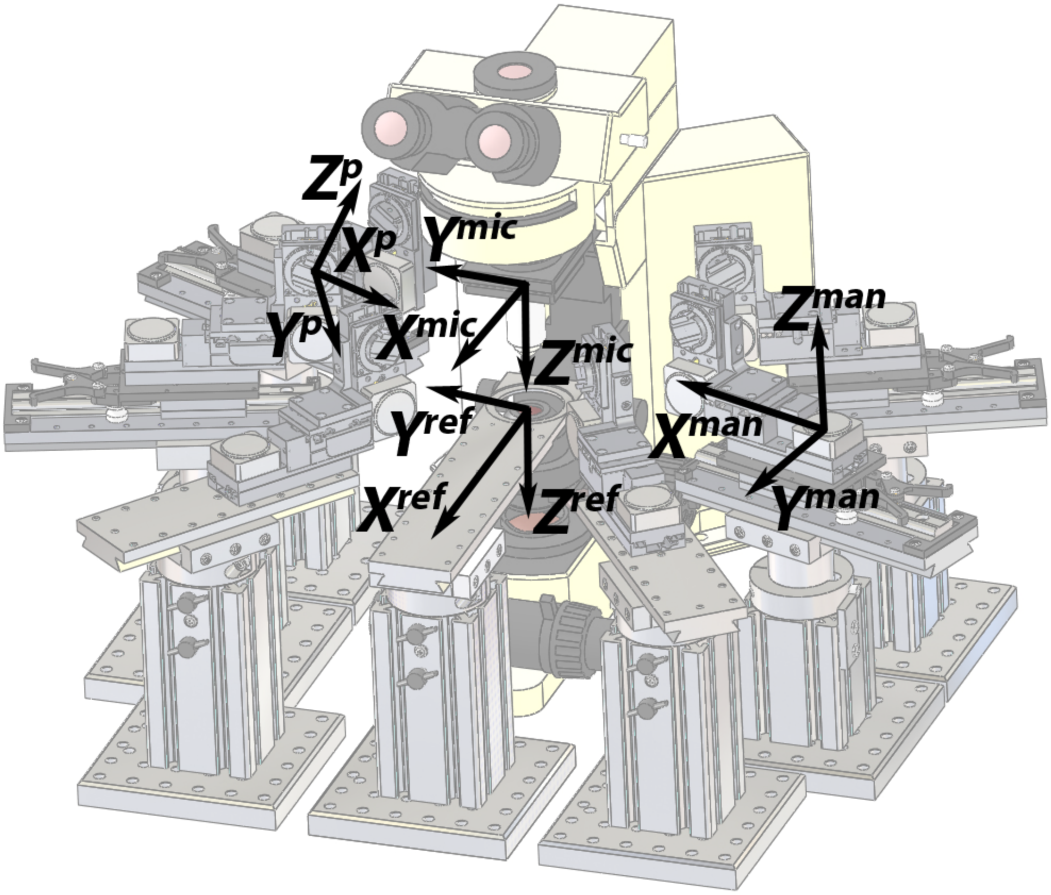

Figure 4.3: Illustration of the coordinate systems on a schematic of the setup: Cartesian coordinate systems $\mathcal{C}^{ref}$ and $\mathcal{C}^{mic}$ and coordinate system $\mathcal{C}^{man}$ for one of the micromanipulators and $\mathcal{C}^{pipette}(X^p, Y^p, Z^p)$ for another manipulator are shown. Background image courtesy of *Sutter Instruments*.

The image coordinate system $\mathcal{C}^{image}$ is attached to $\mathcal{C}^{mic}$, i.e., the center of the image lies on the origin of $\mathcal{C}^{mic}$ with parallel z-coordinates. Therefore, the mapping between the image and microscope coordinates can be stated as:

$$T^{mic^*}_{image} = \begin{bmatrix} s_x \cos\theta_I & -s_y \sin\theta_I & 0 & 0 \\ s_x \sin\theta_I & s_y \cos\theta_I & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

where we have used a simple camera model in which s_x and s_y determine the pixel size in x_I and y_I-directions, respectively; and θ_I is a rotation angle to map (x_I, y_I) to (x_{mic}, y_{mic}). This is defined and measured for the dry objective lens (4× magnification) when the microscope magnification is on 1× (default magnifications). A correction term is used for other objectives/magnifications. Therefore:

$$T^{mic^*}_{mic}(m_{objective}, m_{microscope}) = \begin{bmatrix} \alpha & 0 & 0 & \kappa_x \\ 0 & \alpha & 0 & \kappa_y \\ 0 & 0 & 1 & \kappa_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

where α is the effective magnification factor compared to the default magnification. Due to imperfect optics, we cannot use the nominal magnification factors to calculate α for each case. α and $\kappa_{x,y,z}$ are system dependent parameters that are functions of $m_{objective}$, $m_{microscope}$ which specify the objective and microscope magnification factors, respectively.

The micropipette coordinate system $\mathcal{C}^{pipette_n}$ has a fixed angle Θ_n around the Y-axis compared to $\mathcal{C}^{man_n}$. Therefore, $T^{man_n}_{pipette_n} = T_y(\Theta_n)$ which is a pure rotation without translation around the Y-axis. We also define the *image Jacobian* J_n for each robot. This is to relate displacements/velocities in $\mathcal{C}^{image}$ to $\mathcal{C}^{man_n}$.

$$\delta P^{man_n} = J_n \delta P^{image} \tag{4.7}$$

where δP^{man_n} is the displacement of an object in $\mathcal{C}^{man_n}$ and $\delta P^{image} = [\delta x_I \ \delta y_I \ 0]^T$ is the displacement of the same object in $\mathcal{C}^{image}$. The Jacobian matrix J_n is simply part of the transformation matrix $T^{man_n}_{image}$:

$$T^{man_n}_{image} = T^{man_n}_{ref} T^{ref}_{mic} T^{mic}_{mic^*} T^{mic^*}_{image} = \begin{bmatrix} J_n & D_n \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{4.8}$$

Therefore J_n is a function of the objective and microscope magnification levels.

4.4.1 Calibration Procedure

The system calibration consists of two major steps: (a) coarse calibration, and (b) fine (high-accuracy) calibration. The coarse calibration is performed using macro-scale stereo vision tracking, and the fine calibration is performed using the microscope images.

Coarse calibration

Coarse calibration is performed mainly to bring the micropipettes to the FOV of the microscope. The coarse calibration is not necessary if the user brings the micropipette tips into the FOV or if a previous calibration of the system is still valid and can be used to roughly locate the micropipette tips in the FOV. The overall process of coarse calibration is described here.

We have used a MicronTracker $S60$ optical tracking system (Claron Technologies, Toronto, ON, Canada) to find a coarse estimation of the position and orientation of the micromanipulators and the microscope up to a sub-millimeter accuracy. Black and white printed markers are attached to each part as shown in Figure 4.1. If the stereo camera is used only at a single position, all the markers may not be visible due to occlusions. So the stereo camera is put in at least two positions to be able to detect all of the markers. The camera is mounted on a passive adjustable manipulator and it can be easily relocated. The marker on the substrate is used as a reference to relate coordinates for the different poses of the camera. The transformation of each marker with respect to the substrate marker is calculated. This removes dependence on the location of the stereo camera. Θ_n can be easily estimated for each micromanipulator using this method.

Ideally it should be possible to calibrate the system once and then use the same calibration to roughly locate each micropipette tip in the FOV in order to perform a fine calibration. The problem however rises from the fact that all the micromanipulators use stepper motors with an open-loop control (without encoders). Therefore, if a part of the system is moved manually, then it is not possible to track the motion. The major problem is when the user tries to move the objectives up/down using the coarse focus knob shown in Figure 4.1. The other parameter estimated by the coarse calibration is the change in the vertical position of the objectives. It is assumed that the other parts of the system are not moved manually, so that the previously saved calibration of the system is updated in order to correct for any manual changes in the objective's height.

If the system has been reconfigured, e.g., there has been a change in the micromanipulators base position and/or orientation (which is very unlikely once the system is installed), or if there is no previously saved calibration, it is expected that the user brings all the micropipette tips to the FOV manually and performs a fine calibration.

High-accuracy calibration

Once the micropipettes are in the FOV of the microscope, the fine calibration can be performed fully autonomously or semi-autonomously. In autonomous mode, there is no user interaction but in semi-autonomous mode, the user is responsible for moving the micromanipulators to different locations and clicking at specified micropipette tips. The calibration algorithm is as follows:

1. **Camera Calibration**: estimating T_{image}^{mic} including estimation of the image pixel size and rotation with respect to the base.
    (a) Click on a fixed reference object in the FOV (i.e., the tip of an micropipette); In the autonomous mode, the tip of a specified micropipette is detected as described in [20].

   (b) Record the $3D$ location of the microscope's micromanipulator (P_M) and the $2D$ location of the point in the image (P_I).
   (c) Move the microscope to a different location in the same plane, such that the specified point is still in the FOV. In autonomous mode, this is a random motion in the $X^{mic} - Y^{mic}$ plane. The amount of displacement is generated by a pseudo-random number generator with uniform distribution and the range of displacement depends on the pixel-size, size of the image and the original location of the micropipette in the image.
   (d) Repeat items 1b to 1d at least 3 times (at least 3 of the selected points should not lie on a line, otherwise the problem would become singular).
   (e) Run the RANSAC (RANdom SAmple Consensus) algorithm with least-squares error estimation to find the calibration matrix (refer to Section 4.4.1 for details).

2. **Registration of micromanipulator Coordinate Systems**: registration of micromanipulator coordinates to a reference coordinate system, i.e., estimating $T^{ref}_{man_n}$:
   (a) Autofocus on the micropipette tip.
   (b) Click on the location of the tip of the n^{th} micropipette; In the autonomous mode, the tip is detected automatically as described in [20].
   (c) Record the $3D$ location of the microscope (P_M), the $3D$ location of the micromanipulator (P_R) and the $2D$ location of the point in the image (P_I).
   (d) Move the micromanipulator to a different location, such that the micropipette is still in the FOV and then perform AF to focus on the micropipette tip. In autonomous mode, this is a random motion. The amount of displacement is generated as described in the camera calibration step 1c.

   (e) Repeat items 2b to 2d at least 4 times (at least 4 of the selected points should not lie in a plane, otherwise the problem would become singular).

   (f) Run the RANSAC algorithm with least-squares error estimation to find the registration matrix.

3. **micropipette Calibration**: estimating $T_{pipette_n}^{man_n}$ is equivalent to estimating the angle Θ_n. The estimation of this angle can be simply provided by reading the protractor on the micromanipulator where it holds the headstage (as shown in Figure 4.1) or using the coarse calibration information. Although this estimation may not be very accurate, it is only used to perform coaxial movement along the micropipette axis: (a) to pull an micropipette in/out to avoid collisions with the substrate and objectives during replacements, and (b) to approach the cell membrane when the micropipette tip is in the vicinity of the cell. Small misalignments will not affect these procedures significantly.

Parameter Estimation

We have used the RANSAC algorithm to estimate the parameters of the registration matrices based on the measured data. The RANSAC algorithm can be described as follows [21]:

1. Select randomly the minimum number of points required to determine the model parameters (3 or 4).

2. Solve for the parameters of the model (i.e., using the least-squares error (LSE) optimization algorithm to estimate the parameters).

3. Determine how many points from the set of all points fit with a predefined tolerance.

4. If the fraction of the number of inliers over the total number of points in the set exceeds a predefined threshold, re-estimate the model parameters using all the identified inliers and terminate.

5. Otherwise, repeat steps 1 through 4 (maximum of N times).

We take u as the probability that any selected data point is an inlier and p the probability that at least one of the sets of random samples does not include an outlier. Then we can simply conclude that $1 - p = (1 - u^m)^N$ [22] where N is the number of iterations and m is the minimum number of samples required for estimation. Now if we choose $p = 0.99$ and $u = 0.9$ (which is considered as the worst case when 10% of the selected points are wrong), then we will have $N = \lceil \frac{log(1-p)}{log(1-u^m)} \rceil$. In case of 1, we have $m = 3$ and therefore $N = 4$ and in case of 2, we have $m = 4$ and therefore $N = 5$. In other words, the RANSAC algorithm should converge in 4 and 5 iterations for the camera calibration and micromanipulator registration cases, respectively.

4.5 Micropipette Tip Detection and Tracking

There are several occasions when we need to find a micropipette tip in a microscope image automatically; A major application would be automatic tip detection for replacing user clicks and obtain faster calibration. Figure 4.4 illustrates what a micropipette tip looks like under dry (4X) and water immersion (20X) objective lenses.

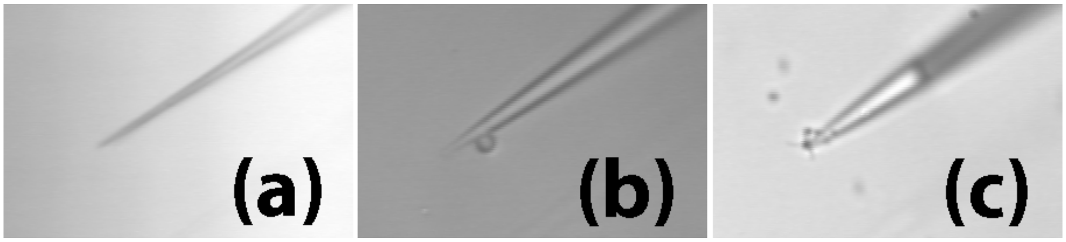

Figure 4.4: Micropipette tip under (a) dry and (b,c) water immersion objective lenses. In (b) the micropipette is filled with water, but in (c) only part of it is filled. There are tiny air bubbles visible in (b,c) and some undesired particles are visible in (c).

Wang et al. [8] have developed a pipette tip detection and tracking algorithm for contact detection in microrobotic cell manipulation, a $5\mu m$ tip micropipette has been used and a submicron accuracy has been reported. Sun et al. [5] have used a sum-of-squared-differences (SSD) optical flow tracking algorithm which is both sensitive to illumination changes and needs an accurate initialization. In our case a micropipette with sub-micron tip should be detected accurately and the tip detection should be robust to noise and illumination changes. We have developed different methods for the dry and water immersion objectives, the details will follow.

4.5.1 Detection under a dry objective

A novel approach has been developed to detect a micropipette tip under dry objective. The tip detection algorithm consists of four steps:

1. **Mask image construction**: In a microscope image, there might be several moving objects other than a micropipette; There might also be intensity changes which are not really associated with any moving object but are created by illumination changes or noise. It is important to recognize the pixels of the image associated with the specified micropipette. To achieve this goal, we define a *mask image* which is a binary image with white pixels on and around the specified micropipette and black pixels otherwise. To construct such a mask image, we use a spatio-temporal filtering technique. A micropipette is moved to within a small range of its current location in the lateral direction (orthogonal to the projection of micropipette's central axis in the image plane) with a fixed frequency $f_v(Hz)$. Images are captured at a rate of $30Hz$ and a band-pass temporal frequency filter is applied to the stream of images, pixel by pixel. The filter passes frequencies around $f_v(Hz)$ and rejects other frequencies. The pixels that pass this filter should specify the points on and around the specified micropipette, because it is the only object vibrating at this frequency. To choose an appropriate frequency f_v, there are different considerations to be taken into account:
    - Our sampling rate is $30Hz$ so we should have $f_v < 15.0$ to satisfy the Nyquist sampling theorem.

- The micromanipulators' maximum operation frequency is $12.57Hz$ due to a $79.5ms$ delay for execution of each command. So we should have $f_v < 12.57$.
- The frequency should be away from the frequency ranges of any mechanical vibrations or optical noises affecting the equipment and images.
- Higher frequencies result in shorter time of detection, but affect the life of the micromanipulator.

Considering all the above-mentioned limitations, we have chosen $f_v = 5.0Hz$ and we only run it for 5 cycles which takes $1.0sec$. This temporal filtering results in a gray-scale image I_n^g where n ($n \in \{1, 2, 3, 4\}$ in this case) is the index for the specified micropipette. This gray-scale image is then thresholded. The threshold level is adjusted to $\tau_n = max\left(\varsigma_n, \frac{1}{2}max(I_n^g)\right)$, where $\varsigma_n = \bar{I}_n^g + 2.5\sigma(I_n^g)$ determines the noise level, $\bar{I}_n^g$ is the average intensity of I_n^g and $\sigma(I_n^g)$ is the standard deviation of the pixel intensities in I_n^g. The result is a binary image I_n^b. Using ς helps to avoid false detection when there is no significant motion. When $\frac{1}{2}max(I_n^g) \leq \varsigma_n$, the threshold is set to ς_n which indicates that there is no significant motion with the specified frequency which usually means that micropipette is out of the field of view, totally out of focus or there is not enough illumination. In such cases, the user is asked to check if micropipette is visible and then repeat the procedure.

The resulting binary image I_n^b might have some pixel noise and also the detected region may have some gaps. A morphological opening and closing is then applied to the binary image I_n^b to remove pixel noise and fill in the gaps in the binary image [23].

$$I_n^m = (I_n^b \circ M_n^o) \bullet M_n^c \tag{4.9}$$

The result is a binary image I_n^m which is called the mask image. M_n^o, M_n^c are 5×5 cross shaped structuring elements for the morphological opening and closing operations.

2. **Orientation Detection**: To detect the orientation of the central axis of a micropipette, we use a technique based on the *Hough* transform [24]. In an arbitrary situation, two images are captured: $I_n(0, -\delta_y)$ which is the image captured when the micropipette is moved $-\delta y(\mu m)$ with respect to its original position in a direction orthogonal to its central axis parallel to focal plane and $I_n(0, \delta_y)$ which is captured when the micropipette is moved $\delta y(\mu m)$ in the reverse direction. After capturing these two images, we construct two new images by thresholding the differences of these images:

   $$\begin{cases} I_{n_{y-}} = \tau\left(I_n(0, -\delta_y) - I_n(0, +\delta_y)\right) \\ I_{n_{y+}} = \tau\left(I_n(0, +\delta_y) - I_n(0, -\delta_y)\right) \end{cases} \tag{4.10}$$

   A Progressive Probabilistic Hough Transform ($PPHT$) [25] is then applied to the resulting binary images $I_{n_{y-}}$ and $I_{n_{y+}}$ to detect the orientation of the micropipette. The orientation of the strongest line segment detected in each of the images is selected and the bisector of these line segments is determined as the orientation of the central axis of the micropipette as shown in Figure 4.5 (c) and (d). Due to the cone-like shape of the micropipette, the two line segments detected in $I_{n_{y-}}$ and $I_{n_{y+}}$ will not be parallel, we use this fact to find the direction that points towards the micropipette tip.

3. **Tip proximity detection**: To detect the proximity of the tip, we use a similar approach as in the previous step. We obtain two images $I_n(0, 0)$ and $I_n(-\delta x, 0)$ representing the images captured when the micropipette is in its original position and when it is moved back $\delta x(\mu m)$ along its central axis, respectively. δx is selected based on the pixel size of the image. Then we calculate the absolute value of intensity difference between these two images and mask it by I_n^m to make sure it is only showing pixels in the proximity of the n^{th} micropipette tip. Then we threshold the

result to obtain a binary image I_{n_x}:

$$I_{n_x} = \tau \left(|I_n(0,0) - I_n(-\delta x, 0)|.I_n^m \right) \tag{4.11}$$

Then a morphological opening and closing is applied to I_{n_x}. The result is a binary image which is white in a small proximity of the tip (as shown in Figure 4.5(d)) and black all over the image.

4. **Accurate tip detection**: To search for the exact location of the tip, the binary image I_{n_x} is searched along the direction of the micropipette detected in previous steps and the ultimate pixel with intensity of 1 is detected as the micropipette tip.

All of this procedure is illustrated in Figure 4.5.

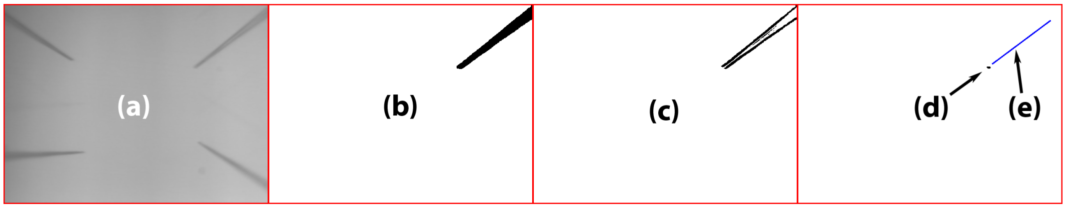

Figure 4.5: Tip detection algorithm – results: (a) original image, (b) mask image (negated for better visibility), (c) overlapped $I_{n_{y-}}$ and $I_{n_{y+}}$, (d) proximity of the micropipette tip as detected in I_{n_x}, negated to increase visibility, and (e) the detected line segment along the central axis of the micropipette.

4.5.2 Detection under a water-immersed objective

It is desirable to use the same tip detection algorithm in both cases but this might reduce the performance and efficiency of such an algorithm because the micropipette tip looks different using each objective lens as shown in Figure 4.4. An *intensity based approach* works for micropipette tip detection under a dry lens because there is enough contrast between the micropipette tip area and its neighborhood, whereas intensity based methods are not very useful for a water-immersed lens because the magnification is 5 times higher in this case so that the inner part of the micropipette becomes visible and the intensity pattern depends on whether it is filled in with water or not. The other issue is that there might be a lot of air bubbles and particles moving in the images obtained by the water immersion lens. Although the higher magnification of the water immersion lens makes it difficult to use an intensity based approach, it makes the cone-like shape of the micropipette tip visible (compare Figure 4.4 (a) and (b)). We take advantage of this feature to detect the micropipette tip using template matching techniques, i.e., searching for a predefined template in the image. The algorithm consists of three steps:

1. **Orientation detection**: This is exactly the same as the orientation detection approach described for the dry objective. The same algorithm works except that the range of motion δy is smaller due to the higher magnification of the water immersion objective.

2. **Initial estimation**: When the orientation detection algorithm is performed, there are two lines detected in $I_{n_{y^-}}$ and $I_{n_{y^+}}$ which are not parallel as described earlier. The intersection of these two lines is taken as the initial estimation.

3. **Template matching**: To obtain a more accurate estimation of the micropipette tip, we use a template of the shape of a micropipette tip aligned with the detected orientation. Then we calculate the cross-correlation between this template and the area of the image around the initial estimation. By detecting the maximum cross correlation, we can obtain an accurate estimation of the tip position. A detailed description of this cross correlation technique is given in Section 4.7.1.

4.5.3 Image-based Tracking of Micropipettes

There are different methods for tracking a moving object in a sequence of images in real time. We have used two different techniques for micropipette tip tracking under dry and water-immersion objectives. The algorithm for tracking under a dry objective is described below and illustrated in Figure 4.6.

- **ROI Extraction**: At iteration k, a region of interest (ROI) is being extracted from the latest captured image frame. The ROI is a rectangular window around the previous micropipette tip $P_n^I(k-1)$. $P_n^I(0)$ is provided by the tip detection algorithm.

- **Background Subtraction**: There are different methods for background subtraction in digital images. We have used a simple method which is suitable for this application. The average intensity is subtracted from each pixel in the ROI and then the absolute value of each pixel intensity is taken.

- **Adaptive Thresholding**: The resulting image is adaptively thresholded. For each pixel, the threshold value is the Gaussian weighted average of a 5×5 block around that pixel.

- **Morphological Opening and Closing**: The resulting binary image might have pixel noise and gaps. A morphological opening/closing sequence is applied to this binary image to obtain a clean segmentation of the micropipette as described in Section 4.5.1.

- **Hough Line Detection**: A Progressive Probabilistic Hough Transform [25] is then applied to detect the central axis of the micropipette.

- **Accurate Tip Detection**: Same as that described in Section 4.5.1. This can be modified to obtain sub-pixel accuracy [8].

Yilmaz et al. [26] have listed different methods for object tracking. One category of image-based object tracking algorithms relies on using registration techniques to register the current image frame to a predefined template. The template can be a region of the previous image around the known position of the object. As a micropipette tip is initially detected using the tip detection algorithm described in Section 4.5.2, micropipette tip tracking can be performed by registering each image frame to its previous frame in a region around the previously known micropipette tip position. This method is partic-

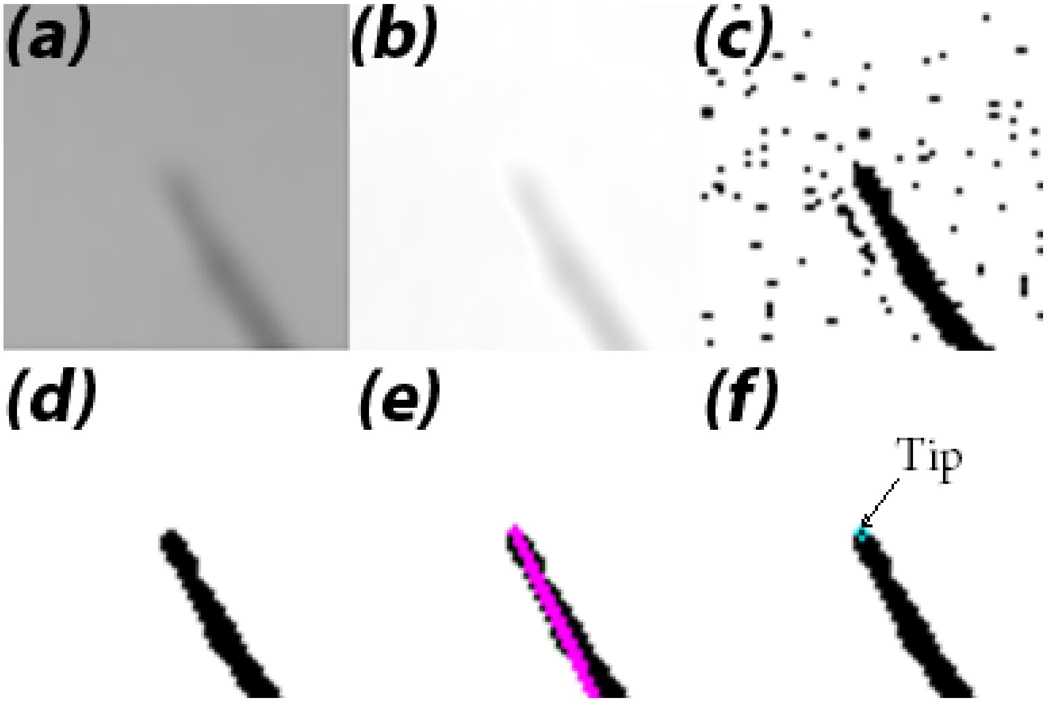

Figure 4.6: Image-based tracking under dry objective: (a) Original Gaussian smoothed ROI, (b) Background subtracted, (c) Adaptive thresholded, (d) Morphological opening and closing, (e) Hough line detection, (f) tip detected. All images except (a) have been negated to increase visibility.

ularly suitable for the water-immersed objective, where the micropipette has a special triangular shape as described in Section 4.5.2.

In our case, the amount of scale change between two consecutive frames (caused by lens distortions) is negligible in each small ROI. The same holds for rotation because the micromanipulators do not rotate and the micropipettes are rigid. A registration method can be used for detecting translation between two consecutive frames in an ROI around the prior known position of the micropipette tip. Phase correlation is an area-based registration method using the shift property of the Fourier transform [27] and it can be used to find translation, rotation and scaling between two images or an image and a template with a pixel or even sub-pixel accuracy [28, 29]. It has gained a lot of popularity in global or local image registration due to its accuracy and robustness to uniform variations of illumination and noise in images [29]. We have used the phase correlation motion estimation technique [30] to track the micropipettes in microscope images in real time. This method consists of the following steps:

- **Initialization**: includes detection of micropipette tip and static calibration to initialize the starting point.

- **Feature extraction**: Using the original images may not result in a robust registration technique because the contrast is not high enough and a big proportion of each image is the background which does not contain enough spectral information while we use an FFT-based technique which relies heavily on comparing 2D image spectra. To achieve more robust and accurate results some features of the images should be extracted.

- **Image augmentation**: To achieve a more robust and reliable performance of the phase correlation technique, we can augment the original images with some of its features to highlight the role of features in the image. In our case, the augmented image is a superposition of the original image and a Canny filtered image. As an example, the original image (I_n) and the results after applying Canny filter and augmentation ($\mathcal{I}_n$) of these images is shown in Figure 4.7.

- **Registration of $\mathcal{I}_n(k)$ to $\mathcal{I}_n(k-1)$:** To detect the displacement of the micropipette between the two image frames, we have used a phase correlation technique. First the $2D$ discrete Fourier transform is calculated for both images:

$$\mathcal{G}_{n,k} = \mathcal{F}\left\{\mathcal{I}_n(k)\right\},\ \mathcal{G}_{n,k-1} = \mathcal{F}\left\{\mathcal{I}_n(k-1)\right\} \tag{4.12}$$

Then the cross-power spectrum is calculated by pixel-wise normalized multiplication of $\mathcal{G}_{n,k}$ into the complex conjugate of $\mathcal{G}_{n,k-1}$: $\mathcal{R}_{n,k} = \frac{\mathcal{G}_{n,k} \cdot \mathcal{G}^*_{n,k-1}}{|\mathcal{G}_{n,k} \cdot \mathcal{G}^*_{n,k-1}|}$. Then we calculate the normalized cross-correlation of the images by taking the inverse discrete Fourier transform of $\mathcal{R}$: $r_{n,k} = \mathcal{F}^{-1}\left\{\mathcal{R}\right\}$. However to obtain a meaningful peak location, we should swap the first and third quadrants and the second and fourth quadrants. Then we determine the peak location in $r_{n,k}$ which gives us the amount of displacement between the consecutive image frames as shown in Figure 4.8(a): $(\delta x_{n,k}, \delta y_{n,k}) = arg\max_{(x,y)} r_{n,k}$. An illustration of the displacement between the two consecutive images is shown in Figure 4.8(b). The calculations are performed simultaneously for all micropipettes or those selected by the user, by processing an ROI around previously known position of each micropipette. An accurate tip detection can be performed once in a while to reset the tracking error.

4.6 Detection of Water Surface Touch

A water immersion lens should touch the water surface before it can be used. The objective lens is moved down slowly until it touches the water surface and then it goes back to its original height while pulling the water via its surface tension. The three steps of this procedure are shown in Figure 4.9.

