IMAGE GUIDED INTERACTION IN MINIMALLY INVASIVE SURGERY

by

Xiaochuan Sun
B.E. Anhui University of Technology, 2008

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE
in the
School of Engineering Science
Faculty of Applied Sciences

© Xiaochuan Sun 2012
SIMON FRASER UNIVERSITY
Spring 2012

All rights reserved.
However, in accordance with the Copyright Act of Canada, this work may be reproduced, without authorization, under the conditions for “Fair Dealing.” Therefore, limited reproduction of this work for the purposes of private study, research, criticism, review and news reporting is likely to be in accordance with the law, particularly if cited appropriately.
APPROVAL

Name: Xiaochuan Sun

Degree: Master of Applied Science

Title of Thesis: Image Guided Interaction in Minimally Invasive Surgery

Examining Committee: Dr. Carlo Menon, PEng
Assistant Professor
Chair

Dr. Shahram Payandeh, PEng
Professor
Senior Supervisor

Dr. Marinko V. Sarunic, PEng
Assistant Professor
Supervisor

Dr. Stella Atkins
Professor
Computing Science
Examiner

Date Approved: January 10, 2012
Declaration of Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection (currently available to the public at the “Institutional Repository” link of the SFU Library website <www.lib.sfu.ca> at: <http://ir.lib.sfu.cahandle/1892/112>) and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author’s written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

While licensing SFU to permit the above uses, the author retains copyright in the thesis, project or extended essays, including the right to change the work for subsequent purposes, including editing and publishing the work in whole or in part, and licensing other parties, as the author may desire.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library
Burnaby, BC, Canada
Abstract

A three-dimensional interactive “mouse” system for minimally invasive surgery (MIS) is proposed in this thesis. The system consists of a novel tracking framework, which uses the images of the existing surgical tools in the field of view to determine their corresponding spatial coordinates. This tracking framework combines the advantages of both Kalman filtering tracking and real-time feature-based tracking, offering a relatively robust system for the later 3D augmentation environment to be laid upon. The designed surgeon-computer interface based on this tracking framework allows the surgeon to access the medical information during surgery without introducing extra pieces of hardware. It also allows manual registration of pre-operative images on the live images. Finally, preliminary results on an optically instrumented surgical tool are presented, which can be used to scan inner cavity or image the inner structure of tissue in MIS.

Keywords: 3D visual tracking; Surgeon-computer interface; 3D non-contact mouse
Acknowledgments

First and foremost, I would like to express my utmost gratitude to my senior supervisor, Dr. Shahram Payandeh, for guiding me through the research with your enthusiasm, inspiration and patience. You are not only my academic teacher, but moral mentor for motivating and cheering my minds when I was upset and lost. This thesis would not have been possible without your sound advice, good teaching and lots of good ideas.

I wish to thank Dr. Marinko V. Sarunic, for his kind assistance with the optical systems and for teaching me the theory of Optical Coherent Tomography in his brilliant class. Especially, your BORG lab has provided me the best and talented colleagues, who are always ready to help with any questions I asked. I would like to give special thanks to Mei Young and Jing Xu for providing me their equipments and sharing their knowledge with me. Thanks to all of you, I was able to contribute one of my main chapters in this thesis.

I am indebted to my lab colleagues for providing a stimulating environment in which to learn and grow. I am especially grateful to Wen Shi, David Tang and Han Xiao for their support and entertainment.

I wish to thank all my friends here, for all the emotional support, comraderie, and caring they provided. I also want to give my special thanks to Jeff Liu, Zhouming Tang and Michelle Cua for sparing time to proofread my thesis.

I am grateful to the secretaries and librarians in the engineering departments of SFU, for helping the departments to run smoothly and for assisting me in many different ways.

Lastly, and most importantly, I wish to thank my parents, Yuantang Sun and Yanli Lu. They bore me, raised me, supported me and loved me. Particularly my father, it is you who let me think different of the world and it is you who told me to be strong all the time. To them I dedicate this thesis.
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D</td>
<td>Three Dimensional</td>
</tr>
<tr>
<td>OR</td>
<td>Operation Room</td>
</tr>
<tr>
<td>AR</td>
<td>Augmented Reality</td>
</tr>
<tr>
<td>MIS</td>
<td>Minimally Invasive Surgery</td>
</tr>
<tr>
<td>SCI</td>
<td>Surgeon Computer Interface</td>
</tr>
<tr>
<td>MHI</td>
<td>Motion History Image</td>
</tr>
<tr>
<td>PF</td>
<td>Particle Filter</td>
</tr>
<tr>
<td>MIL</td>
<td>Matrox Imaging Library</td>
</tr>
<tr>
<td>OpenCV</td>
<td>Open Source Computer Vision Library</td>
</tr>
<tr>
<td>OpenGL</td>
<td>Open Graphics Library</td>
</tr>
<tr>
<td>ROI</td>
<td>Region Of Interest</td>
</tr>
<tr>
<td>HSV</td>
<td>Hue, Saturation, Value</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Component Analysis</td>
</tr>
<tr>
<td>OCT</td>
<td>Optical Coherence Tomography</td>
</tr>
<tr>
<td>He-Ne</td>
<td>Helium-neon</td>
</tr>
<tr>
<td>SS-OCT</td>
<td>Swept Source Optical Coherence Tomography</td>
</tr>
</tbody>
</table>
Contents

Approval ii
Abstract iii
Acknowledgments iv
List of Acronyms v
Contents vi
List of Tables x
List of Figures xii

1 Introduction 1
  1.1 Challenges in minimally invasive surgery . . . . . . . . . . . . . . . . . . . . . 1
  1.2 Motivation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
  1.3 Survey of Relevant work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
    1.3.1 Visual Tracking . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
    1.3.2 Surgeon-computer Interface (SCI) . . . . . . . . . . . . . . . . . . . . . 6
    1.3.3 3D laser scanning and reconstruction . . . . . . . . . . . . . . . . . . . . 7
  1.4 Contributions and layout of the thesis . . . . . . . . . . . . . . . . . . . . . . 9

2 Background Material 11
  2.1 Endoscope calibration method . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
  2.2 Blob Tracking and Ring menu interface . . . . . . . . . . . . . . . . . . . . . 13
  2.3 Tool motion and Gesture recognition system . . . . . . . . . . . . . . . . . . . 14
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4 New Graphical User Interface overview design and preliminaries</td>
<td>15</td>
</tr>
<tr>
<td>2.5 Phantom Experimental Setup</td>
<td>17</td>
</tr>
<tr>
<td>2.5.1 Hardware</td>
<td>17</td>
</tr>
<tr>
<td>2.5.2 Software</td>
<td>19</td>
</tr>
<tr>
<td>3 3-D image tracking: Preliminary studies</td>
<td>20</td>
</tr>
<tr>
<td>3.1 Motion tracking using Motion history image (MHI)</td>
<td>20</td>
</tr>
<tr>
<td>3.2 Tracking using Particle filtering (PF)</td>
<td>22</td>
</tr>
<tr>
<td>3.3 Experimental studies for MHI and PF approaches</td>
<td>27</td>
</tr>
<tr>
<td>3.4 Discussion</td>
<td>27</td>
</tr>
<tr>
<td>4 A new feature-based tracking algorithm</td>
<td>31</td>
</tr>
<tr>
<td>4.1 A new feature-based tracking algorithm overview</td>
<td>32</td>
</tr>
<tr>
<td>4.2 Two dimensional tracking of surgical tool using feature-based tracking</td>
<td>34</td>
</tr>
<tr>
<td>4.2.1 Kalman filter system modeling</td>
<td>34</td>
</tr>
<tr>
<td>4.2.2 Initialization and preprocessing</td>
<td>38</td>
</tr>
<tr>
<td>4.2.3 Tool tracking using Kalman Filter</td>
<td>40</td>
</tr>
<tr>
<td>4.2.4 Tip extraction</td>
<td>43</td>
</tr>
<tr>
<td>4.2.5 Two dimensional tracking of multiple tools</td>
<td>47</td>
</tr>
<tr>
<td>4.3 Three dimensional tracking of surgical tools</td>
<td>48</td>
</tr>
<tr>
<td>4.3.1 3D pose estimation using tip and edges</td>
<td>48</td>
</tr>
<tr>
<td>4.4 Experimental results and discussion for the new proposed feature-based approach</td>
<td>51</td>
</tr>
<tr>
<td>4.4.1 Single tool tracking result</td>
<td>51</td>
</tr>
<tr>
<td>4.4.2 Multiple tracking result</td>
<td>55</td>
</tr>
<tr>
<td>4.4.3 Three dimensional tool tracking result</td>
<td>56</td>
</tr>
<tr>
<td>4.5 Comparison studies between MHI, PF and new proposed feature-based approach</td>
<td>57</td>
</tr>
<tr>
<td>4.6 Discussion</td>
<td>60</td>
</tr>
<tr>
<td>5 3D mouse interactive paradigm</td>
<td>61</td>
</tr>
<tr>
<td>5.1 3D-mouse system overview and its application</td>
<td>61</td>
</tr>
<tr>
<td>5.2 Augmented surgical environment design</td>
<td>63</td>
</tr>
<tr>
<td>5.2.1 Menu design</td>
<td>63</td>
</tr>
<tr>
<td>Chapter</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Interaction design</td>
</tr>
<tr>
<td>5.3</td>
<td>3D-mouse system implementation</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Align physical camera with virtual camera</td>
</tr>
<tr>
<td>5.3.2</td>
<td>3D manipulation with virtual organ</td>
</tr>
<tr>
<td>5.3.3</td>
<td>3D manipulation with virtual thread</td>
</tr>
<tr>
<td>5.4</td>
<td>Testing the accuracy of the 3D-mouse system</td>
</tr>
<tr>
<td>6</td>
<td>Preliminary studies of laser scanner and OCT system</td>
</tr>
<tr>
<td>6.1</td>
<td>Minimally invasive 3D real-time scanning system</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Method overview</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Experimental results of the 3D real-time scanning system</td>
</tr>
<tr>
<td>6.2</td>
<td>Multi-spots laser triangulation system for endoscopic imaging</td>
</tr>
<tr>
<td>6.2.1</td>
<td>System description</td>
</tr>
<tr>
<td>6.2.2</td>
<td>One-mirror, multi-spots system</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Two-mirror, multi-spots system</td>
</tr>
<tr>
<td>6.2.4</td>
<td>Image processing</td>
</tr>
<tr>
<td>6.2.5</td>
<td>Experimental results of the laser triangulation system</td>
</tr>
<tr>
<td>6.2.6</td>
<td>Discussion</td>
</tr>
<tr>
<td>6.3</td>
<td>Minimally invasive OCT diagnosis system</td>
</tr>
<tr>
<td>6.3.1</td>
<td>System Configuration and Setup</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Experimental Results</td>
</tr>
<tr>
<td>7</td>
<td>Conclusion and Future work</td>
</tr>
<tr>
<td>7.1</td>
<td>Conclusion</td>
</tr>
<tr>
<td>7.2</td>
<td>Future work</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
</tr>
<tr>
<td></td>
<td>Appendices</td>
</tr>
<tr>
<td>A</td>
<td>Introduction to endoscopy and surgical equipments</td>
</tr>
<tr>
<td>A.1</td>
<td>Terms related to endoscope</td>
</tr>
<tr>
<td>A.2</td>
<td>Types of endoscopy</td>
</tr>
<tr>
<td>A.3</td>
<td>Optical Properties for endoscopes</td>
</tr>
<tr>
<td>A.4</td>
<td>Surgical instruments</td>
</tr>
</tbody>
</table>
B Pixel format conversion

C Introduction to Optical Coherent Tomography

C.1 Theory of Optical Coherent Tomography

C.2 Theoretical Formulation for Signal Extraction in OCT

C.3 Swept source OCT setup

D Singleton Pattern & its implementation with C++
List of Tables

2.1 Calibration results .......................................................... 13

3.1 Tool tracking result using MHI & PF method ......................... 28
3.2 Initial parameters for PF .................................................... 29

4.1 Experimental results of current feature-based algorithm with different input factors .................................................. 54
4.2 3D tool orientation experimental result ................................ 57
4.3 Function comparison between color-based method, Motion history image method (MHI), Particle filtering method (PF) and Current feature-based algorithm ................................. 58
4.4 2D tracking comparison accuracy(pixel) between color-based method, Motion history image method (MHI), Particle filtering method (PF) and Current feature-based algorithm ........................................... 59
4.5 3D tip tracking comparison accuracy (mm) between the color-based method and the current feature-based algorithm ................................. 59
4.6 Comparison of computational time between color-based method, Motion history image method (MHI), Particle filtering method (PF) and Current feature-based algorithm ........................................... 59

5.1 Interaction Protocol of the current 3D mouse SCI .................... 68
5.2 Experimental results of the augmentation accuracy .................. 79

6.1 The mean and standard error of the three sets of experiments with depth of 90, 100 and 120mm. ................................................. 83
6.2 Data Before and after Correction for a flat surface .................. 91
6.3 Comparison of measured and calculated distance . . . . . . . . . . . . . . . . 95
A.1 Types of endoscopy . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 110
List of Figures

1.1 (a) Typical instrument arrangement for the gall bladder removal surgery. The endoscopic camera and surgical instruments are inserted into the abdomen through small incisions with approximately 5-15 mm diameter. In the upper left corner, the endoscopic view is displayed. (b) Illustration of MIS procedure in a regular operating room. The surgeons have to look at the monitors while doing the operation. ................................................................. 2

1.2 Related commercialized products and research projects. (a) The Gestix system that responds to the surgeon’s gesture commands in real-time (b) 3D stereo needle insertion guidance system (c) ProMIS simulator is able to evaluate surgeons’ performance. (d) Studierstube project provides multiple approaches-augmented reality (e) Annotation interactive system (f) Xbox Kinect system that allows surgeons to access extra data or images through gestures. ................................................................. 8

2.1 The defined world frame of the system. The center of the world frame is defined in the tip of the camera and the X,Y direction is aligned with the image frame. ................................................................. 12

2.2 Pinhole camera model ................................................................. 12

2.3 The ring menu designed to select patient record [21]. By approaching the tool tip to either quarter of the selection, a command is selected. ........ 14

2.4 First prototype SCI with gesture recognition embedded. (a) shows the trajectory of the moving tool. By moving the surgical tool to form a defined pattern (a circle, in this case) [34], the SCI brings up the menu on the right. (b) shows the menu that has been selected, by opening the grasper, a button within the menu could be selected. ................................................................. 15
2.5 Design of surgical instrument and its concept as a 3D mouse. (a) The design of surgical instrument with button and scroll attached on it. (b) The system tracks the tip of the surgical tool, but, the rotation of the surgical tool is not detected in this proposed framework. ............................................. 16
2.6 (a) The phantom set-up with plastic stomach model and real surgical tools. (b) The light source and camera unit. (c) Inside of the stomach model with real surgical tools. ............................................................... 18

3.1 Tracking the surgical tool using MHI. (a) The original input frame of the tool (b) The MHI of the frame (c) The calculated motion gradient of the image (d) The detected bounding box, motion orientation (the circle) by analyzing the MHI. ................................................................. 21
3.2 MHI motion detection flow chart ................................................................. 23
3.3 Tip extraction given ROI of the tracking target. ............................................. 23
3.4 The model of the ellipse used to represent the tip of the tool. ......................... 24
3.5 PF algorithm ............................................................................................. 26
3.6 Multiple targets tracking failed using MHI when tools are crossing together (a) The original input frame of the tool (b) The MHI of the frame (c) The calculated motion gradient of the image (d) The detected bounding box (red), motion orientation (red circle) by analyzing the MHI. ................................................................. 29
3.7 Tip extraction given the ROI detected using MHI (a) The ROI extracted using MHI (b) Canny detection within ROI (c) The extracted edges (red line), middle line (green line) and tip (green spot) .................................................. 30

4.1 Visual tracking module and flowchart .......................................................... 33
4.2 Surgical tool projected on the image plan. ..................................................... 36
4.3 The illustration of the distance calculation between two edges. If the maximum distance of end points of one line to the other line is less than the threshold ($D < Th$), it is assumed to be in the same bucket, otherwise, it belongs to the other bucket. ................................................................. 39
4.4 (a) Original input frame (b) Converted gray image (c) Output of canny’s edge detector used for extracting edges of the tool. (d) Output of edge image detected using canny detector (e) Classify the edge segments into “left” and “right” edge. The yellow segments are belonging to “left” edge, and red one are “right” edge. (f) Merge edge segments together, forming two edges, “left” edge and “right” edge. The green line is the calculated middle line of the two edges.  

4.5 Kalman Filter recursive algorithm  

4.6 Tip extraction search area after applying filters.  

4.7 Tip extraction search method.  

4.8 Tip tracking result. (a) Tip search in $x$ direction (b) Tip search in $y$ direction  

4.9 Two modification of the tracking algorithm to adapt multiple tracking. (a) The initialization is achieved by placing the tool around the two green lines. The two green lines ($E_1(m_1, b_1), E_2(m_2, b_2)$) are initialized as $[-1.0; 550.0]$ and $[1.0; -50.0]$. (b) The tip search area is being modified so that the area between one tool’s middle line intersecting with another tool’s two edges is removed. The blue regions represent the valid search area for each of the tool.  

4.10 (a) 3D projective model (b) The transverse section of the projective model  

4.11 (a) Image with detected 2D edges and tip (b) Reconstructed tool rendered in OpenGL  

4.12 Screenshots of the successful tracking results. (a)-(b) Current algorithm can handle similar objects in the scene by implementing Kalman filtering. (c) The edges belonging to the tool can be recovered even though part of the tool is being occluded. (d)-(e) The edge detection can handle two similar tools having included angle larger than 20-30 degrees. (f) Current tip extraction module can handle some amount of light reflection caused by special material.
4.13 Screenshots of the failed tracking results. (a) Current algorithm failed to detect edges when the light is weak, (under 30/100 in this case). (b) The shadow blurs the image, causing missed detection of the edge. (c) The tip extraction module failed to detect correct tip due to less contrast of the tip with respect to the background. (d) When the tip was occluded, the detection fails. (e) Current algorithm cannot handle fast motion of the tool. (f) The edge detection cannot handle two similar tools having included angle less than 20-30 degrees.

4.14 The multiple tracking result with successful cases. The green lines are the tool’s middle line, the red are their edges and the white spots are the detected tips. (a) Condition when two tools are separate to each other. (b) Condition when tools’ intersection point is on one of the tool’s middle line (c) Condition when two tools are crossing. (d) Condition when tools’ included angle is 30 degree.

4.15 The multiple tracking result with failed cases. The green lines are the tool’s middle line, the red are their edges and the white spots are the detected tips. (a) Condition when one tool’s tip is beneath another tool. (b) Condition when the tools’ included angle is less than 30 degree.

4.16 Plastic slope used in experiment.

4.17 (a) The partial result of 3D vector plot. Red lines are 3D vectors represent the tools. Blue dots are calculated 3D tool tip position. (b) Separate plot of 3D tip position.

5.1 System software architecture

5.2 SCI design and overall structure

5.3 (a) SCI main menu containing “pre-operative images”, “patient record”, “3D effect” and “Exit”. “3D effect” function is selected. (b) Sub-menus under “Pre-operative images” function, including “CT”, “MRI”, “XRAY” and “Return”. “XRAY” is selected; the right upper window shows the XRAY scan once it has been chosen. (c) Sub-menus under “3D effect” function, including “Image Box”, “Thread manipulation” and “Return”.

5.4 “Patient record” function. Demonstration of the Image overlay idea.
5.5 “Image box” function a) Cube is selected by clicking one of its faces b) The 3D object can be rotated and translated according to the tool’s positioning.

5.6 “Thread manipulation” function, the thread is manipulated based on the position of the tips. This function is only valid in two tools tracking system. The number “1” and “2” represent the label of the tools.

5.7 The camera frame.

5.8 The OpenGL frame.

5.9 The OpenGL transformation pipeline.

5.10 The OpenGL screen coordinate and the image coordinate.

5.11 The orientation of the tool tip $V(x, y, z)$ and rotation angle $R(\alpha, \beta, \gamma)$.

5.12 Screenshots of the tool manipulation procedure that try to register a virtual organ to the target.

5.13 Snapshots of the two tools’ thread manipulation.

5.14 Screenshots of the augmentation accuracy result. (a-c) Virtual frames are rendered on top of calibration paper at depth of 60mm, 90mm and 110mm, respectively. (d-f) Virtual cylinders are rendered on top of each intersection at depth of 60mm, 90mm and 110mm, respectively.

6.1 System overview of the real-time laser tracking system.

6.2 The snapshots of the real-time laser tracking system. The yellow dots represent the detected laser spot. The green dot is the detected tip of the tool and the blue lines are extracted edges of the tool.

6.3 The system overview.

6.4 Typical triangulation.

6.5 Distance constraint between the surface and image plane.

6.6 Distance constraint between lens, block paper and surface.

6.7 Design of block paper.

6.8 One-mirror, multi-spots system configuration.

6.9 Two-mirror, multi-spots system setup.

6.10 Simulation before and after correction.

6.11 (a) Overlapped image with segmented perimeter (b) Segmentation with low threshold.

