3D Reconstruction of Colon Structures and Textures from Colonoscopic Videos

by Shuai Zhang

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of
Dr. Liang Zhao,
Prof. Shoudong Huang

University of Technology Sydney
Faculty of Engineering and Information Technology

September 2023

Declaration of Authorship

I, Shuai Zhang declare that this thesis, is submitted in fulfillment of the requirements for the award of Doctor of Philosophy, in the School of Mechanical and Mechatronic Engineering, Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signed: Production Note: Signature removed prior to publication.

Date: 7, September, 2023

Abstract

UNIVERSITY OF TECHNOLOGY SYDNEY

Faculty of Engineering and Information Technology
Robotics Institute

Doctor of Philosophy

by Shuai Zhang

Colonoscopy is considered the most effective method for detecting and removing precancerous polyps in the human colon. This procedure uses an endoscope to examine the internal surface of the entire colon. However, during a standard colonoscopy, it can be challenging for the endoscopist to ensure that the entire colon internal surface is inspected from the colon screening video, which can result in missed polyps and adenomas in uninspected regions. If a 3D map of the colon internal surface with detailed textures can be reconstructed during the colonoscopy procedure, the following two main potential benefits can be achieved: i) uninspected regions can be shown on this map and the endoscopist can navigate the endoscope to these missing regions to ensure more colon surfaces are inspected; ii) the detailed textures on the reconstructed map can help the endoscopist to inspect abnormalities offline.

In this dissertation, we present three works for reconstructing 3D colon maps from colonoscopic videos. Meanwhile, we introduce a colonoscopy simulator developed in Unity that can simulate the procedures of colonoscopy, different levels of colonic surface deformation, and generate synthetic colonoscopy datasets in different scenarios for the development and validation of colon reconstruction algorithms. Furthermore, to foster research in this field, the colonoscopy simulator and source code are made publicly available [1]..

---

[1] `https://drive.google.com/drive/folders/1cypaTsHpi7TRVKI5cYvzk1Ufpmdc0Ets?usp=sharing`

The first work presents a framework for 3D reconstruction of the colonic surface using stereo colonoscopic images. The input comprises a sequence of stereo colonoscopic images and a corresponding colon mesh model, which is segmented from pre-operative CT scans. The final output is the reconstructed and texturized 3D colon maps. The primary contribution of this work is fourfold: (1) Developing a visual odometry for endoscopic camera pose initialization; (2) Using the pre-operative CT-segmented colon model as a global colon map reference to increase the stability and accuracy of the endoscopic camera pose estimation; (3) Developing a joint photometric and geometric constrained scan-to-model registration algorithm for matching 3D scans (point cloud with RGB information and reconstructed from stereo images) to the pre-operative CT-segmented colon model, which can address the inconsistency of the texture matching problem; (4) Developing a barycentric-based texture rendering module for mapping textures from colonoscopic images to the reconstructed colonic surface. Simulation experimental results demonstrate the feasibility and good performance of the proposed 3D colonic surface reconstruction method in terms of accuracy and robustness.

In a clinical setting, the majority of colonoscopes used for colonoscopy procedures are equipped with a monocular camera. Meanwhile, the 3D reconstruction of colonic surface faces the problem of colon deformation. To improve the practicability of the first proposed framework, in the second work, we present a framework that can recover the 3D shape of deformable colon structures with textures from monocular colonoscopic images and a corresponding pre-operative CT-segmented colon mesh model. The novelty of the second work is threefold: (1) Using deep learning techniques to estimate dense depth for monocular colonoscopic images; (2) Developing a non-rigid registration method to address the problem of colon deformation; and (3) Developing the entire framework for the 3D reconstruction of deformable colonic surfaces with high accuracy. Validation by simulation and in-vivo experiments is conducted, and the results demonstrate the practicality of the non-rigid 3D colon reconstruction framework.

The third work is significantly differs from the previous two works, which require pair-wise photometric correspondences and dense geometric correspondences, posing a great challenge for low-textured colonoscopic images. In the third work, we formulate the textured colon reconstruction problem as a bundle adjustment (BA) problem where all the camera poses and the intensities of mesh model vertices are jointly optimized by maximizing the photometric consistency between the pre-operative CT-segmented colon mesh model and multiple views of colonoscopic images. Then, the optimized camera poses are used to render the colon map with textures from colonoscopic images. The novelty of this work is threefold: (1) Formulating simultaneous camera pose estimation as a direct BA problem, where the pre-operative model intensities and all camera poses are jointly optimized, which differs from traditional BA; (2) Directly using intensity information avoids the feature extraction and matching between 2D images in traditional BA, making the proposed method applicable to images lacking salient features, such as colonoscopic images; (3) We prove that when solving the proposed BA problem using the Gauss-Newton (GN) algorithm, the pose estimation result in each iteration of GN is independent of the model intensities in the previous iteration step, thus we propose the camera-only BA algorithm which is equivalent to the proposed direct BA algorithm but with less computational cost. The practicality and accuracy of the proposed direct camera-only BA method are validated using simulation, phantom, and in-vivo datasets.