Water surface contact is detected by a sudden change in illumination of the image. The temporal gradient of the images is calculated at each time step and when it is higher than a pre-defined threshold, the software stops the objective and asks the user to verify if the objective has actually made contact

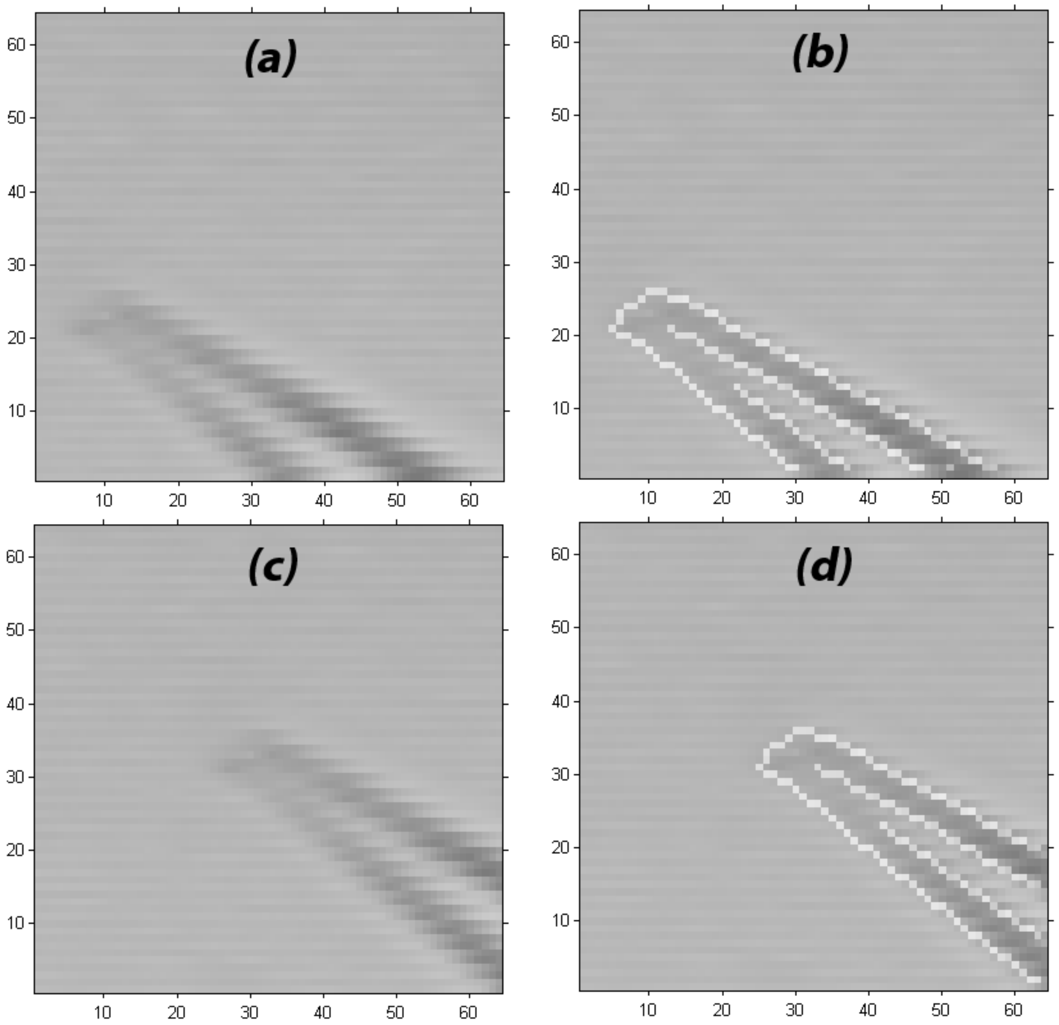

Figure 4.7: Augmentation of images to improve robustness of the phase correlation: (a) Original previous image frame $I_n(k-1)$ and (b) after augmentation $\mathcal{I}_n(k-1)$; (c) Original current image frame $I_n(k)$ and (d) after augmentation $\mathcal{I}_n(k)$.

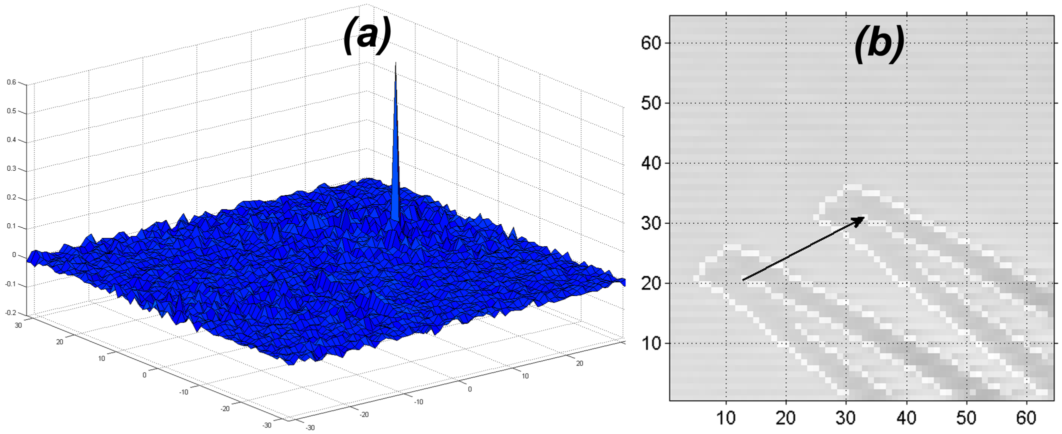

Figure 4.8: Phase correlation results: (a) cross correlation $r_{n,k}$ of the augmented images $\mathcal{I}_n(k)$ and $\mathcal{I}_n(k-1)$ with its peak located at $(20, 10)$; (b) overlapped images with the displacement vector; the augmented images are used to increase visibility.

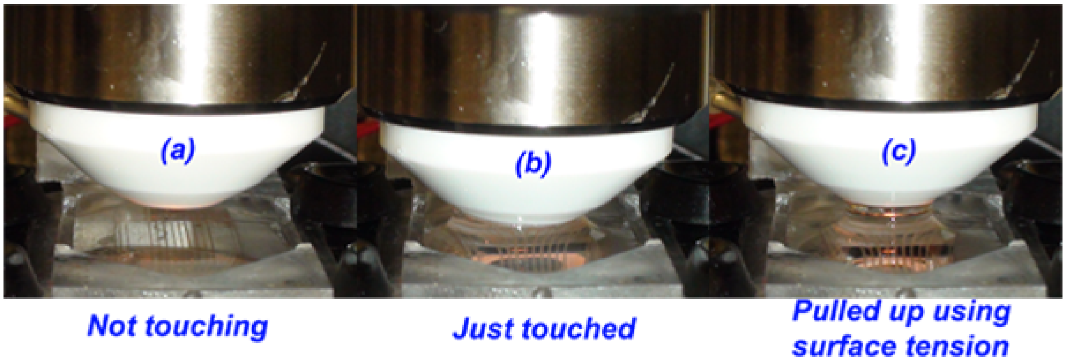

Figure 4.9: Water Surface Touch: (a) Before touching the water surface; (b) while touching; and (c) back up with stretched water surface.

with the water surface. Then the objective is moved up to its original position or it keeps moving down if the goal has not been achieved. The user can stop the motion in case the water surface has been touched but the software has failed to recognize it. We calculate a normalized integrated temporal gradient of the images as $\mathcal{F}_z(\mathcal{I})$:

$$\mathcal{F}_z(\mathcal{I}) = \frac{||\nabla_t \mathcal{I}||}{||\mathcal{I}||} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} |\mathcal{I}_k(i,j) - \mathcal{I}_{k-1}(i,j)|}{\sum_{i=1}^{M}\sum_{j=1}^{N} \mathcal{I}_{k-1}(i,j)} \tag{4.13}$$

The threshold is determined empirically by the user. For our experiments, a threshold of $\tau = 0.05(5\%)$ works well. It should be mentioned that if the initial illumination is too high, i.e., a very bright image is captured out of water, this method may not work properly because the image is already saturated; Also if the initial illumination is too low, there will not be enough light to detect the change. The method may also fail if the user suddenly changes the light intensity when the algorithm is running. $\mathcal{F}_z(\mathcal{I})$ and also the normalized average image intensities ($\bar{\mathcal{I}}(k) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} \mathcal{I}_k(i,j)}{MNI_{max}}$) have been evaluated in four different cases with the initial $\bar{\mathcal{I}}$ changing roughly from 30% to 70% representing a broad range of variation in light intensity. The results are shown in Figure 4.10. The method is robust to different illumination levels. Peaks are easily detectable in $\mathcal{F}_z(\mathcal{I})$ at the moment when the objective touches the water surface as shown in Figure 4.10.

4.7 Image-Based Tracking and Control

Using the dry objective lens with 4× magnification, the patch clamping cannot be performed. The reason is that the pixel size in this case is $> 1.5\mu m$ which makes it very difficult to locate and patch ion-channels in small neurons. A 20× water immersion objective lens is then used to provide a higher magnification as well as a high numerical aperture (NA) due to higher refractive index of water with respect to air [31]. In conventional methods of patch clamping the lower magnification lens is used to locate the target cells and then a higher magnification lens is used to perform the actual patch clamp

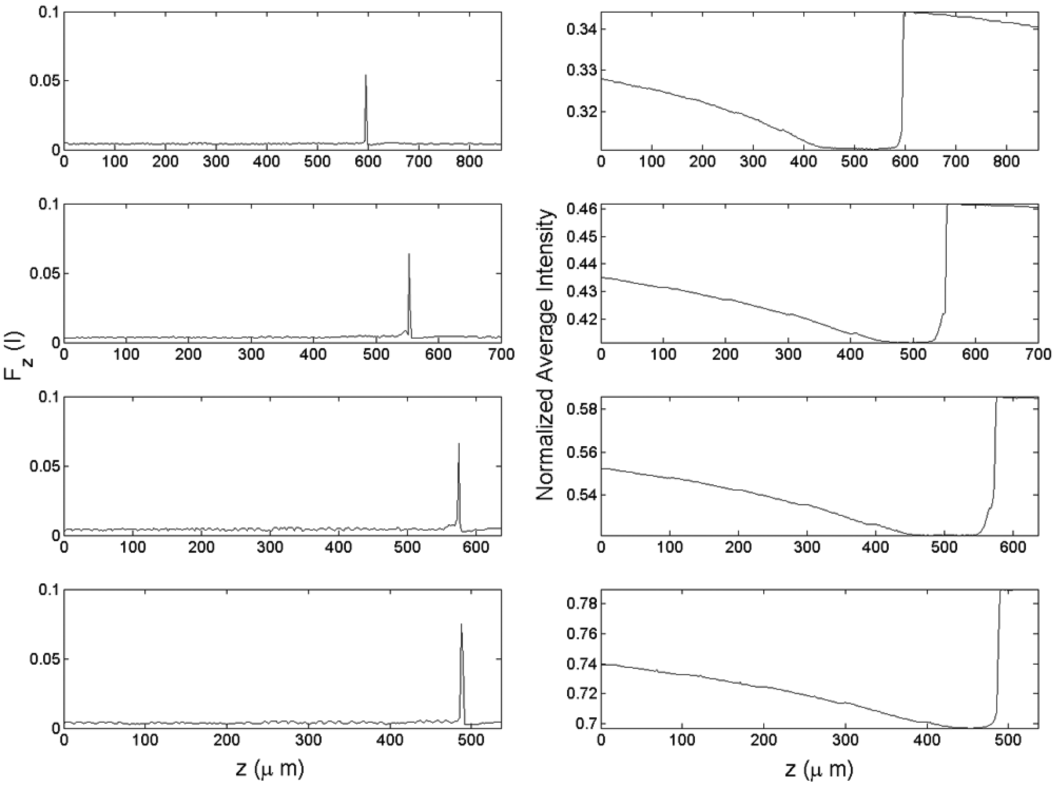

Figure 4.10: The graphs on the left show $\mathcal{F}_z(\mathcal{I})$ vs. z and the graphs on the right show $\bar{\mathcal{I}}$ vs. z for four experiments with different light intensity.

procedure. We have used the same approach: the 4× objective lens is used for initial system calibration and registration of the micromanipulator coordinate systems. The target cells are selected and then the objective is changed to the 20× water immersion lens. The water immersion lens is then placed at the approximate location of the target cells. The software is then used to bring the water immersion objective in contact with the layer of water covering the specimen and it automatically focuses on the target; details are described in [13]. The user clicks on the target cells where he/she wants to do patch clamping. The micropipettes are then brought close to the selected locations and in the FOV. Then the user has the ability to use the haptic device to move each micropipette around and perform patch clamping precisely at the desired locations.

Although the calibration procedure can provide a useful tool for positioning MP tips at desired locations, but the accuracy is limited by the accuracy of the microscope camera calibration. The mismatch of refractive indices of the tissue slice and water (and also the air in the case of the dry objective), causes image distortion. This becomes a bottleneck for the accuracy of microscope camera calibration, considering the fact that the amount of distortion is dependent on: (a) the objective lens's vertical position; (b) the thickness of the tissue slice; and (c) the water level. To compensate for this source of error, there are two main methods that can be applied:

1. Objective Lens distortion correction [32]: There are several models and algorithms developed for lens distortion correction. Most of these methods require a calibration grid of known size and dimensions which is very difficult to apply to our case. Putting such a grid in and out while the setup is working is very difficult if not impossible. Other than a calibration grid, the only existing references in our setup are the micropipettes which can be used to find the distortion pattern. However, there are several disadvantages associated with this method, including: (a) moving an micropipette tip all around the workspace is a time-consuming task; and (b) the distortion pattern changes when the tissue slice is changed.

2. Dynamic tracking of the micropipette tip: The main objective of system calibration was to find the coordinates of the image with respect to the coordinates of the micromanipulators and this is done to control the position of the micropipette tips when the reference target is given in image coordinates. It is possible to track the MP's tip position in real-time without any knowledge of the distortion pattern.

We have justified the application of real-time micropipette tip tracking when a global static distortion map cannot be evaluated or the calibration and/or registration data are not very accurate. The real-time tracking data can be used for visual servoing of the micromanipulators to obtain accurate and robust control of micropipette tips when there is uncertainty in the environment model. The algorithm is described below:

- The micropipette tip detection algorithm is used to detect the initial position of each micropipette tip in a microscope image as described in [20].
- Each micropipette is calibrated individually using the static method described in 4.4. This is done to find an estimation of the registration matrices and the Jacobians.
- Each micropipette tip is tracked in real time. Visual servoing is performed based on the tracking results while there might be uncertainties in the initial estimation of the Jacobians.
- If the micropipette goes out of the FOV, the registration data could be used to move the micropipette or the microscope to bring the tips of the micropipettes into the FOV[3].

4.7.1 Visual Servoing

Figure 4.11 illustrates the visual servoing block diagram where P_t^I, P_n^I and δP_n^I are the target position, the current micropipette tip position estimated

---

[3] The user can activate an option which forces the micropipettes to remain in the FOV by applying virtual forces when the tip of the pipette comes close to the image borders.

by the object tracking algorithm, and the tip position error of n^{th} micropipette produced by the trajectory generation algorithm, all measured in the image coordinate system $\mathcal{C}^{image}$. $I(k)$ and $I(k-1)$ are current and previous image frames, δP^{man_n} is the displacement command applied to the n^{th} micromanipulator controller, J_n is the image Jacobian relating displacement (velocity) in $\mathcal{C}^{image}$ to displacement (velocity) in $\mathcal{C}^{man_n}$, f_r^I is the collision avoidance force in $\mathcal{C}^{image}$ as described in Section. 4.8, α and K are scaling factors adjusting the relative weight of trajectory following versus collision avoidance and λ is a positive constant damping factor in the *damped least squares* method. The whole control loop consists of several parts which are described below:

Trajectory Generation

The robot controller has a closed architecture, i.e., a new move command cannot be issued before the previous displacement has been completed and the controller approves it by sending an acknowledgement signal back to the PC. In other words, the feedback loop for the visual servoing is open when the robot is moving. The amount of delay for completion of each command depends on the amount of displacement as shown in Equation 4.19. To have an effective control on each micropipette tip position, $||\delta P_n^{man}||$ should be small enough to keep the control loop feedback closed.

When a target point P_t^I is specified, the trajectory generation algorithm calculates a displacement δP_n^I based on the current location of the micropipette tip:

$$\delta P_n^I = \eta \frac{P_t^I - P_n^I}{||P_t^I - P_n^I||} \tag{4.14}$$

where η is a constant (in pixels) which can be adjusted by the user. As shown in Equation 4.14, δP_n^I is a displacement vector with a length of η pixels in the direction from the current location of the tip P_n^I to the target position P_t^I.

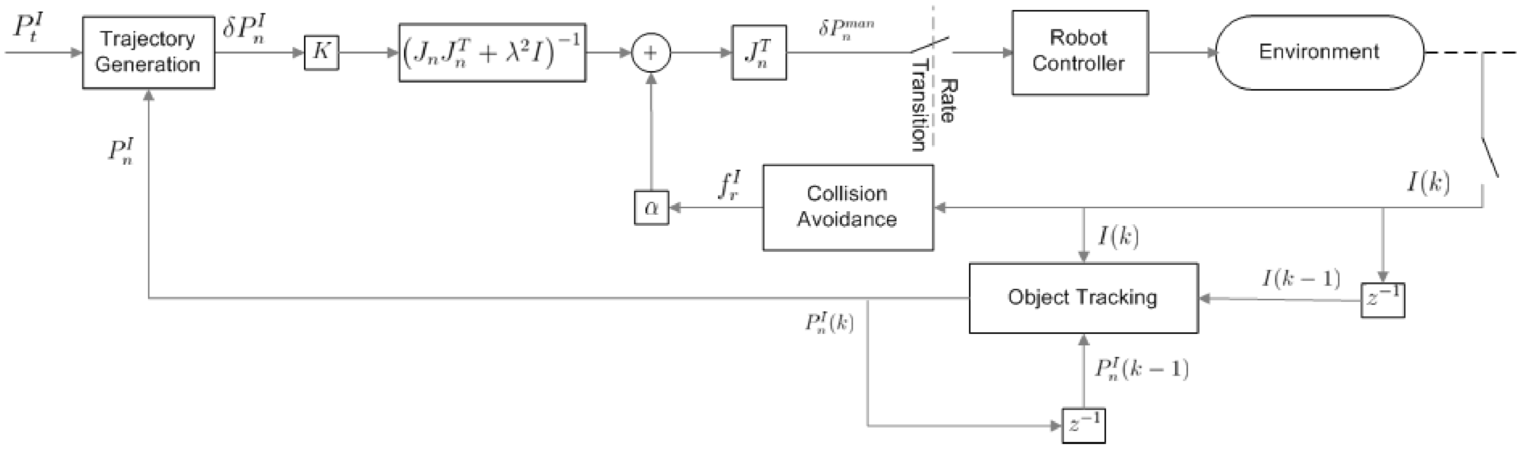

Figure 4.11: Visual servoing block diagram.

Trajectory Tracking

When a trajectory is generated by the trajectory generation block, a controller is required to make the system follow this trajectory. The displacement δP_n^I is measured in $\mathcal{C}^{image}$ while the robot controller works in its own coordinate system $\mathcal{C}^{man_n}$. In other words, we need to convert δP_n^I to δP_n^{man}. This can be stated as an inverse kinematics control problem by assuming $\mathcal{C}^{man_n}$ as the *joint space* for the n^{th} robot and taking $\mathcal{C}^{image}$ as the *task space*. There are different methods to solve an inverse kinematics problem including a direct analytic approach, a *Jacobian inverse/pseudoinverse* approach, a *Jacobian transpose* approach and a *Damped least squares* approach [33].

The inverse kinematics problem in our case is a non-singular problem because the task space and joint space are three dimensional Cartesian spaces which are related by an affine transformation which is invertible. Although such a problem can generally be solved using a direct analytic method or an inverse Jacobian method we have chosen a damped least squares method to do this. The reason is that it is not trivial to incorporate task space forces in an inverse kinematics problem unless the solution uses a Jacobian transpose method or one of its variants. The *damped least squares* technique [33] (also called the *Levenberg-Marquardt* technique) uses a Jacobian transpose based pseudo-inverse method which allows us to involve task-space forces while having a numerically stable inverse kinematics solution.

$$\delta P_n^{man} = J_n^T \left(J_n J_n^T + \lambda^2 I\right)^{-1} \delta P_n^I \tag{4.15}$$

When there is a task-space force f_r^I, the equation can be modified to:

$$\delta P_n^{man} = J_n^T \left[\left(J_n J_n^T + \lambda^2 I\right)^{-1} K \delta P_n^I + \alpha f_r^I\right] \tag{4.16}$$

where K and α are scaling factors which are adjusted by the user. The ratio $\frac{\alpha}{||K||}$ represents the importance of collision avoidance forces with respect to position errors.

The actual image Jacobian varies depending on the current location of the micropipette tip, due to image distortions. The Jacobian matrix is basically an affine matrix in our problem. The uncertainty in the Jacobian matrix consists of scale and rotation uncertainties and can be modeled as:

$$\hat{J}_n = \delta J_n(p) J_n(p) \tag{4.17}$$

where $\hat{J}_n$ is the Jacobian matrix estimated for the n^{th} micromanipulator through system calibration, $J_n(p)$ is the Jacobian matrix when the n^{th} micropipette tip is located at the point p in $\mathcal{C}^{image}$, and $\delta J_n(p)$ is the multiplicative uncertainty of this Jacobian at that point. However, the feedback loop is robust enough to compensate for small uncertainties as shown in the experimental results in Section. 4.12. To achieve a more efficient controller, the Jacobian matrix can be modified using an adaptive algorithm [34].

4.8 Collision Avoidance

Considering the limited space of operation, there is a high chance of collision among the micropipettes or between micropipettes and fixed obstacles. It is not possible to rely on the human operator to avoid collisions manually because the micropipette tip is very small, tips become very close to each other and they may also collide outside the FOV of the microscope.

Artificial potential fields have been used in several different real-time obstacle avoidance applications for robot manipulators and mobile robots [35], [36]. An artificial potential field (APF) algorithm has been implemented to avoid collision among micropipettes and obstacles in the workspace. A virtual force is generated based on this APF. The micropipette under control is assumed as an object with positive charge. Other micropipettes, the microscope lens and the bath are also assumed as objects with positive charges. To keep the algorithm computationally efficient, we have assumed point charges concentrated at a point. The virtually positive-charged point of an obstacle is the closest point of that obstacle to the micropipette under control. The virtual positive charge on the controlled micropipette, is assumed to be located on the closest point of that micropipette to each obstacle. Although it is also possible to extend artificial potential fields to perform full navigation including target tracking and collision avoidance at the same time by assigning an opposite charge to the target [37], we have only used *APF* for collision avoidance to avoid the local minima problem associated with this method.

All calculations are based on the assumption that micropipette tips are located in the focal plane and calculations are done in this plane. A possible scenario is shown in Figure 4.12.

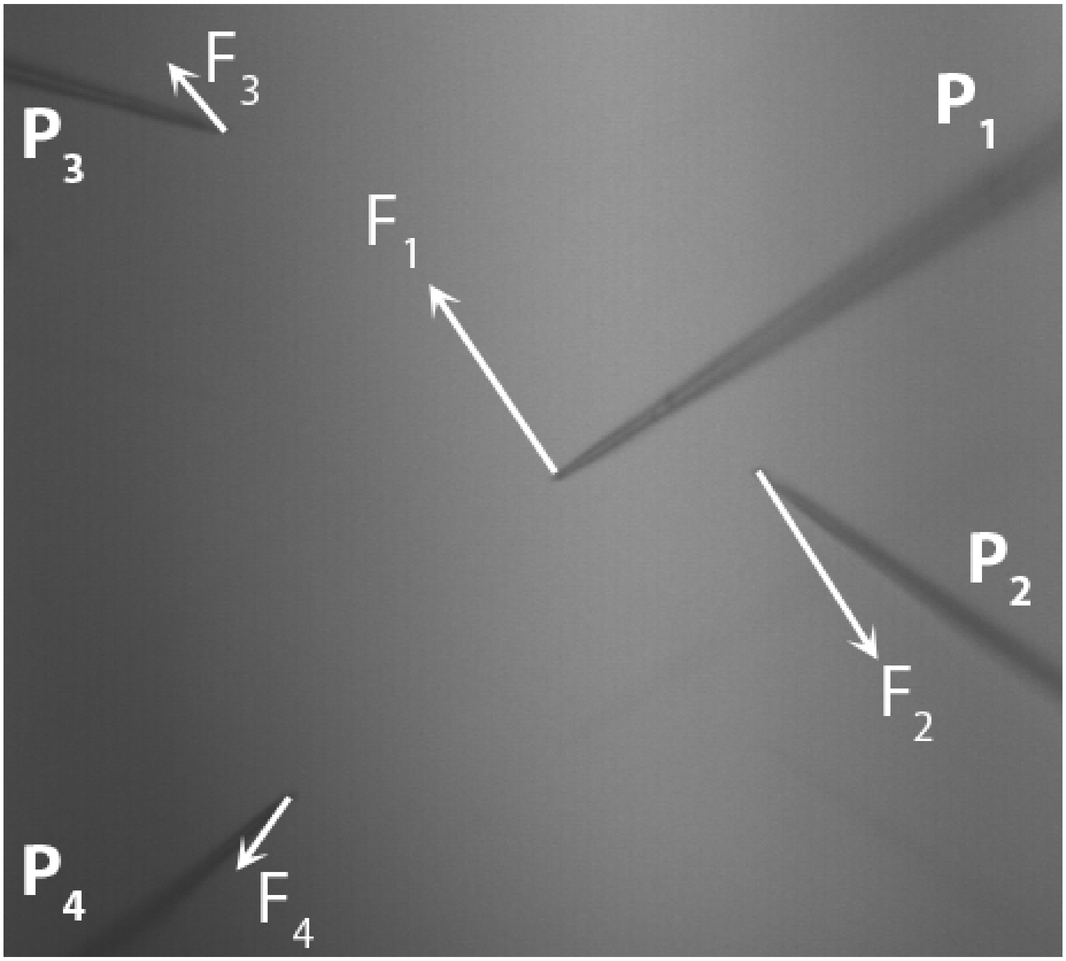

Figure 4.12: Collision avoidance forces are calculated based on the closest distance to other obstacles. P_k shows the k^{th} micropipette. F_i is the repulsion force acting on P_i and generated by other micropipettes.

For each micropipette P_i, $\mathcal{P}_{i,j}$ is the vector connecting the closest points between P_i and P_j, pointing towards P_i. If the vector with minimum length is $\mathcal{P}_{i,k_{min}}$, then:

$$F_i = \gamma_i \sum_{k \in \mathcal{S}_i} \frac{\mathcal{P}_{i,k}}{||\mathcal{P}_{i,k}||^3}, \ \mathcal{S}_i = \{k \ | \ ||\mathcal{P}_{i,k}|| \leq (1+\epsilon_i)||\mathcal{P}_{i,k_{min}}||\} \tag{4.18}$$

where γ_i is the repulsion factor which determines the amount of force used for collision avoidance on P_i and $\mathcal{S}_i$ is the set of indices of the closest obstacles to P_i, and k_{min} is the index of the closest obstacle to P_i. ϵ_i is taken into account to exclude those micropipettes/obstacles which are not closest to P_i. To be on the safe side, it is taken as $\epsilon_i = 0.1, \ \forall i$ to cover up to 10% of the distance to closest obstacle around P_i. Without a small enough ϵ_i, the superposition of the repulsion forces may vanish and a collision may occur. A collision margin is defined by the user in terms of pixels. The collision avoidance force is ignored if the minimum distance is out of this margin.

To mark fixed obstacles (other than micropipettes), user should move an micropipette close to each obstacle manually and the software would register the coordinates of that obstacle in $\mathcal{C}^{ref}$. It is also possible to focus and click to register fixed obstacles (such as the mesh).

4.9 Haptic Interface

A haptic device is used to give the user the opportunity to move a specified micromanipulator while feeling virtual forces which help the user to ensure that the micromanipulators do not collide with each other or with the environment. This is an intuitive and easy to use interface which accelerates the process of positioning the micropipettes. The coordinates of the haptic device are aligned with the coordinates of the live image on the screen which makes it intuitive for a novice to work with the system. An illustration of the coordinates of the system is given in Figure 4.13.

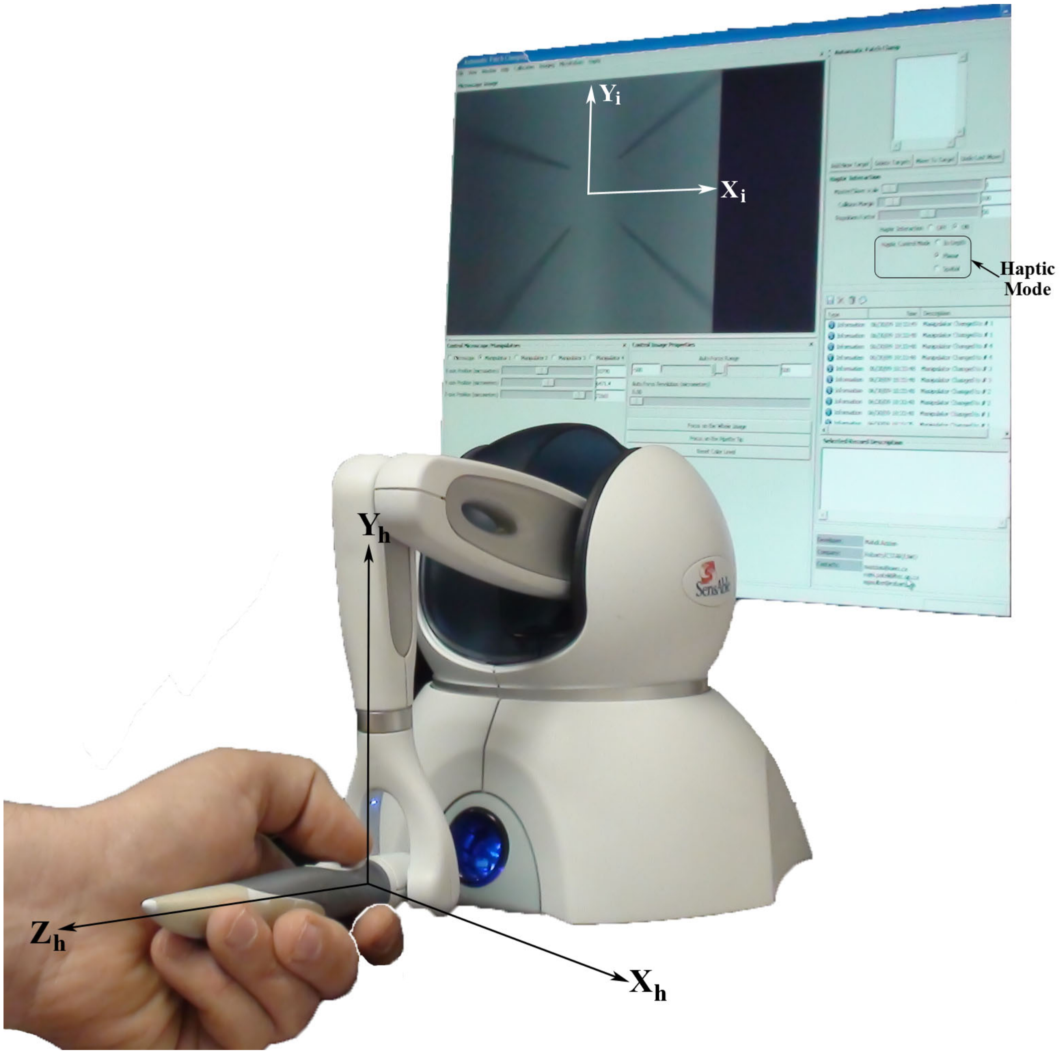

Figure 4.13: Haptic-enabled control: the coordinates of the haptic device are aligned with those of the microscope in such a way that X_h, Y_h are matched with X_i, Y_i, and Z_h is matched with the depth. There are four modes defined for master-slave control: (a) In-depth motion, where the motion of the haptic device is limited along Z_h; (b) planar motion, where the haptic device is limited to move in a plane orthogonal to Z_h and the corresponding micropipette moves almost only in the image plane; (c) 3D motion, where the haptic device can move in 3D and the user has full control over the corresponding micropipette; and (d) coaxial motion where Z_h is mapped on to the X-axis in $\mathcal{C}^{pipette_n}$.