6.12 Simulation result of one-mirror, multi-spots system.
6.13 (a) The modelled surface by scanning a “cup handle” (b) The experimental result of a Barbie face scan, some features appeared on this reconstructed mesh model. From left to the right, they are Barbie’s mouth, nose, and their eyes, respectively. 93

6.14 Point laser system 94

6.15 (a) The Swept Source OCT prototype system. (b) The scan scheme of the OCT system. 98

6.16 The experimental setup of SS-OCT system. (a)-(b) The SS-OCT system contains swept source laser, a balanced photodiode detector with interferometric reference and sample arm. (c) The sample arm and a 1310 nm grin lens is attached to one of the jaws of the surgical tool. (d) The experiment sample of the OCT system - a chicken heart. 99

6.17 The extracted OCT images by scanning the chicken heart tissue. (a) The OCT image of “muscle” tissue. (b) The OCT image of “fat” tissue. (c) Extracted B scan with both “fat” and “muscle” tissue. 100

A.1 (a) Single action jaw graspers (b) Double action jaw graspers (c) Scissors 111

B.1 Pixel format for each of the library. 113

C.1 Schematic of OCT system based on Michelson Interferometer 116

C.2 SS OCT system setup 118
Chapter 1

Introduction

This chapter briefly introduces minimally invasive surgery (MIS), its related benefits and existing problems. Based on the problems, the research goal and its motivation are presented. In this chapter, some of the current technologies in this field are reviewed. Finally, the contributions of this thesis and its layout are described.

1.1 Challenges in minimally invasive surgery

Minimally invasive surgery (MIS) has been replacing the traditional open surgery since the early 1950s. In MIS, operations are performed using an laparoscope and several long, thin, rigid instruments (Figure 1.1(a)) that are inserted into the body through several small incisions[1]. The benefits include reduced pain, shorter hospitalization times, and improved cosmetic results. Due to these advantages, this minimally invasive technique has been applied to many surgical operations, including esophageal tumors, morbid obesity, malignant liver tumors, gallbladder disease, colon cancer, and robotic prostatectomy[2]. The detailed introduction of the endoscopy and surgical equipments is in Appendix A.

As beneficial as MIS may be for the patients, MIS presents several new challenges for surgeons. It requires a different approach from the conventional surgical procedures, since hand-eye co-ordination is not based on direct vision, but more predominantly on images through endoscopes. Surgeons have to operate by looking at the main monitors displaying cavities at awkward angles with restricted range of views (as illustrated in Figure 1.1(b)) and separate auxiliary monitors displaying the patient records operated by assistants. Even when human assistants are not involved, other assistance, for example, verbal commands
CHAPTER 1. INTRODUCTION

and foot pedals are usually used to request the display of pre-operative images, patient records, and other medical information. Such assistance cannot be well communicated and may cause frustration.

Figure 1.1: (a) Typical instrument arrangement for the gall bladder removal surgery. The endoscopic camera and surgical instruments are inserted into the abdomen through small incisions with approximately 5-15 mm diameter. In the upper left corner, the endoscopic view is displayed. (b) Illustration of MIS procedure in a regular operating room. The surgeons have to look at the monitors while doing the operation.

Another common problem in MIS is that the limited set-up only provides two-dimensional vision, without depth perception, which greatly hinders the surgeon’s visual experience. Although binocular and 3D computer vision techniques make depth perception possible by adding an extra camera, this approach increases the hardware required, and thus, is not widely accepted by clinicians. An alternative approach is to let surgeons wear special eye-pieces or head-mounted displays [3, 4]. However, in general, surgeons find wearing eye-pieces blocks their peripheral vision. Most importantly, the need for a head mounted device and the lack of motion parallax has limited the performance of systems based on this method.

Following the above challenges and problems in MIS, my research goal is to assist the surgeons in an intuitive way by designing a surgeon computer interface (SCI). This interface should be easy to operate with least required hardware and inconvenient switching between surgeons and assistance (food pedals or human assistants) during operation. The following sections will describe the motivations in detail.
1.2 Motivation

There are a number of approaches to improve the existing tools and systems used in MIS. Due to visual limitations, it is desirable for surgeons to obtain information regarding to patient’s anatomy during MIS. Thus, surgeons should be able to access medical images without using verbal command but instead via surgical tools in a standard MIS set-up. This will greatly reduce the communication time and inconvenient switching between surgeons and assistants. Another option is to add the ability to manipulate the images or 3D scans for manual registration with the deformable organ of interest. In this way, the surgeons are able to refer to the registered scans directly during operation.

Keeping these options in mind, my research goal is to develop a SCI that allows the surgeons to use the laparoscopic tools as a 3D mouse, thus removing the need for verbal commands or human assistance. The main assumption is that the surgical instruments are the dominating moving objects in the field of view. In order to build such a system, the underlying tracking system should be robust enough and be able to support multiple 3D tool tracking under single camera. The position and orientation of the tool should be resolved in order to support a functional interaction and graphics framework. In this way, without being interrupted in the middle of a task, the surgeons can deliver tasks to the computer by interacting with the surgical instruments. Examples would be the surgeons can manipulate the superimposed virtual images and videos, or access and display text onto live images. Such development can be further enhanced by developing and following a pre-defined user-interaction protocol.

As to the problem of lack of 3D perception, the ability to view the inner cavity is an advantageous procedure for surgical planning. A laser scanning and an OCT optical system provides huge potential as an functional extension to the above SCI. It could offer the ability to model 3D cavity and live-scan the tissues for guiding and diagnostic purpose.

1.3 Survey of Relevant work

The proposed interactive system can be divided into three subsystems namely as: a) visual tracking b) human-machine interface and c) 3D laser scanning and imaging system. A suitable visual tracking approach needs to be identified to map the motion of the tools in physical space to the augmented virtual space such that the virtual graphics and images
can be manipulated by surgical tools. The SCI provides a control and visualization interface between the surgeons and the computer. A suitable 3D scanning and imaging system provides an extra fusion of information. This section reviews the relevant work that has been done in these areas.

1.3.1 Visual Tracking

Tracking systems are used to determine the target’s position in image-guided procedures. Different approaches to visual tracking have been proposed to solve increasingly complex and diverse tracking conditions, especially for multiple target tracking. In general, the tracking approaches can be categorized into bottom-up and top-down procedures. The bottom-up procedure is to localize and track the target for each independent image sequence. It locates the position of a target by extracting characteristics of the target from image sequences, such as edges, corners, and colors. In a tracking scenario, an object can be defined as anything that is of interest for further analysis. For example, points or centroid[21], primitive geometric shapes[22] (rectangles, triangles and ellipse etc.), Object silhouette and contour[23] and more complex articulated shape models[24] can also be considered as “objects”.

Selecting the right features plays a critical role in tracking. In general, the most desirable property of a visual feature is its uniqueness so that the objects can be easily distinguished from the feature space. For example, color is used as a feature for histogram-based appearance representations. A robust color tracking method is presented in [6], using HSV(Hue, Saturation, Value) color mode. In general, color-based tracking is fast, especially for multiple targets. The main drawback is its lack of robustness and stability in case of variant illumination and partial occlusion. Another problem is that it requires artificial color coding on the target.

Unlike color-based approaches, contour-based tracking is reliable and can cope with partial occlusion and light variance. Algorithms that track the boundary of objects usually use edges as the representative feature since edges usually generate strong changes in image intensities. The most popular one is the Canny Edge detector[25].

There are other visual tracking approaches based on motion detection. One method uses frame differences to create a motion history image (MHI) that represents a function of the recent motion in a sequence[9]. By analyzing the motion, the orientation and velocity of the moving target can be extracted.

All the bottom-up approaches shown above extract a set of features from each frame
(typically boundary-like and corner-like features), and then attempt to establish correspondences between the sets of features. However, the drawback of this approach is its inability to deal with multiple moving targets in a clustered and complex environment. The algorithm often fails to label each of the targets, for example two sets of surgical tools.

Another different approach for video tracking is a top-down process, where the filtering and data association is taken into consideration. They not only track the target based on their salient features (such as edge, color blob and shape) but also involve prior information about the object, dealing with object dynamics and evaluation of different hypotheses (such as constant velocity of the object movement). The computational complexity for these algorithms is usually much higher due to matrix operation and probability-based computation. However, it is more practical in a cluttered environment where multiple targets are being tracked. The system is represented using a state-space model, where the state is predicted based on the system model or estimated based on observation. The approach is recursive in nature and operates in two phases: prediction and update. Each state is first propagated according to system model (prediction stage), then corrected in terms of the observations (update stage). Linear probabilistic trackers, Kalman filter \[10, 11\] addresses the general problem of trying to estimate the state of a discrete-time process that is governed by a linear stochastic difference equation. As conditions for visual tracking grow more versatile and sophisticated, non-linearity and non-Gaussian elements are desirable in order to model the underlying system more accurately. Non-linear trackers, such as particle filtering (PF), are therefore being used \[12, 13\].

All top-down methods need an initialization process, requiring the first target position. Moreover, the observation needs to be generated from the camera sensor each frame to update the system model. Therefore, the filter cannot work without cooperating with feature-based detection or motion analysis. The proposed approach will combine both advantages, forming a hybrid system. This system consists of a bottom-up feature detection and top-down Kalman filtering, supporting multi-target tracking in a noisy environment.

Most tracking methods available in the research field are only focused on 2D analysis. Few research mention monocular-based 3D tracking. Novotny et al.\[26\] provides an approach to fully construct the position and orientation of the instrument in 3D space with the help of passive markers placed on the instrument shaft. To localize endoscopic instruments in 3D space, Tonet, et al. \[6, 7\] proposed color segmentation with the help of a colored strip based on real-time images. However, its 3D tool localization is based on empirical estimation.
which reduces the accuracy and limits its application. The ProMIS surgical simulator[16] from Haptica (courtesy of Haptical Inc.) contains a system to track laparoscopic instruments in 3D by covering them with two strips of yellow tape (a marker). Its tracking is accomplished by analysing images captured with three separate cameras.

As shown above, all the approaches require additional markers or extra instruments, which increases the complexity of system and makes it impractical in a standard OR. Unlike the above methods, a 3D reconstruction module is proposed in this thesis using a less expensive yet accurate approach to solve the 3D pose estimation based on camera projection model with less computation and fewer matrix operations. This method utilizes the prior knowledge of the tool, which is a cylindrical shaped tool with known radius, to map the 2D tracking result to 3D space using pure geometrical operation. This approach, however, needs no extra equipment, markers or additional sensors except a single calibrated endoscopic camera.

1.3.2 Surgeon-computer Interface (SCI)

A suitable surgeon-computer interface is crucial for surgical application. In order to assist surgeons in an intuitive way, a large number of potential SCIs have been proposed. A hand gesture system[14] (Figure 1.2(a)) for medical images manipulation called Gestix was developed collaboratively by Ben-Gurion University of the Negev and Washington Hospital Center. The Gestix system responded to the surgeon’s gesture commands in real-time. In an in-vivo experiment, this type of interface did not distract surgeons and allowed them to react rapidly, intuitively, and naturally. UNC Chapel Hill offers an interactive, 3D stereo needle insertion guidance system[15] (Figure 1.2(b)) that helps surgeons with intra-operative needle insertion, where the virtual objects identify the location of the tumor and guide the needle to its target. ProMIS[16] (Figure 1.2(c)) uniquely enables users to interact with virtual and physical models in the same unit while providing accurate and comprehensive feedback on performance. Studierstube[17] (Figure 1.2(d)) developed by Vienna University of Technology allows the user to combine augmented reality, projection displays, and ubiquitous computing to the interface as needed. The environment is controlled by the Personal Interaction Panel, a two-handed, pen-and-pad interface that has versatile uses for interacting with the virtual environment. Researchers at Columbia demonstrated an annotation system[20] (Figure 1.2(e)) with the notion of attaching windows from a standard user interface onto specific locations in the world, or attached to specific objects as reminders. Microsoft Co. is also
promoting new uses for its Kinect device, including stroke patient rehabilitation and cleaner hospital operating theatres. Xbox’s Kinect technology has been adapted to let doctors perform certain tasks during surgery[19] (Figure 1.2(f)). It allows the surgeon to stand in front of the screen, using hand gestures detected by Kinect to move the images on the screen.

Most of the systems mentioned above require either extra equipment or other human assistance to operate. Compared to the available research projects and commercial products above, the design concept of our system has many novel aspects. Firstly, our proposal requires minimal change to the existing surgical setup, without cumbersome equipment. The product is compact and portable. Secondly, our proposed design is a novel combination of video tracking, graphics system and human-interactive design. Therefore, it takes advantage of the various design benefits from above configurations, such as displaying images and manipulating a 3D virtual object hands free. Finally, the system itself has big potential that is not restricted to surgical application. It can be applied to evaluate a trainee’s performance by monitoring the tool motion. It can also be extended to mobile game application, where users are able to use their fingers to interactive with virtual games.

1.3.3 3D laser scanning and reconstruction

As to the problem of lack of 3D perception, Hayashibe et al.[27] developed a laser-pointing endoscope using an optical galvano scanner and 955fps high-speed camera. The laser-pointing endoscopic system acquires and visualizes the shape of the area under the endoscopic camera. Bartosz et al.[28] developed a system based on a contact-free range measuring endoscope that measured distances by the method of laser triangulation. The range measuring endoscope was integrated to the navigation system by attaching an optical tracker and performing a geometrical calibration. In this thesis, we will only examine the basic concept of the 3D inner cavity reconstruction using laser range data in MIS with few modifications based on Xiaoli’s method[33].

Beside 3D surface reconstruction, high-resolution imaging is important in disease diagnosis. Optical-coherence tomography (OCT) is an optical imaging technique that allows high-resolution cross-sectional imaging of tissue micro-structure. The high resolution images provided by OCT has the potential to distinguish normal from cancerous tissue. McLaughlin et al.[29] demonstrated the possibility of diagnosing breast cancer with OCT needle probes. Inoue et al.[30] developed endoscopes that can generate high-quality images of both living
Figure 1.2: Related commercialized products and research projects. (a) The Gestix system that responds to the surgeon’s gesture commands in real-time (b) 3D stereo needle insertion guidance system (c) ProMIS simulator is able to evaluate surgeons’ performance. (d) Studierstube project provides multiple approaches-augmented reality (e) Annotation interactive system (f) Xbox Kinect system that allows surgeons to access extra data or images through gestures.
cancer cells and normal cells in the gastrointestinal tract. Hariri et al.[31] developed the first laparoscopic OCT device to characterize the micro-structural features of human ovaries in-vivo. This thesis focuses on the preliminary implementation of the OCT imaging system in the current set-up, exploring the possibility of integrating such a diagnosis tool in a future interactive paradigm.

1.4 Contributions and layout of the thesis

The proposed method in this thesis is a further development based on several previous systems developed in the “Experimental Robotics and imaging lab” at Simon fraser university(SFU), including “Blob tracking with Ring menu interface” and “Tool motion and Gesture recognition system”. The proposed system in this thesis provides a new tracking framework with the ability to resolve the pose of the tool and to support the multiple tool tracking. It is designed to track the motion and position of surgical instruments in a non-robotic surgical set-up so that the surgeons can use the instruments as a 3D non-contact mouse to interact with an overlaid augmented reality environment. By manipulating the surgical tools, surgeons are able to access medical information and pre-operative 3D image scans during surgery without introducing extra pieces of hardware. Contributions of this thesis are as follows:

- A novel 3D image tracking system using single laparoscopic camera. A simple yet robust tracking system is developed that utilizes Kalman filter implementation to track the 2D position of the tool. The tracking framework then calculates the tools’ spatial position through a 3D reconstruction module. The system is able to track multiple instruments, providing more capability for the future interactive design.

- An enhanced 3D mouse interactive SCI. Taking advantage of the tracking system, an enhanced non-contact mouse interactive system proposed in this thesis is able to provide an intuitive interface to assist the surgeons. It offers a novel 3D mouse that can be manipulated by surgeons, such as to show digital images, virtual organs and patient records during medical procedures by synchronizing the movement of the surgical instrument instead of touching a screen, keyboard or mouse.

- Preliminary research of 3D surface reconstruction and OCT imaging. This thesis provides the basic proof of concept of the laser reconstruction system and OCT imaging
system. It explores the possibility of using such tools for imaging and diagnostic purposes. The proposed real-time laser tracking system also extend the triangulation result based on the 3D tracking system mentioned in Xiaoli’s thesis[33].

The organization of the rest of the thesis is as follows: Chapter 2 gives an overview of the current 3D interactive system. It introduces the framework that this thesis has built on. Chapter 3 presents the preliminary studies of the image tracking system and its experimental results. Chapter 4 introduces the new 3D mouse interactive paradigm, its design issues and implementation. Chapter 5 describes the preliminary studies of the laser imaging system and the corresponding results. Chapter 6 concludes the entire work and proposes the possible future works based on the current achievements.
Chapter 2

Background Material

In this thesis, the proposed interface and tracking system is established based on the previous research in SFU. The first section presents the camera calibration approach, which is the prerequisite for tracking framework. Two interfaces developed previously are reviewed in Section 2.2 and 2.3, followed by the new interface design presented in Section 2.4. The experimental setup is presented in Section 2.5.

2.1 Endoscope calibration method

Defining the pixel-to-world mapping is known as camera calibration. The process of camera calibration is to establish the relationship of the 2D point position in pixel coordinates with its 3D point position in world coordinates. Once the camera has been calibrated, we can use the obtained mapping parameters to transform the 2D tracking elements to their real-world equivalents.

Typical calibration parameters can be classified into two classes: extrinsic parameters and intrinsic parameters. The extrinsic parameters are the coordinate system transformations from 3D world coordinates to 3D camera coordinates. Since the world coordinates are defined at the camera center in this framework and are oriented along the Z axis (as shown in Figure 2.1); the world coordinates coincide with the camera coordinates. Therefore, only intrinsic parameters are considered in this thesis, namely, the effective focal length and image center.

Figure 2.2 shows a basic geometry of pinhole camera model with center of projection O and the principal axis parallel to the Z axis. The image plane is at focus and hence focal
CHAPTER 2. BACKGROUND MATERIAL

Figure 2.1: The defined world frame of the system. The center of the world frame is defined in the tip of the camera and the $X,Y$ direction is aligned with the image frame.

length $f$ away from $O$. $(X,Y,Z)$ the 3D coordinates of an object $P$ in the camera coordinate system. $P_c(u,v)$ are the image coordinates of $P(X,Y,Z)$ projected onto the image plane. $(o_u,o_v)$ is defined as the image center in pixel units with respect to image frame origin. Therefore, when $P_c(u,v)$ and $P(X,Y,Z)$ are known, we have the following relationship:

$$f \frac{Z}{Z} = \frac{u - o_u}{X} = \frac{v - o_v}{Y} \quad (2.1)$$

Figure 2.2: Pinhole camera model

The focal length can be calculated in pixel units as:
\[ f = \frac{(u - o_u)Z}{X} = \frac{(v - o_v)Z}{Y} \]  

(2.2)

Similarly, when \( f, o_u, o_v \) and \( u, v \) are known parameters, we can calculate the depth information, \( Z \), of an object in the field of view.

Using the method proposed in [32], we can estimate the intrinsic parameters as shown in Table 2.1.

Table 2.1: Calibration results

<table>
<thead>
<tr>
<th>Calibration variables</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image center (pixel)</td>
<td>((o_u, o_v) = [320.0; 240.0])</td>
</tr>
<tr>
<td>Focal Length (pixel)</td>
<td>(f = 394.66)</td>
</tr>
</tbody>
</table>

2.2 Blob Tracking and Ring menu interface

The “blob tracking” method is based on color-based tracking. Because of the properties of the linear perspective vision, when a cylindrical shaped tool is projected onto the image plane, its geometrical shape will change according to how far it is located with respect to the center of projection regardless of tools’ relative orientation. As a result, the diameter of the projected tool, along with parameters associated with the calibrated camera and real physical diameter of the tool are used to determine the depth. Zhang and Payandeh [32] use an attached marker to simplify the tracking task.

The depth information is obtained using formula derived from equation:

\[ Z = \frac{fX}{u - o_u} = \frac{fY}{v - o_v} \]  

(2.3)

Salajegheh’s method[21] uses the same procedure except that she applies different colors for recognition, for example, blue and green (whichever has the largest contrast with the background). The algorithm can detect multiple markers at the same time, making multiple tool tracking feasible. The biggest problem of this approach is that the blob pixels cannot be detected correctly with variant light conditions. When light is bright, the pixel is highly saturated to be detected. On the other hand, when the light is dark, the blob color tends to have less contrast with the background, hence, the pixels are also failed to be detected. Both conditions result in partial recognition of the desired marker, resulting in inaccurate tracking.
CHAPTER 2. BACKGROUND MATERIAL

The interface prototype (as shown in Figure 2.3) is designed by Mandana[21] based on the blob tracking framework. It is a ring menu to access four different types of image records. The desired images can be selected by entering the tip of the surgical tools into the corresponding regions. The tool tip coded with blue marker is being tracked by the system as a cursor; whenever the tip is approaching one of the menus, the upper-right sign highlights the image to be displayed. The action is confirmed by holding the selected region for one second or pressing a wireless switch attached to the surgical tool.

Figure 2.3: The ring menu designed to select patient record [21]. By approaching the tool tip to either quarter of the selection, a command is selected.

This “Ring menu SCI” provides one of the initial prototypes of the 3D non-contact mouse idea. Although the simplicity of the algorithm provides a fast system, the drawback is that the tracking itself is not accurate due to the variety of light conditions and possible partial occlusion of tissues or other object. Moreover, due to the lack of information of the orientation of the tool, the SCI prototype cannot be extended further with more complicated interaction, such as simple gesture design or 3D object manipulation.

2.3 Tool motion and Gesture recognition system

To provide a direct and intuitive SCI, Hsu [34] developed another robotic surgical system based on gesture recognition. The system enables surgeons to command the laparoscopic surgery robotic system by performing tool gestures with the surgical tools in hand.