Overall, the three frameworks proposed in this thesis represent a notable advancement in the field of 3D colonic surface reconstruction, using colonoscopic images and a pre-operative CT-segmented colon mesh model. These frameworks undergo validation through rigorous testing with simulation, phantom, and in-vivo datasets, demonstrating their feasibility, accuracy, and practicality. The clinical applications of these frameworks have the potential to enhance the accuracy and efficiency of colonic surface 3D reconstruction, thereby benefiting diagnosis, treatment planning, and surgical navigation in colonoscopy procedures.

Acknowledgements

First, I would like to express my deep appreciation to my supervisors, Dr. Liang Zhao and Prof. Shoudong Huang. Their mentorship has been incredibly valuable to me, not only because of their expert guidance in my research, but also because of their personal support. They have taught me so much, not just in terms of the technical knowledge related to my research, but also in fostering an unwavering passion for academia. Without their patience, vast knowledge, and continuous encouragement, I would not have been able to complete my Ph.D. I feel fortunate to have had the opportunity to work with them, and I hope that we can continue to collaborate in the future. Working alongside them has been an incredibly positive and rewarding experience. I am also grateful to my co-supervisor A/Prof. Hao Qi for giving me the opportunity to work on the colonoscopic project and leading me into the robotics research field.

Besides my supervisors, I would like to thank Dr. Hua Wang, Qi Luo and Kai Pan. It is a pleasure to work closely with them in achieving good publications. I am looking forward to future collaborations.

I would like to thank Dr. Raphael Guenot-Falque, A/Prof. Teresa Vidal Calleja, Dr. Alen Alempijevic, A/Prof. JaimeValls Miro, Prof. Sarath Kodagoda. Thanks a lot for their valuable suggestions on my research.

Many thanks to all my colleagues and friends at the Robotics Institute in University of Technology Sydney. My special thanks go to Yanhao Zhang, Jingwei Song, Yongbo Chen for the numerous support and encouragement, especial at the early stage of my Ph.D. study. I thank my friends Zhehua Mao, Jiaheng Zhao, Kai Pan, Tiancheng Li, Mengya Xu, Shengduo Chen and many other colleagues. Additionally, many thanks to Miao Zhang, Taoping Liu, Huan Yu, Yu He and all my other friends in Sydney. I really enjoy the time with you in Sydney.

Lastly, I want to express my gratitude to my family, particularly to my wife, AXuan Bi. She has been with me every step of the way during my PhD journey, and her constant support and motivation have been instrumental in keeping me motivated and moving forward. I also would like to thank my son, who joined us when I was writing my dissertation, for giving me unlimited happiness and pleasure. To my parents, I am forever grateful for your caring, patient and support, I am happy to make you proud of your son today.

Contents

Declaration of Authorship **i**

Abstract **ii**

Acknowledgements **v**

List of Figures **ix**

List of Tables **xii**

Nomenclature **xv**

1 Introduction **1**
- 1.1 Missed Abnormalities in Colonoscopy . . . 2
- 1.2 3D Reconstruction for Colonoscopy . . . 3
- 1.3 Challenges of Textured 3D Colon Reconstruction . . . 4
- 1.4 Brief Outline of the Developed Colonoscopy Simulator and Proposed Textured Colon Reconstruction Methods . . . 5
  - 1.4.1 Colonoscopy Simulator Development . . . 6
  - 1.4.2 A Template-based 3D Reconstruction of Colon Structures and Textures from Stereo Colonoscopic Images . . . 6
  - 1.4.3 3D Reconstruction of Deformable Colon Structures based on Preoperative Model and Deep Neural Network . . . 7
  - 1.4.4 Direct Camera-Only Bundle Adjustment for 3D Textured Colon Surface Reconstruction Based on Pre-operative Model . . . 8
- 1.5 Thesis and contributions . . . 9
- 1.6 Overview of Chapters . . . 10
- 1.7 List of Publications . . . 10

2 Background and Related Works **12**
- 2.1 3D Rigid Body Transformations . . . 12
  - 2.1.1 Rotation Matrix and the Special Orthogonal Rotation Group $\mathbb{SO}(3)$ 12
  - 2.1.2 The Lie Algebra $so(3)$ Corresponding to the Special Orthogonal Rotation Group $\mathbb{SO}(3)$ . . . 13

2.1.3 The Small Disturbance Model of Lie Algebra Derivation 14
2.1.4 The Geometry of Camera Perspective Projection 15
2.2 3D Non-rigid Body Transformations 16
2.2.1 Embedded Deformation Graph 16
2.2.2 Finite Element Method 18
2.2.3 Regularized Kelvinlet Functions 18
2.2.4 Thin Plate Splines 19
2.3 Colonoscopy Reconstruction Datasets 19
2.4 Reconstruction of Colonic Surface Maps 21
2.4.1 Shape from Shading 22
2.4.2 Structure from Motion 22
2.4.3 Combination of Shape from Shading and Structure from Motion . . 23
2.4.4 Approaches with Restrictive Assumptions 23
2.4.5 Deep Learning-based Depth Prediction 24
2.4.6 Stereo Shape Recovery 25
2.4.7 Visual SLAM Algorithms 25
2.4.8 Combination of Deep Neural Network and Visual SLAM Algorithms 26
2.5 Chapter Summary 27