4.9.1 Master-Slave Control

All of the micromanipulators used in the system have a closed-architecture controller which means that a user-specified control scheme cannot be used. In a robot with a closed-architecture controller, it is not possible to do direct torque or velocity control and it is only possible to issue position or velocity commands and read motor torques and/or forces (contact forces measured by a force sensor) in defined time intervals with specific protocols through a network connection. When a command is issued, the controller replies with an acknowledgement response after accomplishment of the task, or generates an error code when the task cannot be completed. When an inquiry is issued, the controller responds with appropriate data or an error code if the inquiry cannot be completed. Inquiries usually take a certain (fixed) amount of time while control commands can take variable time. For example, if a move command is issued, it takes more time if the requested displacement is longer.

In the system considered here, the closed-architecture robot takes position commands and responds to inquiries on the position of the robot. Each inquiry for the position of the robot takes $T_i = 15.5ms$ and each *move command* takes approximately $T_c(ms)$ to complete, where:

$$T_c = \begin{cases} T_0, \text{ when } ||d|| \leq ||d_0|| \\ T_0 + \frac{||d|| - ||d_0||}{v}, \text{ when } ||d|| > ||d_0|| \end{cases} \tag{4.19}$$

where $T_0 = 64.0ms$ is the minimum delay and v is the constant speed of the corresponding micromanipulator as reported in Appendix D.

Therefore, each iteration of the control loop takes at least $\min(T_c + T_i) = 79.5ms$ for the MP-265. Adding a delay of $33.4ms$ for capturing an image frame and assuming the calculation of collision forces is performed in $1.0ms$, the minimum total delay would be $T_d = 113.9ms$ which gives us a maximum bandwidth of $8.78Hz$ on the slave side. Although large delays in master-slave systems might create serious instability problems [38], [39], the amount of delay in our case lies within the human reaction time and is not expected to create instabilities. Brooks [40] has reported a minimum bandwidth of $3.9Hz$ and a maximum desired bandwidth of $9.7Hz$ (according to $100ms$ human reaction time) based on the consensus of teleoperation experts. We do not expect any stability problems resulting from the inherent delays of the closed-architecture controllers, because:

- There is no significant extra delay induced by the communication channel between the master and the slave (i.e., for master and slave controllers running on the same PC)
- The slave bandwidth ($8.78Hz$) is within the desired range reported in [40]. It is actually close to the maximum desired bandwidth ($9.7Hz$).
- The environment model, including the artificial potential field used for collision avoidance is passive, i.e., there is no active component in the environment.

A block diagram of the master-slave control system is represented in Figure 4.14.

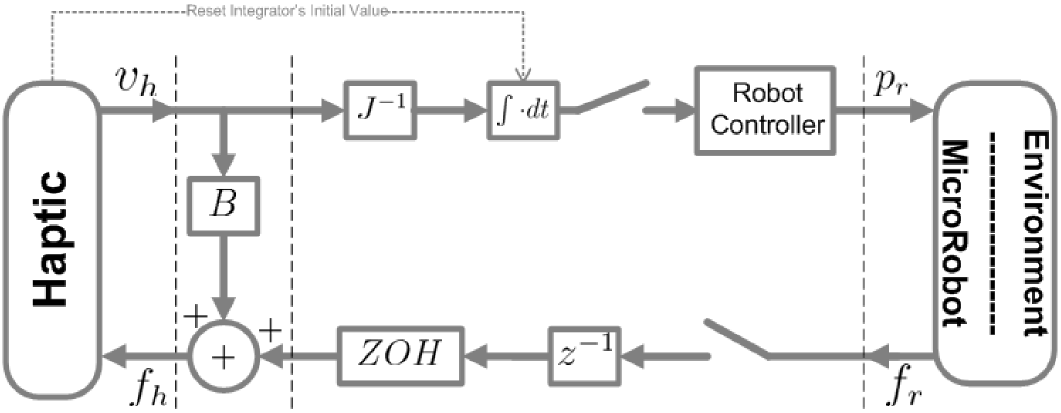

Figure 4.14: Master-Slave control architecture: J^{-1} is the Jacobian inverse as defined in Section 4.4. Each time the user presses the stylus switch on the haptic device, the integrator is reset; This is done to increase the workspace resolution of the haptic device. The user should keep the switch pressed in order to move the slave, this is a useful safety feature.

4.9.2 Studying Performance of the Master-Slave Control

As stated before, we do not expect any instability in the master-slave control due to the passive nature of the environment and the bandwidth of the slave system. The experimental results show that the system is stable and the performance of the master-slave control system is quite acceptable in practice as shown in Section 4.12. To study the performance of the master-slave system, we have recorded the position of the master and the slave as well as the reflecting forces in an actual experiment, when the slave micromanipulator is moving the micropipette through a path close to other micropipettes as shown in Figure 4.31. The forces reflected on the haptic device are a combination of the collision avoidance forces with a small viscosity term created by the damping factor B. As shown in Figure 4.33, the reflected forces are pretty smooth and no jerk was observed in practice. The results are discussed in more detail in Section. 4.12.6.

4.10 Online 3D Visualization

There are applications where micromanipulation of a live cell in tissue is required. Patch clamp electrophysiology is one such application where microelectrodes are used to study ion channels of excitable cells such as neurons and cardiomyocytes [1]. In such problems, there are two major challenges: (a) bringing single/multiple tool(s) close to cell(s) repeatedly and (b) manipulating the cell(s) using the tool(s) once a tool is located close to a target cell. To address the first challenge, we have developed microscope image-guided, robot-assisted techniques to locate micromanipulators close to target cells [20, 41, 42]. A master-slave control scheme has also been developed [41] which makes the user able to manipulate multiple micropipettes using a haptic device. Although the master-slave control of the micromanipulators along with the real-time 2D visual feedback and the autofocusing capability makes it easier for the user to perform cell micromanipulation, part of the second challenge still remains. This is due to (a) lack of depth perception, (b) limitations of $2D$ visual feedback which is always partially defocused because of the limited depth of view in conventional microscopes, and (c) lack of knowledge about the $3D$ shape and relative position of the target cell(s) with respect to the tool(s). This makes cell manipulation difficult and time consuming, e.g., in an application such as patch clamping, the user may tear a cell membrane or may not be able to patch the cell successfully which usually results in the need for changing the micropipette and/or selecting a new target cell, which is undesirable and time consuming. Therefore, real-time $3D$ visual feedback capable of illustrating the $3D$ shape of the target cell(s) and tool(s), can be very helpful in micromanipulation of live cells especially when located in tissue.

Ideally, a 3D volume can be reconstructed by creating a stack of images (optical sections) captured by moving the objective up/down with respect to the specimen. Confocal microscopy is an optical imaging technique which is based on elimination of out of focus light and is used widely in $3D$ reconstruction, however it cannot be used for live cell manipulation [31] and the $3D$ reconstruction is usually performed offline. Conventional wide-field microscopy is normally used for working on live cells; However, the problem is that the out-of-focus light in a conventional wide-field microscope blurs the images in such a way that the $3D$ structures in the specimen will not be clearly visible.

The blurring effect can be modeled by convolution of the $3D$ image into a $3D$ *Point Spread Function* (PSF). There have been several methods developed for deconvolution of the $3D$ reconstructed images such as inverse filtering, Wiener deconvolution, Tikohnov regularization etc., where almost all of them need an accurate estimation of the *PSF*. For a conventional wide-field microscope, it is very difficult to end up with an accurate estimation of the *PSF* due to the existence of noise in experimental methods and the effect of microscope optic aberrations on theoretical methods. Blind deconvolution techniques have been developed to estimate the microscope *PSF* and the $3D$ specimen image simultaneously [43]. However, blind deconvolution techniques are based on iterative approximations. At each iteration, multiple $3D$ convolutions are required which makes the algorithm very slow and therefore not suitable for real-time applications.

To develop a real-time $3D$ reconstruction algorithm, methods should be used which do not need knowledge of the *PSF*. Pure $2D$ (no-neighbor) deblurring methods are very fast and time efficient [44]. These methods are basically spatial high-pass filters while microscope images of most of the specimens, e.g., neuron cells, are combinations of low and high spatial frequency components. The main disadvantage of such methods is that they may remove a part of the specimen as well. In case of live cells in a tissue, there are plenty of undesirable high frequency regions in the images which can not be omitted using no-neighbor methods. This will leave a lot of undesirable tissue texture which will make it very difficult to recognize a cell.

An alternative approach, the nearest neighbor method, is based on subtracting a weighted combination of the nearest neighbor slices on top and bottom of each slice. The out of focus light in each image slice is a combination of the light coming from all other slices, but in the nearest neighbor approach, it is assumed that the main contribution is from the adjacent slices. This method is also very fast and computationally efficient but it has the disadvantage of creating undesirable artifacts in the slices specially when the slices are very close.

Partial $3D$ reconstruction can also be performed using focus/defocus measures [45]. This method is based on estimating the associated depth of each pixel by finding the image slice with maximum focus measure at that pixel [46]. This method cannot be directly applied to our case due to low contrast images where the tissue is also visible along with the cells. Therefore, we have modified the "shape from focus" method and developed an algorithm which is computationally efficient enough to be implemented online. We have then defined a measure of accuracy by comparing the results of our algorithm to manual segmentation of the cell membrane. We have also studied two different methods to reduce the time required for reconstruction by down-sampling and/or processing a region of interest.

4.10.1 $3D$ Surface Reconstruction

We have developed a partial $3D$ surface reconstruction scheme based on focus/defocus measures. For each pixel of the image, all slices are searched to find the slice with the *maximum local sharpness* value [20], then the vertical location of the corresponding slice is recorded. A $3D$ surface can then be reconstructed based on these points but in most of the cases, it will be full of noise and spikes created by the tissue so that the target cell cannot be recognized. Pre-processing and post-processing of the images is required to obtain a smooth surface, free of noise and spikes.

Capturing an Image Volume

In order to capture an image volume, the user specifies z_0, z_l and δz which are the vertical location of the objective with respect to the reference coordinate system for the first and last slices, and the distance between the slices, respectively. The number of captured slices will be $N_z = \frac{|z_l - z_0|}{\delta z} + 1$. A default range of $|z_l - z_0| = 51.0\mu m$ and $\delta z = .2\mu m$ is used in the software; the reason is that usually most of neuron cells fit in such a range, the step size is small enough to capture the required details and also the number of slices will be $N_z = 256$ which fits well for an 8-bit representation; However, the user can modify these settings based on the requirements in each experiment. The microscope objective is moved according to the range and step size set by the user to capture N_z image slices. Samples of three slices captured in one of our experiments are shown in Figure 4.15. However, the raw image volume is useless without processing, because (a) each image slice contains out of focus information, and (b) all parts of the tissue are visible which makes it very difficult to recognize a cell.

Preprocessing and Segmentation

The original captured volume is called $\mathcal{V}(x, y, z)$ which consists of image slices $\mathcal{I}_z(x, y)$ captured at different depth levels of the tissue. To increase the contrast of the images, we equalize the histogram of each image $\mathcal{I}_z$ and call it $\mathcal{I}_z^{eq}$ and the

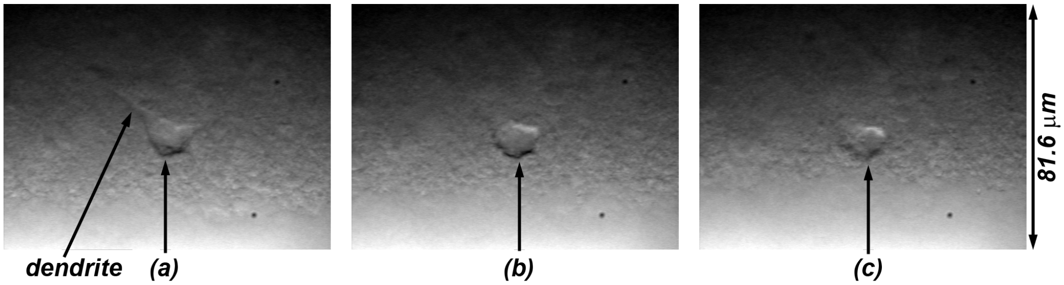

Figure 4.15: A single neuron cell viewed at different depths: (a) bottom slice, (b) middle slice and (c) top slice. All the images have been contrast enhanced by histogram equalization because the original image contrast is very poor (Figure 4.16(a)). The image plane is not parallel with the focal plane and the slice thickness is also not uniform, therefore we observe non-uniform light illumination and out-of-plane focus effects from top to bottom across the image slice.

corresponding volume is called $\mathcal{V}^{eq}$. Then we associate a focus measure to each image slice $\mathcal{I}_z^{eq}$ by evaluating the standard deviation of the pixel intensities in a neighborhood of each pixel resulting in $\mathcal{I}_z^g$ with a corresponding image volume $\mathcal{V}^g(x, y, z)$. A circular neighborhood with N_f pixels radius is used. The user can specify the radius. Increasing N_f will slow down the algorithm while a very small value degrades the results; default value is set to $N_f = 7$. There are different measures that can be used to evaluate the focus of a region in an image [18]. The standard deviation measure creates smooth results which is better suited for depth from focus applications [47]. $\mathcal{V}^g(x, y, z)$ is the called the *local sharpness (focus) measure* at each point (x, y) in each image slice $\mathcal{I}_z$. Results of the sequence of these filters on a sample slice are shown in Figure 4.16.

The corresponding surface can be reconstructed as

$$\mathcal{S}(x, y) = \arg\max_z \mathcal{V}^g(x, y, z) \tag{4.20}$$

At the next step, we adjust the dynamic range of $\mathcal{S}(x, y)$. This process is called normalization: $\mathcal{S}^a(x, y) = \frac{\mathcal{S}(x,y)-z_0}{z_l-z_0}.(2^N - 1)$ where z_0 and z_l are the

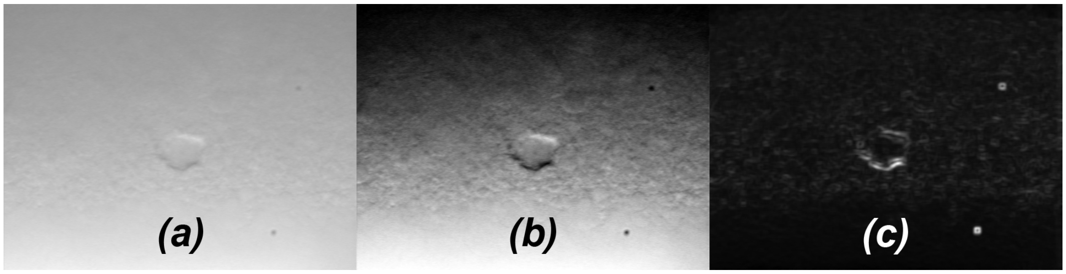

Figure 4.16: Results of the preprocessing filter sequence: (a) Original image $\mathcal{I}_z$, (b) Histogram equalized image $\mathcal{I}_z^{eq}$ and (c) High-pass filtered image $\mathcal{I}_z^g$.

height of the first and last image slices, respectively, and N is the number of digits used for quantization chosen according to $\lceil \log_2 N_z \rceil$; (we have used $N = 8$ as the default value.). Please note that the voxel size is adjusted based on the pixel size of the image and the distance between the slices. This whole procedure is illustrated in Figure 4.17.

The $\mathcal{S}^a(x, y)$ can be considered as an image which is then warped as a surface by associating the grayscale value to a height (in the z-direction) for each pixel. This is shown in Figure 4.18. However the warped surface $\mathcal{S}^a(x, y)$ will be full of spikes and noise as shown in Figure 4.18(b) and the cell structure cannot be recognized in most of the cases. We apply two levels of filtering to remove these spikes and undesirable parts.

Post-processing

We perform two levels of filtering to remove undesirable parts of the reconstructed surface. The overall block diagram of the algorithm is shown in Figure 4.19.

- First-level filtering:

  To construct the first-level filter, we make a mask image:

$$\mathcal{N}(x, y) = \max_z \mathcal{V}^g(x, y, z) \tag{4.21}$$

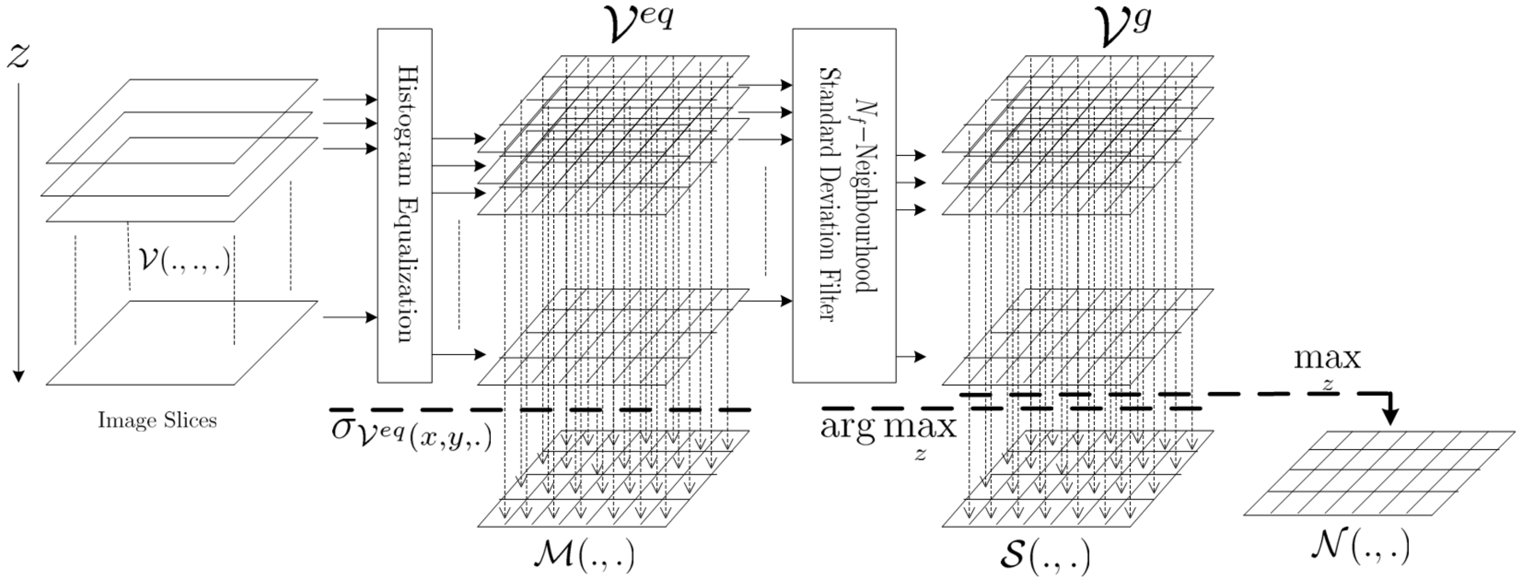

Figure 4.17: The captured image volume $\mathcal{V}(.,.,.)$ is processed to obtain the initial $2D$ shape from focus $\mathcal{S}(.,.)$. Mask images $\mathcal{N}(.,.)$ and $\mathcal{M}(.,.)$ are also calculated.

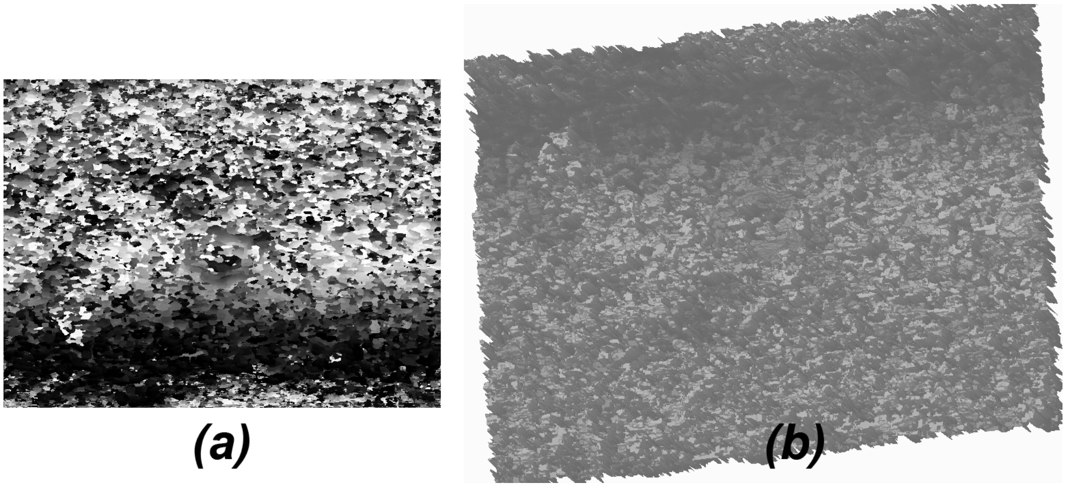

Figure 4.18: (a) $\mathcal{S}^a$ and (b) the corresponding warped surface. The cell cannot be recognized due to the overwhelming existence of undesirable parts, further processing is required.

as shown in Figure 4.17. The mask image has lots of undesirable point as shown in Figure 4.20(a) that should be removed and only parts of the image which are representing the cell structure should remain; In order to do this, the mask image $\mathcal{N}$ is thresholded and the result is denoted by $\mathcal{N}^\tau$. There are different thresholding techniques [48] which are used to determine an optimal threshold value to segment images into foreground and background. We used Kapur's method [49] which is based on maximizing the sum of the background and foreground entropies. The result is shown in Figure 4.20(b).

Then we apply the mask image to $\mathcal{S}^a$ using the *AND* logical operator.

$$\mathcal{S}^a_{\mathcal{N}}(x,y) = \mathcal{S}^a(x,y).\mathcal{N}^\tau \tag{4.22}$$

However, there are still spikes and undesired objects in the reconstructed surface as can be seen in Figure 4.21(b); These are mostly created by parts of the image which have high spatial spectral components in image slices but have a low spatial spectral component across the slices.

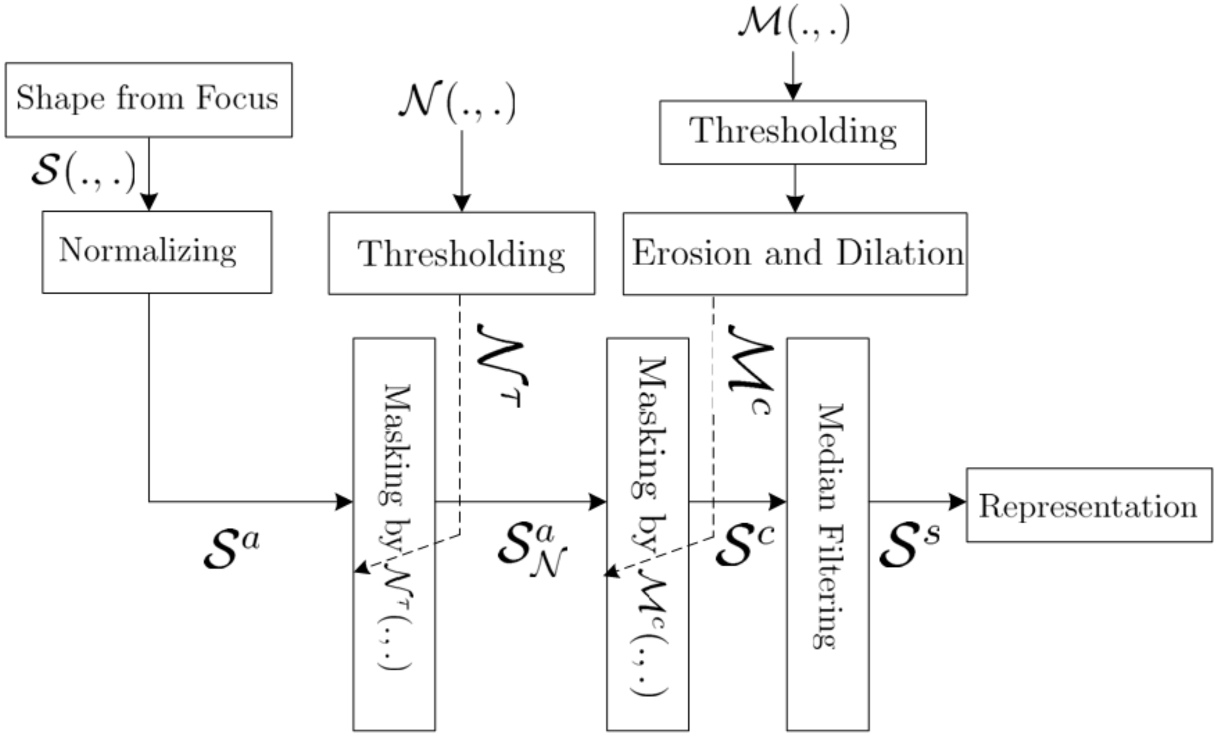

Figure 4.19: Block diagram of the reconstruction algorithm. $\mathcal{S}(.,.)$, $\mathcal{N}(.,.)$ and $\mathcal{M}(.,.)$ are from Figure 4.17.

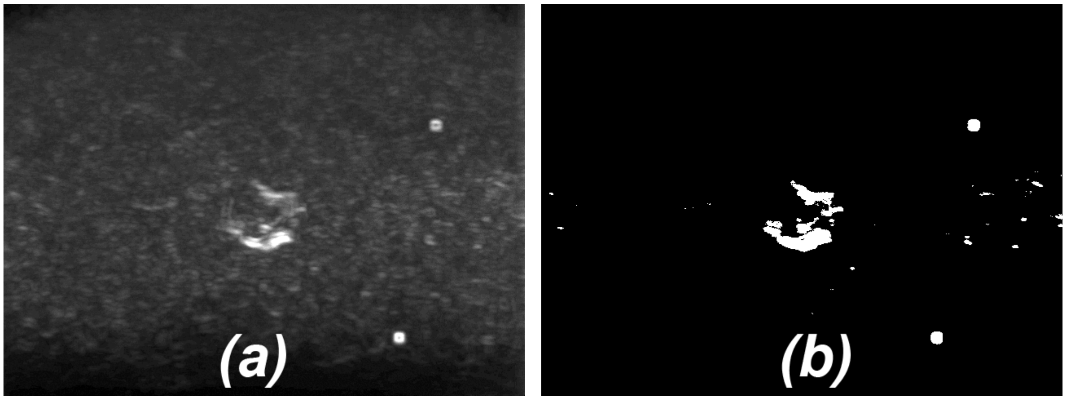

Figure 4.20: First-level mask image $\mathcal{N}$, (a) before and (b) after thresholding ($\mathcal{N}^{\tau}$).

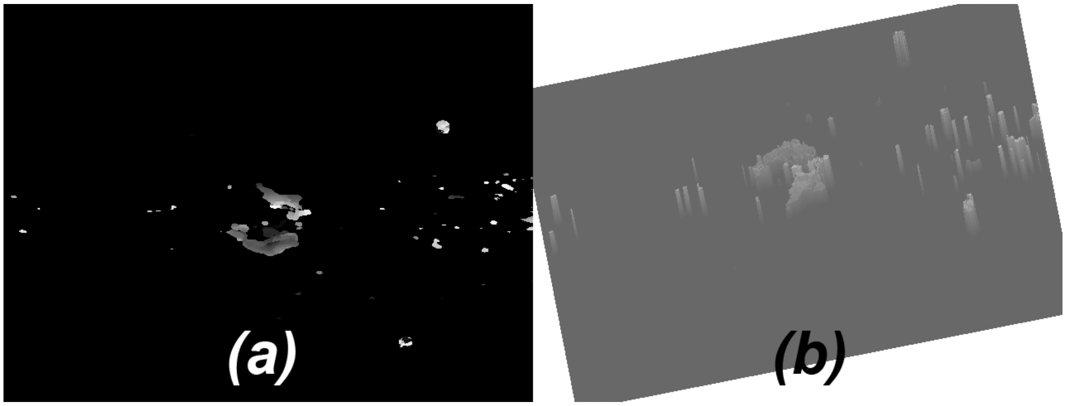

Figure 4.21: (a) $\mathcal{S}^{a}_{\mathcal{N}}$ and (b) the corresponding warped surface.

- Second-level filtering:

  As our second-level filter, we construct a mask image $\mathcal{M}(x, y)$ where each pixel is the standard deviation of the corresponding pixel across all slices.

  $$\mathcal{M}(x, y) = \sigma_{\mathcal{V}^{eq}(x,y,.)} \tag{4.23}$$

  Figure 4.22(a) shows $\mathcal{M}(x, y)$ for a sample case. This mask image is then thresholded, eroded and dilated (as shown in Figure 4.22 to highlight connected parts of the data where there is a significant change across the image slices. The resulting mask image is denoted by $\mathcal{M}^c(x, y)$. We apply the mask image to $\mathcal{S}^a_{\mathcal{N}}(x, y)$.

  $$\mathcal{S}^c(x, y) = \mathcal{S}^a_{\mathcal{N}}(x, y).\mathcal{M}^c(x, y) \tag{4.24}$$

  To reduce noise in the reconstructed surface while preserving the edges, we apply a 2D median filter to $\mathcal{S}^c(x, y)$ resulting in $\mathcal{S}^s(x, y)$, a 7×7 window median filter usually works well for our application.