The interface based on this tool gesture recognition system is designed to use one tool
CHAPTER 2. BACKGROUND MATERIAL

Figure 2.4: First prototype SCI with gesture recognition embedded. (a) shows the trajectory of the moving tool. By moving the surgical tool to form a defined pattern (a circle, in this case) [34], the SCI brings up the menu on the right. (b) shows the menu that has been selected, by opening the grasper, a button within the menu could be selected.

The previous “ring menu interface” and “tool gesture interactive system” only provide the users with limited functionality, namely as 2D image display and tool tip manipulation. In this thesis, a more robust and comprehensive interface is proposed. It enables surgeons to access 2D images similar to the previous systems. Most importantly, the system uniquely
supports the orientation of the tool. The pose information (3D position and orientation), as a result, enable surgeons to manipulate 3D objects and manually register virtual scans. The rest of the section will introduce the overview architecture of the system and its hardware components.

In our proposed set-up, button and scroll wheel are attached to the handle of the standard surgical tool, which have the same analogy to the traditional mouse. The conceptual diagram of our proposed design is shown in Figure 2.5.

![Design of surgical instrument and its concept as a 3D mouse. (a) The design of surgical instrument with button and scroll attached on it. (b) The system tracks the tip of the surgical tool, but, the rotation of the surgical tool is not detected in this proposed framework.](image)

By pressing the button (Figure 2.5(a)), certain tasks can be selected. The scroll wheel can be designed to scroll pages, menus or change the perceptual view of a 3D virtual object. Similar to mouse position tracking, the pivoting motion of the surgical tools inside the abdominal cavity are tracked (cursor location) with respect to the endoscopic camera. Moreover, the tool (Figure 2.5(b)) offers additional degree of freedom compared with the traditional computer mouse - the depth coordinate (motion along the z-axis). This additional motion can be applied to manipulate a 3D augmented virtual object, such as a virtual organ or virtual 3D menu box.

In order to synchronize the movement of the tool with the virtual graphical objects, the designed software tracks the tool motion and re-interprets its real position into its virtual space. Therefore, the system contains two major parts: The first one is the 3D
real-time tracking module. It calculates the 3D position of the tool for each single frame. The underlying tracking framework utilizes a novel Kalman filter to track the 2D position, followed by a 3D reconstruction module to calculate its spatial coordinate. The second part—graphics rendering module—renders the graphics on the screen based on the previous tracking result. Therefore, the interaction is achieved based on the combination of the real-time video tracking and its virtual visual enhancement. Chapter 4 and 5 will introduce each module in detail.

2.5 Phantom Experimental Setup

2.5.1 Hardware

The experimental set-up consists of a plastic stomach model made by Limb & Things Company and surgical tools such as gaspers and scissors as shown in Figure 2.6(a). The rubber organs and rubber tissue inside the plastic stomach mimics a real laparoscopic surgery.

The computer contains a Pentium(R) 4 3.00GHz CPU and 1GB of RAM. The Operating system running on this computer is Microsoft Windows XP.

Images are captured by the Karl Storz Supercam 9050B CCD color camera, as shown in Figure 2.6(b). The CCD sensors inside the camera are electronic systems which transform the real image (photons) into electronic images which can be read on a screen. The Supercam is a small, lightweight, easy to operate, high resolution camera for video monitoring and recording of laparoscopic procedures. The images displayed on the monitor are acquired at a resolution of 640 \times 480 pixels (as shown in Figure 2.6(c)).

Illumination is provided by the Karl Storz 615 xenon light source through a fiber optic cable (as shown in Figure 2.6(b)). The xenon lamp color temperature approximates bright sunlight and is considered unmatched for visual and photographic color rendition. The number 54 displayed on the light source indicates the brightness. Brightness can be adjusted by the switch on the control panel.

The captured images are transferred to the camera unit, which is the box at the bottom in Figure 2.6(b). The camera unit then feeds the images to the Matrox CoronaPlus frame grabber in the image processing computer.

The Matrox CoronaPlus frame grabber is installed to grab live images from camera. It can transfer acquired images to either system (host CPU) memory for processing or display (graphics card) memory for video-in-a-window at live video rates. It handles bandwidth
Figure 2.6: (a) The phantom set-up with plastic stomach model and real surgical tools. (b) The light source and camera unit. (c) Inside of the stomach model with real surgical tools.
CHAPTER 2. BACKGROUND MATERIAL

transfer of up to 32-bit color image over the PCI bus. At the same time, with dual frame
buffer architecture, it can display 32-bit true-color live video with true-color non-destructive
overlay (no influence on the original image). Its RGB digitizer’s sampling rate is up to
33MHz. The video decoder can accept composite(CVBS) and component S-video(Y/C) in
NTSC/PAL formats, and monochrome video in RS-170/CCIR. It also features a 32-bit/33
MHz PCI bus master to reduce CPU usage.

2.5.2 Software

The experiment workstation consists of Visual Studio 6.0 platform for image processing,
acquisition, color analysis, camera calibration and data extraction. Image processing pro-
cure, such as, edge detection, blob analysis and image display is carried out by Matrox
Imaging Library (MIL) and Open Source Computer Vision Library (OpenCV). MIL is a
hardware independent, modular 32-bit imaging library that includes ActiveMIL, a collec-
tion for managing image capture, transfer, processing analysis and display. In general, MIL
can manipulate binary, grayscale, or color images. It enables fast application development
use. OpenCV is a library of programming functions mainly aimed at real time computer vi-
sion and real-time image processing, developed by Intel. The vast variety of functions could
be used, providing a convenient and fast way for our application. Finally, 3D rendering is
accomplished using OpenGL. OpenGL (Open Graphics Library) is a standard specification
defining a cross-language, cross-platform API for writing applications that produce 2D and
3D computer graphics. The interface can be used to draw complex three-dimensional scenes
from simple primitives.
Chapter 3

3-D image tracking: Preliminarily studies

Recently, the field of image tracking has been growing rapidly and has been gaining application in many areas. These include successful attempts in the field such as fiducial-based color tracking[32], feature based tracking [35, 36], movement tracking [35, 36], statistical-based visual servo tracking[38] and PCA(Principal component analysis)-learning based tracking[39]. Each of the tracking algorithms has its applications and environmental constraints. This chapter presents the preliminary studies in the field by analysing two existing tracking approaches, Motion history image(MHI)[40] and Particle filtering (PF) approach[12]. These two methods provide us with the analytical thinking evolved to develop the new tracking method presented in the next chapter. Introduced in Section 3.1, Motion History Image (MHI) defines a region of interest (ROI) and is able to extract the orientation and magnitude of the target by tracking the motion of the tool. Section 3.2 introduces another robust stochastic approach for tracking a color-coded target based on particle filtering.

3.1 Motion tracking using Motion history image (MHI)

This approach uses the idea of frame difference to create a motion history image (MHI) that represents a function of the recent motion in a sequence. A MHI is an image obtained by compressing the image-time volume onto a single image. The intensity values in the MHI indicate the time at which that pixel last registered a motion or presence of the object.
A MHI captures the essence of the underlying motion pattern for the tool by taking the difference of $N$ frames simultaneously over last $N$ time steps[9]. By analyzing the motion, we can extract information such as the motion orientation and velocity of the tool.

The MHI is updated after each sampling period by adding newly detected frame difference. For $N$ frames, the simplest frame difference subtraction is applied. The $N$ image silhouettes are converted into a single template. Every time a new frame arrives, the existing silhouettes are weighted to be less important; brighter MHI values depict more recent motion. Then the new silhouette is overlaid based on this weight, which in this case, the binary intensity. In this way, a MHI is created for further image analysis.

Figure 3.1: Tracking the surgical tool using MHI. (a) The original input frame of the tool (b) The MHI of the frame (c) The calculated motion gradient of the image (d) The detected bounding box, motion orientation (the circle) by analyzing the MHI.
Figure 3.1(b) shows an example of the MHI representing the tool. The more recent movement (left of MHI) has larger intensity than the past (right of MHI) events.

Once the MHI is obtained, motion gradients can be efficiently calculated by convolution with separable Sobel filters in the $x$ and $y$ directions yielding the spatial derivatives\[9\]. $F_x$ and $F_y$ are defined as:

$$
F_x = \frac{1}{8} \begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1
\end{bmatrix},
F_y = \frac{1}{8} \begin{bmatrix}
-1 & -2 & -1 \\
0 & 0 & 0 \\
1 & 2 & 1
\end{bmatrix}
$$

(3.1)

If the resulting response at the pixel location $(x, y)$ is denoted as $F_x(x, y)$ and $F_y(x, y)$ Gradient orientation at each pixel is given by:

$$
\phi(x, y) = \arctan \frac{F_y(x, y)}{F_x(x, y)}
$$

(3.2)

And the magnitude of the gradient is:

$$
Mag(x, y) = \sqrt{F_x^2(x, y) + F_y^2(x, y)}
$$

(3.3)

After calculating the motion gradients (Figure 3.1(c)), we can then extract motion features to varying scales. In this case we can track multiple elements simultaneously similar to the proposed method in \[40\]. As a result, the different tools can be initialized and labelled.

Once the MHI is obtained, a pixel dilation and region growing method is applied to remove the noise. The motion is then segmented relative to object boundaries to find out the moving flow for each of the tools, as shown in Figure 3.1(d). The process flow chart is presented in Figure 3.2. Detailed implementation is in \[40\].

Given the ROI of the tracking target, the tip of the tool can be detected easily. The tip is estimated to be the point of intersection between the middle line of the tool and the bounding box (Figure 3.3). The middle line always intersects with two edges, generating two intersection points. To choose the valid tip location, we made the assumption that the tip is always closer to the center of the view.

### 3.2 Tracking using Particle filtering (PF)

Particle filtering \[12, 41\] is a robust tracking method for a cluttered and occluded environment. The main objective is to track a parameter of interest as it evolves over time in a
CHAPTER 3. 3-D IMAGE TRACKING: PRELIMINARILY STUDIES

Figure 3.2: MHI motion detection flow chart

Figure 3.3: Tip extraction given ROI of the tracking target.
non-linear and multi-modal dynamical system. Due to the nature of the non-linearity, the approach is based on the Monte Carlo integration and uses the posterior probability density function (PDF) by a set of random samples or particles \( \{(x_t^{(i)}, \pi_t^{(i)})|i = 1 \ldots N\} \) with associated weights. Each sample \( x_t^{(i)} \) represents the state of the object, with a corresponding discrete sampling probability \( \pi_t^{(i)} \), where \( \Sigma_{i=1}^{N} \pi_t^{(i)} = 1 \).

The dynamical system is represented using a state-space model, where the unknown state \( x_t \) is predicted based on the system model (posterior density \( P(x_t|z_t) \)) or estimated based on observation \( z_1, \ldots, z_t \) up to time \( t \) (observation density is \( P(z_t|x_t) \)). In our experiment, the unknown state is the location of the tool and the observation is color-based likelihood.

The approach is recursive in nature and operates in two phases: prediction and update. The evolution of the sample set is propagated in each sample according to the system model (prediction stage). Each element of the set is then weighted in terms of the observations (update stage).

The particle filter in our experiment is in color-based context [12], similar to an adaptive color-based PF [41]. We use an ellipse to represent the marker, as shown in Figure 3.4.

![Figure 3.4: The model of the ellipse used to represent the tip of the tool.](image)

The state vector represents the marker which in our case is a 7-tuple vector:

\[
x_t^{(i)} = \{x, y, \dot{x}, \dot{y}, H_x, H_y, \theta\}
\]  

(3.4)

Where \( x, y \) specify the location of the ellipse, \( \dot{x}, \dot{y} \) the velocity and \( H_x, H_y \) the radius of the ellipse used to represent the marker in both directions and \( \theta \) the rotation angle of the marker or the tool. The prediction or propagation is assumed to be:

\[
x_t^{(i)} = Ax_{t-1}^{(i)} + Rw_{t-1}
\]  

(3.5)
Where $A$ and $R$ are the deterministic components of the model, which determine the translation and the rotation of the ellipse. And the $w_{t-1}$ is a multivariate Gaussian random variable. In our experiment, the tool is moving linearly with constant velocity and constant changing of radius and orientation, thus, we can simplify the model to be:

$$x_t^{(i)} = \{x + \dot{x}, y + \dot{y}, \hat{x}, \hat{y}, R(H_x), R(H_y), \theta w_{t-1}\}$$  \hspace{1cm} (3.6)

The update stage is to re-weight the samples by observation. To determine the color distribution inside the elliptic region. We calculate the color distribution histogram for each of the HSV channels.

At each iteration cycle, the samples needs to be reweighted according to the likelihood comparing to the color distribution template with $m$ bins. Here we measure the likelihood function between two distributions $p^{(u)}$ and $q^{(u)}$ ($u = 1, 2, \cdots, m$) by calculating Bhattacharyya coefficient.

$$\rho[p, q] = \sum_{u=1}^{m} \sqrt{p^{(u)} q^{(u)}}$$  \hspace{1cm} (3.7)

Small Bhattacharyya distances correspond to larger weights. Then the weight can be updated as:

$$\pi^{(i)} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{(\rho[p, q] - 1)/2\sigma^2}$$  \hspace{1cm} (3.8)

by a Gaussian model with variance $\sigma$.

Given a particle distribution, an evaluation criterion need to be developed. Three different methods exist: First is the weighted mean $E[x_t] = \sum_{i=1}^{N} \pi^{(i)} x_t^{(i)}$; second, the best particle $\pi_t = (\max(\pi_i) | i = 1 \cdots N)$ can be used, third, the robust mean. In our case, we applied weighted mean method for simplicity. Figure 3.5 shows the basic flow of the particle filtering algorithm. Detailed implementation is in [12].

Note that the particles need to be re-sampled when the number of particles inside the ellipse is lower than defined threshold. The re-sampling process is necessary, otherwise the particles will diverge.
CHAPTER 3. 3-D IMAGE TRACKING: PRELIMINARILY STUDIES

Figure 3.5: PF algorithm

- **Initialization Stage**
  - Establish the color histogram template
  - Initialize the velocity and ellipse model
  - Set Uniform weight to N samples

- **Prediction Stage**
  - Predict next state by applying linear stochastic equation
    \[ x_t^{(i)} = A x_{t-1}^{(i)} + R w_{t-1} \]

- **Update Stage**
  - Calculate Bhattacharyya coefficient
    \[ \rho[p,q] = \sum_{a=1}^{m} \sqrt{p^{(a)} + q^{(a)}} \]
  - Reweight Sample
    \[ \pi^{(i)} = \frac{1}{\sqrt{2\pi\sigma}} e^{-\rho[p,q]-1/2\sigma^2} \]
  - Ignore small weight samples

- **Resampling**

- **Estimate Mean Tool Pose Estimation**

- **Decision**
  - Effective Number of Samples < Threshold?
    - Yes (Y) or No (N)

---

**Figure 3.5:** PF algorithm
3.3 Experimental studies for MHI and PF approaches

Table 3.1 shows the frames of 12, 51, 52, 138 resulting from MHI and PF methods. In frame 52, because of the fast motion of the tool, the bounding box generated in MHI is bigger than the actual border. However, the box always gives the maximum upper-limit bounding of the tool. Thus, it can be utilized for Region of interest (ROI) generation in initialization phase.

The tip and contour of the tool can also be extracted in a simple way, which can be later on set as an initial position for further 3D tool localization. For example, the MHI segmented image can be utilized as the initial template for the later particle filtering or Kalman filter algorithm to trace the tool locations.

The initial set-up for PF tracking is shown in Table 3.2. Since the computational time increases as the number of particles increases, only 100 particles are selected to run using this method, which is experimentally proved to be sufficient. The particle velocity is selected to be 4.5 to adapt the moderate speed of tool movement. The resampling threshold is set to be 60% for particle redistribution. This means that when there are less than 60% of particles, the PF will resample the particles. Different number of bins is selected such that the color property is preserved even if the light is not constant. Finally, the measurement noise is set to be 0.2 as a standard input.

The tracked object model is represented by a weighted histogram which takes both color and shape into account. The approach is very robust against light variance and short absence of tools in the field of view. In frame 51, when the tool marker is absent, the samples diverges quickly; in frame 52, the samples relocate the marker and converged again.

3.4 Discussion

In summary, the MHI is a fast algorithm for multiple target tracking, and it offers a simple approach for the tip extraction. However, when several tools are moving or crossing together, the motion segmentation fails to detect the separate region for each of the bounding box (as shown in Figure 3.6).

Another drawback is that when the tool is moving fast, the ROI is larger than the tool’s actual bounding box, causing the inaccurate estimation of the tip location (as shown in Figure 3.7). Compared to MHI, the PF approach is more robust, and it allows occlusion
Table 3.1: Tool tracking result using MHI & PF method

<table>
<thead>
<tr>
<th>Frame #</th>
<th>MHI</th>
<th>PF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame = 12</td>
<td><img src="image1" alt="MHI Frame 12" /> <img src="image2" alt="PF Frame 12" /></td>
<td><img src="image3" alt="PF Frame 12" /> <img src="image4" alt="MHI Frame 12" /></td>
</tr>
<tr>
<td>Frame = 51</td>
<td><img src="image5" alt="MHI Frame 51" /> <img src="image6" alt="PF Frame 51" /></td>
<td><img src="image7" alt="PF Frame 51" /> <img src="image8" alt="MHI Frame 51" /></td>
</tr>
<tr>
<td>Frame = 52</td>
<td><img src="image9" alt="MHI Frame 52" /> <img src="image10" alt="PF Frame 52" /></td>
<td><img src="image11" alt="PF Frame 52" /> <img src="image12" alt="MHI Frame 52" /></td>
</tr>
<tr>
<td>Frame = 138</td>
<td><img src="image13" alt="MHI Frame 138" /> <img src="image14" alt="PF Frame 138" /></td>
<td><img src="image15" alt="PF Frame 138" /> <img src="image16" alt="MHI Frame 138" /></td>
</tr>
</tbody>
</table>

The table shows the PF and MHI results with the same video frames. The first column of the image shows the single tool tracking by using MHI-based tracking. These two sets of snapshots are the tracking results together with their MHI when motion of a single tool is being tracked in the view of the stationary camera. The green rectangle represents the bounding box of the whole object and the red circle with a clock in it represents the motion flow. The second column of the image shows the tracking result by applying PF. The yellow ellipse represents the expected object location. Red dots are the distributed particles. The green curves record the trajectory of the object. By tracking the red marker, we can trace the desired marker efficiently.
CHAPTER 3. 3-D IMAGE TRACKING: PRELIMINARILY STUDIES

Table 3.2: Initial parameters for PF

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Particles</td>
<td>100</td>
</tr>
<tr>
<td>Particle Velocity</td>
<td>4.5</td>
</tr>
<tr>
<td>Resampling Threshold</td>
<td>60</td>
</tr>
<tr>
<td>Number of bins in $H$</td>
<td>6</td>
</tr>
<tr>
<td>Number of bins in $S$</td>
<td>4</td>
</tr>
<tr>
<td>Number of bins in $V$</td>
<td>4</td>
</tr>
<tr>
<td>Measurement Noise</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Figure 3.6: Multiple targets tracking failed using MHI when tools are crossing together (a) The original input frame of the tool (b) The MHI of the frame (c) The calculated motion gradient of the image (d) The detected bounding box (red), motion orientation (red circle) by analyzing the MHI.
CHAPTER 3. 3-D IMAGE TRACKING: PRELIMINARILY STUDIES

Figure 3.7: Tip extraction given the ROI detected using MHI (a) The ROI extracted using MHI (b) Canny detection within ROI (c) The extracted edges (red line), middle line (green line) and tip (green spot).

of the tool in a short period of time. Its disadvantage is that, due to PF’s non-linearity and Monte-Carlo sampling process, the computational cost of the problem with a large set of particles will be significantly large. To achieve real-time tracking, multi-thread or parallel computing has to be considered.

These two approaches have their own pros and cons; the proposed approach of developing a simple yet robust algorithm is to combine their advantages. Although PF is computationally costly, however, a similar approach like “Kalman filter” offers a compromising alternative (presented in Chapter 4). Although MHI is not reliable when targets are cluttered together or there is fast movement of the target, it can be used in the initialization procedure as a future work.
Chapter 4

A new feature-based tracking algorithm

The purpose of the tracking algorithm is to calculate the position and orientation of the surgical tool in 3D space. Assuming the tool is uniformed cylindrical shape with known physical diameter, we need at least its edge position and tip position to characterize a tool in 3D space. In Chapter 3, two tracking approaches are presented: MHI (Motion History Image) and PF (Particle Filtering). However, neither of these methods can solve the three-dimensional reconstruction of tools without combining the information of model-based information, namely edges and tip. Therefore, a more robust and novel 3D tracking system is proposed in this chapter combining both stochastic approach and feature-based recognition. Our assumption here is that the camera is static and it is calibrated beforehand. Another assumption is that the physical size of the tool is given. In this thesis, the tracking module serves as the base unit for future 3D surgeon-computer interactive system. However, the instrument tracking is very challenging, we summarizes as follows:

- **Few textures on the instrument.** Most tracking algorithms use texture analysis to detect/track the object, which tends to be more robust and feasible. Without textures, few tracking algorithms can be adapted into this problem.

- **Dynamic environment of the visual scene in real surgical operation.** The real surgical operation has a complicated background, such as organs, tissues, multiple surgical tools, etc, making the recognition of the surgical instrument very challenging, especially for multiple tool tracking. Another challenge is the light reflection of the scene,
which makes recognition problem even more difficult.

- **Partial occlusion and fast movement of the surgical instruments.** The surgical instrument can be contaminated by blood or partially occluded by smoke during surgical operation. Moreover, the fast movement of the surgical instrument blurs the images, causing loss of the tracking. A high speed camera can solve the problem, however, it adds the complexity and cost of the OR set-up.

- **The constraint of the surgical application.** The tracking must be accurate and fast enough (10-60 fps) to be applied to surgical routine.

- **To make minimum changes of the real surgical set-up, a single camera is used.** To map 2D images to a 3D position, multiple cameras are required; otherwise, the model of the target has to be given. In this case, with a single camera, various assumptions have to be made to implement 3D tracking.

Due to the above reasons and challenges, the surgical tracking becomes extremely difficult. To overcome the problems, we first apply Kalman Filter approach (similar to PF) to add robustness to the system and at the meantime support the multiple tracking. Then, with a designed edge/tip extraction module, we are able to partially solve the occlusion problem, due to the usage of long edges instead of markers. Finally, the 3D reconstruction module allows us to take advantage of the single camera to map 2D information to 3D. This chapter will introduce the new system with above features. Section 4.1 gives an overview of the proposed tracking system. Section 4.2 introduces the 2D tracking of the surgical tool, where the edges and tip are detected in the video sequences. In section 4.3, 3D reconstruction of the tool is presented. The experimental results of the new proposed method are presented in Section 4.4. Comparison with the previous tracking algorithms developed in Experimental Robotics and imaging lab are presented in Section 4.5. The experimental studies include the comparison of functionality, accuracy and robustness in a dynamic and noisy environment.

### 4.1 A new feature-based tracking algorithm overview

A motion based MHI is able to extract the motion of the target. The particle filter is able to track multiple targets in a noisy and dynamic environment. However, neither of
the approaches above can be applied to analyze a target in 3D. The reason is that the full geometrical reconstruction needs not only the position of the tip but also the two edges assuming the cylindrical shape of the rigid tool. This thesis proposes a simple algorithm which solves the 3D tracking problems with both 2D tracking and 3D reconstruction module. Various comparisons are made at the end showing the advantage over the previously proposed tracking algorithms.

![Visual tracking module and flowchart](image)

The proposed visual tracking algorithm is different from the previous methods. This method uses the shape and feature analysis instead of color-based tracking, as such, no marker is needed. The overview of the current tracking algorithm (as shown in Figure 4.1) is as follows: The program obtains the 2D position of the edge and tip of the surgical tool as an initial input for Kalman filter tracking. Once the initialization is completed, the tracking algorithm initiates. First it handles the pre-processing of the image frames. Then it tracks the tool orientation vector (middle line of the tool) and updates its Kalman state. If the tool is lost during the tracking, the system re-initializes its state; otherwise, the system continues its tracking. Based on the orientation estimation of the middle line, edges that don’t belong to the tool are filtered. Then, the tool tip extraction module extracts the tip position based on the middle line of the tool. Finally, the 3D reconstruction module takes the two valid edges and tip position as the input and geometrically reconstructs the position and orientation of the tool in 3D space. The remain section will further highlight some of
the details about the proposed method.

4.2 Two dimensional tracking of surgical tool using feature-based tracking

In order to calculate the 3D pose of the tool, its 2D information have to be extracted first, namely edges and tip of the tool. The tracking algorithm has three major modules. The first module handles the initialization and pre-processing of the image frames, which is a necessary step for later Kalman detection. The second module is Kalman filter module, whose purpose is to track the middle line of the tool and filter out the edges (coming from background or other objects) that don’t belong to the tool. The last module is to extract the 2D edges and tip location. The module is processed in the order of Canny edge detection, Hough transform, edges post-processing and tip extraction. Canny edge detection, Hough transformation and edges post-processing serves the purpose of edge extraction. After the edges of the tool is extracted, the middle line of the tool can be calculated. Since the tip is lying on the middle line of the tool assuming the rigid shape of the tool is symmetrical, the tip can be extracted at the end using tip extraction module. The following section will elaborate each of the module and functionality.

4.2.1 Kalman filter system modeling

In order to reconstruct the tool configuration in 3D, the edges and tip position has to be calculated at each frame. The simplest approach is to extract edges and tip using Canny edge detection and corner detection for each frame individually. However, this approach is not robust in a variant environment where light reflection and partial object occlusion exists. Moreover, edge detection will fail when more than one tool appears within the scene having multiple edges that cannot be labelled. As an alternative, Kalman filter can be used to address the tracking problem assuming the tool is moving smoothly and slowly. This top-down tracking approach is able to estimate and track the current frames based on the previous one, even if the tool is partially occluded or disappears in a short period of time. The reason we choose Kalman filter over Particle filter is its simplicity and less computation expense.

The Kalman filter has long been regarded as the optimal solution to many tracking and
data prediction tasks. The purpose of filtering is to extract the required information from a signal. It is recursive so that new measurements can be processed as they arrive.

The dynamical system is represented using a state-space model where the unknown state is predicted based on the system model or estimated based on the observation/measurements. Linear Kalman filters are based on linear dynamic systems discretized in the time domain. The Kalman filter addresses the problem of trying to estimate the state \( x \in \mathbb{R}^n \) of a discrete-time controlled process that is governed by the linear stochastic difference equation:

\[
x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}
\]

with a measurement \( z \in \mathbb{R}^m \) that is

\[
z_k = Hx_k + v_k
\]

The random variables \( w_k \) and \( v_k \) represent the process and measurement noise, respectively. They are assumed to be independent (of each other), white, and with normal probability distributions:

\[
p(w) \sim N(0, Q), \ p(v) \sim N(0, R)
\]

We assume that the process noise covariance \( Q \) and measurement noise covariance \( R \) matrices are constant here.

The \( n \times n \) matrix \( A \) in the difference equation 4.1 relates the state at the previous time step to the state \( k - 1 \) at the current step \( k \), in the absence of either a driving function or process noise. The \( n \times l \) matrix \( B \) relates the optional control input \( u \in \mathbb{R}^l \) to the state \( x \). The \( m \times n \) matrix \( H \) in the measurement equation 4.2 relates the state to the measurement \( z_k \).

In current tracking system, the Kalman state is modeled as the position and velocity of the middle line of the tool. The position of the middle line (Green line in Figure 4.2) is parametrized as the slope \( (m) \) and axis-intercept \( (b) \) of the projected line on the image plane. The velocity is modelled as the first derivatives of the slope and \( y \)-intercept, respectively. Thus, the state \( x \) in frame \( k \) is modelled as:

\[
x_k = [m, b, m', b']^T
\]

Where the middle line is represented as:
The time update equations can also be thought of as a predictor equation, which in this case, is the step to predict the location of the middle line for the current frame. The current frame $k$ can be predicted from frame $k-1$ as:

$$
\begin{bmatrix}
  m_k \\
  b_k \\
  m'_k \\
  b'_k
\end{bmatrix} =
\begin{bmatrix}
  1 & 0 & 1 & 0 \\
  0 & 1 & 0 & 1 \\
  0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  m_{k-1} \\
  b_{k-1} \\
  m'_{k-1} \\
  b'_{k-1}
\end{bmatrix} + w_{k-1}
$$

(4.6)

where $w_{k-1}$ represents the process noise coming from the sudden tool motion, modelling error, etc. $(m_k, b_k)$ represents the middle line of the tool in the current frame, same as $(m_{k-1}, b_{k-1})$ for previous frame, $(m'_k, b'_k)$ and $(m'_{k-1}, b'_{k-1})$ denotes the first derivative of the slope-intercept parameters. The above equation implies that the derivative of the parameter is kept constant through the transition; in another words, the position parameter is linearly increasing or decreasing. The main assumption used in the above equation is that the impending velocity of tools can be considered slow in terms of its transition.

From the above equation, we extract the transition matrix model as:
CHAPTER 4. A NEW FEATURE-BASED TRACKING ALGORITHM

\[ A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \] (4.7)

Another part of the Kalman filter mechanism is to define a measure for the system observation in order to correct the system model. In our case, the measurement is the location of the middle line:

\[ \text{Meas} = [m, b]^T \] (4.8)

Since, the measurement noise \( v_k \) is applied between our observation and ground truth, the measurement equation is given as:

\[
\begin{bmatrix} m_{\text{meas}_k} \\ b_{\text{meas}_k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} m_k \\ b_k \\ m'_k \\ b'_k \end{bmatrix} + v_k \] (4.9)

Thus, the measurement matrix \( H \) is modeled as:

\[ H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \] (4.10)

The system and measurement matrices \( H \) and \( A \) need to be initialized beforehand, the same as the noise model, including process noise \( (w_k) \) and measurement noise \( (v_k) \). The noise variance is defined as:

\[ Q = \begin{bmatrix} 80 & 0 & 0 & 0 \\ 0 & 80 & 0 & 0 \\ 0 & 0 & 80 & 0 \\ 0 & 0 & 0 & 80 \end{bmatrix} \] (4.11)

\[ R = \begin{bmatrix} 0.005 & 0.005 \\ 0.005 & 0.005 \end{bmatrix} \] (4.12)

If the measurement noise covariance \( R \) is small, the final calculated state of the middle line position is nearly the same as the measurement obtained from image processing.
Therefore, the $R$ is defined small in this experiment assuming small noise comes from measurement. Since we do not have the ability to directly observe the process, the determination of the process noise covariance $Q$ becomes very difficult. In this experiment, big process noise covariance is injected to tolerate enough uncertainty throughout the process. By empirically testing different $Q$ and $R$ off-line, the estimation error covariance $P_k$ and the Kalman gain $K_k$ stabilizes quickly and remains constant (see the filter update equations in 4.19, 4.20 and 4.21). Through the experiment, 3-6 iterations later, the error covariance remain in steady state, providing reliable selection of the noise covariance parameter.

### 4.2.2 Initialization and preprocessing

The first problem is to detect and localize the middle line of the tool for the first frame, parameterized as $m$ and $b$. Our initialization method is semi-automatic, which requires some amount of user cooperation. The user is required to place the surgical tool inside the camera view, ready for initialization. The system observes the scene and detects the sharp edges (the edges with maximum peaks in the Hough transformation). Since the edges detected are edge segments and are not necessarily only two edges, the system needs to be able to merge the edges to either the left or right part of the tool boundaries. The algorithm is summarized as follows:

1. Grab and load an image that was captured by frame grabber (Figure 4.4(a)).

2. Convert the colored image to be gray image so that the following canny detector can be applied (Figure 4.4(b)).

3. Use canny detector to detect the edges of the image. Small segments of canny output are directly discarded (Figure 4.4(c)).

4. Use Hough transform to detect the straight lines appearing in the images. Several line segments are detected at the same time (Figure 4.4(d)).

5. Label the edge segments. Since the tool has two edges as a whole, each segment of the lines has to be labelled such that it belongs to either side of the edges. In order to label these lines, two buckets representing the two edges are initialized to store the lines segments. The algorithm is performed as follows: The first line detected belongs to the first bucket. Then the algorithm compares the second line with the
first line, if the maximum distance between the end points of the second line to the first line (Figure 4.3) is larger than a threshold (in this case 10 pixels), it is labelled to be another bucket, namely, another edge; otherwise, it belongs to the first bucket. Similarly, comparison is performed between the rest of the segments. Finally, each of the buckets should store the corresponding line segments representing either side of the edges. To decide the distance between a random point in 2D space \( P(x_0, y_0) \) and a line \( \vec{L}(ax + by + c = 0) \), the following equation is used:

\[
Dist(P, \vec{L}) = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}
\]  

(4.13)

Figure 4.3: The illustration of the distance calculation between two edges. If the maximum distance of end points of one line to the other line is less than the threshold \( D < Th \), it is assumed to be in the same bucket, otherwise, it belongs to the other bucket.

The following is the pseudo code of the classifier:

\begin{verbatim}
Algorithm 1 Classifier (lines[ ])
1: bucket1.insert(lines[1])
2: for i = 2 to n do
3:   curline ← lines[i]
4:   if isClose(curline, bucket1[1]) then
5:     bucket1.insert(curline)
6:   else
7:     bucket2.insert(curline)
8: end if
9: end for
\end{verbatim}
CHAPTER 4. A NEW FEATURE-BASED TRACKING ALGORITHM

Note that \( isClose() \) function is to determine if the distance between two lines is within the threshold. The distance is calculated from equation 4.13.

6. Once the edges have been classified into two buckets, joining the line segments becomes the next step. For each bucket or edge, the algorithm is to join the leftmost point with the rightmost point together, forming an single edge. Figure 4.4(f) shows the result of this method: the leftmost point (green) and rightmost point (red) is joined forming a red line.

Up till now, the two edges of the tool are extracted, denoted as \( \vec{E}_1(m_1, b_1) \) and \( \vec{E}_2(m_2, b_2) \). The middle line of the two edges then becomes:

\[
\vec{E}(m, b) = \left( \frac{m_1 + m_2}{2}, \frac{b_1 + b_2}{2} \right)
\]

(4.14)

Figure 4.4(f) shows the final result of the initialization, where the green line is the extracted middle line of the tool.

7. The program runs the initialization until the user presses “Enter” key to start tracking mode.

In this section, the semi-automatic initialization is accomplished by performing edge detection and human interruption. It is performed at the beginning of the tracking, as Kalman initialization. The next step is to start tracking, described in the following section.

4.2.3 Tool tracking using Kalman Filter

After initialization phase, the system initiates tracking of the tool in real-time until the program exits. The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state and then obtains feedback in the form of (noisy) measurements. As such, the equations for the Kalman filter fall into two groups: time update equations and measurement update equations. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the \( a \) priori estimates for the next time step. The measurement update equations are responsible for the feedback - i.e. for incorporating a new measurement into the \( a \) priori estimate to obtain an improved \( a \) posteriori estimate. The time update equations can also be regarded as predictor equations, while the measurement update equations can be thought of as corrector
CHAPTER 4. A NEW FEATURE-BASED TRACKING ALGORITHM

Figure 4.4: (a) Original input frame (b) Converted gray image (c) Output of canny’s edge detector used for extracting edges of the tool. (d) Output of edge image detected using canny detector (e) Classify the edge segments into “left” and “right” edge. The yellow segments are belonging to “left” edge, and red one are “right” edge. (f) Merge edge segments together, forming two edges, “left” edge and “right” edge. The green line is the calculated middle line of the two edges.
CHAPTER 4. A NEW FEATURE-BASED TRACKING ALGORITHM

equations. The final estimation algorithm resembles that of a predictor-corrector algorithm for solving numerical problems as shown in Figure 4.5.

\[ K_k = P_k H^T (HP_k H^T + R)^{-1} \]

\[ \hat{x}_k = \hat{x}_k + K_k (z_k - H\hat{x}_k) \]

\[ x_k = Ax_{k-1} \]

\[ P_k = AP_{k-1}A^T + Q \]

Figure 4.5: Kalman Filter recursive algorithm

The first task during the measurement update is to compute the Kalman gain, \( K_k \). The next step is to actually measure the process to obtain \( z_k \), and then to generate an a posteriori state estimate by incorporating the measurement as in Figure 4.5. The final step is to obtain an a posteriori error covariance estimate. After each time and measurement update pair, the process is repeated with the previous a posteriori estimates used to project or predict the new a priori estimates. Figure 4.5 presents a conceptual operation of the filter.

Based on the system modelling, the equations for the prediction phase are as follows, with priori state estimate \( x_k \) and its error covariance \( P_k \) as:

\[ x_k = Ax_{k-1} \] (4.15)

\[ P_k = AP_{k-1}A^T + Q \] (4.16)

After the initial prediction, through the proposed image processing, the location of the middle line is measured before the correction phase. The algorithm is the same as the edge detection algorithm in the initialization phase except that it takes advantage of the predicted state to filter out the edges that are unlikely to belong to the tool. Even though there are two tools or similar objects within the screen, the edges that belong to other objects won’t
be detected. The filter uses the simple comparison between the edge and the middle line. If
the difference in the slopes between the measured edge and the predicated middle line is
smaller than a threshold (Equation 4.17) and its edge point is close enough to the middle
line (Equation 4.18), we assume this edge belongs to the tool; otherwise, it is filtered:

\[
|m_{\text{middleline}} - m_{\text{edge}}| < \epsilon \tag{4.17}
\]

\[
|d_{\text{max}}| < \epsilon \tag{4.18}
\]

\(m_{\text{middleline}}\) is the slope of the middle line and the \(m_{\text{edge}}\) is the slope of the edge. \(d_{\text{max}}\) is
the maximum distance among the edge points and the middle line.

Once the edges are filtered, we classify the remaining lines to be either left or right edge
and convert them again to the slope-intercept form \(m\) and \(b\) (detailed algorithm please refer
to section 4.2.2). These two parameters provide the measurement output for the Kalman
filter. After finding out the measurement \(Meas = [m, b]^T\), the state model is corrected as
follows:

\[
K_k = P_k^{-1} H^T (HP_k^{-1}H^T + R)^{-1} \tag{4.19}
\]

\[
x_k = x_k^- + K_k(z_k - Hx_k^-) \tag{4.20}
\]

\[
P_k = (I - K_kH)P_k^- \tag{4.21}
\]

Finally, from the above steps, we are able to extract the edges from the 2D image.

### 4.2.4 Tip extraction

Once the two edges have been extracted, the only information that is left to be determined
is the tip of the tool. In order to localize the tip, we first get the 1D profile of the image
along the middle line for each tool. By performing 1D edge detection on the image profile,
the tip, which corresponds to the largest change of the data, can be extracted.

Theoretically, the tip is localized where there is a sudden change of the intensity since
the tool is uniformly black while the background is brighter. However, when light changes
and reflection occurs on the tool, the algorithm might mistakenly interpret the tip to be the
CHAPTER 4. A NEW FEATURE-BASED TRACKING ALGORITHM

reflection spot instead of the expected location. To prevent such situation, which occurs quite often in the real surgical operation, we designed filters to avoid such search areas.

There are three filters added to minimize the noise. The first one is to extract the HSV (Hue, Saturation, Value) of the pixel; when the $V$ value is larger than certain threshold, which means that the location has high possibility of being light reflection, the algorithm eliminates this possible location and starts to search the next possible pixel on the middle line. As shown in Figure 4.6, the “Light reflection area” will be avoided by applying this filter. The second filter is used to decide whether the extracted possible tip location is valid given the edge orientation. In another word, the tip is always along the side where the edges are intersecting. When the tip is within the location of the edges or on the wrong side, the tip must be invalid. Finally, the last filter is designed to ensure that the tip location won’t be miss-detected to be the border of the endoscope since the sharp intensity change occurs between the background and the circle borderer of the camera. In Figure 4.6, the arrow indicating the “Possible search area” shows the effect by applying the last two filters. The green circle is the border of the image, and the possible search area starts from the end point of the edges (whichever is further away along the middle line) and ends at the border. By applying the second and third filter, the possible search area minimize to be within an edge point and the borderer. In this way, we are able to minimize the noise and side effect in a light variant environment.

![Figure 4.6: Tip extraction search area after applying filters.](image-url)
The following is the pseudo code for the tip extraction:

**Algorithm 2** Tip Localization \((E, P)\)

1. if \(\text{ABS}(E_m) \geq 0 \& \text{ABS}(E_m) \leq 1\) then
2. \(tip \leftarrow \text{FindTipX}(E, P)\)
3. else
4. \(tip \leftarrow \text{FindTipY}(E, P)\)
5. end if

**Algorithm 3** FindTipX \((E)\)

1. for \(x_i \leftarrow 1\) to \(\text{SizeX}\) do
2. \(y_i \leftarrow E_m x_i + E_b\)
3. if \(y_i \geq 0 \& y_i < \text{SizeY}\) then
4. \(\text{profile}[x_i] \leftarrow \text{frame}[x_i][y_i]\)
5. else
6. \(\text{profile}[x_i] \leftarrow 0\)
7. end if
8. end for
9. \(\text{diffMax} \leftarrow \text{ABS}(\text{profile}[2] - \text{profile}[1])\)
10. \(\text{tipLoc} \leftarrow 0\)
11. for \(x_i \leftarrow 1\) to \(\text{SizeX} - 1\) do
12. \(y_i \leftarrow E_m x_i + E_b\)
13. \(\text{diff} \leftarrow \text{ABS}(\text{profile}[i + 1] - \text{profile}[i])\)
14. if \(\text{diffMax} < \text{diff} \& \text{isReflectance}([x_i, y_i]) \& \text{isValidGeoX}([x_i, y_i], E) \& \text{isInsider}([x_i, y_i])\) then
15. \(\text{diffMax} \leftarrow \text{diff}\)
16. \(\text{tipLoc} \leftarrow i\)
17. end if
18. end for

Notice from the above pseudo code, to search for the tip location, we can search from \(x\) direction, which is to calculate the intensity difference from the leftmost pixel to the rightmost. However, when the tool is almost vertical in the scene, this approach is very inaccurate since the searchable area for tip location suddenly becomes very small; in which case, a \(y\) search is applied. To make the algorithm work better, we split the two situations depending on its slope or orientation (Figure 4.7). Therefore, no matter how the tool is oriented, the tip localizer is always able to search the valid tip location. Figure 4.8 shows two screenshots from the image frames. One of the tools is oriented horizontally, another one is vertically inserted, both of which are detected successfully.
CHAPTER 4. A NEW FEATURE-BASED TRACKING ALGORITHM

Figure 4.7: Tip extraction search method.