Improving the processing speed

As reported in Section. 4.12.7, processing takes $49.35sec$ for 201 image slices with a size of 640×480 pixels. There are several methods which can be used to improve the processing speed of the proposed reconstruction algorithm. Some of these methods are categorized as:

1. Reducing the size of the input data
    - Down-sampling the input image volume
    - Processing on a region of interest (ROI)
2. Using a different processing technique
    - Using a faster processor.

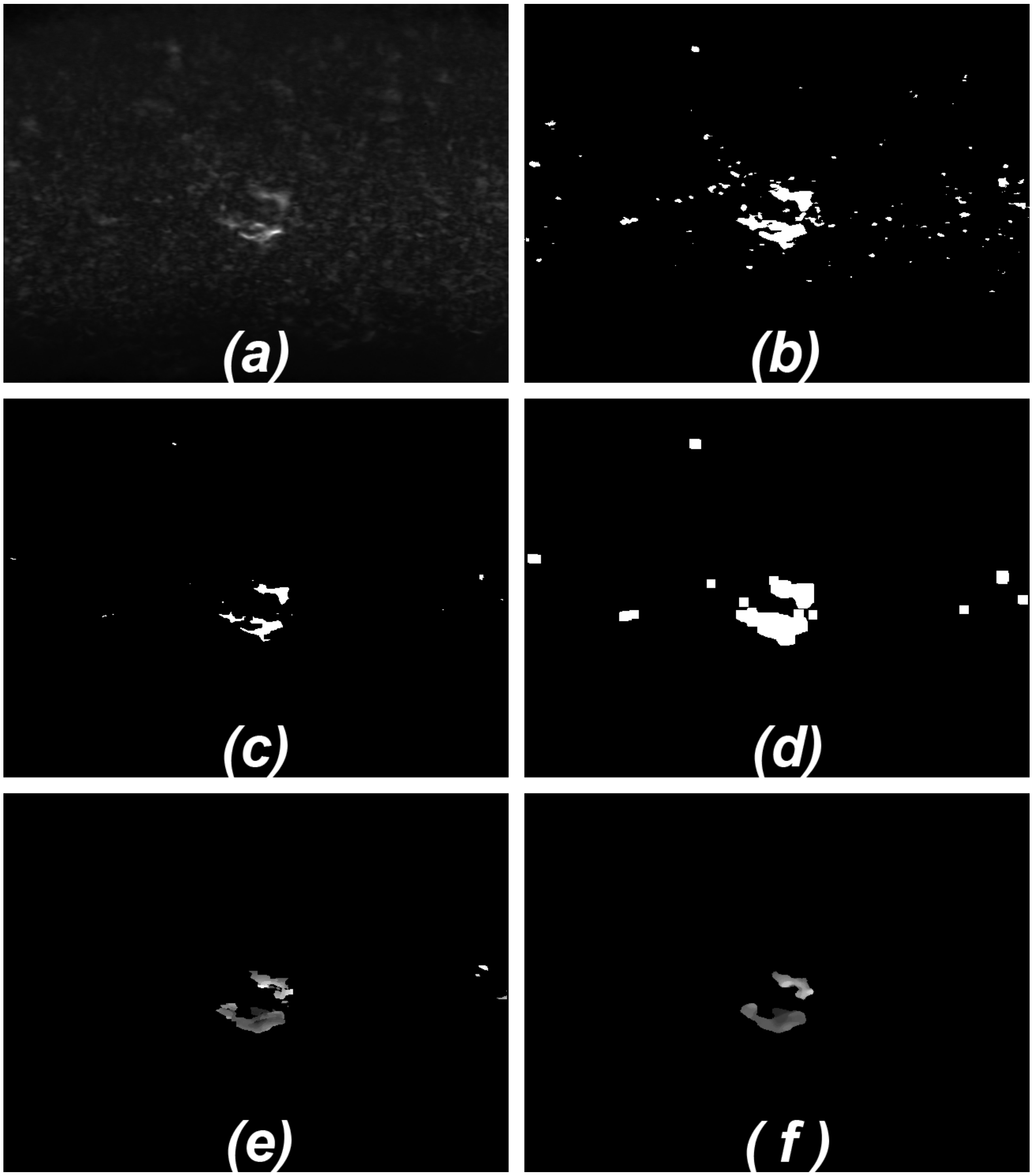

Figure 4.22: Second-level mask image: (a) $\mathcal{M}$, (b) $\mathcal{M}^{\tau}$, (c) $\mathcal{M}^{\tau}$ after morphological erosion, (d) $\mathcal{M}^{c}$ which is $\mathcal{M}^{\tau}$ after morphological erosion and dilation, (e) $\mathcal{S}^{c}$ (masked) and (f) $\mathcal{S}^{s}$ (median filtered).

- Re-implementation of the algorithm for a parallel processing environment such as a multi-core CPU or GPU.
- Performing parts of processing while capturing images.

We have developed methods according to the first category to reduce the size of the input data. The second category is mostly related to implementation of the algorithm and is the subject of further research.

- Down-sampling:

  We have used a *Gaussian Pyramid* technique for down-sampling images [50]. Each slice is down-sampled by a power of two and the slice thickness is also adjusted accordingly. This method reduces the processing time due to smaller size of the input data. It will also reduce the time required for capturing image volumes because of the smaller number of slices are being captured due to increased slice thickness. A sample slice on a *Gaussian Pyramid* is shown in Figure 4.23.

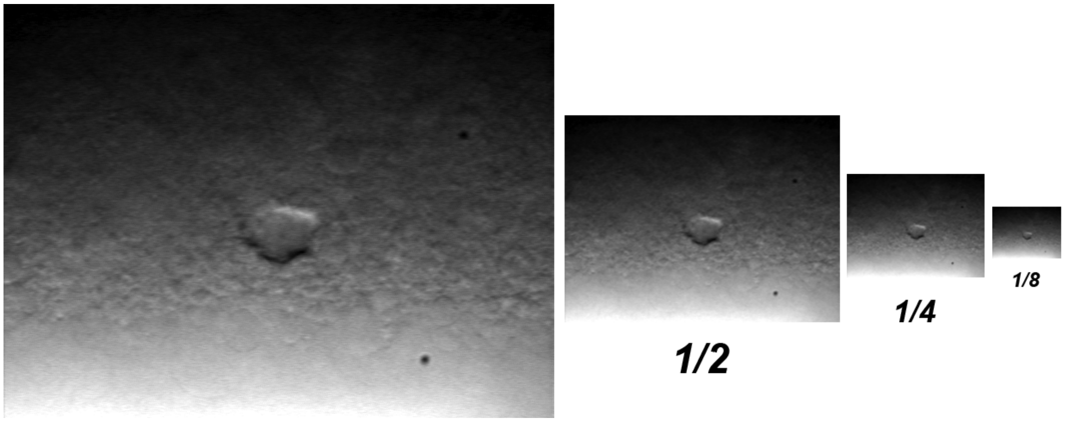

Figure 4.23: Down-sampling on a Gaussian Pyramid.

  This method results in faster processing but it decreases the accuracy of the algorithm. The trade-off between accuracy and the speed of the algorithm is studied in Section. 4.12.7.

- Region of interest processing:

Another method to reduce the processing time is to decrease the size of the input data by processing a region of interest instead of the entire image. The advantage of this method is that it does not sacrifice the spatial resolution while it adds another step to the algorithm to obtain region(s) of interest where cells are located. This can be done manually by asking the user to click on to cells or through an automatic 2D segmentation of the cells.

We have developed a simple segmentation algorithm to automatically determine the region of interest where a cell is located. This can be used when minimum user interaction is desired. The slice with the maximum overall sharpness, $\mathcal{I}_{z_{max}}$ is selected where:

$$z_{max} = \arg\max_{z} \left\{ \sum_{x} \sum_{y} |\mathcal{V}^{g}(x, y, z)| \right\} \tag{4.25}$$

Then a circular Hough transform (CHT) was used to determine the location of cells in the corresponding image slice, $\mathcal{I}_{z_{max}}$. The results are shown for the sample image in Figure 4.24.

The thresholding method used for the whole image masks $\mathcal{N}, \mathcal{M}$ is not suitable for processing a region of interest around the cell. The entropy maximization removes parts of the cell structure in order to make a balance between background and foreground entropies. To solve this problem, we have used Otsu's thresholding method to minimize the weighted sum of pixel variances in background and foreground clusters [51]. This method is fit for the cases where the image is almost equally divided into two clusters [48] and therefore provides satisfactory results for a region of interest but no good result for the whole image.

We have tested this algorithm on three different image volumes. In each case 26 image slices around the $\mathcal{I}_{z_{max}}$ have been used. Location of the center of the detected Hough circle was determined with an average standard deviation of 5.2566 pixels (equal to $0.8936\mu m$). An average error of 6.1927 pixels (equal to $1.0528\mu m$) was obtained compared to a

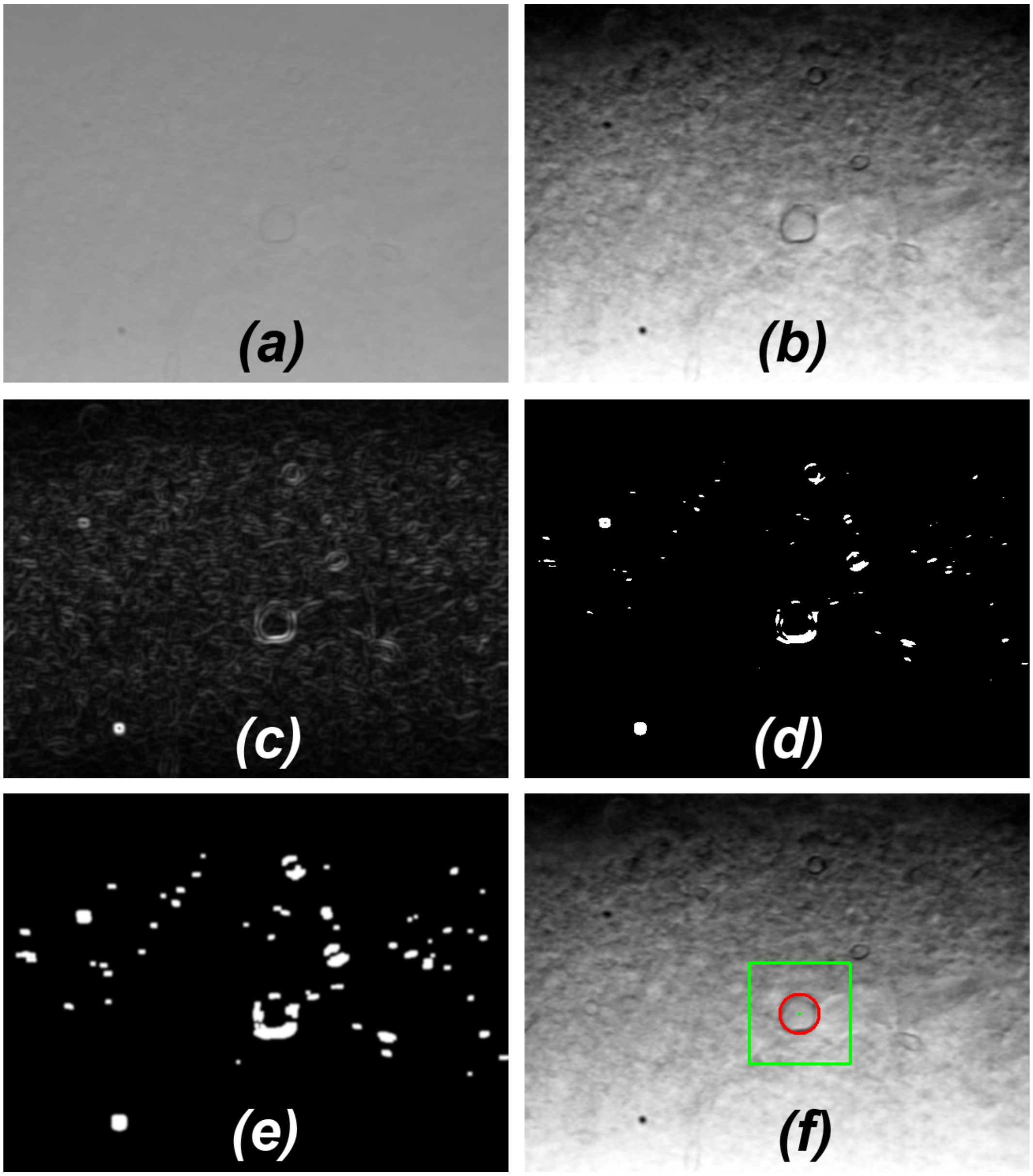

Figure 4.24: Finding the region of interest around a cell: (a) original image $\mathcal{I}_{z_{max}}$, (b) $\mathcal{I}^{eq}_{z_{max}}$, (c) $\mathcal{I}^{g}_{z_{max}}$, (d) after thresholding, (e) after a sequence of dilation, erosion, dilation, Gaussian smoothing and (f) circular Hough transform. The region of interest is selected as a 128×128 pixels square co-centered with the detected circle.

manual reference when a user was asked to draw a circle on the target cell.

Representation

The resulting warped surface is shown in Figure 4.25. However, the warped $3D$ surface representation may have some limitations specially when we want to combine the reconstructed 3D structure with a live 2D microscope image. We have also created an alternative representation by adding voxels at each $\mathcal{V}^r(x, y, z) = (x, y, \mathcal{S}(x, y))$ where $\mathcal{S}^s(x, y) > 0$. We have applied a linear cubic voxel interpolation to get a better representation of the reconstructed volume. The result is shown in Figure 4.26(b).

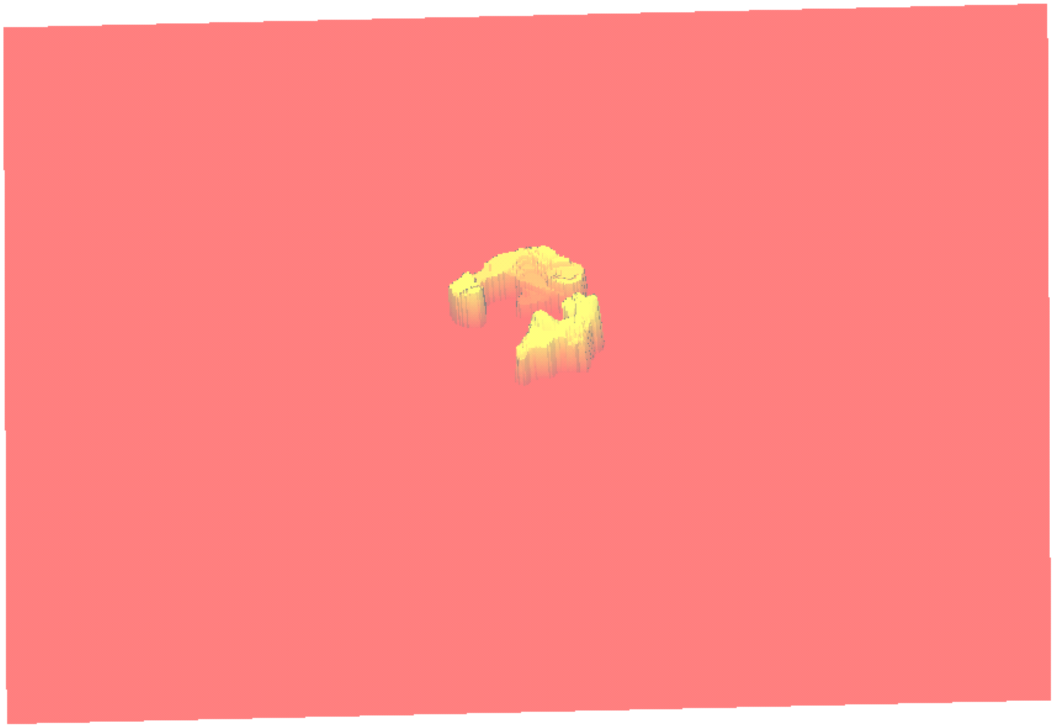

Figure 4.25: Reconstructed surface as a result of warping the $\mathcal{S}^s(x, y)$.

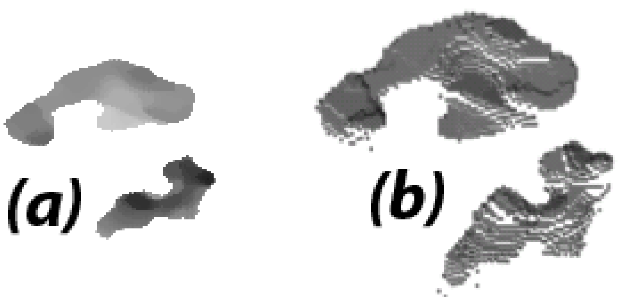

Figure 4.26: (a) Part of the gray-scale image $\mathcal{S}^s(x, y)$ before warping, colors inverted to increase visibility, and (b) snapshot of the 3D representation of interpolated voxels in $\mathcal{V}^r(x, y, z)$.

4.10.2 Master-Slave Control

A haptic device was used as the master and is capable of measuring user hand motions and at the same time applying forces on his/her hand. The haptic device is used for two different purposes:

1. Master-slave control of each manipulator.
2. Changing the camera pose of the 3D scene.

The registration information along with the virtual camera position and orientation is used to coordinate the motion of the haptic device with the motion of the $3D$ tools displayed in the $3D$ image. The haptic device coordinates are aligned with the display coordinates in such a way that the user can intuitively move the specified micropipette in the desired orientation. A collision avoidance algorithm is used to generate haptic forces to prevent the user from going too close to another micropipette. For more details please see [41].

4.10.3 Real-time Visual Feedback

The reconstructed scene, either the volume or the warped surface is displayed and rendered in real-time along with the micropipettes. The position/orientation of each micropipette is calculated based on its current position and the calibration/registration information. The $3D$ scene is updated and rendered in real-time. The user can select to look at the $3D$ reconstructed volume, the warped surface or the $2D$ image. The user can also superimpose the live 2D microscope image on the $3D$ reconstructed volume. The system is capable of stereoscopic image display and the user can activate this mode and put on stereoscopic goggles to get better depth perception as shown in Figure 4.27.

4.11 Software Architecture

The developed system involves several hardware access routines as well as user interaction and visualization processes; Hence it is very important to use a versatile software design that avoids any collision among hardware access routines, provides real-time visual feedback to the user and at the same time incorporate user commands during real-time processes. As an example, when the Haptic mode is enabled, the user interacts with the software through the haptic device and the graphical user interface (GUI) at the same time. The haptic device controller, visual servoing, micromanipulator control and visualization and rendering are running on the same system and accessing the same resources. We have developed an event-driven multi-threaded software environment that handles all of the required tasks and takes care of mutual access problems and real-time coordination among different threads.

This software has different operation modes, including the AF mode, different haptic interaction modes, the calibration mode, etc. To have coordination among different operation modes, we have used a finite state-machine architecture which prevents any confusion among the different modes and tasks. Each program thread may have a different functionality at each different state.

We have used VTK (Visualization Toolkit) for graphics and visualization purposes [52] and KWWidgets (a GUI Toolkit based on VTK) to create the graphical user interface [53]. A snapshot of the GUI can be seen Figure 4.13.

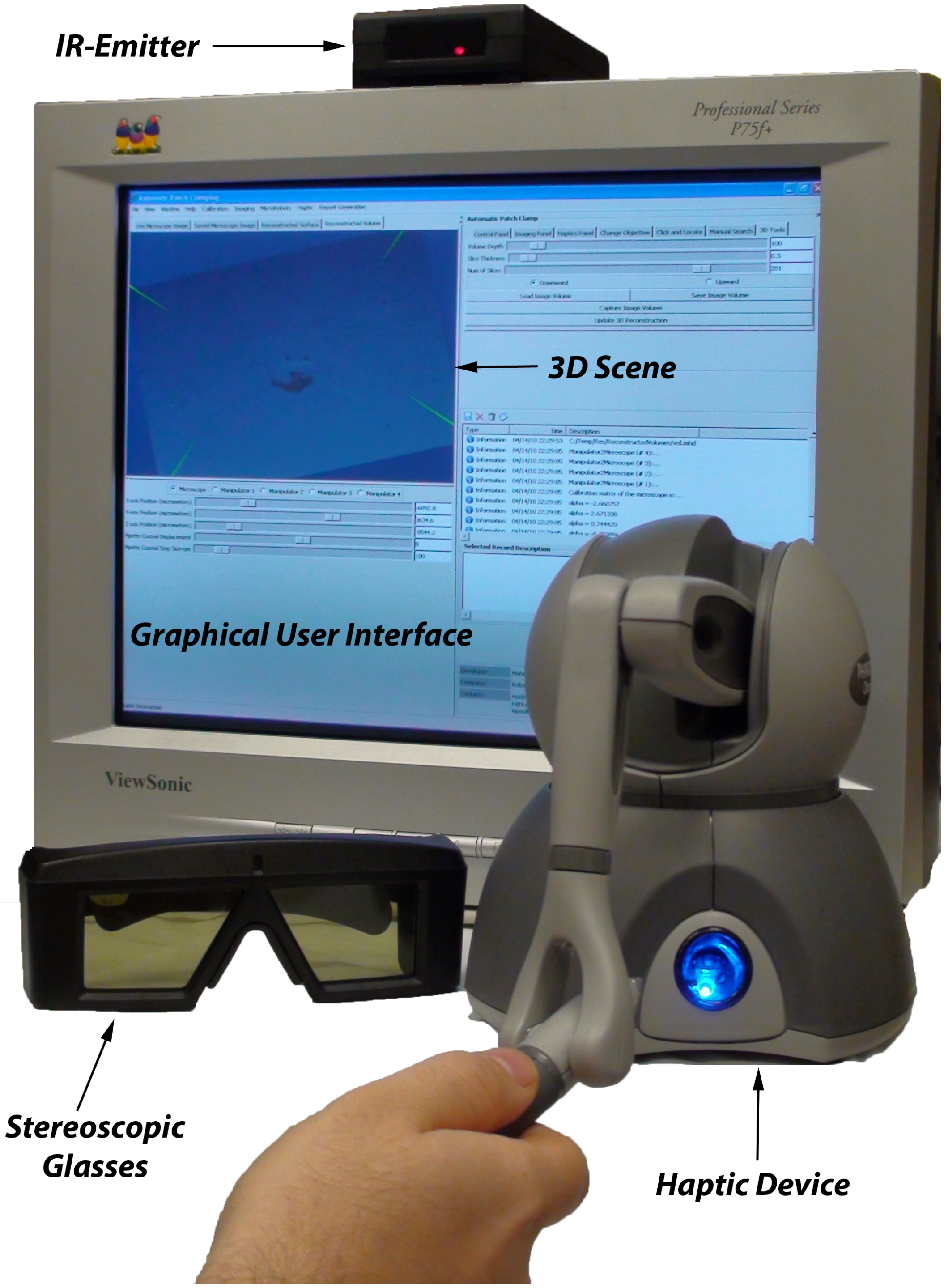

Figure 4.27: Graphical user interface, haptic device, and stereoscopic goggles are shown for a sample experiment. The represented 3D scene is an augmented reality environment consisting of the partial $3D$ cell reconstruction, the live 2D microscope image and a virtual representation of the micropipettes.

4.12 Experimental Results

The procedure for patch clamping using the developed platform is as follows: The user places the brain slice (or other tissue to be studied) on the substrate and performs the conventional preparation routine. Then the micropipettes are placed in the field of view of a dry lens ($4X$ in this case) under the microscope either manually or using the rough calibration information. Accurate calibration and registration routines are then performed either automatically or semi-automatically (by user intervention) as described in Section. 4.4. Then the user moves the slice under the microscope to locate the region of the slice that contains the appropriate cells. This is an initial guess and can be modified later. The software can be used to autofocus on this region and register the coordinates of this location. Micropipettes are then moved close to the specified region at a height that is slightly above the top layer of the slice. The location of each micropipette is registered, the micropipettes are pulled out slightly to avoid collisions during the lens change and then the objective lens is switched to the higher magnification water immersion lens using a motorized lens slider [13]. Software is then used to bring the objective over the previously registered location. The lens-water contact detection algorithm is used to moved the objective down until it comes into contact with the water surface (almost nothing can be seen before that) and then it is moved up again to its previous position. The micropipettes are automatically moved back to their registered locations. Then the objective and micropipettes are moved down slowly until the cells in the slice are visible. The user can move the objective around to search for good cells while the micropipettes follow the objective or are brought into the field of view after appropriate cells are found by the user. Then the haptic device is used to bring each micropipette into close proximity to a cell and move it diagonally (using coaxial mode) to touch the cell membrane. An artificial potential field algorithm is used to prevent any collisions between micropipettes [19]. Once the membrane is touched, suction is applied to achieve a *Gigaseal*. If the resistance is high enough, recording of voltage and current clamping can be performed [19], [13]. The results obtained using the algorithms described in the previous sections are discussed below.

4.12.1 Autofocusing

We have evaluated some features of the autofocusing algorithm. Autofocusing is repeatedly performed on the same scene and the mean and standard deviation of z_{opt} (vertical position of objective lens when it is in focus) are determined. The results for various methods are represented in Table 4.1. The precision was measured to be $2.680\mu m$ for the focusing on the micropipette tip in Section. 4.3. To measure the accuracy of the autofocusing algorithm on a micropipette tip, the position of the micropipette was used as a reference. We repeated the autofocusing task several times and measured the average value of z_{opt}. We then moved the micropipette vertically with sub-micron accuracy and repeated the measurements. This whole process was performed 100 times. The difference between the average z_{opt} in two consecutive experiments was compared to the amount of movement of the micropipette. The average value of $|\delta z_{objective} - \delta z_{micropipette}|$ shows the accuracy of autofocusing which is estimated to be $2.6\mu m$ for the dry objective. This method ignores any fixed offset in focusing since a fixed offset is not important in our application because we are using focusing as a method for relative depth estimation. Better accuracy and precision can be achieved for the water immersion objective because of the higher magnification.

4.12.2 System Calibration

There are several factors that cause errors in calibration, including but not limited to, (a) micropipette tip detection error, (b) unmodeled image distortions, (c) vibrations in the setup, and (d) limited accuracy of the micromanipulators. We have used a method to evaluate the calibration error and the results of this evaluation are reported below.

An micropipette is moved on a 7×5 grid in the FOV. At each node the micropipette tip is detected using the tip detection algorithm. The error between the detected tip point and the tip point calculated using T_{man}^{image} is obtained at each point. This is repeated for 10 different objective lens heights. The error on this set of 350 points is then processed to evaluate the calibration/registration algorithm. The average square root error in the X and Y directions ($mean(\sqrt{e_x^2 + e_y^2})$) is estimated to be $5.642 \mu m$ (dry lens, 1× magnification).

We have evaluated the magnification adjustment parameters as shown in Equation 4.6. Table 4.2 lists these parameters for the dry objective lens. The magnification adjustment parameters for the water immersion objective are reported in the last row in Table 4.2. A detailed description for measurement of these parameters can be found in [13]. It should be noted that the κ_z value in the last row of this table depends on the water level; more details are given in [13].

Magnification		Parameters			
Objective	Microscope	α	$\kappa_x(\mu m)$	$\kappa_y(\mu m)$	$\kappa_z(\mu m)$
4×	1×	1.0	0	0	0
	2×	1.91	46	-68	-123
	4×	3.81	73	-68	-167
20×	1×	5.03	20	323	3534

Table 4.2: Objective and microscope magnification adjustment parameters.

Table 4.3 gives the magnification adjustment parameters for different microscope magnification levels in the case of the water immersion objective. The values in this table are measured with respect to the same objective with microscope magnification at 1×.

Magnification		Parameters			
Objective	Microscope	α	$\kappa_x(\mu m)$	$\kappa_y(\mu m)$	$\kappa_z(\mu m)$
20×	1×	1.0	0	0	0
	2×	1.89	9	-9	-8
	4×	3.68	16	-9	-10

Table 4.3: Objective and microscope magnification adjustment parameters for the 20× water immersion objective.

To measure the magnification adjustment parameters, we used the AF

algorithm to focus on an micropipette tip. We also used a modified visual servoing algorithm to locate the micropipette tip at the center of the image, by moving the microscope micromanipulator after changing the microscope magnification. The overall amount of the microscope micromanipulator motion determines the $\kappa_{x,y,z}$ values in Tables 4.2 and 4.3. The value of α was determined by dividing the average pixel sizes after performing camera calibration for each microscope magnification separately.

4.12.3 Tip Detection Error

To determine the accuracy of tip detection, the software moves a micropipette all over the image and the user is asked to click on the tip of the micropipette. The tip detection algorithm is also performed for each case and the results are recorded and processed. An average absolute error of 0.4706 pixels is achieved in X direction and an average absolute error of 0.4314 pixels in Y direction. Maximum error in each direction is 1 pixel. The average norm of error ($mean(\sqrt{e_x^2 + e_y^2})$) is 0.7871 pixels which is within 1 pixel error of the reference user clicks. Detailed results are represented in Table 4.4.

$mean(e_x)$	0.0784	$mean(e_y)$	0.1176
$mean(\|e_x\|)$	0.4706	$mean(\|e_y\|)$	0.4314
$std(e_x)$	0.6883	$std(e_y)$	0.6526

Table 4.4: Tip detection errors chart

To evaluate the precision of the tip detection algorithm, a random point is chosen, the micropipette tip detection is performed a 100 times for that point and the standard deviation of the measurement is evaluated. The process is repeated for 12 points on a 4 × 3 grid all over the image and the precision is calculated as the average of the measurement standard deviations for all points. The tip detection precision is 0.2412 pixels in X direction, 0.2600 pixels in Y direction and 0.3125 pixels in both direction.

4.12.4 Image-Based Tracking

To evaluate the image-based tracking algorithm described in Section 4.7.1, the user was asked to move a micropipette using the haptic device on an arbitrary path in the field of view. The tracking algorithm has been running during the experiment in a separate thread and the results have been recorded. The tracking results are shown in Figure. 4.28 along with the results calculated using calibration information and encoder readings of the micropipette at each instant of time.