Figure 4.8: Tip tracking result. (a) Tip search in $x$ direction (b) Tip search in $y$ direction
Up till now, all the information that is needed for the following 3D reconstruction has been extracted. The tip and edges are detected and extracted for each frame. The calculated edges are involved in the Kalman filter loop as the “measurement value”, along with the tip, which are the three necessary parameters for the 3D reconstruction. This module serves as the input for the following 3D reconstruction module, which is used to obtain the final 3D pose of the instrument.

4.2.5 Two dimensional tracking of multiple tools

Tracking multiple tools is important in order to build a more complicated and functional interface. Two tools tracking, for example, can be utilized similar to two hands. By manipulating two tools, the users are able to grab, rotate and translate an overlaid image/3D object by moving the two “hands”. A suitable tracking system has to be adopted in order to support multiple tool tracking. Mandana[21] was able to track two tools using blob tracking of different colors. However, the approach is not robust when the tip is occluded. Additionally, because of lack of ability to reconstruct the tools orientation, it cannot provide a sophisticated interface. Motion history image approach (discussed in Chapter 3) can track multiple tools, but when the tools are crossing together, the detection fails.

The proposed algorithm supports multiple surgical tool tracking based on Kalman filtering procedure. The edge that doesn’t belong to the tool is filtered during image processing, and as a result, the multiple tracking can be applied individually without confusion.

To support multiple tool tracking, the algorithm is going to be enhanced to adapt to the second tool tracking. There are two minor modifications based on the previous program. First, a second Kalman filter is added to the program. For simplicity, the program is initialized with a predefined \( m \) and \( b \) value. Figure 4.9(a) shows the initialization procedure by placing the tool around the two green lines. The two green lines \((E_1(m_1, b_1), E_2(m_2, b_2))\) are initialized as \([-1.0; 550.0]\) and \([1.0; -50.0]\). The second modification is that the tip extraction procedure adds one more filter. It’s purpose is to remove the intersecting region between two tools’ middle lines (as shown in Figure 4.9(b)), where the intensity changes occur.
CHAPTER 4. A NEW FEATURE-BASED TRACKING ALGORITHM

Figure 4.9: Two modifications of the tracking algorithm to adapt multiple tracking. (a) The initialization is achieved by placing the tool around the two green lines. The two green lines ($E_1(m_1, b_1)$, $E_2(m_2, b_2)$) are initialized as $[-1.0; 550.0]$ and $[1.0; -50.0]$. (b) The tip search area is being modified so that the area between one tool’s middle line intersecting with another tool’s two edges is removed. The blue regions represent the valid search area for each of the tool.

4.3 Three dimensional tracking of surgical tools

This section proposes an approach for obtaining depth and relative position information between instruments. The proposed image analysis extracts information from the projective geometry of the surgical setting in the image to determine laparoscopic tool’s pose. The cylindrical geometry of the instruments and their projective contours in the image are utilized to estimate the tool’s orientation and spatial position.

4.3.1 3D pose estimation using tip and edges

This module utilizes the information obtained from the previous module (2D edges and tip) to reconstruct the 3D pose of the tool with respect to the endoscope. It takes advantage of the geometrical properties of the rigid cylindrical-shaped instrument to calculate the 3D tip location as well as the orientation (or 3D vector) of the tool axis. Another prior knowledge is that the camera is fully calibrated and the physical diameter ($d_{physical}$) of the tool is given.

Due to the projective geometry of the laparoscopic scene[32], the cylindrical-shaped tool is visualized as a trapezoid in the image. To reconstruct the 3D pose of the tool, the
following equations are derived using the projective geometry[42] (Figure 4.10). E1 and E2 are two edges of the surgical tool projected on the image plane. $T$ is the tip of the tool in the 2D image, with 3D coordinate as $P$. The camera center is denoted as $C$. $\theta$ is the angle between the physical tool with respect to the image plane.

We defined two tangential planes $\Omega_1$ and $\Omega_2$ that include one of the edges on the tool and the camera vector to the edge. The normal vector $(u_{\Omega_1}, u_{\Omega_2})$ to each tangential plane is obtained through cross product between the edges $(\vec{u}_{E_1}, \vec{u}_{E_2})$ and the camera vector $(\vec{u}_{CE_1}, \vec{u}_{CE_2})$ to the edge.

$$\vec{u}_{\Omega_i} = \vec{u}_{E_i} \times \vec{u}_{CE_i}, \quad i = 1, 2 \quad (4.22)$$

As shown in Figure 4.10(b), the angle $\lambda_N$ between one of the symmetrical plane to the $CN$ vector can be expressed by addition of the normal vector of $(u_{\Omega_1}$ and $u_{\Omega_2}$):

$$\tan \lambda_N = \frac{|\vec{u}_{\Omega_1} + \vec{u}_{\Omega_2}|}{|\vec{u}_{\Omega_1} - \vec{u}_{\Omega_2}|} \quad (4.23)$$

To determine the length of the $CN$ vector (Figure 4.10(b)), namely, the distance from the camera to the tool axis, we can apply the prior knowledge of the tool radius as:

$$|CN| = \frac{d_{\text{physical}} \cdot 0.5}{\sin \lambda_N} \quad (4.24)$$
The orientation of the $CN$ vector can be simply derived as:

$$\vec{u}_{CN} = |CN| \cdot \frac{\vec{u}_{\Omega 1} + \vec{u}_{\Omega 2}}{|\vec{u}_{\Omega 1} + \vec{u}_{\Omega 2}|}$$

Finally, the $CP$ vector is determined as:

$$\vec{CP} = \frac{|CN| \cdot \vec{u}_{CT}}{\vec{u}_{CT} \cdot \vec{u}_{CN}}$$

Note that $P$ can be any point on the tool, in this case, the tip of the tool. Therefore, $\vec{CP}$ represents the vector from the camera origin $C$ to the tip, meaning that the 3D location of the tool is extracted. Moreover, the $NP$ vector which represents the orientation of the tool can be calculated as follows:

$$\vec{NP} = -\vec{CN} + \vec{CP}$$

As $\vec{NP}$ is known, any other point of the tool can be obtained by knowing the physical distance between $P$ and the desired point. Figure 4.11 shows the 2D edge and tip information along with the reconstructed 3D pose rendered by OpenGL. Up to this point, all the information needed are extracted. The only degree of freedom we cannot detect is the rotation of the instrument around its axis (roll). Other than this, the left-right and forward-backward rotations of the instrument around the incision point (yaw and pitch) can be extracted.

Figure 4.11: (a) Image with detected 2D edges and tip (b) Reconstructed tool rendered in OpenGL
4.4 Experimental results and discussion for the new proposed feature-based approach

This section presents the 2D and 3D result of the current feature-based algorithm introduced in section 4.2 to 4.3. The 2D algorithm takes the advantage of the Kalman filtering to filter the invalid edges. The 3D algorithm uses the detected edges and tip to reconstruct the tool position and orientation. The multiple tool tracking result is also presented below to show the feasibility of this approach.

4.4.1 Single tool tracking result

This subsection presents the tracking results for a single tool in the scene. The Kalman filter implementation allows the correct recognition of edges even though there are similar objects in the scene (Figure 4.12(a), 4.12(b)). Without the filtering process, in other words, detecting the edges for each frame individually, the detection will certainly fail. Another advantage of the algorithm is that when the tool is partially occluded (Figure 4.12(c)), the algorithm is able to recover the edges. The final advantage of the system is that it can handle certain amount of light reflection. Due to the filters added in the tip extraction algorithm, the light is mostly excluded, allowing successful recognition of the tip. Figure 4.12(f) shows the algorithm is robust against the confusion caused by the bright light occurring on the metal part of the tool.

To test the algorithm’s ability of handling the partial occlusion, similar objects and light reflection, we categorize each of the situations and explain the degenerated case for the detection.

When the light is weak, the tool cannot be recognized properly since there is not enough contrast with respect to the background. Moreover, the shadow of the surgical tools will blur the sharp edge of the tool, making the edge detection fail as shown in Figure 4.13(a) to 4.13(c).

The partial occlusion occurs occasionally in real surgical circumstance, such as tool occluded by organs and tissues. There are successful cases for handling the occlusion. However, when the tool’s tip is occluded, the algorithm performs poorly, as shown in Figure 4.13(d).

When the tool moves fast enough (≥ 8cm/s), the movement causes blurring of the images, thus, the algorithm failed to detect the edges (as shown in Figure 4.13(e)). In
Figure 4.12: Screenshots of the successful tracking results. (a)-(b) Current algorithm can handle similar objects in the scene by implementing Kalman filtering. (c) The edges belonging to the tool can be recovered even though part of the tool is being occluded. (d)-(e) The edge detection can handle two similar tools having included angle larger than 20-30 degrees. (f) Current tip extraction module can handle some amount of light reflection caused by special material.
Figure 4.13: Screenshots of the failed tracking results. (a) Current algorithm failed to detect edges when the light is weak, (under 30/100 in this case). (b) The shadow blurs the image, causing missed detection of the edge. (c) The tip extraction module failed to detect correct tip due to less contrast of the tip with respect to the background. (d) When the tip was occluded, the detection fails. (e) Current algorithm cannot handle fast motion of the tool. (f) The edge detection cannot handle two similar tools having included angle less than 20-30 degrees.
general cases, however, the algorithm can be robust enough to capture the images when they move moderately fast.

The detection also fails when a similar object approaches the tracking target and is oriented the same way or the opposite way, with included angle less than 20-30 degrees. However, this will not happen in the real surgery, where the tools are generally farther apart separated by incision points. Figure 4.12(d) and 4.12(e) are two successful tracking results with two objects having most tolerable angles. However, when the two objects are getting closer, as shown in Figure 4.13(f), the tracking is lost. The algorithm mistakenly recognizes the edge belonging to another tool, causing incorrect edge and tip extraction.

In order to test the robustness and correctness of the current algorithm, disturbances resulting from tissue occlusion, the physical movement of the tool and light variance are taken into consideration in the following experiments. Results of the image analysis have been grouped with respect to the input factors, as shown in Table 4.1. It has been assumed that the tool’s tip is found correctly if the position error is not greater than twice of the width of the tool and the tool’s orientation is found within 20 degrees of error range. The number of frames in the experiment is 50 frames per sub-test tracking under the corresponding input environments.

Table 4.1: Experimental results of current feature-based algorithm with different input factors

<table>
<thead>
<tr>
<th>Input factors</th>
<th>Range</th>
<th>Successful rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool Speed(cm/s)</td>
<td>[0, 4]</td>
<td>82%</td>
</tr>
<tr>
<td></td>
<td>(4, 8]</td>
<td>54%</td>
</tr>
<tr>
<td></td>
<td>[9, ∞)</td>
<td>32%</td>
</tr>
<tr>
<td>Light (Brightness rate)</td>
<td>[0, 30]</td>
<td>22%</td>
</tr>
<tr>
<td></td>
<td>(30, 45]</td>
<td>52%</td>
</tr>
<tr>
<td></td>
<td>(46, 100]</td>
<td>89%</td>
</tr>
<tr>
<td>Distraction from similar objects</td>
<td></td>
<td>95%</td>
</tr>
</tbody>
</table>

The results show that the current algorithm can handle some amount of noise and can be adopted as a moderately reliable tracking subsystem especially when the condition is perfect, with tool speed as [0, 4], brightness rate to be (46, 100] and with no similar objects within the scene.
4.4.2 Multiple tracking result

Figure 4.14 shows the screenshots of successful multiple tracking results. The algorithm works well even though one tool is on top of the other tool. Though some part of the tool is excluded, the edge detection procedure is able to recover the edges based on the assumption that the tool is cylindrical shaped.

![Figure 4.14: The multiple tracking result with successful cases. The green lines are the tool’s middle line, the red are their edges and the white spots are the detected tips. (a) Condition when two tools are separate to each other. (b) Condition when tools’ intersection point is on one of the tool’s middle line (c) Condition when two tools are crossing. (d) Condition when tools’ included angle is 30 degree.](image)

However, one issue of this algorithm is that the tip extraction fails if it is excluded or it is located beneath or on top the other tool (as shown in Figure 4.15(a)). Another problem is that when the tools’ included angle is small, the filtering failed to categorize the edges and the detection fails (as shown in Figure 4.15(b)).


4.4.3 Three dimensional tool tracking result

We evaluate the 3D reconstruction result by using the known inclined board (Figure 4.16).

The 3D points are pre-defined along the board as the ground truth. Two sets, with ground truth of 30 degree and 60 degree are tested in this experiment. Each set has 100 frames taken from the perfect environment as discussed above. A total of 200 frames are evaluated and compared with the ground truth. Figure 4.17 shows part of the result generated from above sequence.

The video has been analyzed to calculate the mean and standard derivation. The result
is shown in Table 4.2; the error is about 5 degrees, which is within expected range. The result shows the relatively high accuracy of the above algorithm with only monocular view. Theta is the angle between the tools with respect to the horizontal plane (parallel to camera image plane, shown in Figure 4.10(a)). The distance represents the distance between camera and the tool vector.

<table>
<thead>
<tr>
<th>Feature-based tracking</th>
<th>Mean Error</th>
<th>Std. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theta θ (Degree)</td>
<td>4.0</td>
<td>5.8</td>
</tr>
<tr>
<td>Distance (Camera to tool) (mm)</td>
<td>3.8</td>
<td>8.2</td>
</tr>
</tbody>
</table>

The above tests are all based on the assumptions that the tip and orientation are correctly detected; improper detection or missing detection in the real time sequence are taken out. The following section will look into detail of more experimental results and comparison studies with the other methods (MHI and PF) presented in Chapter 3.

### 4.5 Comparison studies between MHI, PF and new proposed feature-based approach

This section presents the comparison result of the four possible methods for localization of the surgical instruments: motion history images (MHI), particle filter (PF) tracking,
CHAPTER 4. A NEW FEATURE-BASED TRACKING ALGORITHM

color-based tracking (results shown in [21]) developed previously in “Experimental Robotics and Imaging Laboratory” and the current feature-based method. The MHI provides the motion analysis and PF offers robust color tracking limited in 2D space. The previous color-based blob tracking[21] offers the quickest and simplest way for multiple tracking, however, it cannot capture the orientation of the tool. The following table 4.3 is the functional comparison between the different tracking methods.

Table 4.3: Function comparison between color-based method, Motion history image method (MHI), Particle filtering method (PF) and Current feature-based algorithm

<table>
<thead>
<tr>
<th>Function(Y/N)</th>
<th>Color-based tracking</th>
<th>MHI</th>
<th>PF</th>
<th>Feature-based tracking</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D tip location</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>2D edge extraction</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>3D tip location</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>3D tool orientation</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Multiple tools tracking</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

To validate the current algorithm and compare with the previously proposed methods, several experiments have been conducted to estimate the accuracy error of estimating the tip or orientation of the tool.

The first experiment is conducted by measuring the 2D performance of the algorithm, which is to measure the accuracy of the $x$ and $y$ coordinate of the tip location. The ground truth is recorded by manually selecting the tip from 100 real-time frames taken inside the plastic stomach model. As shown in Table 4.4, the feature based blob tracking has better performance than the other three methods. The reason is that the color-based tracking and PF highly depend on the light condition. The light reflection and lower intensity will cause failure of the detection. Thus, the error is larger than the feature-based algorithm. As for MHI tracking, the ROI bounding box is larger than the actual tool location when the tool is moving fast. The tip, which is estimated based on ROI, is inaccurate as the result.

To further validate the above algorithms, the 3D localization result is compared in the physical space. In this experiment, MHI and PF are not tested because of their limited tracking in 2D space. Five trials are experimented with the ground truth to be 150mm, 130mm, 110mm, 90mm and 70mm (using ruler as the reference). Each test has 80 frames of data taken from real-time images under plastic stomach model. As shown in Table 4.5, the feature-based motion tracking outperforms the color-based tracking especially in $z$
Table 4.4: 2D tracking comparison accuracy (pixel) between color-based method, Motion history image method (MHI), Particle filtering method (PF) and Current feature-based algorithm

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean(x axis)</th>
<th>Std.(x axis)</th>
<th>Mean(y axis)</th>
<th>Std.(y axis)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color-based blob tracking</td>
<td>5.9</td>
<td>4.9</td>
<td>3.3</td>
<td>3.1</td>
</tr>
<tr>
<td>MHI</td>
<td>5.4</td>
<td>3.3</td>
<td>5.2</td>
<td>3.1</td>
</tr>
<tr>
<td>PF</td>
<td>2.3</td>
<td>1.8</td>
<td>1.9</td>
<td>1.8</td>
</tr>
<tr>
<td>Feature-based tracking</td>
<td>2.0</td>
<td>2.4</td>
<td>1.9</td>
<td>3.5</td>
</tr>
</tbody>
</table>

As for the color-based tracking, the z coordinate is measured using projective model and Pixel-to-Millimeter conversion look-up table, which makes the measurement fairly inaccurate. The current algorithm, on the other hand, uses the projective geometry to reconstruct the 3D pose of the tool, which limits the error within expected range.

Table 4.5: 3D tip tracking comparison accuracy (mm) between the color-based method and the current feature-based algorithm

<table>
<thead>
<tr>
<th>Tip coordinate error (mm)</th>
<th>x Mean(Std.)</th>
<th>y Mean(Std.)</th>
<th>z Mean(Std.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color-based blob tracking</td>
<td>1.3(1.0)</td>
<td>0.7(0.6)</td>
<td>10.2(7.7)</td>
</tr>
<tr>
<td>Feature-based tracking</td>
<td>0.4(0.7)</td>
<td>0.4(0.7)</td>
<td>3.2(6.3)</td>
</tr>
</tbody>
</table>

In addition to the accuracy and robustness test, we experimented with the computing time for different approaches under the same workstation and set-up. The hardware setup please refer to Chapter 2: Phantom Experimental Setup. Table 4.6 shows the computational result.

Table 4.6: Comparison of computational time between color-based method, Motion history image method (MHI), Particle filtering method (PF) and Current feature-based algorithm

<table>
<thead>
<tr>
<th>Method</th>
<th>Frame rate (fps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color-based blob tracking</td>
<td>22</td>
</tr>
<tr>
<td>MHI</td>
<td>25</td>
</tr>
<tr>
<td>PF</td>
<td>8</td>
</tr>
<tr>
<td>Feature-based tracking</td>
<td>13</td>
</tr>
</tbody>
</table>

This table indicates that, comparing with the color-based blob tracking and MHI, the current algorithm is more computationally expensive. However, in a 3D mouse paradigm, 13 fps is reasonable for delivering a comfortable real-time tracking framework.
4.6 Discussion

This chapter presents the new tracking framework based on Kalman filter and feature-based detection. The combination of these top-down and bottom-up approaches provides a robust system with multiple tracking support. This system is superior to color-based tracking, previously proposed by Mandana [21]. The advantages are the following: 1) The current system can handle large light variance and partial occlusion, due to its edge-based detection. Even some pixels on the edge cannot be detected, the remaining pixels are able to reconstruct the edges. In color-marker-tracking, on the other hand, the detection fails if the tool tips get occluded. Moreover, the blob tracking is inaccurate in a light variant environment. If some of the pixels are not detected, the calculated center of the blob shifts to unexpected location. 2) The Kalman filter implementation allows successful detection even when several tool are cluttered in the environment. It enables tracking of a defined tool or label the tools in the multiple tools application. However, color-based tracking fails if an object with similar color is present in the scene. 3) The proposed algorithm is able to calculate the 3D orientation of the tool, providing additional information over the color-based tracking. 4) The proposed tracking approach does not need any markers on the tool. No modification on the instrument manufacturing process is needed.

However, the disadvantage of the current algorithm is that it is slower than the MHI and color-based tracking. And tracking cannot be processed if the tools are moving fast. Moreover, the tracking fails when the tools are oriented in the same way. However, in surgical applications, the degenerated cases illustrated above generally do not occur.

To summarize, the result of the proposed system is promising in terms of robustness and accuracy. It is feasible to be applied as a base framework in a functional and relatively complicated surgeon-computer interface presented in the next chapter.
Chapter 5

3D mouse interactive paradigm

This chapter presents the preliminary 3D interactive system based on the proposed tracking procedure. Section 5.1 gives an overview of the current 3D mouse interactive system. Section 5.2 presents the SCI design with details including menu design and mouse interactive design. The implementation of the system is introduced in Section 5.3, covering the procedures of image format conversion, augmented reality alignment, virtual object rendering and manipulation. The performance of the proposed 3D-mouse system, the augmentation tests, and the discussion are presented in Section 5.4.

5.1 3D-mouse system overview and its application

Information technology has dramatically changed medical practice in the past three decades, particularly in the areas of patient record management and preoperative planning. In general, a typical operating theater is crowded with surgeons, assistants, anesthesiologists, and supporting nurses. In addition, there are surgical equipment and tools and tables which require an efficient workflow and use of space mandatory. In the operating room (OR), however, computers tend to be used sparingly. Although there are numerous reasons for this (equipment bulkiness/clutter, software reliability, etc.), a primary factor is the lack of convenient and well-designed surgeon-computer interaction.

Here we propose a simple SCI for allowing the surgeon to access the medical information and pre-operative images during the surgery by using the existing surgical tools operated by the surgeons. In this way, our proposed system does not require any extra pieces of hardware and uses the existing imaging system to provide this novel surgeon-computer interface.
Instead of using stereoscopic head-tracked displays, we use mono-camera and augmented reality (AR) to allow surgeons to interact with virtual reality. AR combines computer graphics with images of the real world. This can be accomplished through the use of video mixing, 3D laparoscopic reconstruction, accurate motion tracking, and creation of live images that combine computer-generated graphics with the surgeons’ live view of a patient. We believe that the use of AR and 3D mouse paradigm can significantly simplify both learning and performing of minimally invasive interventions in image guided navigation and surgery. Our initial prototype and phantom experimental evaluation have shown promising results that also require further refinement for realizing further clinical evaluation.

Our current system setup is shown in Figure 5.1. Color images (320*240, 24-bit) are acquired from the endoscopic camera. After image acquisition, the tracking subsystem (details please see Chapter 4) is implemented in three steps: edge extraction, tip localization, and 3D pose reconstruction. We use a combination of feature-based recognition and Kalman filtering process to achieve a robust, reliable instrument tracking. However, if the tool is lost during the tracking, the program recognizes the miss-detection and displays an flag on the overlaid interface. In case the tracking fails, the user need to exit the program and press tool button to restart the interface again. The initialization is achieved by placing
the tool around two green lines. The procedure, in average, takes approximately 5 seconds to re-initialize. The next step is to perform interactive functions with tool manipulations. The system recognizes the position and orientation of the tool to call a certain function, for example, click a button or rotate a 3D virtual cube. The final step of the SCI is to render the virtual environment based on the mouse interactions. The following sections present the last two steps of the overall software architecture, including SCI design and rendering implementation.

5.2 Augmented surgical environment design

In Chapter 2, we reviewed two SCIs developed previously, called “Ring menu” [21] and “Gesture-based recognition system” [34]. These two interfaces provide an intuitive design; however, neither of them could achieve 3D manipulation of the object by using such tools. This thesis proposes a more comprehensive design that enables surgeons to access not only 2D images but also 3D organ scans. Additionally, the system provides a preliminary result for two-tool SCI system to manipulate a virtual thread as “two hands”.

5.2.1 Menu design

The proposed SCI contains three levels, including “Main menu”, “Sub-menu” and actual function page. Each menu is comprised of different buttons. The main menu contains “Preoperative images”, “Patient record” and “3D effect” sub-menus. Within each sub-menu, there are different options. For example, “Image box”, “Thread manipulation” buttons are under “3D effect” sub-menu. Each sub-menu provides the option to “Return” to the upper menu or “Exit” the program. A display page is rendered for a specific function call. The example would be “Patient record” function page and “Image box” function page. The overview of the menu design is shown in Figure 5.2.

5.2.2 Interaction design

To operate the system, several interactive functions are defined, which translate the manipulation of the tool to a virtual interaction. By using these functions, we send the commands to the computer and it triggers rendering of the page accordingly.

All the buttons are designed to be semi-transparent. Our interactive system supports
Figure 5.2: SCI design and overall structure
the “wait to click” paradigm. By tracking the tip location (i.e. $x, y$ coordinates) of the instrument, a certain task can be selected. In another words, once the system observes the user pointing the surgical tip (like a cursor) within the button, the button is highlighted. After the button is highlighted for 2 seconds, the button is considered to be clicked and the corresponding function is called. Then, the system displays its sub-menu containing lower-level set of buttons such as CT, MRI and XRAY under the “Pre-operative images” menu.

To demonstrate the interactive concept, Figure 5.3(a) shows the virtual main menu of our proposed SCI, which comprises of several buttons to direct surgical tasks, including the pre-operative images, 3D effect and patient record. The left main window shows the tracking result, virtual menus and objects. The upper right window shows additional images or patient information and the bottom right window shows the 3D reconstructed tool drawn using OpenGL. For example, in Figure 5.3(b), the upper right window shows the selected “X-ray” image scan of the patient; in Figure 5.3(c), the bottom right window shows the reconstructed position of the two tools. Additionally, under each sub-menus or displayed page, the user is able to return to the main menu or exit from the interactive system by clicking on “Return” or “Exit” button.

Figure 5.4 is a display of the patient record function. Critical images and diagnosis are displayed to facilitate surgeons. When necessary, the record will be pulled out from the database and displayed. The designed interface of this function is to translate and move the patient record image according to the tip position. The tip is fixed in the center of the record. Additional information and record could be displayed in the upper right window.

Figure 5.5 shows the 3D display of a virtual organ within a cube under “Image box” function. By selecting one of the faces of the cube (Figure 5.5(a)), the face is highlighted and is stuck onto the tool tip. The translation of the tip determines the position of the cube in 3D space (Figure 5.5(b)).

Figure 5.6 shows the 3D display of a thread. By manipulating the thread using two tools, the thread is deformed and translated according to the tip of the tools.

To summarize, each function corresponds to a specific interaction procedure, as shown in Table 5.1. The notation of the position $(x, y, z)$ and tool orientation $(\alpha, \beta, \gamma)$ is shown in Figure 5.11. The detailed implementations of the above functions are presented in section 5.3.
CHAPTER 5. 3D MOUSE INTERACTIVE PARADIGM

Figure 5.3: (a) SCI main menu containing “pre-operative images”, “patient record”, “3D effect” and “Exit”. “3D effect” function is selected. (b) Sub-menus under “Pre-operative images” function, including “CT”, “MRI”, “XRAY” and “Return”. “XRAY” is selected; the right upper window shows the XRAY scan once it has been chosen. (c) Sub-menus under “3D effect” function, including “Image Box”, “Thread manipulation” and “Return”. 
Figure 5.4: “Patient record” function. Demonstration of the Image overlay idea.

Figure 5.5: “Image box” function a) Cube is selected by clicking one of its faces b) The 3D object can be rotated and translated according to the tool’s positioning.
Figure 5.6: “Thread manipulation” function, the thread is manipulated based on the position of the tips. This function is only valid in two tools tracking system. The number “1” and “2” represent the label of the tools.

Table 5.1: Interaction Protocol of the current 3D mouse SCI

<table>
<thead>
<tr>
<th>Function</th>
<th>Mouse function</th>
<th>Interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buttons</td>
<td>Move the tool tip within the button region.</td>
<td>Wait two seconds and the according function is triggered.</td>
</tr>
<tr>
<td>Patient records</td>
<td>Tip position ((x, y, z))</td>
<td>The record is stick on the tip. Patient record move according to the tip location.</td>
</tr>
<tr>
<td>Image box function</td>
<td>Tip location is on one of the cube faces</td>
<td>Stick on the tool and the manipulation begins.</td>
</tr>
<tr>
<td></td>
<td>Tip position ((x, y, z))</td>
<td>Cube translation in (x, y, z) direction</td>
</tr>
<tr>
<td></td>
<td>Tool orientation ((\alpha, \beta, \gamma))</td>
<td>Cube rotation along (x, y, z) axis</td>
</tr>
<tr>
<td>Thread manipulation</td>
<td>Two tips ((x_1, y_1, z_1), (x_2, y_2, z_2))</td>
<td>The head and tail of the thread move according to the tool tip.</td>
</tr>
</tbody>
</table>
5.3 3D-mouse system implementation

In order to realize an augmented reality (AR) interactive system for surgical application, the graphical display according to the designed interface is a crucial part. To align the graphics system with the physical world system, we have to convert camera parameters to OpenGL parameters. This section addresses this problem. It also presents the detailed implementation of the interaction introduced above, for example, the rendering of the virtual cube and the thread.

5.3.1 Align physical camera with virtual camera

In the current setup, MIL is used for image capturing, OpenCV is used for major image processing and tracking module, with few functions written in MIL. The entire graphics system is displayed using OpenGL. Thus, the final displayed buffer is OpenGL window. To let the third party libraries be compatible with each other, the pixel format needs to be considered in order to render the processed image and virtual graphics. For detailed implementation please refer to Appendix B.

Even though the pixel formats are compatible with each other, another problem for graphics display is to map the physical camera parameters with the virtual camera parameters. Since the 3D Graphics libraries, such as OpenGL, do not use the same parameters for rendering a virtual scene, we cannot apply the intrinsic and extrinsic parameters of a camera directly to the virtual object augmentation. In the following section, we explain the relationship between the physical and virtual camera parameters and provide the solution to align the real world to the virtual entities. The method is similar to Li’s method [47]. The experimental results are shown in Section 5.4.

The physical camera model has a right-hand coordinate system as shown in Figure 5.7. Mathematically, projection can be written as the following equation, assuming the optical center of the camera is the origin of the world frame:

\begin{equation}
    u_{\text{real}} = f_x \frac{X}{Z} + u_0 \tag{5.1}
\end{equation}

\begin{equation}
    v_{\text{real}} = f_y \frac{Y}{Z} + v_0 \tag{5.2}
\end{equation}

For virtual camera implemented in OpenGL, the coordinate system is defined such that
the front of the camera is pointing in the -Z direction. The +Y direction is the upward vector of the camera. The OpenGL frame is shown in Figure 5.8.

In OpenGL, a 3D location is transformed to a 2D image pixel through the following transformation (Figure 5.9).

The Modelview matrix transforms a 3D vertex to eye coordinates, which is the same as a camera’s local coordinates. Thus, Modelview matrix corresponds to the extrinsic parameters of a physical camera. Since, the world frame is defined at the optical center of the camera, we only need to set-up the projection matrix, which is used to transform the vertex in the eye coordinate system to the clip coordinates. gluPerspective function is used here to setup our virtual camera. The syntax of the function call is:

\[
\text{gluPerspective}(\text{fovy}, \text{aspect}, \text{Near}, \text{Far}) ;
\]

There are four parameters for gluPerspective function: fovy, aspect, Near and Far. The transformation matrix is as follows:

\[
M_p = \begin{bmatrix}
\frac{\cot(\frac{\text{fovy}}{2})}{\text{aspect}} & 0 & 0 & 0 \\
0 & \frac{\cot(\frac{\text{fovy}}{2})}{\text{aspect}} & 0 & 0 \\
0 & 0 & \frac{z_{\text{Far}} + z_{\text{Near}}}{2} & \frac{z_{\text{Near}} - z_{\text{Far}}}{z_{\text{Far}} - z_{\text{Near}}} \\
0 & 0 & -1 & 0 \\
\end{bmatrix}
\] (5.3)

Finally, we need to set up viewport transformation to determine the region of image to be rendered on the screen. The viewport transformation is set by the function glViewport.

\[
\text{glViewport}(x_0, y_0, \text{width}, \text{height}) ;
\]

The corresponding transformation matrix transforms the vertex in clip coordinate system to window coordinates:

\[
M_v = \begin{bmatrix}
\frac{\text{width}}{2} & 0 & \frac{\text{width}}{2} + x_0 \\
0 & \frac{\text{height}}{2} & \frac{\text{height}}{2} + y_0 \\
0 & 0 & 1 \\
\end{bmatrix}
\] (5.4)

The OpenGL setup is completed by multiplying all the matrices together:

\[
\begin{bmatrix}
u \\
v \\
1 \\
\end{bmatrix} = M_v \cdot M_p \cdot \begin{bmatrix}
X \\
Y \\
Z \\
\end{bmatrix}
\] (5.5)
CHAPTER 5. 3D MOUSE INTERACTIVE PARADIGM

Figure 5.7: The camera frame.

Figure 5.8: The OpenGL frame.

Figure 5.9: The OpenGL transformation pipeline.
Finally, we get the pixel coordinates of a 3D point in OpenGL.

\[
u_{\text{virtual}} = -\frac{\cot(\frac{\text{fovy}}{2}) \cdot \text{width} \cdot X}{2 \cdot \text{aspect} \cdot Z} + \frac{\text{width}}{2} + x_0 \tag{5.6}
\]

\[
v_{\text{virtual}} = -\frac{\cot(\frac{\text{fovy}}{2}) \cdot \text{height} \cdot Y}{2 \cdot \text{aspect} \cdot Z} + \frac{\text{height}}{2} + y_0 \tag{5.7}
\]

There is another problem before we can equate the virtual and real world entities together. The OpenGL coordinates are rotated with respect to the image coordinates. For example, the coordinate \((0, 0, 1)\) in physical camera corresponds to \((0, 0, -1)\) in virtual camera, due to the constraint of right-hand coordinate system, our image coordinate system is different (see Figure 5.10).

Given the virtual and actual pixel coordinate and the frame relationship above, we can equate the two correspondences together:

\[
f_x \frac{X}{Z} + u_0 = f(\frac{\text{fovy}}{2}) \cdot \text{width} \cdot X \cdot \frac{\text{width}}{2} + x_0 \tag{5.8}
\]

\[
f_y \frac{Y}{Z} + v_0 = \frac{\text{height}}{2 \cdot \text{aspect} \cdot Z} - \frac{\text{height}}{2} - y_0 \tag{5.9}
\]

Finally we get the parameters for rendering 3D points in OpenGL:

\[
x_0 = u_0 - \frac{\text{width}}{2} \tag{5.10}
\]

\[
y_0 = \frac{\text{height}}{2} - v_0 \tag{5.11}
\]
\[ fovy = 2 \cdot \arctan \left( \frac{\text{height}}{2f} \right) \] (5.12)

\[ \text{aspect} = \frac{\text{width} \cdot f_y}{\text{height} \cdot f_x} \] (5.13)

Through these equations, we can calculate OpenGL parameters from the real camera parameters. The detailed results are described in Section 5.4.

### 5.3.2 3D manipulation with virtual organ

The “image box” function is developed to simulate a potential system which superimposes a 3D pre-operative scan on top of a physical object. This can be used for various image guided surgical diagnostics and procedures. Examples would be to access and display information of the graphical entities, zoom in the virtual object, and live-scan to visualize an abnormal tissue or even a tumor. This design could allow users to translate, scale, and rotate the virtual organ. To simplify the design, we assume that the organ is within a 3D cube. By manipulating the cube, the organ is moved accordingly. Another example of this application is to help surgeons manually register the scan to a real “organ”. Since the pre-operative scans and live endoscopic images of the organ are not necessarily taken from the same viewing perspective, the user might use the system to superimpose the scan on top of the live images to enhance the visualization during MIS.

The cube overlay program is written in C++ using OpenGL. As shown in Figure 5.5, the program allows the user to click one face of the virtual cube with the surgical tool to start the manipulation. The cube is originally drawn at 80\text{mm} depth away from the camera. The length is 40\text{mm} with zero rotation around its axis. In order to have a natural way of displaying the virtual objects, the interface preserved the perspective view of the cube. In another words, the cube becomes smaller as it gets farther and larger as it gets closer to the camera.

When the surgical tool gets close to the vicinity of one of its faces, the face is highlighted. If the user holds the tool for another 2 seconds, the face is clicked and stick on the tool tip. It means that the grabbed box will shift along the tool tip and the organ inside the cube will move accordingly.

In order to control the cube movement, the cube manipulation is divided into translation and rotation part along \(x\), \(y\) and \(z\) axis in the camera frame. Both translation and rotation
are based on the start pose of the tool as $T_0(x_0, y_0, z_0)$ and $R_0(\alpha_0, \beta_0, \gamma_0)$. As the manipulation begins, the new pose of the tool in $n$ frame becomes $T_n(x_n, y_n, z_n)$ and $R_n(\alpha_n, \beta_n, \gamma_n)$. The cube translation and rotation along each axis are as the following:

$$T_{nw} = T_n - T_0 = (x_n - x_0, y_n - y_0, z_n - z_0)$$ (5.14)

$$R_{nw} = R_n - R_0 = (\alpha_n - \alpha_0, \beta_n - \beta_0, \gamma_n - \gamma_0)$$ (5.15)

Where the $T$ is the position of the tool tip and $R$ is the including angle of the tool orientation vector with respect to $Y$-$Z$, $X$-$Z$ and $X$-$Y$ surface, respectively (as shown in Figure 5.11).

The angle is calculated from the following equation:

$$\alpha = \arctan\left(\frac{x}{\sqrt{y^2 + z^2}}\right)$$ (5.16)

$$\beta = \arctan\left(\frac{y}{\sqrt{x^2 + z^2}}\right)$$ (5.17)
\[ \gamma = \arctan\left( \frac{z}{\sqrt{x^2 + y^2}} \right) \] (5.18)

By manipulating the tool, the organ can be zoomed in and out by translating the tool in Z axis. It can also be rotated by moving the tool vector in different directions. To prove the concept of the organ registration idea, we set a “target” position by pressing the button. By manipulating the cube, the user can place the organ to align with the target organ as shown in Figure 5.12. When the organ’s axis is close enough to the target’s axis, the two objects collide and stick together (Figure 5.12(d)).

Figure 5.12: Screenshots of the tool manipulation procedure that try to register a virtual organ to the target.
5.3.3 3D manipulation with virtual thread

The “thread manipulation” function simulates a suture and a suturing task, where two tools work as two needles to manipulate the thread. The single-dimension mass-spring model is used to model the sutures and the strings. The behavior of the string is dominated by the internal forces acting on each mass point or element. The internal forces include the friction force, linear spring, linear damper, tensional spring, tensional damper and swivel damper. Various behaviors of the string can be modelled such as bending and twisting, knotting and unknotting. During graphical rendering, the cylinders as suture segments are connected to each other between two successive points. The shape of the suture is calculated using explicit Euler method. The thread we modelled in this function is originally from [49]. Several screenshots are shown in Figure 5.13, simulating the thread manipulation with stretching, pulling and crossing using two surgical tools.

![Figure 5.13: Snapshots of the two tools’ thread manipulation.](image)
5.4 Testing the accuracy of the 3D-mouse system

To examine the augmentation result, we rendered virtual objects on three calibrated images shown in Figure 5.14. The difference between adjacent intersections is 20mm. Figure 5.14 shows the arrow representing the local frame of each intersection and a rendered virtual object on top of the intersection. The experiments were conducted in three groups; each has a fixed depth between camera to the image pattern, including 60mm, 90mm and 110mm. The virtual objects were rendered at the same location in the virtual space.

As shown in the Figure 5.14, the augmentation results are visually reasonable. However, in order to measure the errors, we project and measure the points with real camera parameters and with OpenGL’s. We calculate the error in the following equation:

\[ e_x = u_{\text{real}} - u'_{\text{virtual}} \]

\[ e_y = v_{\text{real}} - v'_{\text{virtual}} \]

(5.19)

(5.20)

The table shows the mean and standard deviation of projection error. It shows the error is within 1 pixel in both \( x \) direction and \( y \) direction.
Figure 5.14: Screenshots of the augmentation accuracy result. (a-c) Virtual frames are rendered on top of calibration paper at depth of 60mm, 90mm and 110mm, respectively. (d-f) Virtual cylinders are rendered on top of each intersection at depth of 60mm, 90mm and 110mm, respectively.
Table 5.2: Experimental results of the augmentation accuracy

<table>
<thead>
<tr>
<th>Depth</th>
<th>Experiment Results</th>
<th>Ground Truth</th>
<th>Mean error ((x, y))-pixel</th>
<th>Std. error ((x, y))-pixel</th>
</tr>
</thead>
<tbody>
<tr>
<td>60mm</td>
<td>(206, 125)</td>
<td>(206, 121)</td>
<td>((-0.1, 1.8))</td>
<td>(1.5, 1.4)</td>
</tr>
<tr>
<td></td>
<td>(320, 125)</td>
<td>(322, 124)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(434, 125)</td>
<td>(434, 125)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(206, 240)</td>
<td>(207, 238)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(320, 240)</td>
<td>(320, 240)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(434, 240)</td>
<td>(435, 239)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(206, 355)</td>
<td>(203, 352)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(320, 355)</td>
<td>(319, 352)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(434, 355)</td>
<td>(435, 353)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90mm</td>
<td>(230, 150)</td>
<td>(230, 150)</td>
<td>((0.4, 1.3))</td>
<td>(1.5, 1.4)</td>
</tr>
<tr>
<td></td>
<td>(320, 150)</td>
<td>(320, 151)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(410, 150)</td>
<td>(408, 149)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(230, 240)</td>
<td>(231, 238)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(320, 240)</td>
<td>(320, 240)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(410, 240)</td>
<td>(408, 238)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(230, 330)</td>
<td>(231, 328)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(320, 330)</td>
<td>(321, 327)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(410, 330)</td>
<td>(407, 327)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>110mm</td>
<td>(252, 172)</td>
<td>(251, 172)</td>
<td>((0.1, 0.8))</td>
<td>(0.8, 0.8)</td>
</tr>
<tr>
<td></td>
<td>(320, 172)</td>
<td>(320, 170)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(388, 172)</td>
<td>(387, 171)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(252, 240)</td>
<td>(252, 238)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(320, 240)</td>
<td>(320, 240)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(388, 240)</td>
<td>(389, 239)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(252, 308)</td>
<td>(252, 308)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(320, 308)</td>
<td>(321, 307)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(388, 308)</td>
<td>(387, 308)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 6

Preliminary studies of laser scanner and OCT system

MIS provides a number of benefits to patients over open surgery, including lower risk of infection and swifter recovery times. However, due to the lack of depth cues, surgeons face significant visual obstacles. Therefore, it is important to develop a visualization tool to provide 3D model of the inner cavity. In order to have a synchronized positioning tool, Section 6.1 presents a real-time 3D laser tracking system to localize a laser point in 3D space. It is based on the results of the previous tracking scheme. Section 6.2 proposes a laser scanner using triangulation theory. This system can be used as an “pre-operative” scanning tool for surface reconstruction. To distinguish normal tissue from possible cancerous tissue, another optically instrumented surgical tool is proposed in Section 6.3 to discover the potentiality of adopting OCT to therapy routines.