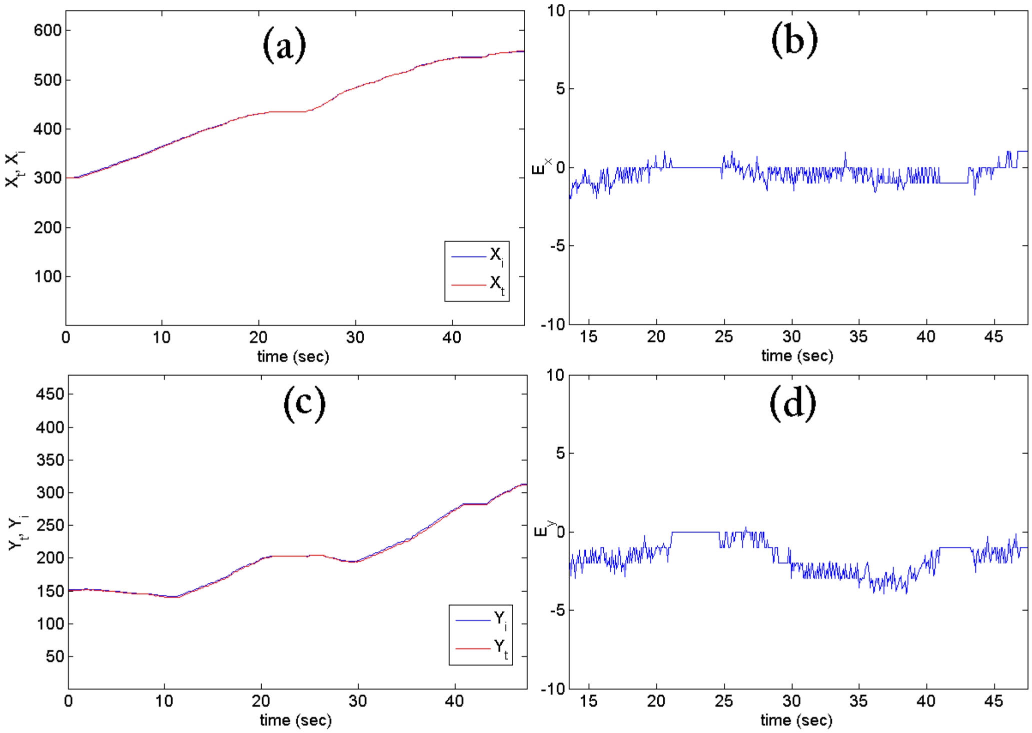

Figure 4.28: Image tracking results showing tracked P_t vs. calculated P_i coordinates: (a) X_t vs. X_i, (b) $E_X = X_t - X_i$, (c) Y_t vs. Y_i and (d) $E_Y = Y_t - Y_i$.

The average absolute error is X and Y directions are 0.8642 and 1.4073 pixels respectively. The average square root error in both directions is 1.8594 pixels. The worst-case maximum error is 4 pixels in each direction. The results were obtained using the dry objective. The tracking algorithm for the water immersion objective also shows acceptable. However, the calibration/registration for water immersion objective is not very accurate and therefore cannot be used as a reference to evaluate the tracking results. This is mainly due to image distortions, variable water levels and other factors such as difference of refraction index between the tissue and water. In each case, the tracking algorithm can be reset using the tip detection algorithm if the tracking is lost for any reason.

4.12.5 Visual Servoing

Test results for the visual servoing algorithm are shown for an arbitrary path in proximity to an obstacle (another micropipette). The parameters have been adjusted to $\lambda = 0.1$, $K = I$ (identity matrix), $\alpha = 0.5$, $\eta = 2.0$, and the collision safety margin has been set to 150 pixels. The results are shown in Figure 4.29.

Figure 4.30 represents the starting and target points and the straight path between them, the obstacle, moving micropipette at the start and the end of the path, and the virtual collision avoidance forces at each point of the path. As observed in Figure 4.30, the Jacobian transpose controller shown in Figure 4.11 takes a curved path around the obstacle instead of following a straight path to the target. To increase the speed, η can be increased.

4.12.6 Master-Slave Control

Master-slave control of an micropipette in proximity to another micropipette is performed for an arbitrary path selected by the user. In other words, the user is able to move each micropipette around while the virtual collision avoidance forces are reflected to user's hand and prevent him/her from approaching the obstacles. Figure 4.31 represents the path followed by an micropipette during an actual experiment. The micropipette under control is shown at the start, in the middle and at the end of the path along with a fixed obstacle which is another micropipette in this case. The collision avoidance forces are also illustrated as arrows in Figure 4.31. The slave coordinates during the experiment are shown in Figure 4.32 in both $\mathcal{C}^{man}$ and $\mathcal{C}^{image}$.

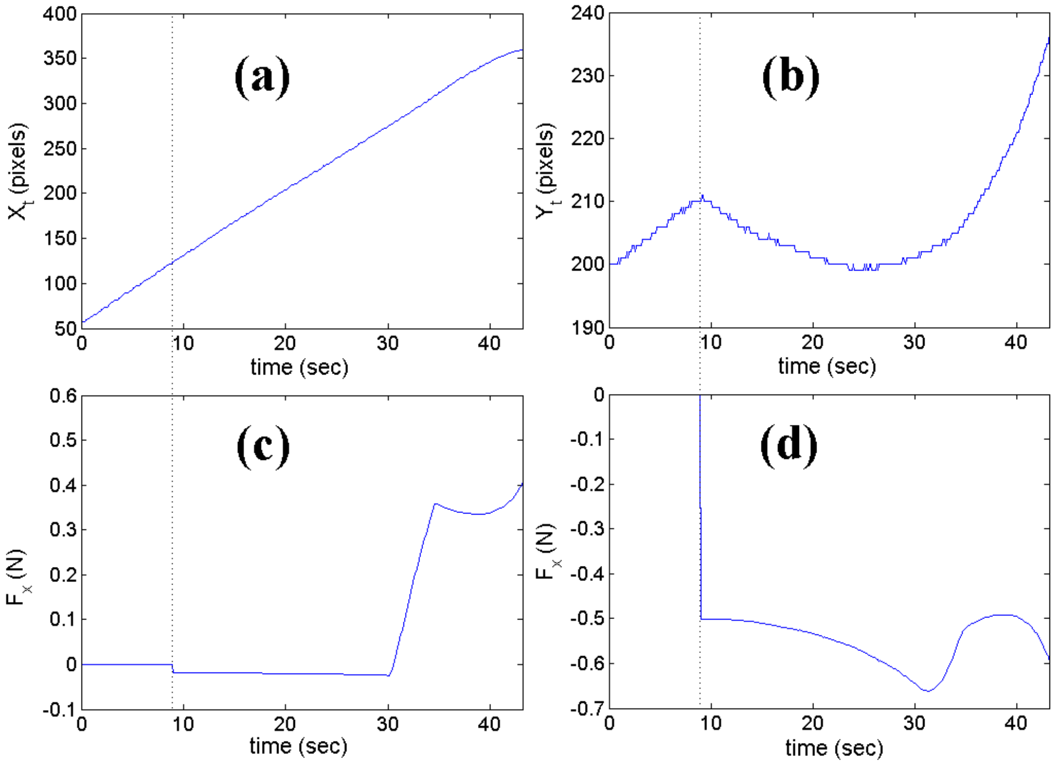

Figure 4.29: Results of visual servoing in proximity of an obstacle: (a), (b) represent X_t and Y_t which are the image-based tracking results and (c), (d) represent the collision avoidance forces in each direction. As observed in (c,d), the virtual forces are zero during the first 9.0 seconds because the minimum distance between the moving micropipette and the obstacle is beyond the specified margin.

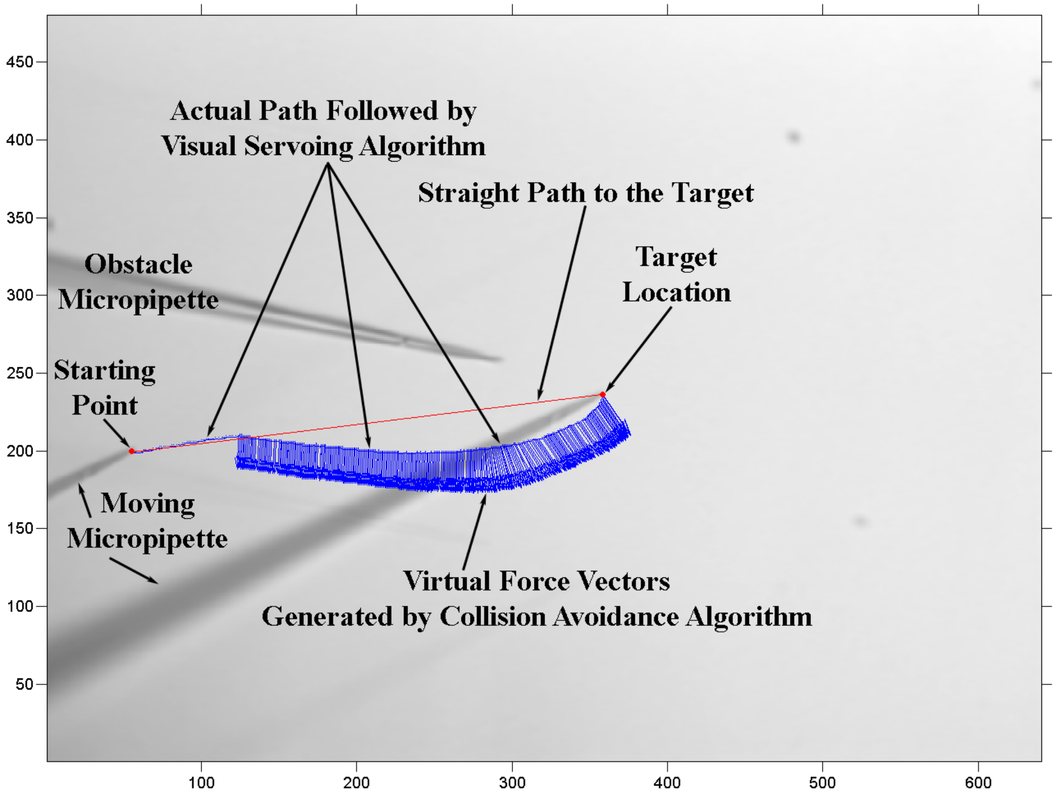

Figure 4.30: Visual servoing in proximity to an obstacle. The moving micropipette goes from the starting point to the target location under visual servoing control while collision avoidance forces cause the path to bend around the obstacle instead of following a straight line.

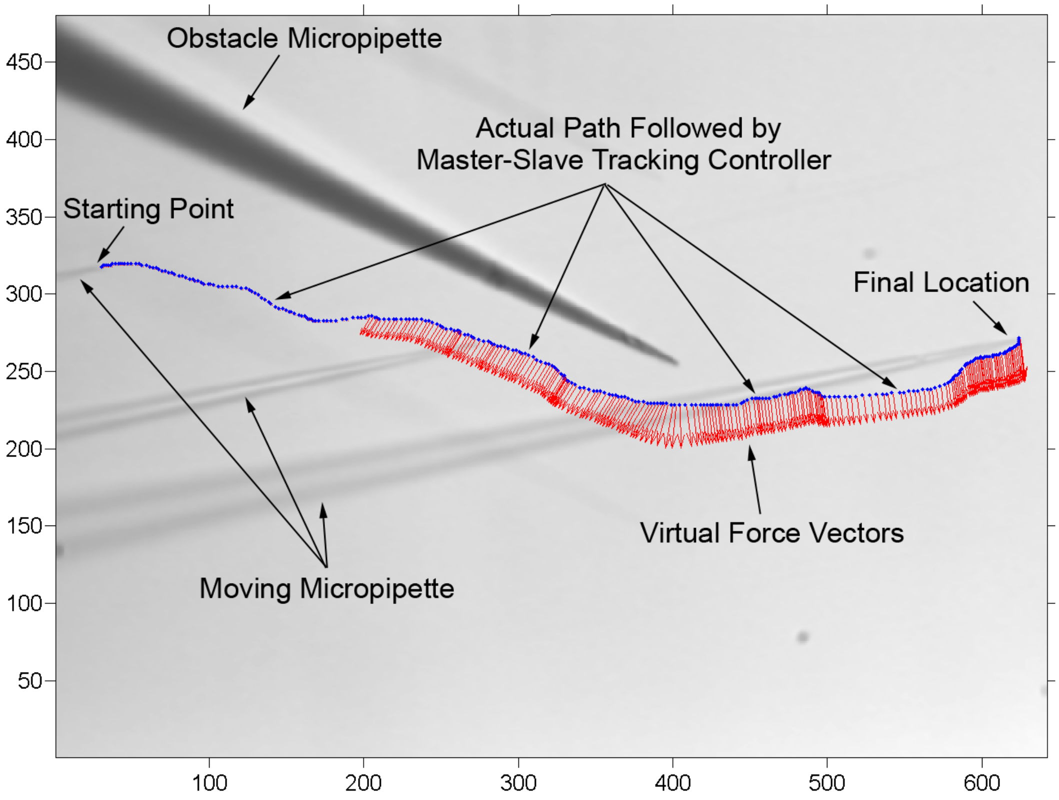

Figure 4.31: The trajectory of an micropipette while passing close to another micropipette is shown. Collision avoidance forces are represented as arrows. The length and orientation of each arrow shows the magnitude and orientation of the repulsion force at each point on the trajectory. The collision force is not applied to the haptic device when the minimum distance is out of a range as described in Section. 4.8. The controlled micropipette is shown at the beginning, in the middle and at the end of the trajectory. The 4× dry objective is used in this experiment.

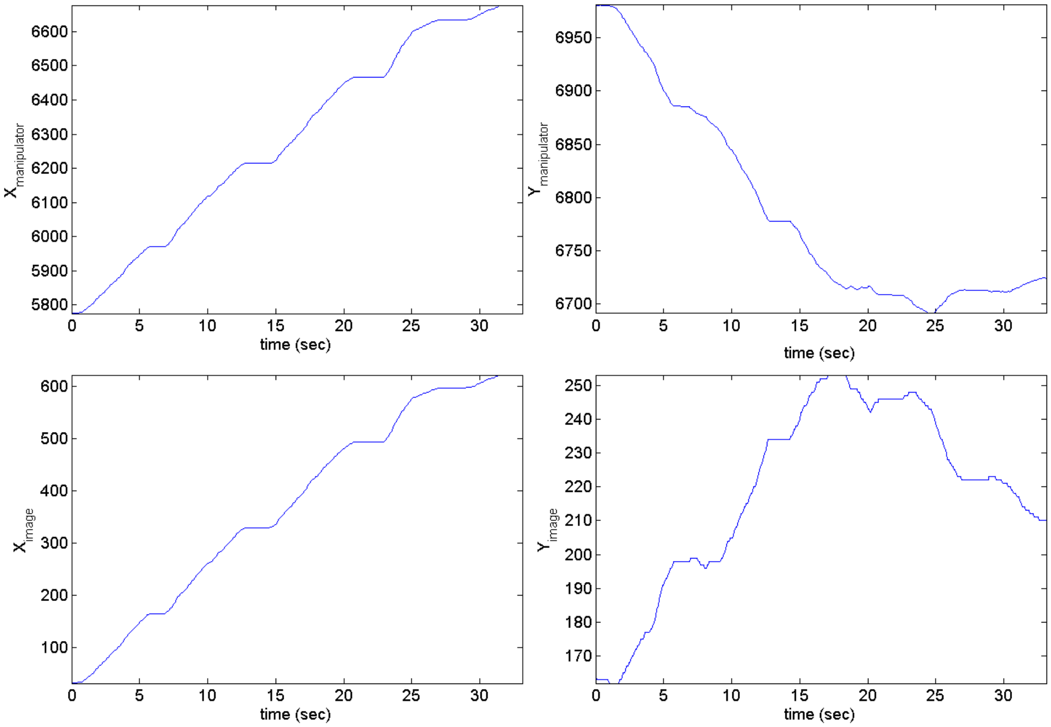

Figure 4.32: Slave coordinates in $\mathcal{C}^{man}$ and $\mathcal{C}^{image}$.

The master-slave tracking results and the forces applied to the user's hand are shown in Figure 4.33. The results show very good tracking in the X and Y directions except for the fact that an inherent delay of around $100msec$ is observed. This is due to the closed architecture of the micromanipulator's controller. The slave does not follow the master along the Z direction as observed in Figure 4.33 because the system is in a planar control mode which is supposed to keep the micropipette in the focal plane. The force F_{h_z} is applied to the user's hand to limit the haptic device motion in a plane.

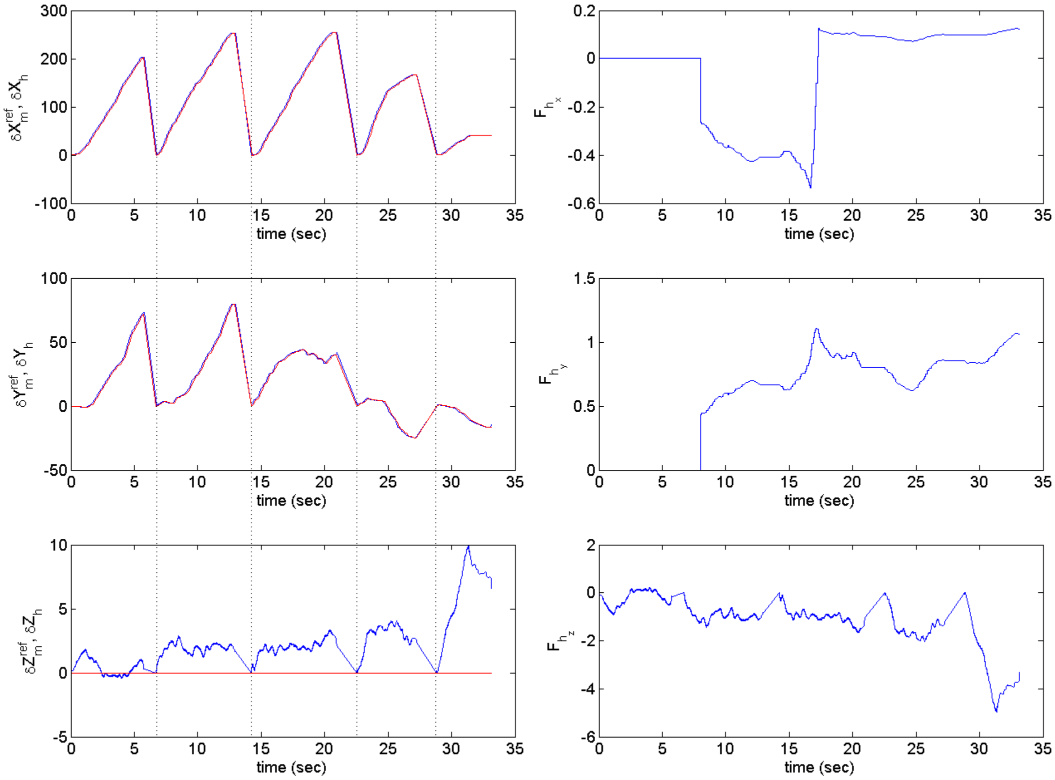

Figure 4.33: Master-slave tracking results: the left column shows the master (blue) vs. the slave (red) displacements along the X, Y and Z axes, the right column shows the haptic force along the same axes. Master-slave control was performed in planar mode where tracking is only done in the X and Y directions. The master coordinates are in mm while the slave coordinates are in μm.

4.12.7 3D Reconstruction

We have tested the developed system on a $200\mu m$ coronal mouse brain slice (piriform cortex). Brain slicing, incubation and slice perfusion were done according to a published methodology [54]. To show the validation of the algorithm, we report a sample case of $3D$ reconstruction by capturing 201 image slices (640×480, $0.2\mu m$ steps).

Accuracy

To validate the reconstruction results, two different references can be used: (a) a morphological reconstruction of the same cell by a confocal microscope after the experiments, and (b) manual segmentation of the image slices. The problem with the first method is that the reconstruction is performed after the experiments, i.e., after the cell goes through deformations when it is being touched by a micropipette and has been fixed for more than 24 hours; It is also necessary that the cell is patched successfully and the contrast dye is spread in the cell. Therefore, the first method cannot provide a reliable reference to evaluate the accuracy of our algorithm. Considering these limitations, we chose the second approach. The user segments the cell borders in each image slice by moving the mouse pointer and clicking on the images. This method provides a more reliable reference, although it requires some time and effort to generate the reference point sets.

We consider the result of automatic reconstruction and manual segmentation as sets of points in three dimensional space, as $\mathcal{W}_a$ and $\mathcal{W}_m$ respectively. The reconstruction error is then defined as:

$$\mathcal{E} = \frac{1}{|\mathcal{W}_a|} \sum_{p_a \in \mathcal{W}_a} \mathcal{D}(p_a, \mathcal{W}_m) \tag{4.26}$$

where $|\mathcal{W}_a|$ is the cardinality (number of members) of $\mathcal{W}_a$ and $\mathcal{D}(p_a, \mathcal{W}_m)$ is the distance between the point p_a and the set $\mathcal{W}_m$ defined as:

$$\mathcal{D}(p_a, \mathcal{W}_m) = \min_{p_m \in \mathcal{W}_m} ||p_a, p_m||_2 \tag{4.27}$$

user id	1	2	3
$\mathcal{E}(\mu m)$	0.9227	1.0302	0.4615

Table 4.5: Accuracy with respect to manual segmentation. The results are averaged for 3 different cases.

It is difficult to find a very exact reference to evaluate the reconstruction error because the cell borders are not sharp. It can be observed in Figure 4.15 that there is not a single contour that can be considered as cell membrane in each slice. Therefore we need to study the robustness of the defined error measure $\mathcal{E}$. We asked 3 different users (non-experts) to manually segment the images in the slices for 3 different image volumes and then we calculated the errors, the results are reported in Table 4.5. Each user chose a different strategy for segmentation: (1) The first user chose to segment the cell in slices where it was fully visible, (2) the second user chose to segment the cell every 4 slices (as adjacent slices are similar), (3) the third user segmented the cell in every slice. The users did not have a priori knowledge of the automatic segmentation results.

In all three different cases, the segmentation error is around $1\mu m$ or less. The manual segmentation performed by the third user is used as a reference as it is performed on all of the slices in the corresponding image volume. The reconstruction error $\mathcal{E}$, only evaluates the accuracy of the automatic segmentation and it does not determine what portion of the cell surface has been reconstructed. The number of reconstructed points is also reported to provide a rough estimate of the portion of the reconstructed cell membrane.

Speed

It only takes $T_0 = 0.837sec$ for the robot to move the microscope $40\mu m$ vertically. If the images are captured while the robot is moving, we can capture the whole volume very quickly in about $6.7sec$ which is $\max(T_0, N/f)$, where $N = 201$ is the number of slices and $f = 30Hz$ is the capturing frequency. However, there are major drawbacks that prevent us from using this method: (a) There will be motion artifacts in the images, (b) the vertical location of each image slice will depend on the time when it is captured. Therefore, the microscope is moved vertically and stops to capture each image slice. Capturing 201 images takes $53.527 sec$ which is mainly because the robot should stop 200 times to capture an image when the microscope is not in motion.

Accuracy vs. Speed

There is a trade-off between the accuracy and the speed of the algorithm. We have studied this trade-off for the two cases of down-sampling and ROI processing.

- Down-sampling:

  To increase the speed of the algorithm we can (a) down-sample the images in $2D$, and (b) increase the slice thickness (i.e., the distance between two image slices). This will reduce the time required for computation and also reduce the capturing time. If the image volume is down-sampled by a factor of n, the spatial resolution is decreased by the same factor and it is expected that $\mathcal{E}$ will also increase. If n is increased more than a certain threshold, the cell will not be detectable anymore and the algorithm fails to detect the cell membrane and the error increases drastically. The trade-off between the accuracy and speed of the algorithm is represented in Table 4.6, the results are averaged for three different image volumes.

  The reconstruction algorithm generates an empty set for $n = 8$ as the cell is not detectable in a 80×60 image. When n is increased, $\mathcal{E}$ increases while T_t decreases, as observed in Table 4.6; Therefore there should be an optimal scaling factor to compromise between reconstruction error and time. We define an optimization criterion to determine the optimal scaling factor.

  $$n_{opt} = \arg\min_{n} E(n), \ \ E(n) = \mathcal{E}(n).\sqrt{T_t(n)} \tag{4.28}$$

  The optimization criterion is chosen to minimize the reconstruction error and the total reconstruction time simultaneously. The square root of

scale $(1/n)$	1	1/2	1/4	1/8	ROI processing
$\mathcal{E}(\mu m)$	0.4615	0.6749	1.0424	∞	2.1531
$T_c(sec)$	53.527	26.845	13.495	7.571	53.527
$T_r(sec)$	49.35	6.682	1.363	0.699	4.213
$T_t(sec)$	102.877	33.527	14.858	8.27	57.74
E	4.6809	3.9047	4.0180	∞	Not defined
N_a	2442	687	96	0	6565

Table 4.6: Accuracy vs. speed for the reconstruction algorithm, n is the downsampling factor, $\mathcal{E}$ is the reconstruction error, T_c, T_r are the capturing and reconstruction time respectively. T_t is the overall time required for capturing and reconstruction and E is the optimization criterion defined in Equation. 4.28. The last column reports the results for the case of processing a region of interest. The last row represents the number of points in the reconstructed surface for each case; Number of manual segmentation points was $N_m = 19441$.

scale $(1/n)$	Volume size	Voxel Size
1	$640 \times 480 \times 201$	$.17 \times .17 \times 0.2\mu m^3$
1/2	$320 \times 240 \times 100$	$.34 \times .34 \times .4\mu m^3$
1/4	$160 \times 120 \times 50$	$.68 \times .68 \times .8\mu m^3$
1/8	$80 \times 60 \times 25$	$1.32 \times 1.32 \times 1.6\mu m^3$
ROI	$128 \times 128 \times 201$	$.17 \times .17 \times .2\mu m^3$

Table 4.7: Volume sizes for the reconstruction algorithm in different cases reported in Table 4.6

the overall reconstruction time T_t is used to put more emphasis on the reconstruction error $\mathcal{E}$. As observed in the last row of Table 4.6, $n_{opt} = 2$ results in an optimal value. It is a good compromise compared to $n = 1$, because we gain almost 4 times faster reconstruction time T_t while having only 30% degradation of the accuracy $\mathcal{E}$.

- *ROI* processing:

  We have used the generalized Hough transform to roughly segment a cell membrane as a circle. This can also be replaced by asking the user to click on a point where a cell is located. We can then select a region of interest containing the target cell and perform the processing only on that specific region. We have selected a 128×128 pixels square as the region of interest. This covers a $21.76\mu m \times 21.76\mu m$ area which covers most of neuron cells[4].

  This method does not reduce the time required for capturing images (T_c) and only affects the processing time T_r. The results are shown in the last column of Table 4.6. The resulting warped surface is also represented in Figure 4.34.

  This method results in reconstruction of more points and shows a more complete 3D shape of the cell as shown in Figure 4.34. At the same time, it results in increased reconstruction error. This method can be used in combination with down-sampling to speed up the reconstruction even further.

4.12.8 Patch Clamping Results

The procedure of patch clamping with the assistance of the developed platform includes: (a) system calibration using the dry objective; (b) locating the region of the slice which contains the appropriate cells; (c) autonomous placement of the micropipettes at the user specified locations by using calibration/registration information, visual servoing or the haptic device or a

---

[4] Average diameter of neuron cells is around $10\mu m$.

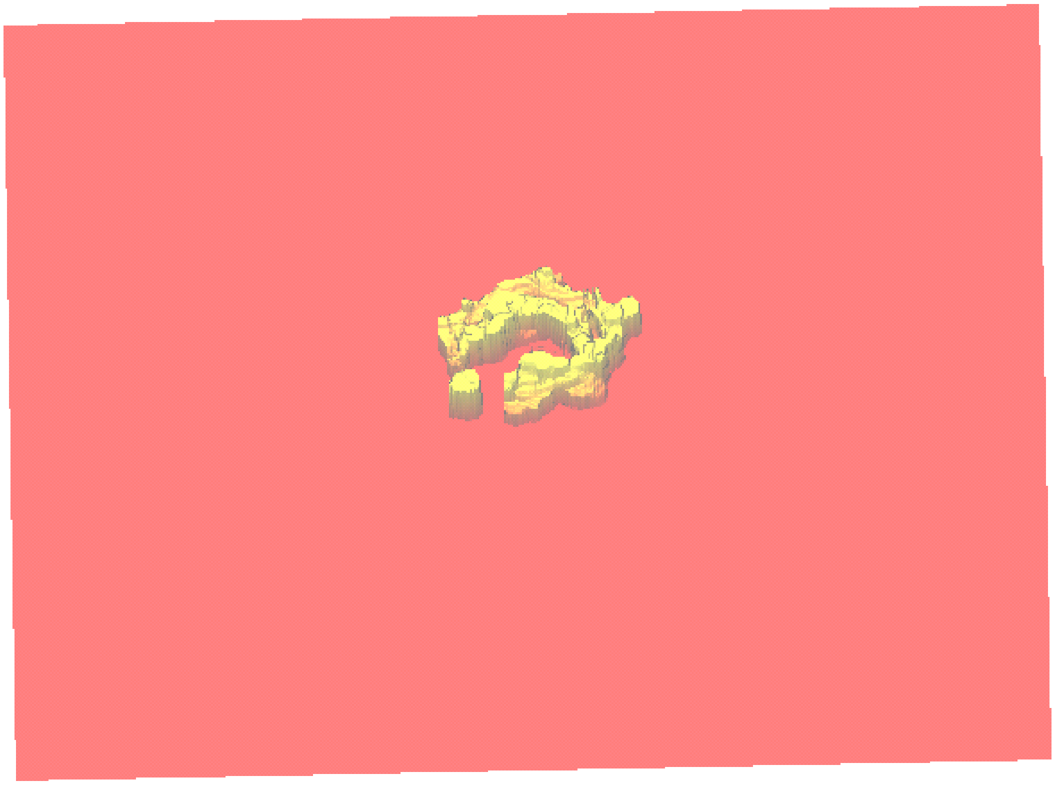

Figure 4.34: Reconstructed surface in the case of *ROI* processing.

combination of these methods; (d) changing the objective to water immersion lens automatically [13]; (e) bringing the objective and micropipettes down close to target cells automatically; and (f) using the haptic device to approach each cell membrane individually in order to perform patch clamping.