6.1 Minimally invasive 3D real-time scanning system

This section proposes a real-time laser scanner system, combining the previous tool tracking system presented in Chapter 4 with an additional laser tracking module. This system offers a portable and functional tracker to point-scan the surface in real-time. The system can be used to position an robot arm (holding endoscope camera) to follow a desired path or to a defined position. Moreover, the position of a shone laser point allows the surgeons to estimate the relative position of a surgical tool with respect to the inner cavity of the
6.1.1 Method overview

This section presents an overview of the laser scanning system. The proposed method is shown in Figure 6.1. The system has a standard MIS set-up with additional laser mounted on the tool tip. The laser shines light on the surface and forms a laser spot. The endoscope camera is used to capture the image of both the surgical tools and the laser spot. If either the tool and the laser spot is not captured, the system stops tracking until the next arrival frame with both information available. By tracking the surgical tool and the laser, the 3D spatial position of the laser spot can be calculated. The following section presents the geometry calculation of this procedure.

The laser spot 3D position is denoted as \((x_l, y_l, z_l)\), with 2D projective position \((u_l, v_l)\) (Figure 6.1). For each frame, blob detection is performed to localize the shone laser. As shown in Figure 6.2, the yellow circle is the 2D blob tracking result superimposed on the actual laser spot. The centroid of the blob represents the laser point location \((u_l, v_l)\).

In order to map this 2D image coordinate to its 3D equivalent \((X_l, Y_l, Z_l)\), we use the image projection formula as the following:

\[
\begin{align*}
(x_l, y_l, z_l) & \rightarrow (u_l, v_l) \\
X_l & = x_l + e_x \cdot z_l \\
Y_l & = y_l + e_y \cdot z_l \\
Z_l & = z_l
\end{align*}
\]
Figure 6.2: The snapshots of the real-time laser tracking system. The yellow dots represent the detected laser spot. The green dot is the detected tip of the tool and the blue lines are extracted edges of the tool.

\[
\begin{bmatrix}
    u_l \\
    v_l \\
    1
\end{bmatrix} = \begin{bmatrix}
    f & 0 & u_0 \\
    0 & f & v_0 \\
    0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
    X_l \\
    Y_l \\
    Z_l
\end{bmatrix} / Z_l = \begin{bmatrix}
    f \frac{X_l}{z_l} + u_0 \\
    f \frac{Y_l}{z_l} + v_0 \\
    1
\end{bmatrix}
\]

However, this equation alone cannot localize the 3D laser position \( P_{laser}(X_l, Y_l, Z_l) \) with 3 unknowns. Nevertheless, the previous tracking system in Chapter 4 is able to localize the 3D position and orientation of the surgical tool, as \((x_t, y_t, z_t)\) and \((e_x, e_y, e_z)\), respectively. Since the laser spot is on the line of the 3D orientation vector \( \vec{v}(e_x, e_y, e_z) \) (as shown in Figure 6.1), the position of the laser spot can be written as a linear combination of the tool tip coordinate and the unit vector of the orientation of the tool. Therefore, we can derive the following equation:

\[
\begin{bmatrix}
    X_l \\
    Y_l \\
    Z_l
\end{bmatrix} = \begin{bmatrix}
    X_t \\
    Y_t \\
    Z_t
\end{bmatrix} + \Delta d \begin{bmatrix}
    e_x \\
    e_y \\
    e_z
\end{bmatrix}
\]

Given equation 6.1 and equation 6.2, we are able to calculate the spatial coordinate of the laser spot as the following:

\[
Z_l = \frac{f e_y x_t - f e_x y_t}{(u_t - u_0)e_y - (v_t - v_0)e_x},
X_l = \frac{(u_t - u_0)Z_l}{f},
Y_l = \frac{(v_t - v_0)Z_l}{f}
\]
CHAPTER 6. PRELIMINARY STUDIES OF LASER SCANNER AND OCT SYSTEM

From above equation, the system is able to reconstruct the position of the laser spot using the knowledge of the previous tracking system and 2D projective location of the laser spot. The system is processed only when both the laser spot and the tool appear within the image. The following section will present the experimental results of this tracking system.

6.1.2 Experimental results of the 3D real-time scanning system

From Equation 6.3, the $X$ and $Y$ coordinates are depending on the $Z$ coordinate, hence, we will only test the $Z$ coordinate (estimated depth) to validate this system. Three sets of experiments are conducted with depth of 90, 100 and 120mm with respect to the camera frame. The ground truth of the depth is obtained using a regular ruler. Each experiment set comprises of 50 frames with different position and orientation of the surgical tool. The video frames are captured by moving the tool consistently until 50 frames are collected. Table 6.1 shows the mean and standard error of the experiment.

<table>
<thead>
<tr>
<th>Ground truth (mm)</th>
<th>Mean error(mm)</th>
<th>Std. error(mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z = 90$</td>
<td>1.5</td>
<td>8.4</td>
</tr>
<tr>
<td>$Z = 100$</td>
<td>2.2</td>
<td>7.5</td>
</tr>
<tr>
<td>$Z = 120$</td>
<td>2.1</td>
<td>7.8</td>
</tr>
</tbody>
</table>

The large standard error (about 8mm) means that the calculated $Z$ coordinate has unstable estimation. It is due to the fact that the 3D reconstruction error is largely depending on the accuracy of the image processing such as edge detection and tip positioning. For example, a small shift of the tip position calculation might result in 5-10mm of 3D reconstruction error. Moreover, the blob tracking of the laser spot might have slight variance due to the light condition. As a result, the error accumulates from the image processing, calibration, 3D pose reconstruction and blob tracking, forming a rough estimation of the laser point location in 3D space. This estimation, however, is enough for an augmented reality navigation and rough registration in the minimally invasive surgery.
6.2 Multi-spots laser triangulation system for endoscopic imaging

Unlike the previous real-time laser tracker system, this system is designed for pre-operative scanning of the inner cavity. A single laser spot is useful for positioning and localization, however, in terms of 3D modelling, another scanning system has to be designed. This section presents a multi-spot laser tracking system for 3D inner surface reconstruction. This system allows 1D scan of a surface once at a time. Additional mirror is also used for manually scan the surface, comprising a complete 2D scan. By using “triangulation” method, the depth of the surface with respect to camera can be extracted, finally reconstructing the 3D surface.

In this section, we first introduce the “triangulation” method, the overall optical system and its experimental set-up. Then, the solutions concerning this approach, such as optical instrument selection and image processing, are described in detail.

6.2.1 System description

Triangulation is the process of determining the location of a point by measuring angles to it from known points at either end of a fixed baseline, rather than measuring distances to the point directly.

The proposed scanning system is based on the previous studies from Zhang[33] using the idea of triangulation to pinpoint the location of a laser spot. To extend Zhang’s idea, our system splits a single laser spot to multiple spots to retrieve multiple range data instead of a point. Furthermore, with a scanning mirror added in the system, we are able to manually scan a surface and reconstruct its 3D model.

In this thesis, we enhanced Zhang’s idea[33] by replacing the horizontal manual scan with multi-spots laser system. This improvement allows the acquisition of multiple range data simultaneously from a single video frame.

Figure 6.3 shows an overview of this multi-spots system. The laser is attached at the tip of the surgical tools. A series of cylindrical lenses transform the laser point to a line. A “block paper” (paper with cut apertures), then, divides this line into multiple spots. When the spots are projected onto the surface, they are observed by camera.

Figure 6.4 shows the typical triangulation scheme. Spot $C$, laser source, $A$ and camera optical center, $B$, forms a unique triangle, in which $\alpha$ can be measured and $\beta$ can be calculated.
CHAPTER 6. PRELIMINARY STUDIES OF LASER SCANNER AND OCT SYSTEM

Figure 6.3: The system overview

Figure 6.4: Typical triangulation
The relationship between $\beta$ and image location is as follows:

$$\beta = \tan\left(\frac{f}{x}\right)^{-1} \quad (6.4)$$

$X$ is the distance from target spot to image center in pixel numbers and $s$ is the number of pixels in each millimeter. $f$ is the focal length of the camera. Given $\alpha$, $\beta$ and $L$, depth $z$ can be derived from the following equation:

$$z = \frac{L \sin \alpha \sin \beta}{\sin(\alpha + \beta)} \quad (6.5)$$

The $x$, $y$ coordinates can be easily extracted given camera parameters (Equation 2.1). The depth derived from Equation 6.5, along with $x$, $y$ coordinates are used to reconstruct the target surface.

To build such multi-spots system, the optical selection is an important step. The basic equipments are listed as the following:

1. Helium-neon(He-Ne) laser source.
2. A series of cylindrical lens (diverging).
3. Block paper (9 apertures).
4. A prototype represents object surface.
5. Converging lens.
6. CCD camera

There are several constraints for equipment selection in order to build a reasonable system. According to the camera’s product specifications, the maximum horizontal distance of the image plane is $6.4\, mm$. As shown in Figure 6.5, the focal length is $35\, mm$ and the scanning surface is $15\, cm \sim 20\, cm$ away from the optical center. Thus, the camera only allows $2.7\, cm \sim 3.6\, cm$ horizontal space as the target surface. However, due to the loss of intensity by blocked paper, this space shrinks to approximately $2\, cm \sim 3\, cm$, as shown in Figure 6.6. Therefore, the calculated diverging angle is limited within the range of $5.7^\circ \sim 11.4^\circ$.

Another constraint is that the block paper should have a reasonable distance to the cylindrical lens. If it is too close to the lens, it fails to form a line. In another extreme
CHAPTER 6. PRELIMINARY STUDIES OF LASER SCANNER AND OCT SYSTEM

condition, when it is close to the specimen, the paper will block the beam path. A reasonable position is 10 ∼ 16cm away from lens. Therefore, a block paper is designed within 0.6cm ∼ 1.5cm wide and 1.5cm ∼ 2.5cm high.

According to the above technical constraints, the following subsection presents the selected equipments.

1. He-Ne laser source.

He-Ne laser is the most common type of gas laser. It has the advantages in laser beam stability, laser modes, beam diameter, coherence length, output power, and most importantly, the lower price. Thus, we choose He-Ne for testament.

2. Converging lens

In order to force CCD camera to focus on the image plane, we choose a converging lens with a focal length of 35mm. To test this lens, a point laser is shone orthogonally to a flat surface. A spot with radius of 0.5mm is captured, provided that it is small enough to be employed in the following experiment.

3. Cylindrical lens

To spread the light into a line, a Plano-convex cylindrical lens with negative focal length is chosen. As stated in the design constraint, the diverging angle is \( \theta = 5.7^\circ \sim 11.4^\circ \). The relationship between focal length and converging angle is as follows:

\[
f = \frac{\omega_0}{\tan(\frac{\theta}{2})}
\]  

(6.6)

A typical He-Ne laser waist \( \omega_0 \) is 0.4mm, requiring focal length to be \( f = -4.0 \sim -8.0mm \), which is beyond regular lens capability. Therefore, series of lenses are placed.

Figure 6.5: Distance constraint between the surface and image plane
CHAPTER 6. PRELIMINARY STUDIES OF LASER SCANNER AND OCT SYSTEM

88

together to add up the powers (reciprocal of focal length). The relationship of the total focal length is:

$$\frac{1}{f_1} + \frac{1}{f_2} + \frac{1}{f_3} + \frac{1}{f_4} + \cdots = \frac{1}{f} \quad (6.7)$$

Finally, using two $FL = 50.8\text{mm}$ and three $FL = 25.4\text{mm}$, we build up a lens system with $f = 6.35\text{mm}$ and $\theta = 7.2^\circ$.

4. Block paper

According to design constrains, block paper is cut with $1\text{cm} \times 2.5\text{cm}$ size, 9 apertures, as shown in Figure 6.7.

6.2.2 One-mirror, multi-spots system

In this subsection, we start with a typical triangulation. In order to scan surface vertically, a mirror is added in this configuration, as shown in Figure 6.8. The center of this mirror is kept aligned with the optical center. The testing surface is a wedge with a slope angle of $45^\circ$.

This set-up is a typical triangulation configuration with constant baseline $L = 60\text{mm}$. The angle of the central beam is $\alpha = 70^\circ$. The total diverging angle is $7.2^\circ$ for block paper with 8 apertures, which is separated as about $1^\circ$ for each beam.

6.2.3 Two-mirror, multi-spots system

Another mirror is added in the second set-up (Figure 6.9) to enlarge the scanning area, but the mirror is not aligned with the optical center. This case often occurs in actual

![Figure 6.6: Distance constraint between lens, block paper and surface](image)

Figure 6.6: Distance constraint between lens, block paper and surface
CHAPTER 6. PRELIMINARY STUDIES OF LASER SCANNER AND OCT SYSTEM

Figure 6.7: Design of block paper

Figure 6.8: One-mirror, multi-spots system configuration
surgical operation because the surgical tools are randomly positioned. Therefore, the laser spot moves accordingly, forcing inconstant $L$. To solve this problem, a look-up table is introduced to correct distortion caused by this “non-typical” triangulation.

In theory, the calculated distance $z$ should be the same for all spots shining on a flat surface. However, experiment shows that the computing distance changes as baseline varies. To extract the actual distance, a flat surface with a uniform depth is used to reconstruct a look-up table between angles before and after correction, as shown in Table 6.2. Afterwards, these table will be used to apply displacements for later scanning tests.

After obtaining this table, we apply them to a tested “cup handle”. Figure 6.10 shows the correction effect after the look-up table has been applied.

The disadvantage of this method is that it only applies to a specific distance. When the distance changes, the angle changes accordingly, thus, another look-up table has to be applied. However, only small difference $\pm 5\%$ exists within a small range and the feasibility of this approach has been verified in the preliminary experiment.

### 6.2.4 Image processing

The goal of image processing is to segment the spots and calculate their center coordinates. The following steps are the general processes for laser spot segmentation using Matlab Imaging Toolkit.
CHAPTER 6. PRELIMINARY STUDIES OF LASER SCANNER AND OCT SYSTEM

Table 6.2: Data Before and after Correction for a flat surface

<table>
<thead>
<tr>
<th>Angle before correction (degree)</th>
<th>Distance before correction (mm)</th>
<th>Distance after correction (mm)</th>
<th>Angle after correction (degree)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4.0</td>
<td>124.4</td>
<td>126.8</td>
<td>-5.6</td>
</tr>
<tr>
<td>-3.0</td>
<td>125.1</td>
<td>126.8</td>
<td>-4.2</td>
</tr>
<tr>
<td>-2.0</td>
<td>125.6</td>
<td>126.9</td>
<td>-2.8</td>
</tr>
<tr>
<td>-1.0</td>
<td>126.4</td>
<td>126.8</td>
<td>-1.3</td>
</tr>
<tr>
<td>0</td>
<td>126.8</td>
<td>126.8</td>
<td>0</td>
</tr>
<tr>
<td>1.0</td>
<td>127.1</td>
<td>126.8</td>
<td>1.2</td>
</tr>
<tr>
<td>2.0</td>
<td>127.6</td>
<td>126.8</td>
<td>2.5</td>
</tr>
<tr>
<td>3.0</td>
<td>128.1</td>
<td>126.7</td>
<td>3.9</td>
</tr>
<tr>
<td>4.0</td>
<td>128.5</td>
<td>126.8</td>
<td>5.1</td>
</tr>
</tbody>
</table>

Figure 6.10: Simulation before and after correction.
CHAPTER 6. PRELIMINARY STUDIES OF LASER SCANNER AND OCT SYSTEM

1. Read original image
2. Improve contrast
3. Binarize image
4. Fill image regions and holes
5. Morphologically open image (remove small particles)
6. Segment image

The result is a binary image containing the contour of the spots. As shown in Figure 6.11(a), the process cannot segment the spots with low intensity, such as, spot No.8 and 9. However, by changing the threshold, these spots appeared, as shown in Figure 6.11(b). As a trade-off, the lower threshold causes joint contours of spots No.1 to No.6,

![Figure 6.11: (a) Overlapped image with segmented perimeter (b) Segmentation with low threshold](image)

Figure 6.11: (a) Overlapped image with segmented perimeter (b) Segmentation with low threshold

Regarding this problem, 5 ~ 10 thresholds are being applied to iterate segmentation process. The final central coordinate is the average of all effective coordinates.

6.2.5 Experimental results of the laser triangulation system

In this section, both one-mirror-multi-spot system and two-mirror-multi-spots system are experimented to evaluate the proposed method. The depth information can be obtained using Equation 6.5.

1. One-mirror, multi-spots system

This configuration uses a mirror to manually scan a wedge with an inclination angle of 45°. The simulated result is shown in Figure 6.12. To evaluate the accuracy of this
system, we calculate the slope angle of this simulated model. The measured angle is $42.3^\circ$, showing acceptable accuracy. Notice that in Figure 6.12, some of the points are miss detected from the plane. This error is caused by low intensity spots and imperfect spot shape.

![Simulation result of one-mirror, multi-spots system](image)

Figure 6.12: Simulation result of one-mirror, multi-spots system

2. Two-mirror, multi-spots system

The following Figures 6.13 are the experimental results for the second configuration. The testing surfaces are a cup handle and a Barbie face, respectively. The scanning area is about $50mm \times 50mm$. As shown in the Figures 6.13, the reconstructed surface is close to the appearance of the actual surface.

![Experimental results](image)

Figure 6.13: (a) The modelled surface by scanning a “cup handle” (b) The experimental result of a Barbie face scan, some features appeared on this reconstructed mesh model. From left to the right, they are Barbie’s mouth, nose, and their eyes, respectively.
To further evaluate this system, we validate this approach using a separate set-up as shown in Figure 6.14. A micrometer calliper is used in this set-up to move the surface at different distances and at the meantime record the ground truth.

![Point laser system diagram](image)

Figure 6.14: Point laser system

The measured and calculated distance is shown in Table 6.3. The calculated system error is within 3mm.

### 6.2.6 Discussion

This section presents the triangulation-based laser-scanning system. One of the advances is that it divides one point laser into several spots in order to simultaneously extract depth, avoiding horizontal scan. The simulated result is close to the appearance of the actual surface; however the error always exists due to inaccurate physical set-up, material difference, image processing, and imperfect spot.

The result of this research can be applied to surgical operations in the future. However, it has several drawbacks: The first problem is that the image processing and the model reconstruction process are obtained semi-automatically. This is due to the complexity of
Table 6.3: Comparison of measured and calculated distance

<table>
<thead>
<tr>
<th>Measured Distance (mm)</th>
<th>Calculated Distance (mm)</th>
<th>Error (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>153.0</td>
<td>151.4</td>
<td>1.6</td>
</tr>
<tr>
<td>153.5</td>
<td>152.1</td>
<td>1.4</td>
</tr>
<tr>
<td>154.0</td>
<td>152.8</td>
<td>1.2</td>
</tr>
<tr>
<td>154.5</td>
<td>153.5</td>
<td>1</td>
</tr>
<tr>
<td>155.0</td>
<td>154.2</td>
<td>0.8</td>
</tr>
<tr>
<td>155.5</td>
<td>154.8</td>
<td>0.7</td>
</tr>
<tr>
<td>156.0</td>
<td>155.5</td>
<td>0.5</td>
</tr>
<tr>
<td>156.5</td>
<td>156.3</td>
<td>0.2</td>
</tr>
<tr>
<td>157.0</td>
<td>157.0</td>
<td>0</td>
</tr>
<tr>
<td>157.5</td>
<td>157.7</td>
<td>-0.2</td>
</tr>
<tr>
<td>158.0</td>
<td>158.4</td>
<td>-0.4</td>
</tr>
<tr>
<td>158.5</td>
<td>159.1</td>
<td>-0.6</td>
</tr>
<tr>
<td>159.0</td>
<td>159.9</td>
<td>-0.9</td>
</tr>
<tr>
<td>159.5</td>
<td>160.6</td>
<td>-1.1</td>
</tr>
<tr>
<td>160.0</td>
<td>161.3</td>
<td>-1.3</td>
</tr>
<tr>
<td>160.5</td>
<td>162.0</td>
<td>-1.5</td>
</tr>
<tr>
<td>161.0</td>
<td>162.7</td>
<td>-1.7</td>
</tr>
<tr>
<td>161.5</td>
<td>163.4</td>
<td>-1.9</td>
</tr>
<tr>
<td>162.0</td>
<td>164.2</td>
<td>-2.2</td>
</tr>
<tr>
<td>162.5</td>
<td>164.9</td>
<td>-2.4</td>
</tr>
<tr>
<td>163.0</td>
<td>165.7</td>
<td>-2.7</td>
</tr>
</tbody>
</table>
CHAPTER 6. PRELIMINARY STUDIES OF LASER SCANNER AND OCT SYSTEM

the image, the variant light intensity and imperfect equipments (especially manually cut paper block). Another problem is that the non-typical configuration requires a look-up table for each distance, making this method time-consuming and tedious to be applied in real surgical operations. Finally, it is difficult to build a portable and flexible system consists of all the optical equipments applied in this experiment. A multi-spot endoscopic projection system would be used to replace this set-up eventually. Further consideration has to be taken to reduce the space and constraints in order to adopt this technology into surgical routines.

6.3 Minimally invasive OCT diagnosis system

Optical engineering and imaging have played a major role in the evolving field of minimally invasive surgery by making it possible to visualize the manipulation of tissue at remote internal sites. Optical Coherence Tomography (OCT) is one of the advanced optical imaging techniques that offers a non-contact, non-invasive medical imaging modality that can acquire sub-surface high-resolution cross sectional images of biological tissue. The advantage of OCT will allow for detection of cancerous tissues without cutting them in the surgical diagnosis procedure. The purpose of this section is to use the existing OCT system for tissue imaging and to test the potentiality of translating such technology to MIS application in the near future.