4.12.9 Brain slicing procedures and tissue maintenance

Coronal rat brain slices ($350\mu m$; 1.5 to $-0.3mm$ relative to bregma) were performed according to published methodology [54]. Slicing, incubation, and storage are all performed in choline solution. Ringer's solution used during electrical recordings is similar to choline solution except that pyruvate and ascorbate are removed, equimolar $NaCl$ replaces the choline Cl, and $MgCl_2$ is used at a $2mM$ concentration. All solutions are maintained at pH 7.4 and bubbled with $5\% - CO_2$ / $95\% - O_2$ (carbogen).

4.12.10 Electrophysiology

Patch electrodes were pulled from borosilicate glass capillaries ($10cm$ length, $1.5mm$ O.D. and $1.17mm$ I.D.) and filled with K-gluconate solution (300 mOsm, pH 7.3-7.4). Voltage-gated currents and excitability of cells in layers II and III of piriform cortex were monitored by means of voltage-clamp and current-clamp protocols (PulseFit v 8.0; Heka, Germany). The results are shown in Figures 4.35.

4.12.11 Immunohistochemistry and image acquisition

In order to reconstruct the morphology and understand where the recordings are made, biocytin was included in patch microelectrode solution. After the completion of a recording, the slice was removed from the microscope chamber and fixed in 4% paraformaldehyde (PFA) for at least 24 h. After rinsing [55] the slices were incubated in streptavidin-conjugated Alexa Fluor-594 ($5\mu g/ml$; Molecular Probes) and mounted onto Fisher SuperFrost slides and mounted with glass coverslips in Prolong Gold Antifade mounting medium (Molecu-

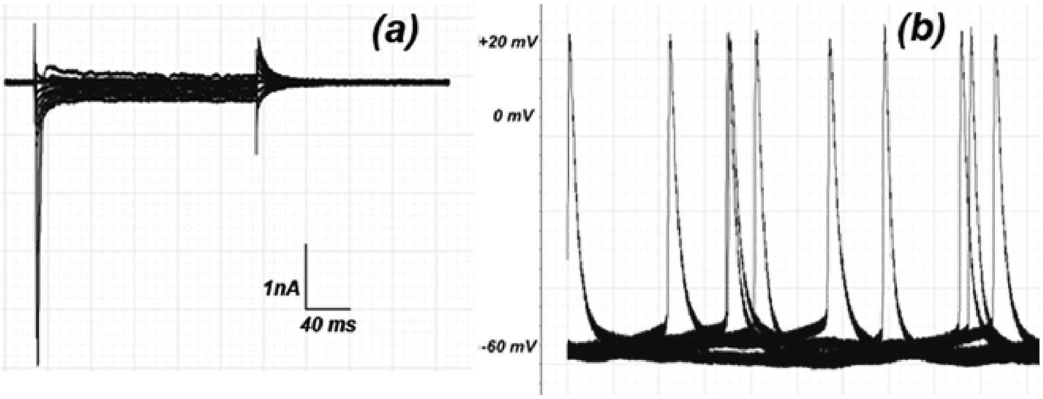

Figure 4.35: (a) Voltage clamp recording of a patch clamped interneuron (downward traces showing sodium current flow) and (b) Neuronal firing pattern of a patch clamped interneuron (upward traces indicate action potentials) in layer III of rat piriform cortex.

lar Probes, Eugene, OR). Confocal images were taken on an Olympus IX 60 inverted microscope outfitted with a Perkin Elmer Spinning Disk Confocal attachment with a 20× ($N.A. = 0.50$) objective. The microscope was equipped with a Hamatsu Orca ER CCD camera (1300×1030 pixels), and images were acquired in Volocity software (Improvision, Lexington, MA). Each image represents a stack of 40–50 images $0.2 \mu m$ apart in the z-plane. For morphological reconstruction of the dendritic arborization of patched neurons, the stacks of confocal images were deconvolved with AutoQuant software (AutoQuant Imaging, Media Cybernetics, Inc., Bethesda, MD) and then processed with Imaris Filament Tracer module in *surpass mode* (Bitplane, Zurich, Switzerland). To mark the cell bodies, an *Isosurface* was then created. This process creates a cell body from the stack of images that is then merged with the dendritic morphology (Figure 4.36).

The system calibration and visual servoing technique are used to bring MP tips close to target cells and the master-slave control is used to touch the membranes in order to patch them. Figure 4.37 shows the four patched micropipettes in the tissue slice after the recording. Figure 4.38 shows an micropipette tip pushing against a cell membrane when the cell is ready to be patched. Figure 4.39 shows four micropipettes patching four neurons while

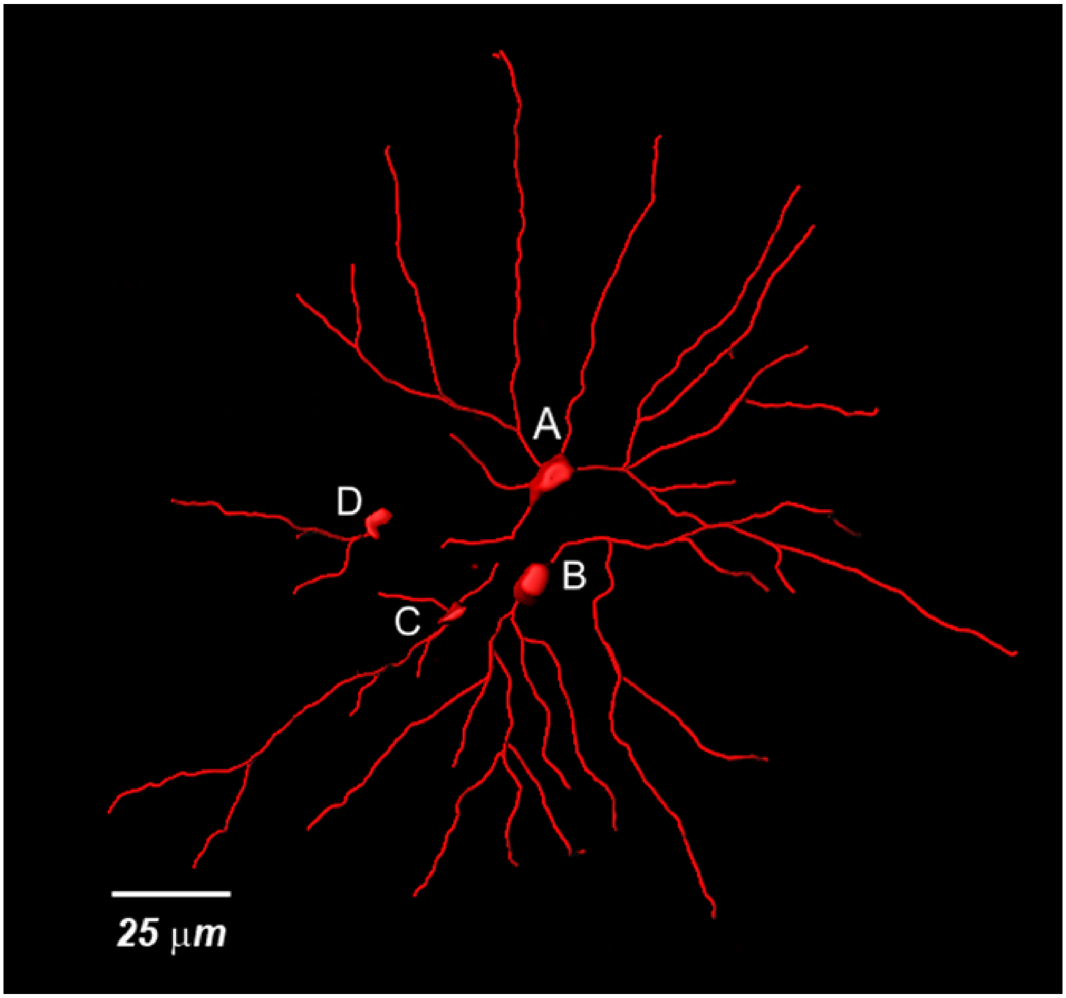

Figure 4.36: Morphological reconstruction of four interneurons in layer III of piriform cortex as shown in Figure 4.37. Partial cell body and dendritic arborization reconstruction for cell C and D indicate a lighter biocytin staining of these neurons.

the recording is being performed.

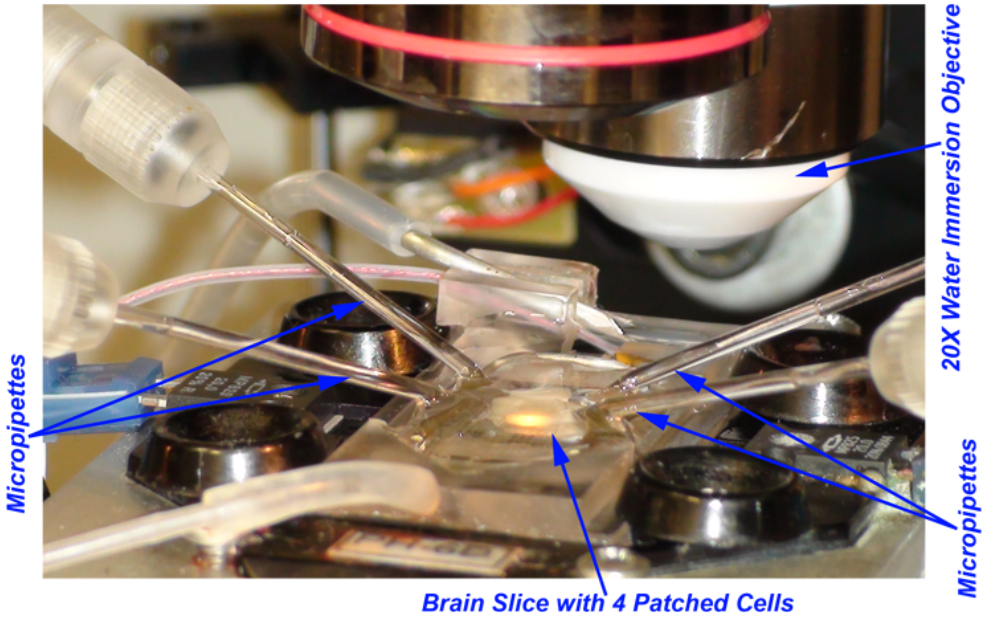
20X Water Immersion Objective
Micropipettes
Micropipettes
Brain Slice with 4 Patched Cells

Figure 4.37: Closer view of micropipettes, objective lenses and the substrate when four pipettes are patched. The objective lens is changed in order to increase visibility.

4.13 Conclusion

To the best of our knowledge, this is the first computer-integrated robot-assisted patch clamping system that is able to work with existing patch clamp setups while requiring minimal hardware modifications. The experimental results show very accurate autofocusing for depth estimation and system calibration. This makes it easy for the user to find and focus on desired objects under the microscope. The micropipette tip detection algorithm helps in automating the calibration and coordinate registration procedure. The developed real-time micropipette tip tracking along with the calibration information, makes it pos-

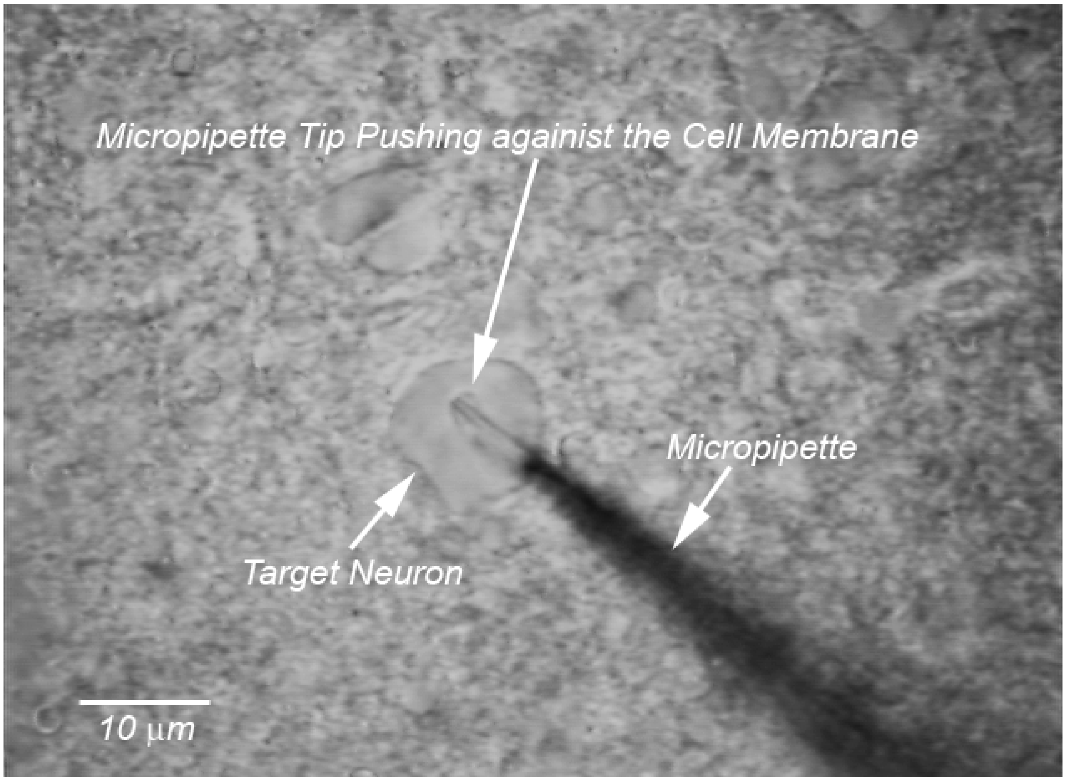

Figure 4.38: Micropipette tip is pushing against a cell membrane, the cell is ready for patch clamping.

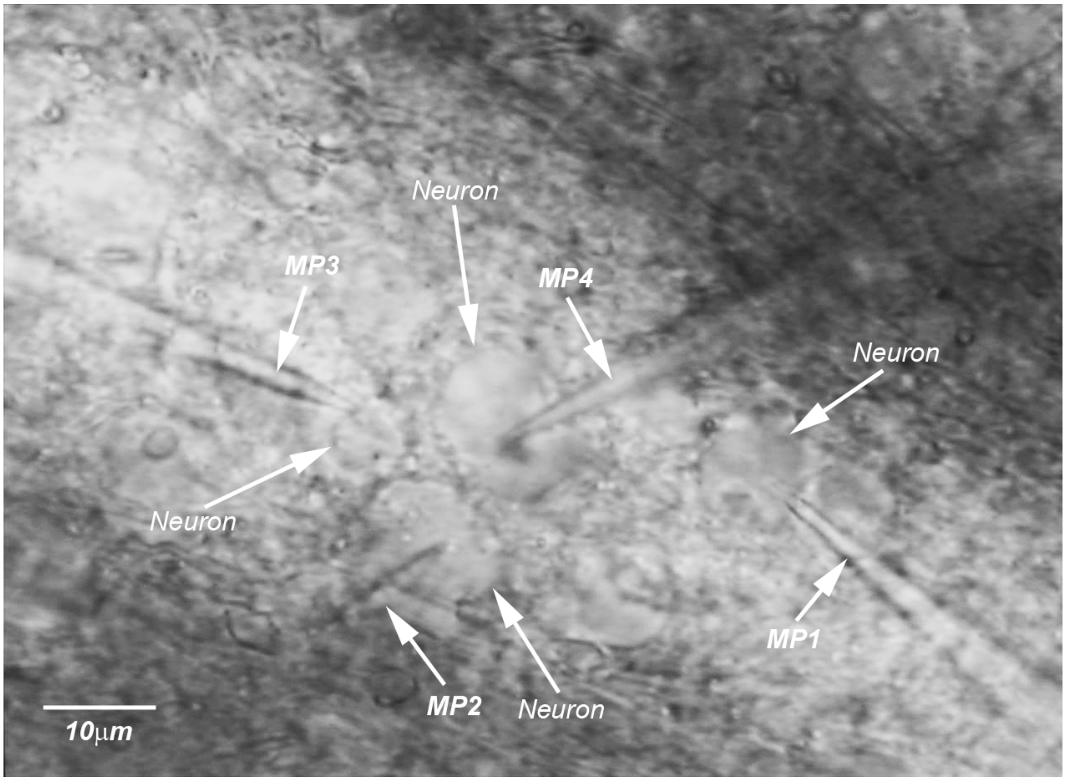

Figure 4.39: Four neurons patched.

sible to locate the micropipette tip at a desired location by just clicking on that point. The overall system makes it easy for a user to perform successful patch clamping very quickly by providing automated/semi-automated tools to select the desired cells and move the micropipettes into close proximity of the cells and then using the haptic device to patch the cell membranes.

Bibliography

[1] Areles Molleman. *Patch Clamping: An Introductory Guide to Patch Clamp Electrophysiology*. Wiley, 2003.

[2] Erwin Neher and Bert Sakmann. Single-channel currents recorded from membrane of denervated frog muscle fibres. *Nature*, 260:799–802, 1976.

[3] Cecilia Farre, Michael George, Andrea Bruggemann, and Niels Fertig. Ion channel screening - automated patch clamp on the rise. *Drug Discovery Today: Technologies*, 5(1):e23 – e28, 2008.

[4] Kenneth A. Jones, Nicoletta Garbati, Hong Zhang, and Charles H. Large. *Automated Patch Clamping Using the QPatch*, volume 565 of *Methods in Molecular Biology*, chapter 10, pages 209–223. Humana Press, 2009.

[5] Y. Sun and B. J Nelson. Biological Cell Injection Using an Autonomous MicroRobotic System. *The International Journal of Robotics Research*, 21(10-11):861–868, October 2002.

[6] Z. Zhang, M.a. Ferenczi, and C.R. Thomas. A micromanipulation technique with a theoretical cell model for determining mechanical properties of single mammalian cells. *Chemical Engineering Science*, 47(6):1347–1354, 1992.

[7] H Mashmoushy, Z Zhang, and C R Thomas. Micromanipulation measurement of the mechanical properties of baker's yeast cells. *Biotechnology Techniques*, 12(12):925–929, 1998.

[8] W.H. Wang, X.Y. Liu, and Y. Sun. Contact Detection in Microrobotic Manipulation. *The International Journal of Robotics Research*, 26(8):821–828, August 2007.

[9] W.H. Wang, X.Y. Liu, and Y. Sun. Autonomous Zebrafish Embryo Injection Using a Microrobotic System. In *Proceedings of IEEE International Conference on Automation Science and Engineering*, pages 363–368, 2007.

[10] Yu Sun and B.J. Nelson. Microrobotic cell injection. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 620–625, 2001.

[11] Wenhui Wang, Xinyu Liu, Danielle Gelinas, Brian Ciruna, and Yu Sun. A fully automated robotic system for microinjection of zebrafish embryos. *PloS one*, 2(9):e862, January 2007.

[12] Yi Zhang, Kok Kiong Tan, and Sunan Huang. Vision-Servo System for Automated Cell Injection. *IEEE Transactions on Industrial Electronics*, 56(1):231–238, January 2009.

[13] M. Azizian, R. Patel, C. Gavrilovici, and M. Poulter. Image-guided robot-assisted microscope objective lens positioning: Application in patch clamping. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[14] P.T. Yap and P. Raveendran. Image focus measure based on chebyshev moments. In *IEE Proceedings on Vision, Image and Signal Processing*, volume 151, pages 128–136, April 2004.

[15] Muhammad Riaz, Seungjin Park, Muhammad Bilal Ahmad, Waqas Rasheed, and Jongan Park. *Generalized Laplacian as Focus Measure*, volume 5101 of *Lecture Notes in Computer Science*, pages 1013–1021. Springer Berlin / Heidelberg, 2008.

[16] Loren Shih. Autofocus survey: a comparison of algorithms. volume 6502. SPIE, 2007.

[17] Yu Sun, S. Duthaler, and B.J. Nelson. Autofocusing Algorithm Selection in Computer Microscopy. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, pages 419–425, 2005.

[18] Murali Subbarao and Jenn Tyan. Selecting the optimal focus measure for autofocusing and depth-from-focus. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20:864–870, 1998.

[19] M. Azizian, R. Patel, C. Gavrilovici, and M. Poulter. Computer-assisted patch clamping. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 4131 –4136, 2010.

[20] Mahdi Azizian, Rajni Patel, Cezar Gavrilovici, and Michael O. Poulter. Image processing techniques in computer-assisted patch clamping. In *SPIE Imaging, Manipulation, and Analysis of Biomolecules, Cells, and Tissues VIII*, volume 7568, pages 75681E1–12, 2010.

[21] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*, pages 117–121. Cambridge University Press, 2nd edition, 2003.

[22] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Transactions of ACM - Graphics and Image Processing*, 24(6):381–395, 1981.

[23] John Russ. *Image Processing Handbook*, pages 472–476. CRC Press Inc., 5th edition, 2007.

[24] Rafael C. Gonzalez and Richard Eugene Woods. *Digital Image Processing*, pages 733–738. Prentice-Hall, Inc., 3rd edition, 2006.

[25] C. Galambos, J. Kittler, and J. Matas. Progressive probabilistic hough transform for line detection. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 560 Vol. 1, 1999.

[26] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4), 2006.

[27] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003.

[28] B.S. Reddy and B.N. Chatterji. An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing*, 5(8):1266–1271, 1996.

[29] H. Foroosh, J.B. Zerubia, and M. Berthod. Extension of phase correlation to subpixel registration. *IEEE Transactions on Image Processing*, 11(3):188–200, 2002.

[30] Hongshi Yan and Jian Guo Liu. Robust phase correlation based motion estimation and its applications. In *British Machine Vision Conference (BMVC)*, 2008.

[31] James B. Pawley. *Handbook of Biological Confocal Microscopy*, pages 145–161. Springer, 3rd edition, 2006.

[32] Qiang Wu, Fatima Merchant, and Kenneth Castleman. *Microscope Image Processing*. Elsevier Academic Press, 2008.

[33] Samuel R. Buss. Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares Methods. `http://math.ucsd.edu/~sbuss/ResearchWeb/ikmethods/iksurvey.pdf`, 2004.

[34] C.C. Cheah, C. Liu, and J.J.E. Slotine. Adaptive jacobian tracking control of robots based on visual task-space information. In *Robotics and Automation, ICRA. Proceedings of the IEEE International Conference on*, pages 3498–3503, 2005.

[35] Oussama Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.

[36] Jing Ren, Kenneth A. McIsaac, and Rajni V. Patel. Modified newton's method applied to potential field-based navigation for nonholonomic robots in dynamic environments. *Robotica*, 26(1):117–127, 2008.

[37] N.J. Cowan, J.D. Weingarten, and D.E. Koditschek. Visual servoing via navigation functions. *Robotics and Automation, IEEE Transactions on*, 18(4):521–533, 2002.

[38] Peter F. Hokayem and Mark W. Spong. Bilateral teleoperation: An historical survey. *Automatica*, 42(12):2035–2057, December 2006.

[39] M. Tavakoli, A. Aziminejad, R.V. Patel, and M. Moallem. Discrete-time bilateral teleoperation: modelling and stability analysis. *Control Theory and Applications, IET*, 2(6):496–512, 2008.

[40] Thurston L. Brooks. Telerobotic Response Requirements. In *Systems, Man and Cybernetics, 1990. Conference Proceedings., IEEE International Conference on*, pages 113–120, Nov 1990.

[41] M. Azizian, R. V. Patel, C. Gavrilovici, and M. O Poulter. Computer-assisted patch clamping. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[42] M. Azizian, R. V. Patel, C. Gavrilovici, and M. O Poulter. Image-guided robot-assisted microscope objective lens positioning: Application in patch clamping. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[43] J. Markham and J. Conchello. Parametric blind deconvolution: a robust method for the simultaneous estimation of image and blur. *Journal of the Optical Society of America*, 16(10):2377–2391, 1999.

[44] Jonathan R. Monck, Andres F. Oberhauser, Thomas J. Keating, and Julio M. Fernandez. Thin-section ratiometric ca2+ images obtained by optical sectioning of fura-2 loaded mast cells. *The Journal of Cell Biology*, 116(3):745–759, 1992.

[45] S.K. Nayar and Y. Nakagawa. Shape from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):824 –831, 1994.

[46] Murali Subbarao and Tae Choi. Accurate revovery of three-dimensional shape from image focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:266–274, 1995.

[47] Markus Niederost, Jana Niederost, and Jiri Scucka. Automatic 3D reconstruction and visualization of microscopic objects from a monoscopic multifocus image sequence. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIV-5/W10, 2003.

[48] Mehmet Sezgin and Bulent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–168, 2004.

[49] J.N. Kapur, P.K. Sahoo, and A.K.C. Wong. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing*, 29(3):273 – 285, 1985.

[50] Peter J. Burt, Edward, and Edward H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31:532–540, 1983.

[51] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62 – 66, 1979.

[52] Visualization Toolkit. `http://www.vtk.org`.

[53] KWWidgets GUI Toolkit. `http://www.kwwidgets.org`.

[54] Cezar Gavrilovici, Sabrina D'Alfonso, Mathew Dann, and Michael O. Poulter. Kindling-induced alterations in gabaa receptor mediated inhibition and neurosteroid activity in the piriform cortex of rat. *European Journal of Neuroscience*, 24(5):1373–1384, 2006.

[55] Cezar Gavrilovici, Sabrina D'Alfonso, and Michael O. Poulter. Diverse interneuron populations have highly specific interconnectivity in the rat piriform cortex. *Journal of Comparative Neurology*, 518(9):1570–1588, 2009.

Chapter 5

Concluding Remarks and Future Work

Each of the three parts of the thesis in Chapters 2, 3 and 4 includes a conclusions section. In this chapter, overall concluding remarks and suggestions for future work are given. We have studied the application of image-guided robotics to provide shared control between a human operator (e.g., a cardiologist and a neurobiologist) and a computer. This was applied to three different cases in biomedical engineering - intravascular and intracardiac interventions and cell micromanipulation. The existing imaging modalities in each case (i.e., x-ray fluoroscopy, ultrasound and microscope imaging) have been efficiently used to provide feedback to the robot and/or the human operator. Master-slave control as well as supervised autonomous or semi-autonomous control schemes were developed to enable the user to perform the task according to the requirements of any specific task. Robot manipulators were controlled at scales ranging from multiple centimeters to sub-micrometers.

5.1 Remarks

Concluding remarks for the research described in each of the Chapters 2, 3 and 4 are included here.

5.1.1 Angioplasty

An image-guided robotic system was developed for catheter insertion in intravascular interventions. An image-based tracking algorithm was developed to track the distal tip of a catheter in real time. Image-based and magnetic tracking of the catheter tip were deployed through a sensor fusion scheme using Kalman filtering to provide more robust and reliable feedback of the catheter tip position. Feedback control (mainly based on visual servoing) was implemented for accurate positioning of the distal tip of a catheter inside a vascular model. An autonomous guidance algorithm was developed to guide the catheter to a target point in the vasculature. Shared control between the human operator and the computer is implemented which allows for different control modes including supervised autonomous control or teleoperated master-slave control. To the best of our knowledge, this is the first time visual servoing has been demonstrated for catheter insertion control with shared control for intravascular interventions. The results show accurate tracking and control of the distal tip of a catheter. Robot-assisted insertion of the catheter with autonomous guidance and master-slave control reduces the X-ray exposure to the clinicians significantly. However, the tracking algorithms need to be modified to compensate for beating heart and respiratory motions. A model-free controller has been used for catheter tip position control. Using a model that describes the nonlinear behavior of a flexible catheter and its interaction with vasculature, could help to improve the performance of the tip position controller.

5.1.2 Cardiac Ablation

A master-slave controlled robotic system was developed for actuation of conventional steerable ablation catheters. Online 3D visualization was also developed based on transthoracic ultrasound images captured using a motorized ultrasound probe. The position of the distal end of the catheter is controlled to follow the motion of a master device while live visual feedback of the beating heart and the catheter is provided to the user. The position of the ultrasound probe and the distal part of the catheter are measured using magnetic trackers. This is used to control the position of the catheter tip as well as orienting the master device coordinates with the visual feedback. The system described in this part of the thesis provides the necessary tools for implementing bilateral teleoperation with haptic feedback to provide a sense of touch to the user while performing cardiac ablation.

5.1.3 Patch Clamping

A microscope image-guided robotic system was developed for multiple electrode patch clamping. The developed system includes: (a) master-slave control of micropipettes with sub-micron motion resolution, (b) microscope image-based detection and tracking as well as visual servoing of the micropipette tips, (c) partial 3D reconstruction and online 3D visualization of neurons, (d) algorithms for collision avoidance and haptic feedback to avoid collisions among micropipettes (or between micropipettes and obstacles), (e) an autofocusing algorithm to provide relative depth perception, and (f) a graphical user interface that provides intuitive interaction with the user to control the system. This work also included the design and implementation of an objective lens changing mechanism as well as algorithms for calibration and registration. To the best of our knowledge, this is the first computer-integrated robot-assisted patch clamping system that is able to work with existing patch clamp setups while requiring minimal hardware modifications. This system provides tools to facilitate multiple-electrode patch clamping, however the interaction with the cell membrane is done manually. Automatic control of rupturing pressure by using resistance measurements and visual feedback can make the overall procedure faster.

5.2 Future Research

Possible directions for future research in each of the topics of the thesis are discussed below.

5.2.1 Angioplasty

A possible area of future work is the development of an adaptive data fusion scheme which can update the noise characteristics ($Q(k)$ and $R(k)$) based on the varying errors in the observed values. This will help to penalize the sensor having more uncertainty and favour the sensor with less uncertainty. The image-based tracking can also be modified to compensate for beating heart and respiratory motion in order to use it for clinical cases in percutaneous coronary angioplasty applications. The catheter tip position controller was a model-free PID controller. The accuracy of the controller and the amounts of overshoot/undershoot at the distal tip of a catheter can be improved by model-based compensation of some of the nonlinearities including deadband. It would also be helpful to obtain an estimation of friction forces between the catheter and the vasculature during insertion, based on the robot end-effector force sensor readings as well as the shape of the vasculature (as provided by the roadmap). Active compensation of this friction force can result in much smoother catheter insertion. Although the proof of concept was done using a commercial robot, it would be worthwhile to design a catheter insertion robot small enough for clinical tests in an operating room.

5.2.2 Cardiac Ablation

Evaluation of the system by comparing experimental results of this system to those obtained by conventional methods using pre-operative images can be part of future research. The system should be evaluated through several experiments and the results should be compared at least for three different cases: (a) with and without stereo vision, (b) with and without intra-operative imaging, and (c) with master-slave control and with manual actuation of the catheter. Positioning accuracy should be studied in each case. It is anticipated that the developed system will improve the positioning accuracy in each case.

We have used simple and fast 2D pre-processing of ultrasound image slices. However, 3D volumetric processing after 3D reconstruction is shown [1] to provide better quality 3D ultrasound images compared to 2D image processing techniques applied before reconstruction. Volumetric processing is more time consuming and computationally intensive; therefore a possible area of future research could be using Graphics Processing Units (GPUs) to achieve higher quality reconstructed 3D ultrasound images in real time.

It is also expected that modeling the nonlinearities of steerable catheters results in more accurate control of the distal tip position. Future research could include modeling the nonlinear behavior of the steerable catheter and design of controllers to compensate for these nonlinearities. Deployment of a force sensing catheter and providing force feedback to the user can also be a future research direction.

5.2.3 Patch Clamping

Studying the effect of light refraction by brain tissue slices is one possible research topic. This has a nonlinear effect on the accuracy of depth estimation when the micropipette tip moves inside the tissue. Currently the system provides facilities for moving micropipettes to desired locations, but the actual patch needs to be done by a human operator. Development of a more complicated control scheme to perform patch clamping automatically by visually guided contact force control between a micropipette and a cell can also be the subject of future research.

5.3 Towards supervised automation in biomedical robotics

Although a majority of the applications of visual servoing have been aimed at enabling the user (e.g. a clinician) to perform a procedure more efficiently, there is interest in making certain procedures autonomous using visual servoing and under supervision of a human operator. Development of visual servo controllers depends greatly on the robustness of image-based detection and tracking algorithms. While there have been many developments in image processing and computer vision techniques, there are several drawbacks in terms of using imaging to automate certain surgical procedures. As an example Nageotte et al. [2] have studied the kinematic analysis of an optimal stitching task which can be used to automate complicated tasks such as suturing and stitching, through visual servoing. The same phenomenon can be applied to cell micromanipulation and other biomedical applications. On the other hand, visual servoing can provide reliable feedback control without using complicated nonlinear models for flexible tools such as catheters and tool/tissue interactions. Availability of robust and reliable feedback of the position, orientation or deformations measured at the same location as the controlled variables, makes it possible to perform efficient feedback control with simplistic linear models or even without any model of the tool or tool/tissue interaction. We expect that advances in imaging and processing technologies will make it possible to meet the requirements listed in Section 5.3.1 and therefore, result in more robust and reliable visual servoing techniques. This will provide increasing applications of supervised automation of biomedical robotic procedures.

5.3.1 Visual servoing

A common theme in this thesis, mainly in Chapters 2 and 4 is the application of visual servoing. The results and issues in each case have been discussed in the corresponding chapters. We include a brief overall discussion on the main requirements and challenges of using visual servoing for biomedical robotic applications. The following is a list of these major requirements and challenges:

- Direct access to image data in real time
- Imaging resolution and tracking accuracy
- Computational efficiency of the image processing algorithms
- Image capture rate which is usually much lower than servo rate
- Robustness and reliability of image-based detection and tracking algorithms

- Effect of latency caused by the time required for image capture, transfer and processing in the control loop

Most of these items depend on the technology used for imaging and image capturing and processing. New technologies provide higher capturing and processing rates and less latency in the loop. However, the accuracy, robustness and reliability of image-based feedback is very much dependent on the algorithms used. Future research can be focused on developing more accurate and robust image-processing algorithms, data fusion and intelligent detection of faults in the algorithms while using new technologies which provide higher quality images and faster image capturing and processing.

5.3.2 Open access to raw imaging data

As Bachta et al. [3] have suggested, direct access to the medical imaging data in real time can boost the application of visual servoing techniques in various areas of medical robotics. Although few of the medical imaging products provide an open architecture for accessing the raw imaging data, specialized hardware and privileged access is required to access raw data for a majority of the currently available medical imaging devices [4]. Therefore several visual servoing applications still rely on analog video outputs which pose limitation such as a maximum $30Hz$ frame rate, low resolution and no control over the imaging system parameters. Novotny et al. [5] incorporated real-time processing of 3D ultrasound image volumes using a Philips Sonos 7500 machine through a privileged access which is not commercially available to the general research community. Open access to raw imaging data can provide more flexibility in the design of algorithms for tracking tools and/or organs.

Bibliography

[1] Flemming Forsberg, Vincenzo Berghella, Daniel A. Merton, Keith Rychlak, Joann Meiers, and Barry B. Goldberg. Comparing image processing

techniques for improved 3-dimensional ultrasound imaging. *Journal of Ultrasound in Medicine*, 29(4):615–619, 2010.

[2] F. Nageotte, P. Zanne, C. Doignon, and M. de Mathelin. Stitching Planning in Laparoscopic Surgery: Towards Robot-assisted Suturing. *The International Journal of Robotics Research*, 28(10):1303–1321, May 2009.

[3] Wael Bachta, Pierre Renaud, Ezio Malis, Koichi Hashimoto, and Jacques Gangloff. Visual Servoing for Beating Heart Surgery. *Visual Servoing via Advanced Numerical Methods*, 401:91–114, 2010.

[4] A. Krupa, G. Fichtinger, and G. D. Hager. Real-time Motion Stabilization with B-mode Ultrasound Using Image Speckle Information and Visual Servoing. *The International Journal of Robotics Research (IJRR)*, 28(10):1334–1354, May 2009.

[5] Paul M Novotny, Jeff a Stoll, Nikolay V Vasilyev, Pedro J del Nido, Pierre E Dupont, Todd E Zickler, and Robert D Howe. GPU based real-time instrument tracking with three-dimensional ultrasound. *Medical image analysis*, 11(5):458–64, October 2007.

Appendix A

Mitsubishi PA10 Robot

An ARCNET card (PCI-20U from Contemporary Controls Inc.) has been used to communicate with the Mitsubishi PA10-7CE control unit. Considering θ_i, $i = 0, 1, 2, 3, 4, 5, 6$ as the joint angles of the robot, we define $s_i = \sin(\theta_i)$ and $c_i = \cos(\theta_i)$ to make the equations concise.

A.1 Jacobian

Entries of the 6×7 Jacobian matrix, $J_{i,j}$ are calculated as follows:

$J_{0,0} = -a_3 s_6 c_5 c_4 s_1 c_2 c_3 - a_3 s_6 c_5 c_4 c_1 s_3 + a_3 s_6 c_5 s_1 s_2 s_4 + a_3 s_6 s_5 s_1 c_2 s_3 - a_3 s_6 s_5 c_1 c_3 - a_3 c_6 s_4 s_1 c_2 c_3 - a_3 c_6 s_4 c_1 s_3 - a_3 c_6 s_1 s_2 c_4 - a_2 s_4 s_1 c_2 c_3 - a_2 s_4 c_1 s_3 - a_2 s_1 s_2 c_4 - s_1 s_2 a_1$

$J_{0,1} = -c_1 (a_3 s_6 c_5 s_2 c_3 c_4 + a_3 s_6 c_5 c_2 s_4 - a_3 s_6 s_2 s_3 s_5 + a_3 c_6 s_2 c_3 s_4 - a_3 c_6 c_2 c_4 + a_2 s_2 c_3 s_4 - a_2 c_2 c_4 - c_2 a_1)$

$J_{0,2} = -a_3 s_6 c_5 c_4 c_1 c_2 s_3 - a_3 s_6 c_5 c_4 s_1 c_3 - a_3 s_6 s_5 c_1 c_2 c_3 + a_3 s_6 s_5 s_1 s_3 - a_3 c_6 s_4 c_1 c_2 s_3 - a_3 c_6 s_4 s_1 c_3 - a_2 s_4 c_1 c_2 s_3 - a_2 s_4 s_1 c_3$

$J_{0,3} = -a_3 s_6 c_5 s_4 c_1 c_2 c_3 + a_3 s_6 c_5 s_4 s_1 s_3 - a_3 s_6 c_5 c_1 s_2 c_4 + a_3 c_6 c_4 c_1 c_2 c_3 - a_3 c_6 c_4 s_1 s_3 - a_3 c_6 c_1 s_2 s_4 + a_2 c_4 c_1 c_2 c_3 - a_2 c_4 s_1 s_3 - a_2 c_1 s_2 s_4$

$J_{0,4} = -a_3 s_6 (s_5 c_4 c_1 c_2 c_3 - s_5 c_4 s_1 s_3 - s_5 c_1 s_2 s_4 + c_5 c_1 c_2 s_3 + c_5 s_1 c_3)$

$J_{0,5} = -a_3 (-c_6 c_5 c_4 c_1 c_2 c_3 + c_6 c_5 c_4 s_1 s_3 + c_6 c_5 c_1 s_2 s_4 + c_6 s_5 c_1 c_2 s_3 + c_6 s_5 s_1 c_3 + s_6 s_4 c_1 c_2 c_3 - s_6 s_4 s_1 s_3 + s_6 c_1 s_2 c_4)$

$J_{0,6} = 0$

$J_{1,0} = a_3s_6c_5c_4c_1c_2c_3 - a_3s_6c_5c_4s_1s_3 - a_3s_6c_5c_1s_2s_4 - a_3s_6s_5c_1c_2s_3 - a_3s_6s_5s_1c_3 + a_3c_6s_4c_1c_2c_3 - a_3c_6s_4s_1s_3 + a_3c_6c_1s_2c_4 + a_2s_4c_1c_2c_3 - a_2s_4s_1s_3 + a_2c_1s_2c_4 + c_1s_2a_1$

$J_{1,1} = -s_1(a_3s_6c_5s_2c_3c_4 + a_3s_6c_5c_2s_4 - a_3s_6s_2s_3s_5 + a_3c_6s_2c_3s_4 - a_3c_6c_2c_4 + a_2s_2c_3s_4 - a_2c_2c_4 - c_2a_1)$

$J_{1,2} = -a_3s_6c_5c_4s_1c_2s_3 + a_3s_6c_5c_4c_1c_3 - a_3s_6s_5s_1c_2c_3 - a_3s_6s_5c_1s_3 - a_3c_6s_4s_1c_2s_3 + a_3c_6s_4c_1c_3 - a_2s_4s_1c_2s_3 + a_2s_4c_1c_3$

$J_{1,3} = -a_3s_6c_5s_4s_1c_2c_3 - a_3s_6c_5s_4c_1s_3 - a_3s_6c_5s_1s_2c_4 + a_3c_6c_4s_1c_2c_3 + a_3c_6c_4c_1s_3 - a_3c_6s_1s_2s_4 + a_2c_4s_1c_2c_3 + a_2c_4c_1s_3 - a_2s_1s_2s_4$

$J_{1,4} = -a_3s_6(s_5c_4s_1c_2c_3 + s_5c_4c_1s_3 - s_5s_1s_2s_4 + c_5s_1c_2s_3 - c_5c_1c_3)$

$J_{1,5} = -a_3(-c_6c_5c_4s_1c_2c_3 - c_6c_5c_4c_1s_3 + c_6c_5s_1s_2s_4 + c_6s_5s_1c_2s_3 - c_6s_5c_1c_3 + s_6s_4s_1c_2c_3 + s_6s_4c_1s_3 + s_6s_1s_2c_4)$

$J_{1,6} = 0$

$J_{2,0} = 0$

$J_{2,1} = -a_3s_6c_5c_2c_3c_4 + a_3s_6c_5s_2s_4 + a_3s_6c_2s_3s_5 - a_3c_6c_2c_3s_4 - a_3c_6s_2c_4 - a_2c_2c_3s_4 - a_2s_2c_4 - s_2a_1$

$J_{2,2} = s_2(a_3s_6c_5s_3c_4 + a_3s_6c_3s_5 + a_3c_6s_3s_4 + a_2s_3s_4)$

$J_{2,3} = a_3s_6c_5s_2c_3s_4 - a_3s_6c_5c_2c_4 - a_3c_6s_2c_3c_4 - a_3c_6c_2s_4 - a_2s_2c_3c_4 - a_2c_2s_4$

$J_{2,4} = a_3s_6(s_5s_2c_3c_4 + s_5c_2s_4 + s_2s_3c_5)$

$J_{2,5} = a_3(-c_6c_5s_2c_3c_4 - c_6c_5c_2s_4 + c_6s_2s_3s_5 + s_6s_2c_3s_4 - s_6c_2c_4)$

$J_{2,6} = 0$

$J_{3,0} = 0$

$J_{3,1} = -s_1$

$J_{3,2} = s_2c_1$

$J_{3,3} = -c_3s_1 - c_2c_1s_3$

$J_{3,4} = c_2s_4c_1c_3 - s_3s_1s_4 + s_2c_4c_1$

$J_{3,5} = -c_2c_4s_5c_1c_3 - c_3c_5s_1 - c_5c_2c_1s_3 + s_2s_5s_4c_1 + s_3s_5s_1c_4$

$J_{3,6} = s_6c_5c_4c_1c_2c_3 - s_6c_5c_4s_1s_3 - s_6c_5c_1s_2s_4 - s_6s_5c_1c_2s_3 - s_6s_5s_1c_3 + c_6s_4c_1c_2c_3 - c_6s_4s_1s_3 + c_6c_1s_2c_4$

$J_{4,0} = 0$

$J_{4,1} = c_1$

$J_{4,2} = s_2 s_1$

$J_{4,3} = -c_2 s_1 s_3 + c_3 c_1$

$J_{4,4} = c_2 s_4 s_1 c_3 + s_2 c_4 s_1 + s_3 c_1 s_4$

$J_{4,5} = c_3 c_5 c_1 - c_5 c_2 s_1 s_3 - s_3 s_5 c_1 c_4 + s_2 s_5 s_4 s_1 - c_2 c_4 s_5 s_1 c_3$

$J_{4,6} = s_6 c_5 c_4 s_1 c_2 c_3 + s_6 c_5 c_4 c_1 s_3 - s_6 c_5 s_1 s_2 s_4 - s_6 s_5 s_1 c_2 s_3 + s_6 s_5 c_1 c_3 + c_6 s_4 s_1 c_2 c_3 + c_6 s_4 c_1 s_3 + c_6 s_1 s_2 c_4$

$J_{5,0} = 1$

$J_{5,1} = 0$

$J_{5,2} = c_2$

$J_{5,3} = s_2 s_3$

$J_{5,4} = -s_2 s_4 c_3 + c_2 c_4$

$J_{5,5} = s_2 c_4 s_5 c_3 + s_5 c_2 s_4 + c_5 s_2 s_3$

$J_{5,6} = -s_6 c_5 s_2 c_3 c_4 - s_6 c_5 c_2 s_4 + s_6 s_2 s_3 s_5 - c_6 s_2 c_3 s_4 + c_6 c_2 c_4$

where $a_0 = 0.317$, $a_1 = 0.450$, $a_2 = 0.480$ and $a_3 = 0.35$.

A.2 Gravity Compensation

τ_g is a vector of torques that compensates for the effect of gravity on the robot at each joint.

$\tau_g[0] = 0.0$

$\tau_g[1] = -(g(s_2(l_3(m_3 + m_4 + m_5 + m_6 + m_7) - m_3 ry_3 + m_2 rz_2 + c_4(l_5(m_5 + m_6 + m_7) - m_5 ry_5 + m_4 rz_4 + c_6(m_6 rz_6 + m_7(l_7 + rz_7))) - c_5(m_6 rz_6 + m_7(l_7 + rz_7)) s_4 s_6) + c_2(c_3(l_5(m_5 + m_6 + m_7) - m_5 ry_5 + m_4 rz_4 + c_6(m_6 rz_6 + m_7(l_7 + rz_7))) s_4 + (m_6 rz_6 + m_7(l_7 + rz_7))(c_3 c_4 c_5 - s_3 s_5) s_6)))$

$\tau_g[2] = g s_2((l_5(m_5 + m_6 + m_7) - m_5 ry_5 + m_4 rz_4 + c_6(m_6 rz_6 + m_7(l_7 + rz_7))) s_3 s_4 + (m_6 rz_6 + m_7(l_7 + rz_7))(c_4 c_5 s_3 + c_3 s_5) s_6)$

$\tau_g[3] = -(g(l_5(m_5 + m_6 + m_7) - m_5 ry_5 + m_4 rz_4 + c_6(m_6 rz_6 + m_7(l_7 + rz_7)))(c_3 c_4 s_2 + c_2 s_4)) + c_5 g(m_6 rz_6 + m_7(l_7 + rz_7))(-(c_2 c_4) + c_3 s_2 s_4) s_6$

$\tau_g[4] = g(m_6 rz_6 + m_7(l_7 + rz_7))(c_5 s_2 s_3 + (c_3 c_4 s_2 + c_2 s_4) s_5) s_6$

$\tau_g[5] = -(g(m_6 rz_6 + m_7(l_7 + rz_7))(c_6(c_5(c_3 c_4 s_2 + c_2 s_4) - s_2 s_3 s_5) + (c_2 c_4 - c_3 s_2 s_4) s_6))$

$\tau_g[6] = 0.0$

where $g = 9.81\frac{m}{s^2}$ is the gravity constant and $m_2 = 8.41Kg$, $m_3 = 3.51Kg$, $m_4 = 4.31Kg$, $m_5 = 3.45Kg$, $m_6 = 1.46Kg$, $m_7 = 0.24Kg + m_L$ are the link masses for PA10 robot and $m_L = 1.82Kg$ is the effective mass for the catheter actuation mechanism. $rz_2 = 0.06325m$, $ry_3 = 0.08944m$, $rz_4 = 0.04609m$, $ry_5 = 0.1647m$, $rz_6 = -0.03m$, $rz_7 = -0.029m$ are the centers of mass for the PA10 robot links. $l_1 = 0.115m$, $l_3 = 0.450m$, $l_5 = 0.50m$ and $l_7 = 0.08m$ are the offset distances of the robot links.

Appendix B

SMA-Actuated Catheter

The contents of this appendix has been published in [1] and [2].

An active catheter has been developed for easy guidance of the catheter into a branch based on an input received from the image processing algorithm, as shown in Fig. B.1. Shape memory alloy (SMA) wires have been employed in developing the active catheter. Three SMA wires, spaced around the catheter at 120^0, have been employed to obtain the bending of the active catheter in all directions. Using an average energy of $420J/cm^2$, these wires are microwelded to stainless steel pads in an Argon atmosphere, as shown in Fig. B.2. The stainless steel pads are obtained by laser machining a tube of inner diameter equal to the outer diameter of the catheter. The pads have an arc length of $1.5mm$ ensuring a gap of $0.67mm$ between each pad. These pads are then glued in place by means of medical grade epoxy glue. Multiple sections of SMA actuators have been integrated into the catheter to obtain varying bending angles and radii. It should be noted here that the developed active catheter is for proof-of-concept. The final design of the active catheter would have two lumens - one channel would serve as a guide for the wires of the SMA actuators and the second lumen would be hollow to allow for the passage of a stent as well as the balloon used for angioplasty. In an actual application, care needs to be taken to shield the SMA actuators from cardiac tissue. It has been reported in the literature that epithelial cells and cardiac tissue can withstand a maximum of 50^0C [3]. There are two ways to ensure that tissue is not damaged by the active catheter, first, by using an SMA of lower transition temperature from Martensite to Austenite phase [4] and second, by appropriate packaging of the active catheter to ensure that the tissue is insulated from the generated heat. We have packaged the catheter with a high temperature resistant polytetrafluoroethylene (PTFE) film (HM350 film from Saint Gobain Performance Plastics Corporation) to ensure that the SMA wires are appropriately shielded from the artery walls. Several tests were performed to check the insulation capability of the film. The temperature inside the insulation and on the outer surfaces of the film were monitored to check the performance of the PTFE film during operation of the active catheter in water at 37^0C. The active catheter was actuated to its maximum possible deformation for 10 seconds and cooled to an ambient temperature and this process was repeated for 5 cycles. The maximum temperature attained on the outer surface of the catheter was 43^0C and that inside the insulation was 39^0C, although the Martensite to Austenite transformation temperature was 85^0C, thereby confirming the insulation capability of the PTFE film. The model and control developed in this paper, however, remain the same for both cases with changes made only to the parameters of the model. The strain in each SMA wire changes continuously, therefore the active catheter is capable of bending in any direction in a continuous manner rather than in discrete steps. The closed loop control designed for the active catheter ensures that the orientation of the active catheter follows the desired reference, as determined by the autonomous guidance algorithm. In order to control bending in the active catheter, a mathematical description of the behavior of SMAs is required. A model based on the physics of the process has been proposed by Jayender et al. [5], [6].

B.0.1 Closed-loop Control

A closed-loop control of the active catheter using the robust $\mathcal{H}_\infty$ loop-shaping controller was developed by Jayender et al. [5], [6]; The results of which are summarized in Table B.1. The difference between this controller and the previously developed controller in [6] is that the strain in the SMA actuators

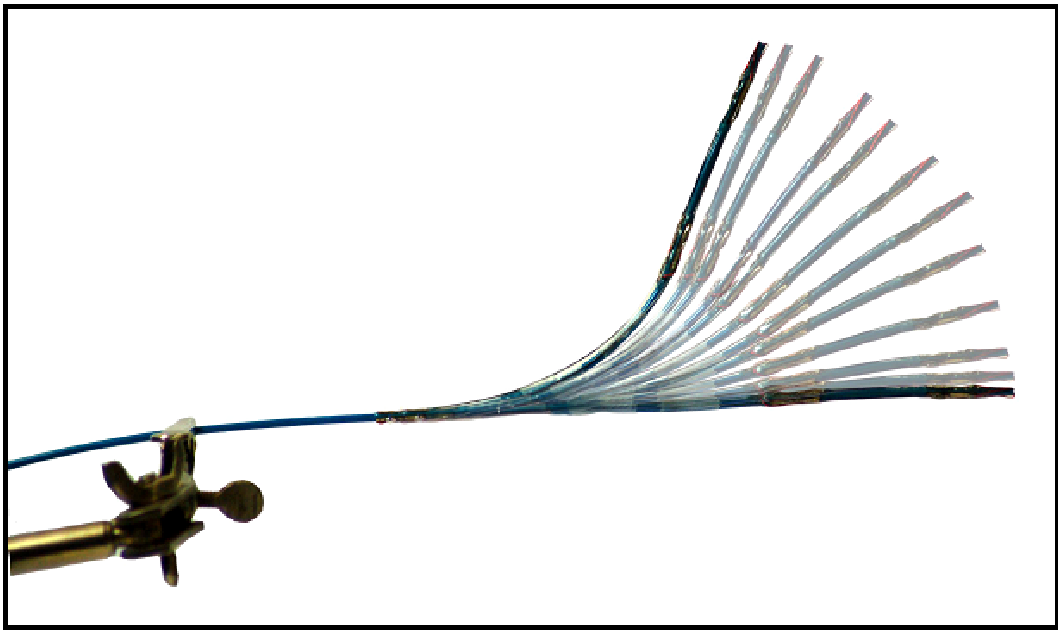

Figure B.1: Active catheter

is not directly fed back to the control loop. A 5-DOF magnetic sensor (model Aurora from NDI Inc.) placed at the tip of the catheter provides feedback on the orientation of the catheter. The transformation from the orientation in the active catheter to the strain in the SMA is inaccurate due to machining intolerances. Therefore an outer PID loop is necessary to ensure that the orientation of the catheter follows the desired trajectory, as shown in Fig. B.3. The output of the PID block is given by the following equation

$$u_\beta = K_p(e_b) + K_d(\dot{e_b}) + K_i \int e_b dt \tag{B.1}$$

$$u_\gamma = K_p(e_g) + K_d(\dot{e_g}) + K_i \int e_g dt \tag{B.2}$$

where e_b is the error between the desired yaw orientation β_d and the orientation of the active catheter β. Similarly e_g is the error between the desired pitch orientation γ_d and the orientation of the active catheter γ. K_p, K_i and K_d are the Proportional, Integral and Derivative gains respectively of the closed-loop

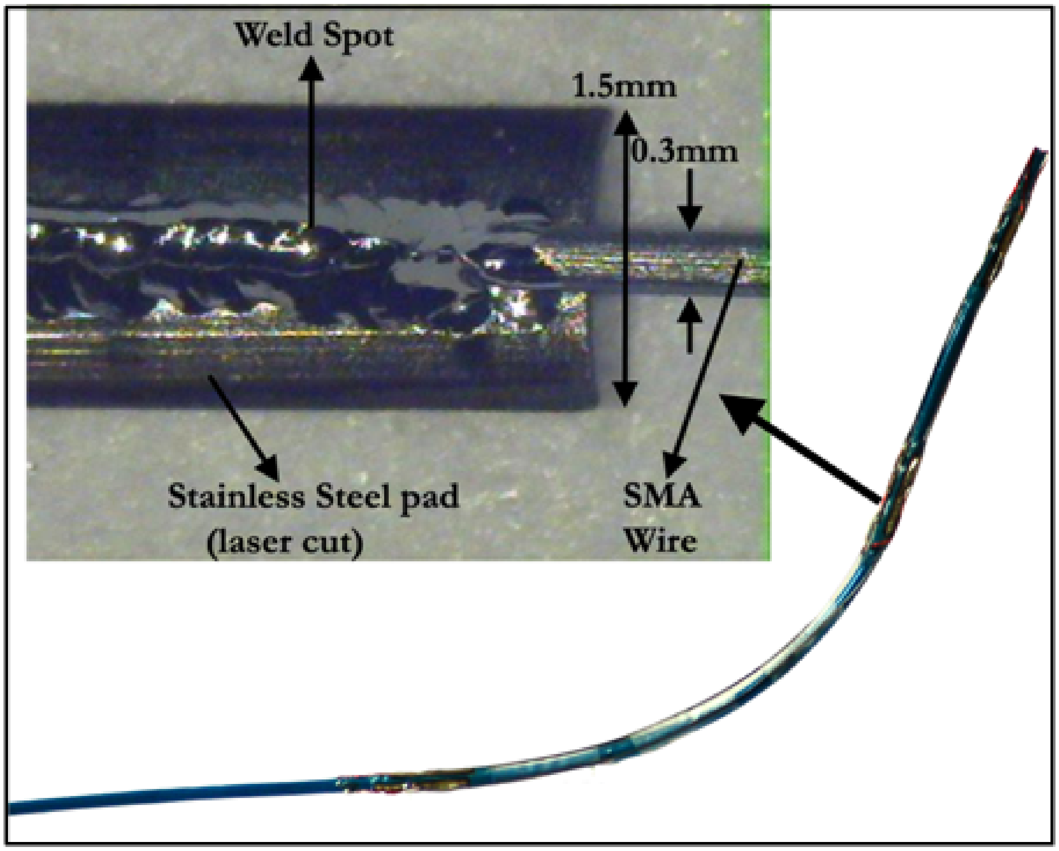

Figure B.2: SMA wires laser micro-welded to stainless steel pads

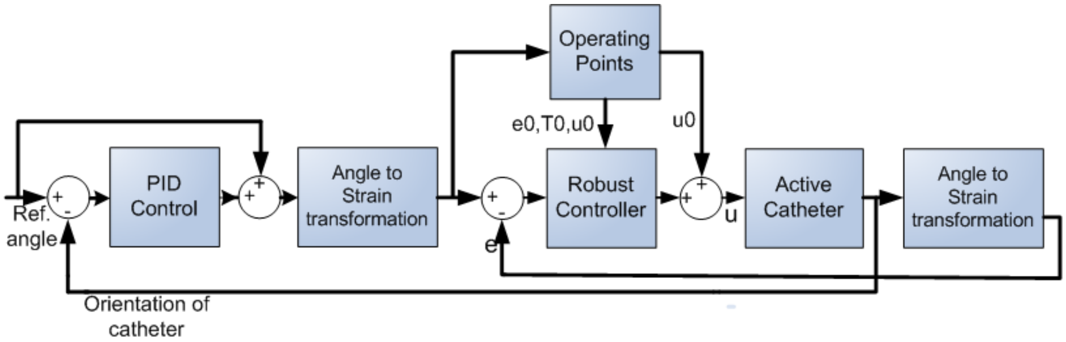

Figure B.3: Closed-loop control for the Active Catheter

controller. A linear transformation was chosen to map the orientation angle to strain in the SMA actuators. Accordingly the strain in the SMA actuators is resolved according to the following equations:

$$\begin{aligned}\epsilon_1 = -(1-u(\gamma_i))(\gamma_i/\gamma_{max})\epsilon_{max} - \\ (1-u(\beta_i))(\beta_i/\beta_{max})\epsilon_{max}/\cot(60^0) + \\ u(\beta_i)(\beta_i/\beta_{max})\epsilon_{max}/\cot(60^0)\end{aligned}$$

$$\begin{aligned}\epsilon_2 = u(\gamma_i)(\gamma_i/\gamma_{max})\epsilon_{max}/\sin(60^0) - \\ (1-u(\beta_i))(\beta_i/\beta_{max}))\epsilon_{max}/\sin(60^0)\end{aligned}$$

$$\begin{aligned}\epsilon_3 = u(\gamma_i)(\gamma_i/\gamma_{max})\epsilon_{max}/\sin(60^0) - \\ u(\beta_i)(\beta_i/\beta_{max})\epsilon_{max}/\sin(60^0)\end{aligned} \tag{B.3}$$

where ϵ_1, ϵ_2 and ϵ_3 are the strains in the three SMA actuators, γ_i and β_i are the input orientation to the "Angle to Strain Transformation" block and $u(.)$ is a unit step function.

	Perturbation	Error
Simulation results	No (Air)	0.29
Experimental results	No (Air)	0.73
	Load (Air)	2.78
	Cooling-rate (Air)	1.03
	No (Water)	1.21
	Cooling-rate (Milk)	2.69

Table B.1: RMS error (%) for H_∞ control

The transformation block in the outer-loop takes $\gamma_c = \gamma_{ref} + u_\gamma$ and $\beta_c = \beta_{ref} + u_\beta$ as the input orientation and generates the reference strain for the three SMAs, ϵ_{ref_1}, ϵ_{ref_2} and ϵ_{ref_3}. A saturation limit of ϵ_{max} was placed for the three reference strains. Three separate robust $\mathcal{H}_\infty$ loop-shaping controllers ensure accurate tracking of the reference strains generated for the three SMAs. Each robust controller has the same structure as described in [6]. The output of the robust controller u is provided to the three SMA actuators which make the active catheter bend in the desired direction. The robust controller ensures that the strain in the SMA actuators follows the desired reference generated by the outer loop and compensates for any load or cooling rate disturbances which affects the performance of the SMA actuators. The output of the "Active Catheter" block is the orientation of the catheter, γ and β, which is measured by means of a 5-DOF magnetic sensor at the tip of the catheter. The strain in the SMA actuators is estimated by means of the transformation (B.3) by providing γ and β as the input orientation to complete the feedback in the inner loop.