6.3.1 System Configuration and Setup

The overview of the OCT system is shown in Figure 6.15(a). Axial cross sectional data (called A-scans) are generated by interfering backscattered light with a reference reflection. A sequence of A-scans acquired while transversely scanning the beam is called a Brightness (B)-scan, using the grasper of the surgical tools. The light back reflected from the sample and reference arms are recombined in the fiber coupler and interfere to produce fringes related to the optical path length difference. A typical swept-source OCT contains the following components: swept source or tunable laser (SS), beamsplitter (BS), reference mirror (REF), sample (SMP) and photodetector (PD). The sample arm of the OCT system (the surgical tool) contained a grin lens as the final focusing element and was attached to one of the jaws of the grasper. As shown in Figure 6.15(b), each depth scan(A-scan) is generated by scan a single point of the organ. The A-scan is the information in the form of one dimensional.
In order to obtain a B-scan, the users need to “open” and “close” the jaw of the grasper to scan through the organ, providing a two-dimensional, cross-sectional view of the organ.

We used a Swept Source OCT prototype system (built in Biomedical Optics Research Group lab in SFU) operating at a central wavelength of 1310 nm with a 68 nm 3dB bandwidth [50], as shown in Figure 6.16(a). The experiment was performed using a typical SS-OCT system, which generally consists of a swept source laser, a balanced photodiode detector with interferometric reference and sample arm. The axial scan rate of the system was 8 kHz. The detailed OCT theory and system configuration are presented in Appendix C.

The scan mechanism in this system is to use the surgical grasper to operate “opening” and “closing” to scan through the tissue instead of a galvanometer mirror. The sample in this experiment here is chicken heart as shown in Figure 6.16(d).

6.3.2 Experimental Results

The procedure of the experiment was to scan the sample by “opening” and “closing” the jaws of the grasper. We captured the images of the fat and muscle sections of the chicken heart, as shown in Figure 6.17.

Real-time data acquisition was performed using a software package developed at SFU called OCTViewer. The software package is written in C++ for rapid image acquisition, processing and display. 2D images were acquired by manually scanning the beam using the grasper. The effective penetration depth is about 1 ∼ 2 mm. Figure 6.17 shows several screenshots of the OCT images. As shown in these figures, the fat tissue of the OCT image is loosely structured and appeared brighter. Whereas, the muscle tissue has a denser structure with dark appearance. This comparison reveals the possibility of practical applications of this technology on cancerous tissue diagnosis assuming their inner structures are different. Since this experiment is at its preliminary stage, only fat and muscle tissues are examined for proof of concept. Further experiments have to be implemented in the future for developing a valid clinical system.
CHAPTER 6. PRELIMINARY STUDIES OF LASER SCANNER AND OCT SYSTEM

Figure 6.15: (a) The Swept Source OCT prototype system. (b) The scan scheme of the OCT system.
CHAPTER 6. PRELIMINARY STUDIES OF LASER SCANNER AND OCT SYSTEM

Figure 6.16: The experimental setup of SS-OCT system. (a)-(b) The SS-OCT system contains swept source laser, a balanced photodiode detector with interferometric reference and sample arm. (c) The sample arm and a 1310 nm grin lens is attached to one of the jaws of the surgical tool. (d) The experiment sample of the OCT system - a chicken heart.
CHAPTER 6. PRELIMINARY STUDIES OF LASER SCANNER AND OCT SYSTEM

Figure 6.17: The extracted OCT images by scanning the chicken heart tissue. (a) The OCT image of “muscle” tissue. (b) The OCT image of “fat” tissue. (c) Extracted B scan with both “fat” and “muscle” tissue.
Chapter 7

Conclusion and Future work

7.1 Conclusion

In this thesis, we first present the 3D image tracking module for the 3D-mouse paradigm. The proposed method is based on the assumption that the endoscope motion is quasi-static. This assumption can be applied to most cases of surgical procedures where the endoscope needs to be stationary during a particular surgical task such as cauterizing, cutting and suturing. Due to the dynamic environment of the real surgical operation, developing a robust visual tracking algorithm is an enormous challenge. A number of factors are involved in the surgical environment, such as, extreme light conditions, sudden rapid motion, object reflectance, surgical smoke or partial occlusions of the tool. Under these conditions, therefore, it may be very difficult to detect and track an object in a video sequence or image frames. To overcome these problems, a combined algorithm that takes advantage of both top-down and feature-based approach is proposed. The Kalman filter is responsible for tool edge filtering and the feature-based approach provides the necessary 2D information for individual frame. The multiple tool tracking becomes possible by adapting this combined algorithm. The result of the proposed system is promising in terms of robustness and accuracy.

Another part of this thesis presents a 3D mouse interactive interface developed on top of the tracking framework. The objective of this prototype is to design an intuitive SCI using the most general and common set-up in the MIS. Our proposed system fills the demands requested by surgeons in the following aspects: (1) less device requirements, (2) more information fusion (3) man-machine interactions. The graphics system provides the surgeons with an intuitive environment to call functions including accessing a piece of patient record, an
image or 3D manual organ manipulation. However, for the preliminary studies, this system is a proof of concept. This initial prototype and phantom experimental evaluation requires further refinement for realizing clinical evaluation.

We also investigate several optical systems for 3D reconstruction and diagnostic purpose. These preliminary experiments provide an entry toward integrating such systems in the future paradigm for real-time navigation. The laser triangulation method enables simultaneous multi-spots scan, followed by the proposed real-time laser tracker that offers live cavity scan during surgical operation. Last but not least, the OCT-imaging system provides a way for real-time inner tissue imaging without harming and cutting the tissue, making the diagnosis quicker and contact-free. The above three systems are not yet integrated to the current 3D paradigm, but it proves the feasibility of such optical systems and its promising future.

7.2 Future work

The proposed 3D-mouse paradigm is a comprehensive system including the work of visual tracking and graphics display, with extra optical insight. However, it is at the preliminary stage and needs to be improved and extended in the future. The scope of future work of this thesis is listed as follows:

- **Extended SCI.** The thesis focuses on the tracking subsystem; however, a more complicated interactive system should be developed to help surgeons in the real operation. Furthermore, the multiple tracking interface can be extended, adding more functionality and possibility to the current SCI. Several ideas can be applied, such as using two tools to zoom in/out a target, grab a 3D scan or simulate a suture task.

- **In-vivo clinical trials and user studies.** To evaluate the usability of the non-contact mouse, we should test a variety of interaction modalities with professional surgeons. The current experimental studies are based on phantom environment; further studies should apply the system to in-vivo environment where more noise and complexity are involved.

- **Integrate optical system into the current framework.** The current optical system shows a promising result, but is not yet integrated to the current paradigm. Thus, in the
future, we should combine this imaging method with 3D reconstruction technologies to build a more comprehensive surgical navigation system.

- *Explore the gesture-based interface.* Gesture-based interactive surgical system can be designed to respond to user gesture, such as moving the tool to form a circle, opening or closing the tool grasp, etc. A similar system with “iPhone” gesture-based human interaction system can be applied to the current interface.

- *Extend the framework to other applications.* The whole system not only fits into medical applications, it can also be easily transported to education, military, entertainment and the game industry, which facilitates human daily life.
Bibliography


Appendix A

Introduction to endoscopy and surgical equipments

A.1 Terms related to endoscope

- **Flexible endoscopes** General term that can refer to any flexible endoscope for Industrial or Medical purposes.

- **Endoscopy** Usually used to reference Medical Endoscopes. These are often related to more outpatient type procedures rather than surgical.

- **Laparoscopes** Strictly surgical Endoscopes (includes veterinary use as well), used in various disciplines, general, gynae, urology, etc.

A.2 Types of endoscopy

Endoscopy is a broad term used to describe the inspection of the inside of the body using a flexible or rig instrument called an endoscope, with a camera and light at one end and a viewing monitor or eyepiece at the other. Examples of Endoscopy are listed in Table A.1

A.3 Optical Properties for endoscopes

- **Direction of view**
  
  Endoscopes offer various viewing directions with different angles of view. Depending
Table A.1: Types of endoscopy

<table>
<thead>
<tr>
<th>Endoscopy</th>
<th>Examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arthroscopy</td>
<td>Joint (knee, hip, wrist, shoulder, etc.)</td>
</tr>
<tr>
<td>Bronchoscopy</td>
<td>Mouth, voice box, bronchi and bronchioles</td>
</tr>
<tr>
<td>Colonoscopy</td>
<td>Rectum and colon</td>
</tr>
<tr>
<td>Cystoscopy</td>
<td>Bladder</td>
</tr>
<tr>
<td>Laparoscopy</td>
<td>Abdomen</td>
</tr>
<tr>
<td>Laryngoscopy</td>
<td>Voice box (larynx) and vocal cords</td>
</tr>
<tr>
<td>Proctoscopy</td>
<td>Anal canal</td>
</tr>
<tr>
<td>Thoracoscopy</td>
<td>Lung</td>
</tr>
</tbody>
</table>

on the optical structure, the endoscope can provide “around the corner” or even reverse viewing.

- **Angle of view (field of vision/aperture angle)**
  The angle of view is the measurement of the visible section of the image, which is given in degrees. This value is used to determine whether the endoscope used is a wide angle or standard scope.

- **Field of view**
  As opposed to microscopes, endoscopes do not have a definite magnification factor. The magnification of an endoscope depends on the distance between the front lens and the object being viewed.

### A.4 Surgical instruments

The surgical instruments include grasper, scissors, or forceps. This grasper may have single action jaw (Figure A.1(a)) or double action jaw (Figure A.1(b)). Single action jaw open less than double action jaw but close with greater force thus, most of the needle holders are single action jaw. The necessary wider opening in double action jaw is present in grasper and dissecting forceps. Single action graspers and dissectors are used where more force is required.

Scissors (Figure A.1(c)) are used to perform many tasks in open surgical procedure but its use in minimal access surgery is restricted. In minimal access surgery scissors require greater skill because in inexperienced hands scissors may cause unnecessary bleeding and damage to important structures.
Figure A.1: (a) Single action jaw graspers (b) Double action jaw graspers (c) Scissors
Appendix B

Pixel format conversion

In order to realize augmented reality (AR) interactive system for surgical application, the graphical display according to the designed interface is a crucial part. In the current setup, MIL is used for image capturing, OpenCV is used for major image processing and tracking module, with few function written in MIL. The entire graphics system is displayed using OpenGL. Thus, the final displayed buffer is OpenGL window. To let the third party libraries be compatible with each other, pixel format needs to be considered in order to render the processed image and virtual graphics.

To process the frames taken from frame grabber, the image data need to be copied from host address of the buffer rather than query it directly from OpenCV using cvCaptureFromCAM function. To access the image data, Matrox Image Library provides the following function:

\[
\text{MbufInquire(MilImage, M\_HOST\_ADDRESS, \&m\_hostaddr)}
\]

To process the image and to render the virtual scene, the libraries we used include MIL, OpenCV and OpenGL, each of which has its own pixel format. As Figure B.1 shows, MIL has a 4-channel $RGBx$ format. To convert the MIL image format to OpenCV $RGB$ format, an image converter function is written to get rid of the fourth channel of MIL image. To visualize the converted data or to access the OpenCV image data, $frame \rightarrow image\_Data$ is copied to the desired OpenCV buffer and being accessed by OpenGL buffer later on. The following is the function to copy the data buffer to a desired OpenCV image:

\[
\text{//memory copy of image data to OpenCV frame buffer.}
\]
Figure B.1: Pixel format for each of the library.

```c
memcpy(frame->imageData, data, frame->imageSize);
```

Similarly, to access the image buffer, we can do:

```c
uchar *data = (uchar*)frame->imageData;
```

Finally, we have to convert OpenCV image to OpenGL image in order to display processed images, together with virtual images in OpenGL window. Both OpenGL and OpenCV image format are RGB, however, the difference has more to do with little endian and big endian. Intel and Intel compatibles use little endian where the least significant byte (LSB) is stored first. OpenGL came from Silicon Graphics machines, which are mostly big endian, and thus the OpenGL standard required the bitmap format to be in big endian format. Therefore, to convert LSB to MSB, we have to convert the image from RGB to BGR so that it can be recognized in OpenGL. OpenCV has its own function to convert image format as the following:

```c
//To convert from RGB to GBR which is the OpenGL format
cvCvtColor(frame, glFrame, CV_RGB2BGR);
```

Now, having the data being extracted from MIL and OpenCV, the final step is to display the image in OpenGL window.

There are three ways to visualize the image in OpenGL, one is to draw each pixel on the OpenGL window, using `gldrawpixel()` call, which is very slow, especially when the number of pixels to be drawn approaches the total number of pixels on the display window. To achieve better performance, the more efficient way is to draw into a pixel buffer, then set a surface with it (each frame), so the video hardware only needs to send a single texture for each frame using `glTexImage2D()`. By texturing the image each frame than drawing each pixel, the frame rate increase dramatically. However, creating a texture may be more
computationally expensive than modifying an existing one. Instead of creating a texture for each frame, \texttt{glTexSubImage2D()} repeatedly replace the texture data with new image frame, which creates the most efficient way of displaying the video sequences. Another important reason to use \texttt{glTexSubImage2D} is that not only it is faster on many OpenGL implementations, but the target area are not necessarily required to be a power of 2. This gives the flexibility to display the image frame at its original aspect, rather than distorting or clipping each frame to fit the texture dimensions.

After the above procedures, the frames are converted to the desired format for both image processing and rendering. The same procedures take place for each frame and its image data are updated in order to achieve real time processing.
Appendix C

Introduction to Optical Coherent Tomography

C.1 Theory of Optical Coherent Tomography

Optical Coherence Tomography (OCT) is a form of low coherence interferometry. It is a medical imaging modality with resolution in the μm range and depth of imaging in the mm range. OCT uses the principle of low coherence interferometry to perform a depth resolved axial scan. OCT can be used to image various aspects of biological tissues. The most typical OCT imaging is to measure the local reflectivity of the tissue.

Although the operating mechanisms of time-domain OCT (TD-OCT) and Fourier-domain OCT (FD-OCT) systems, differ as different broadband sources and detection and signal processing schemes are employed, the basic principle is the same and can be explained with the help of the same schematic block diagram with a standard Michelson interferometer shown in Figure C.1.

Light from the broadband source is split into two arms of the Michelson interferometer using a beam splitter or a 50/50 fiber optic coupler. The light incident on the tissue undergoes partial backscattering due to the presence of discrete as well as a continuum of reflection sites at different depths within the tissue. At the output of the interferometer the backscattered light from the tissue is then recombined with the light from the reference arm and the interference signal recorded at the detection end is then used to extract axial structural information of the tissue, which can be represented as an A-scan. The beam
incident on the tissue is scanned laterally and a series of A-scans are collected and then used to obtain the cross-sectional image of the tissue.

The OCT systems can be broadly classified into:

- Time domain (TD) OCT systems
- Spectral domain (SD) OCT systems
- Swept source (SS) OCT systems

The most popular type of OCT system is FD-OCT, including SD-OCT and SS-OCT. The main advantage of FD-OCT is that it has increased sensitivity, resolution and acquisition speed. The use of a fast spectrometer or swept source can achieve high-speed acquisition without any moving parts compared to traditional TD-OCT. Further, in shot noise limit, the theoretical signal-to-noise (SNR) ratio of the SD-OCT system is independent of the spectral bandwidth of the light source. Therefore, the axial resolution of the system, which is dependent on the bandwidth of the source, could be increased without any deterioration of the SNR.

SD-OCT uses a low coherence broadband light source and a spectrometer which uses a diffraction grating to disperse the interferometric signal spectrum across an array detector such as a Charge-Coupled Device (CCD). In swept-source OCT, the broad bandwidth optical source is replaced by a rapid-scanning laser source. By rapidly sweeping the source wavelength over a broad wavelength range, and collecting all the scattering information at each wavelength and at each position, the composition of the collected signal is equivalent
to the FD-OCT technique. In any case, once the data is captured the operations of the SS-OCT are the same as in SD-OCT. Collected spectral data is then inverse Fourier transformed to recover the spatial depth-dependent information. Swept-source OCT systems are advantageous for their extremely fast scan rates, on the order of 50,000 to several MHz axial scans per second. Here the advantage lies in the proven high SNR detection technology, while swept laser sources achieve very small instantaneous bandwidths (=linewidth) at very high frequencies (20-200 kHz).

C.2 Theoretical Formulation for Signal Extraction in OCT

To understand how the interferometric signal acquired at the detector, we illustrate the acquisition of the signal at the detector. Assuming a low coherence source, a single reflector from the sample arm and ignoring most of the constants, the equation for the recombined electric field at the detector \(E_{\text{det}}\) is

\[
E_{\text{det}} = E_R e^{-j2kz_R} + E_S e^{-j2kz_S} \tag{C.1}
\]

where \(E_R\) is the electric field component from the reference arm, \(E_S\) is the electric field component from the sample arm, \(z_R\) is the length of the reference arm and \(z_S\) is the length of the sample arm. But the detector can only measure the intensity of the spectrum, so the spectral intensity detected at the detector is given by:

\[
I_{\text{det} \propto} \langle E_{\text{det}} E_{\text{det}}^* \rangle \tag{C.2}
\]

Combining Equation C.1 and C.2 and representing the intensity profile of the source as a function of \(k\), \(S(k)\), the intensity at the detector is:

\[
I_{\text{det}} \propto S(k)(I_R + I_S + \sqrt{I_R I_S} e^{-j2k(z_R-z_S)} + \sqrt{I_R I_S} e^{-j2k(z_S-z_R)}) \tag{C.3}
\]

Using the trigonometric identity of the cosine function and the assumed single reflector, the signal acquired at the detector can be represented by:

\[
I_{\text{det}}(k) = S(k)(I_R + I_S + 2\sqrt{I_R I_S} \cos(2\Delta k)) \tag{C.4}
\]

FD OCT uses a stationary reference arm reflector, and is encoded by the frequency of the interferometric fringes as a function of the source spectrum, \(S(k)\), acquiring the optical
Figure C.2: SS OCT system setup

spectrum of back-reflected light at different depths simultaneously rather than sequentially. The signal acquired at the detector is a function of the wavenumber given by Equation C.4. To extract the location of the single reflector, we take the Fourier transform of the equation and obtain:

$$F^{-1}\{I_{det}\} \propto F^{-1}\{S(k)\} \otimes ((I_R + I_S)\delta(z) + \sqrt{I_RI_S}(\delta(z + \Delta z) + \delta(z - \Delta z))) \quad (C.5)$$

where the first term on the right hand side is the Fourier transform of the source, the first term in the parenthesis is the DC term where $\Delta z = 0$ and the second term in the parenthesis represents the location of the reflector and its complex conjugate.

### C.3 Swept source OCT setup

A typical SS OCT setup (Figure C.2) consists of a wavelength tuned laser, an optical circulator, a balanced photodiode detector configured into an interferometric topology.

Light in an OCT system is broken into two arms—a sample arm (containing the item of interest) and a reference arm (usually a mirror). The combination of reflected light from the sample arm and reference light from the reference arm by the coupler gives rise to an interference pattern, but only if light from both arms have travelled the “same” optical distance (“same” meaning a difference of less than a coherence length).
In the SS-OCT, the swept source laser doesn’t provide a linear signal in $k$ space. Thus, in addition to the typical OCT setup, the Balanced Photo-diode Detector and Laser Driver Controller are added in order to re-sample the signal to a linear pattern. The photo-diode detector is responsible to output spikes, with a sample taken by A/D. This signal can be used as an external pixel clock for the ADC to sample the signal at the balanced detector evenly in wavenumber. By rapidly sweeping the source wavelength over a broad wavelength range, and collecting all the scattering information at each wavelength and at each position, the composition of the collected signal is equivalent to the FD-OCT technique.
Appendix D

Singleton Pattern & its implementation with C++

Due to the complexity of the OOP class design and the frequent sharing of the data throughout the lifetime of an application, there is a need to repeatedly use a single object of a class within different classes and files. In C++, it is possible to declare a global object, which can be used anywhere inside the program. But a good object oriented design strictly prohibits the use of global variables or methods, since they are against the fundamental principles of object orientation like data encapsulation or data hiding.

To avoid this situation, we can declare a class, which contains only static methods. The lifetime of this static class will starts from the execution of the program until it ends. However, it is not a good object oriented design either since a class of static methods unfortunately breaks down to a list of functions or utilities.

The best way to share the lifetime data/object throughout the program is to implement Singleton design pattern. The Singleton Pattern comes under that classification of Creational Pattern, which deals with the best ways to create objects. The Singleton Design pattern is used, where only one instance of an object is needed throughout the lifetime of an application. The Singleton class is instantiated at the time of first access and same instance is used thereafter till the application quits.

In our program, there are two OOP class that has been designed based on Singleton Pattern, which are CVip class and WndID class. CVip class needs to exchange data with OpenCV image processing data with OpenGL graphics data. WndID class is responsible to
share information between display subwindows.

/*
SINGLETON Pattern
*/

#include <iostream>
using namespace std;

class Singleton
{
private:
    static bool instanceFlag;
    static Singleton *single;
    Singleton()
    {
        //private constructor
    }
public:
    static Singleton* getInstance();
    ~Singleton()
    {
        instanceFlag = false;
    }
};

bool Singleton::instanceFlag = false;
Singleton* Singleton::single = NULL;
Singleton* Singleton::getInstance()
{
    if(!instanceFlag)
    {
        single = new Singleton();
        instanceFlag = true;
        return single;
    }
else
APPENDIX D. SINGLETON PATTERN & ITS IMPLEMENTATION WITH C++  122

```c
{
    return single;
}
```