For the initial experiments, the reference trajectory for the active catheter is provided from a 6-DOF haptic device. The orientation of the haptic device γ (pitch) and β (yaw) are used to orient the catheter in space. The real-time implementation of the SMA model and control is done on a Windows based PC at a sampling rate of 30Hz. The results of the closed-loop tracking of the reference generated by the haptic device are shown in Fig. B.4. The figures show excellent tracking of the reference trajectory generated by the haptic device with an RMS error of 0.4^0.

Bibliography

[1] J. Jayender, M. Azizian, and R. V. Patel. Autonomous image-guided robot-assisted active catheter insertion. *IEEE Transactions on Robotics*, 24(4):858–871, 2008.

[2] Jagadeesan Jayender. *Haptics Enabled Robot-Assisted Active Catheter Insertion.* PhD thesis, The University of Western Ontario, 2007.

[3] J. Heisterkamp, R. van Hillegersberg, and J. N. IJzermans, "Critical temperature and heating time for coagulation damage: Implications for interstitial laser coagulation (ILC) of tumors," *Lasers in Surgery and Medicine*, vol. 25, pp. 257–262, 2000.

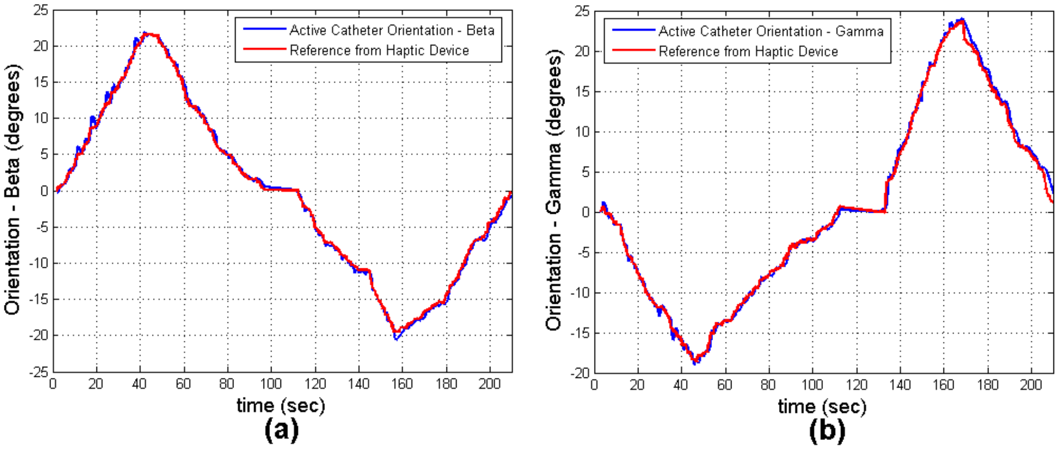

Figure B.4: Closed-loop control: Tracking a reference in (a) the β (yaw) orientation and (b) the γ (pitch) orientation

[4] *http://www.shape-memory-alloys.com/ni-ti_alloys.html*

[5] J. Jayender, R. V. Patel, S. Nikumb, and M. Ostojic, "Modeling and Control of Shape Memory Alloy actuators," *IEEE Transactions on Control Systems Technology*, vol. 16, pp. 279 – 287, March 2008.

[6] ——, "H_∞ Loop Shaping Controller for Shape Memory Alloy Actuators," in *44th IEEE Conference on Decision and Control and European Control Conference*, Dec. 2005.

Appendix C

Augmented Hybrid Impedance Control

The contents of this appendix has been published in [1] and [2].

For performing catheter insertion, the robot should have the ability to control the amount of force being exerted while inserting the catheter and should also precisely follow a pre-defined trajectory in Cartesian space. An Augmented Hybrid Impedance Control (AHIC) scheme has been implemented on a Mitsubishi PA 10-7C robot to control the force of insertion and the position of the end-effector in Cartesian space. The task space in AHIC is divided such that force control is performed in the direction of insertion, moment control in the orthogonal directions and position control in all the remaining directions. A brief overview of the algorithm is given below.

In the AHIC scheme, there are basically two control loops - the outer loop generates the position and force profiles that have to be tracked in real-time and determines the desired Cartesian acceleration that is fed to the inner loop. The inner loop converts the Cartesian acceleration to a joint level acceleration and the desired torques for each of the links are generated to track both the desired position and force profiles. The block diagram is given in Fig. C.1.

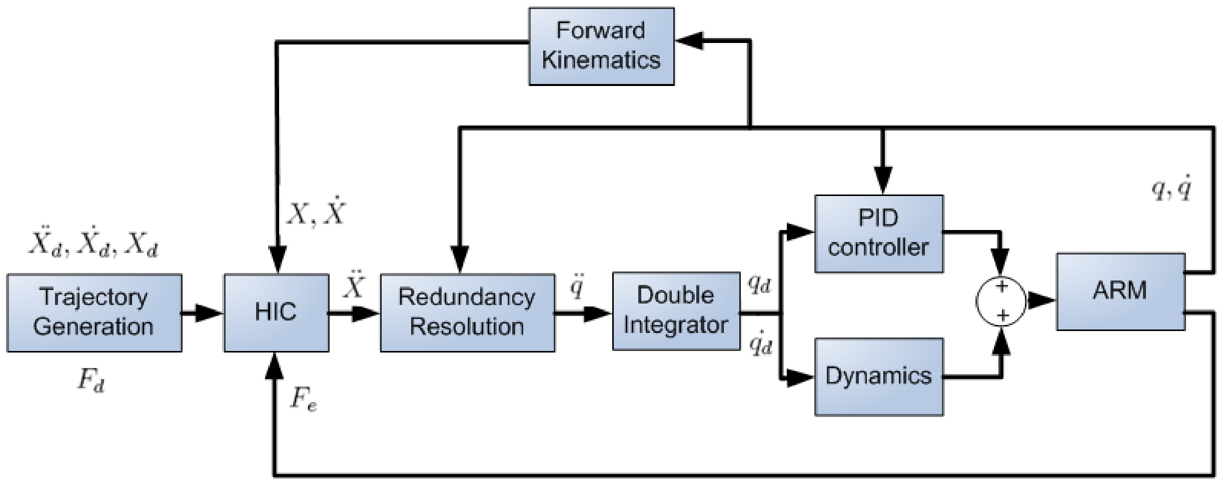

Figure C.1: Augmented Hybrid Impedance Control

C.1 Augmented Hybrid Impedance Controller

The AHIC module can be defined by the following equations:

$$M^d\ddot{X}^t = [-F_e + (I - S)F^d - B^d(\dot{X}^t - S\dot{X}^d) - K^dS(X^t - X^d)] + S\ddot{X}^d \tag{C.1}$$

$$\ddot{X}^r = \ddot{X}^t + K_v(\dot{X}^t - \dot{X}) + K_p(X^t - X) \tag{C.2}$$

where M_d and B_d are the desired mass and damping parameters, F_d and F_e are the desired force and environment contact force, X, $\dot{X}$ and $\ddot{X}$ are the Cartesian position/orientation, and the corresponding velocity and acceleration respectively. The matrix **S** denotes the selection matrix which defines the force and position controlled subspaces. The constraints of the catheter insertion problem are reflected in the **S** matrix. It should be noted that both linear and angular trajectories are specified in equations (C.1) and (C.2). The terms K_p and K_v in (C.2) are diagonal matrices with positive diagonal entries which are chosen to ensure good tracking performance.

C.2 Redundancy Resolution

The inner loop which consists of the redundancy resolution module converts the Cartesian acceleration to a desired joint level acceleration which is provided to the joint-based controller. Since the Mitsubishi PA 10-7C robot has 7-DOFs, the Jacobian is not square. As a result an additional task, which fixes the redundant joint S3, is included to make the Jacobian square [3]. A damped least-squares solution at the acceleration level is implemented to damp out the joint velocities in the null-space of the Jacobian as given by the following equation:

$$\ddot{\theta}^t = [J_e^T W_e J_e + J_c^T W_c J_c + W_v]^{-1}[J_e^T W_e(\ddot{X}^r - \dot{J}_e\dot{\theta}) + J_c^T W_c(\ddot{Z}) - W_v \lambda \dot{\theta}] \quad \text{(C.3)}$$

where J_e and J_c are the Jacobian matrices corresponding to the primary and the secondary tasks, W_e and W_c are weight matrices, W_v is the singularity robustness factor and λ is the velocity damping factor. The joint accelerations are integrated to obtain the desired joint velocities and positions and fed to the joint control module after canceling the gravity term.

C.3 Joint based controller

Each of the 7 joints is controlled to follow a certain desired trajectory. The dynamic model for a rigid-link manipulator is given by:

$$\tau = M(\theta)\ddot{\theta} + V(\theta,\dot{\theta}) + G(\theta) + f(\theta,\dot{\theta}) \quad \text{(C.4)}$$

where $M(\theta)$ is the mass matrix, $V(\theta,\dot{\theta})$ is the vector of Coriolis and centrifugal terms, $G(\theta)$ is the vector of gravity terms and $f(\theta,\dot{\theta})$ is the joint friction vector, which has not been modeled. For medical robots, the joint velocities are generally quite small. Therefore the $V(\theta,\dot{\theta})$ term can be assumed to be negligible. In addition, since there is very little change in the robot's configuration during catheter insertion, the $M(\theta)$ term can be assumed to be constant. The gravity terms are obtained in closed-form. The joint level controller therefore simplifies to

$$\tau = \ddot{\theta}^t + K_{p_j}(\theta^t - \theta) + K_{d_j}(\dot{\theta}^t - \dot{\theta}) + G(\theta) \quad \text{(C.5)}$$

where θ corresponds to the joint angles, θ^t and $\dot{\theta}^t$ are the desired joint angles and velocities, and K_{p_j} and K_{d_j} are the proportional and derivative gains of the joint level controller.

In order to determine the accuracy of the PA10-7C robot, experiments were performed to follow desired trajectories with continuous position and velocity profiles in Cartesian space. The robot followed the reference trajectories accurately with an average RMS position error of 0.27mm in x, 0.30mm in y and 0.43mm in z directions. The average RMS errors for tracking the orientation trajectories were 0.18 degrees, 0.06 degrees and 0.17 degrees for α, β, and γ Euler angles respectively. The results showed accurate tracking of the reference trajectories even in the presence of inaccuracies in the model. Force control experiments conducted with the PA10-7C robot indicated that the robot end-effector was able to track force trajectories with an average RMS error of 0.05N. An accurate dynamic model of the PA10-7C robot can be used to obtain greater robustness to external disturbances [4], [5]. In a recent paper, Kennedy and Desai [4] have developed a model for the PA10-7C robot and have performed position control using inverse kinematics. An RMS error of 2cm for following a position trajectory in Cartesian space has been reported. However, improvements in the model have been made in [5] and an RMS error of 0.8mm in position tracking has been reported. The controller in [5] does not regulate the force exerted by the robot on its environment since redundancy resolution is performed at the velocity level.

Bibliography

[1] J. Jayender, M. Azizian, and R. V. Patel. Autonomous image-guided robot-assisted active catheter insertion. *IEEE Transactions on Robotics*, 24(4):858–871, 2008.

[2] Jagadeesan Jayender. *Haptics Enabled Robot-Assisted Active Catheter Insertion*. PhD thesis, The University of Western Ontario, 2007.

[3] H. Seraji, “Configuration control of redundant manipulators,” *IEEE Journal of Robotics and Automation*, vol. 5, pp. 472–490, 1989.

[4] C. W. Kennedy and J. P. Desai, “Modeling and Control of the Mitsubishi PA-10 Robot Arm Harmonic Drive System,” *IEEE/ASME Transaction on Mechatronics*, vol. 10, pp. 263 – 274, 2005.

[5] N. A. Bompos, P. K. Artemiadis, A. S. Oikonomopoulos, and K. J. Kyriakopoulos, "Modeling, Full Identification and Control of the Mitsubishi PA-10 Robot Arm," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2007, pp. 1–6.

Appendix D

Micromanipulators in Patch Clamping Setup

Specifications for the micromanipulators used in the system described in Chapter 4 are given in the following table.

	M1	M2	M3(1..4)	M4(1..4)
Max. Speed	2.9	4.0	4.0	5.0
Max. Resolution	40	50	62.5	50
Motion Range (xyz)	25.4	60	$25, 12.5, 25$	150
Interface	RS-232	RS-232	USB	RS-232
Manufacturer	Sutter	Zaber	Sutter	Zaber
Model	MP-265	T-LA60A-S	MPC-200	T-LSR150A
What's Moved?	Microscope	Objectives	MPs	M3(1..4)

Table D.1: Specifications of micromanipulators used in the patch clamping setup.

Appendix E

Lens Changing Mechanism

A prismatic objective lens changing mechanism was designed. A cross section of the CAD design is shown in the figure below. A miniature linear actuator was used to push a lens holding plate on a set of linear bearings. Two springs were used to pull the plate back. The linear actuator, T-LA60A-S, Zaber technologies [1] was used which had a maximum travel range of $60mm$, minimum step size of $0.05\mu m$ with accuracy of $\pm 16\mu m$, repeatability of $1\mu m$ and backlash of $4\mu m$ with a maximum continuous force of $15N$ ($25N$ peak). The maximum linear speed was $4mm/sec$. The model was made using a 3D printing machine (Dimension Elite [2]) using ABS thermoplastic [3]. The stainless steel extension springs had an overall length of $2.044in$, outside diameter of $0.24in$, wire diameter of $0.022in$ with a $75N/m$ spring constant.

Bibliography

[1] `http://www.zaber.com`

[2] `http://www.dimensionprinting.com/`

[3] `http://en.wikipedia.org/wiki/Thermoplastic`

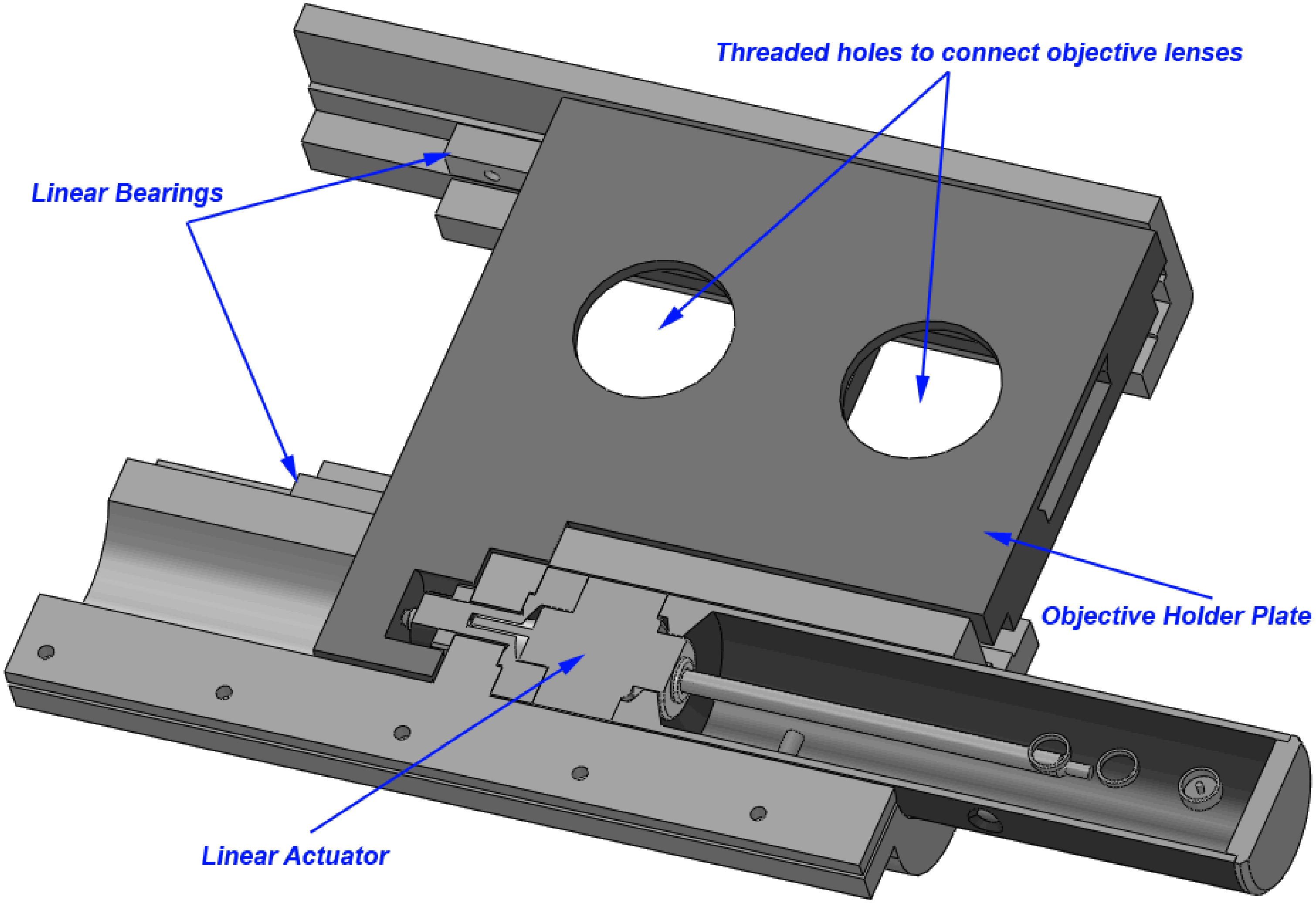

Figure E.1: Cross-section of the lens changing mechanism

Appendix F

Steerable Catheter Actuation Mechanism

A CAD model of the 2-DOF catheter actuation mechanism is shown in Figure 3.6. DC motors, gear heads and encoders were chosen from Maxon motors (Maxon Precision Motors, Inc., Fall River, MA, US).

For the rotary actuator, a rotary DC motor (RE-max 29) with a planetary gearhead (GP 26) with a reduction rate of 128 was used. An optical encoder (Encoder MR, Type ML, 3 Channels) with 1000 counts per turn was chosen to provide feedback.

For the prismatic actuator, a rotary DC motor RE-max 24 was used along with a spindle drive (GP 22) for rotary to linear motion conversion. An optical encoder (Encoder MR, Type M, 2/3 Channels) with 512 counts per turn was used.

Appendix G

SensAble Haptic Devices

Two haptic devices were used, both manufactured by SensAble Technologies [1]. The OpenHaptics Toolkit [2] was used to interface to the haptic devices. A PHANTOM Omni haptic device [3] was used for the works reported in Chapters 2 and 4 and a PHANTOM Desktop [4] was used for the work reported in Chapter 3. The haptic device control loop was updated at a rate of $1KHz$. The properties of these haptic devices are listed in the following table.

	PHANTOM Omni	PHANTOM Desktop
Position Resolution	$0.055mm$	$0.023mm$
Maximum Exertable Force	$3.3N$	$7.9N$
Continuous Exertable Force	$0.88N$	$1.75N$
Workspace (WxHxD-mm)	160x120x70	160x120x120
Position Sensing	6DOF	6DOF
Force Exertion	3DOF	3DOF
Apparent Mass	$45g$	$45g$
Interface	IEEE1394 (Firewire)	Parallel (EPP mode)

Bibliography

[1] `http://www.sensable.com`

[2] `http://www.sensable.com/products-openhaptics-toolkit.htm`

[3] `http://www.sensable.com/haptic-phantom-omni.htm`

[4] `http://www.sensable.com/haptic-phantom-desktop.htm`

Appendix H

Magnetic Tracking

For the work reported in Chapters 2 and 3, we have used the NDI Aurora electromagnetic measurement system [1] (Northern Digital Inc., Waterloo, ON, Canada). The Aurora magnetic tracking system consists of sensor coils and a sensor interface unit (SIU) for each sensor coil, a field generator device and a system control unit. The system control unit is connected to the computer through an RS-232 link. It also provides the power for the field generator. Sensor coils are connected to the system control unit via sensor interface units. 5-DOF miniature magnetic tracking sensors ($0.5mm$ diameter, $8.0mm$ length) were used. The NDI magnetic tracking system connects through a RS-232 connection at a baud rate of up to $115Kbps$. The sensor position/orientations can be updated at rates of up to $40Hz$. The spatial range of the magnetic tracking system is greater than a cube of $500 \times 500 \times 500mm^3$ around the field generator. Position and orientation sensing with an accuracy of $0.9mm$ and 0.3^o respectively and precision of $0.6mm$ and 0.1^o respectively, is achieved. The magnetic tracking system is sensitive to electromagnetic interference and ferromagnetic materials in its workspace.

Bibliography

[1] `http://www.ndigital.com/medical/aurora.php`

Appendix I

Auto Focus Measures

Different autofocusing measures (indices) used in Section 4.3 are defined in this appendix.

- Scharr: $\Phi_{Scharr}(I) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\sqrt{(\mathcal{C}_x * I)_{i,j}^2 + (\mathcal{C}_y * I)_{i,j}^2}$, where: Kernel: $\mathcal{C}_x = \begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix}$, $\mathcal{C}_y = \mathcal{C}_x^T$

- Sobel: $\Phi_{Sobel}(I) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\sqrt{(\mathcal{S}_{k_x} * I)_{i,j}^2 + (\mathcal{S}_{k_y} * I)_{i,j}^2}$, where:

  $$\mathcal{S}_{3_x} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \mathcal{S}_{5_x} = \begin{bmatrix} -1 & -2 & 0 & 2 & 1 \\ -4 & -8 & 0 & 8 & 4 \\ -6 & -12 & 0 & 12 & 6 \\ -4 & -8 & 0 & 8 & 4 \\ -1 & -2 & 0 & 2 & 1 \end{bmatrix}, \mathcal{S}_{7_x} = \begin{bmatrix} -1 & -4 & -5 & 0 & 5 & 4 & 1 \\ -6 & -24 & -30 & 0 & 30 & 24 & 6 \\ -15 & -60 & -75 & 0 & 75 & 60 & 15 \\ -20 & -80 & -100 & 0 & 100 & 80 & 20 \\ -15 & -60 & -75 & 0 & 75 & 60 & 15 \\ -6 & -24 & -30 & 0 & 30 & 24 & 6 \\ -1 & -4 & -5 & 0 & 5 & 4 & 1 \end{bmatrix},$$

  $\mathcal{S}_{k_y} = \mathcal{S}_{k_x}^T$, $k = 3, 5, 7$

- Laplacian: $\Phi_{Laplacian}(I) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}||\Delta I(i,j)||$, where: $\mathcal{L}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$,

  $$\mathcal{L}_3 = \begin{bmatrix} 2 & 0 & 2 \\ 0 & -8 & 0 \\ 2 & 0 & 2 \end{bmatrix}, \mathcal{L}_5 = \begin{bmatrix} 2 & 4 & 4 & 4 & 2 \\ 4 & 0 & -8 & 0 & 4 \\ 4 & -8 & -24 & -8 & 4 \\ 4 & 0 & -8 & 0 & 4 \\ 2 & 4 & 4 & 4 & 2 \end{bmatrix}, \mathcal{L}_7 = \begin{bmatrix} 2 & 8 & 14 & 16 & 14 & 8 & 2 \\ 8 & 24 & 24 & 16 & 24 & 24 & 8 \\ 14 & 24 & -30 & -80 & -30 & 24 & 14 \\ 16 & 16 & -80 & -160 & -80 & 16 & 16 \\ 14 & 24 & -30 & -80 & -30 & 24 & 14 \\ 8 & 24 & 24 & 16 & 24 & 24 & 8 \\ 2 & 8 & 14 & 16 & 14 & 8 & 2 \end{bmatrix}$$

- Morphological Gradient: $\begin{cases} \nabla_{Morph} I = I \oplus \mathcal{M} - I \ominus \mathcal{M} \\ \Phi_{Morph} I = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}||\nabla_{Morph} I(i,j)|| \end{cases}$

  where $\oplus$ and $\ominus$ denote the dilation and the erosion, respectively and $\mathcal{M} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ is the *Structural Element*.

Appendix J

Ultrasound Imaging

In the work reported in Chapter 3, we have used a *SonixTouch* ultrasound machine [2] (Ultrasonix, Richmond, BC, Canada) to provide real-time imaging in the second part of the project for cardiac ablation applications. A motorized curvilinear transducer $4DC7-3/40$ with 104^o field of view and $55mm$ length of array, was used to provide online $3D$ ultrasound images. This probe can work on any of $2.5, 4.0, 5.0MHz$ B-Mode frequencies. The probe was used to provide an online 3D apical view of the heart phantom, which mimics transthoracic echocardiography (TTE) [3]. The Ulterius SDK [1] was used to control the ultrasound machine and stream the images over the network in real time using an ethernet connection.

Bibliography

[1] `http://www.ultrasonix.com/wikisonix/index.php?title=Ulterius`

[2] `http://ultrasonix.com/products/sonixtouch/research`

[3] `http://en.wikipedia.org/wiki/Transthoracic_echocardiogram`

Appendix K

X-ray Fluoroscopy

In the work reported in Chapter 2, we have used an *OEC 9900 Elite* X-ray fluoroscopy machine to test the image-based tracking algorithms. It can provide 1.0 megapixels images at a rate of $30fps$(frames per second). However we used the analog video output and a frame grabber to capture 640×480 pixels images at $30fps$. A Matrox Morphis [2] frame grabber was used for this purpose. The robotic tests were performed using a digital camera simulating X-ray fluoroscopy imaging. A Dragonfly2 camera [3] (Point Grey Research, Richmond, BC, Canada) was used for simulating real-time X-ray imaging. The camera has a Firewire (IEEE 1394) interface and can capture $0.78MP$ images at a rate of $30fps$.

Bibliography

[1] `http://www.gehealthcare.com/euen/surgery/products/oec-9900-elite`

[2] `http://www.matrox.com/imaging/en/products/frame_grabbers/morphis/morphis`

[3] `http://www.ptgrey.com/products/dragonfly2`

Vita

NAME:	Mahdi Azizian
PLACE OF BIRTH:	Damghan, Semnan, Iran
YEAR OF BIRTH:	1979
POST-SECONDARY EDUCATION AND DEGREES:	University of Western Ontario London, Ontario Ph.D, 2011
	Amirkabir University of Technology (Tehran Polytechnic) Tehran, Iran B.E.Sc., 2002
HONOURS AND AWARDS:	MITACS Elevate Industrial Postdoctoral Fellowship, 2011
	Alexander Graham Bell Canada Graduate Scholarship, (NSERC CGS), 2009-2011.
	Ontario Graduate Scholarship (OGS), 2009, declined in favor of the NSERC CGS.
	Nominated for the best Graduate Teaching Assistant, The University of Western Ontario, 2009.
	Precarn Scholarship, 2008.
	Best Presentation Award, Western Engineering Graduate Symposium, 2007.
	Best Graduate Teaching Assistant in the ECE Department, University of Western Ontario, 2007.

RELATED WORK EXPERIENCE:

Research Assistant, Canadian Surgical Technologies & Advanced Robotics (CSTAR), 2007-2011

Research Assistant, Dr. M.O. Poulter’s Lab,
Robarts Research Institute, 2009-2011.

Research Assistant, Robotics Lab, ECE Dept.,
The University of Western Ontario, 2005-2008.

Graduate Teaching Assistant, ECE Dept.,
The University of Western Ontario, 2006-2011.

Project Manager, Technology Pioneers Co. Ltd., 2002-2005

Research Assistant, Iran Telecom. Research Center (ITRC), 2001-2003

Undergraduate Teaching Assistant, EE Dept.,
Amirkabir University of Technology, 2000-2002.

Undergraduate Research Assistant, EE Dept.,
Real-Time Systems Research Laboratory (Dr. Talebi),
& Mobile Robot Research Laboratory (Dr. Raei),
Amirkabir University of Technology, 1999-2002.

Publications:

- M. Azizian, R. V. Patel, C. Gavrilovici, and M. O. Poulter, ”Semi-Automated Patch Clamp Electrophysiology”, under review in IEEE Transactions on Automation Science and Engineering, 2010.
- M. Azizian, R. V. Patel, C. Gavrilovici, and M. O. Poulter, ”Image-Guided Patch Clamp Electrophysiology”, under review in SPIE Journal of Biomedical Optics, 2010.

- M. Azizian, R. V. Patel, ”Intraoperative 3D Stereo Visualization for Image-Guided Cardiac Ablation,” in Medical Imaging 2011: Visualization, Image-Guided Procedures, and Modeling, edited by Kenneth H. Wong, David R. Holmes III, Proceedings of SPIE Vol. 7964 (SPIE, Bellingham, WA 2011) 79640F.

- M. Azizian, R. V. Patel, ”Data Fusion for Catheter Tracking using Kalman Filtering: Applications in Robot-Assisted Catheter Insertion,” in Medical Imaging 2011: Visualization, Image-Guided Procedures, and Modeling, edited by Kenneth H. Wong, David R. Holmes III, Proceedings of SPIE Vol. 7964 (SPIE, Bellingham, WA 2011) 796413.

- M. Azizian, R. V. Patel, C. Gavrilovici, and M. O. Poulter, ”Image-Guided Robot-Assisted Microscope Objective Lens Positioning: Application in Patch Clamping,” in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp.6149-6154, Oct. 2010.

- M. Azizian, R. V. Patel, C. Gavrilovici, and M. O. Poulter, ”Computer Assisted Patch Clamping,” in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp.4131-4136, May. 2010.

- M. Azizian, R. Patel, C. Gavrilovici, and M. Poulter, ”Image Processing Techniques in Computer-Assisted Patch Clamping,” in Imaging, Manipulation, and Analysis of Biomolecules, Cells, and Tissues VIII, edited by Daniel L. Farkas, Dan V. Nicolau, Robert C. Leif, Proceedings of SPIE Vol. 7568 (SPIE, Bellingham, WA 2010) 75681E.

- J. Jayender, M. Azizian, R. V. Patel, ”Autonomous Image-Guided Robot-Assisted Active Catheter Insertion,” IEEE Transactions on Robotic, 24 (4), pp. 858-871, 2008.

- J. Jayender, M. Azizian, and R. V. Patel, ”Bilateral Telemanipulation of a Flexible Catheter in a Constrained Environment,” in Proceedings of the

IEEE International Conference on Robotics and Automation (ICRA), pp.649-654, May. 2008.

- J. Jayender, M. Azizian, and R. V. Patel, "Autonomous Robot-Assisted Active Catheter Insertion Using Image Guidance," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp.889-894, Oct. 2007.
- M. Azizian, J. Jayender, and R. V. Patel, "Image processing algorithms for real-time tracking and control of an active catheter", in Proceedings of the European Control Conference (ECC), Jul. 2007.