3 Colonoscopy Simulator Development 29
3.1 Colon Segmentation and Mesh Optimization 32
3.2 Mesh Texture and Colon Surface Material Generation 33
3.3 3D visualization and Interaction with Unity 36
3.4 Virtual Camera Design and Configuration 37
3.5 Colon Deformation Simulation and Post Processing Effects 38
3.6 Obtanning Synthetic Datasets from the Simulator 39
3.7 Chapter Summary 44

4 A Model-based 3D Reconstruction of Colon Structures and Textures from Stereo Colonoscopic Images 46
4.1 Overview of the Framework 47
4.2 Technical Details 48
4.2.1 3D Scan Reconstruction from Stereo Images 48
4.2.2 Sparse Key Correspondences and Camera Pose Initialization 50
4.2.3 Scan to Colon Model Registration 53
4.2.4 Optimization Details 54
4.2.5 Texture Mapping using Barycentric Coordinates 57
4.3 Experiments and Results 60
4.3.1 Evaluation of RGB-D and Stereo SLAM Systems on Colonoscopic Images 61
4.3.2 Colon 3D reconstruction on Simulator-planned Camera Flight Paths 65
4.3.3 Colon 3D Reconstruction on Manually Flown Paths 69
4.3.4 Colon 3D reconstruction on fully inspected colon 70
4.3.5 In-Vivo Experiments 71

4.4 Chapter Summary 71

5 3D Reconstruction of Deformable Colon Structures based on Preoperative Model and Deep Neural Network 78

5.1 Framework Overview 79

5.2 Technical Details 80

5.2.1 Generating Ground Truth Dataset 80

5.2.2 Dense 3D Scan Reconstruction using DNN 80

5.2.3 Sparse Key Correspondences and VO Based Camera Pose Initialization 84

5.2.4 Non-rigid Registration using ED graph 85

5.2.5 Optimization Details 87

5.3 Experiments and Results 88

5.3.1 Validation using Simulation Datasets 89

5.3.2 Comparison Between Our approach and RNNSLAM on Simulation Datasets 91

5.3.3 Validation using In-vivo Datasets and Compared to RNNSLAM . . 91

5.4 Chapter Summary 93

6 Direct Camera-Only Bundle Adjustment for 3D Textured Colon Surface Reconstruction Based on Pre-operative Model 95

6.1 Methodology 96

6.1.1 Problem statement and Mathematical Formulation 96

6.1.2 Determining Visibility 98

6.2 Camera-Only BA Solution 99

6.2.1 Solving BA using Iterative GN Method 99

6.2.2 BA Sparsity and Schur Trick 100

6.2.3 Pose Only Optimization without Intensity 102

6.2.4 Pre-computaion of Gridded Intensity and Gradient Field 103

6.3 Experiments and Results 104

6.3.1 Proposed Theorem Validation 104

6.3.2 Simulation Experiments 105

6.3.3 Phantom Experiments 107

6.3.4 In-vivo Experiments 109

6.4 Chaper Summary 111

7 Conclusion and Future Work 112

7.1 Summary of Contributions 112

7.2 Future Work 113

Appendices 116

Bibliography 116

List of Figures

1.1 Colonoscopy Procedure. 2
1.2 3D Reconstruction for Colonoscopy from Colonoscopic Images. 3

3.1 The Snapshot of the Developed Colonoscopy Simulator. 30
3.2 Schematic Diagram of the Developed Colonoscopy Simulator Framework. . 31
3.3 Errors Deletion from the CT-segmented Colon Mesh. 32
3.4 Colon Mesh Errors Fixing and Further Polishment. 33
3.5 Colon Mesh Down-sampling. 33
3.6 Cutting the Mesh Into Several Parts for Better Performance in Unity. . . . 34
3.7 UV Mesh Creation for the Colon Mesh Model. 34
3.8 Baked Ambient Occlusion Into Vertex Colors. 35
3.9 Colon Model Shader and Material. 35
3.10 Vessels Texture Map Creation 36
3.11 The Snapshot of the Developed Colonoscopy Simulator Working in Stereo Mode 38
3.12 A Default Camera Flying Path 39
3.13 Visual Comparison between Simulated and Real Colonoscopic Images . . . 40
3.14 Launch the Simulator and Control the Virtual Camera 40
3.15 Examples of Colonoscopic Images Obtained from the Simulator 43

4.1 The Framework of Reconstructing and Texturing 3D Colon Structures From Stereo Colonoscopic Videos. 47
4.2 3D Scan Reconstruction from Disparity Map Through SGM. 48
4.3 Examples of reconstructed scans and ground truth. 49
4.4 VO: Sparse Key Correspondences and Camera Pose Initialization. 50
4.5 2D SIFT Matches Between Consecutive Colonoscopic Images. 51
4.6 2D SIFT Matches Between Consecutive Real Colonoscopic Images. 52
4.7 Barycentric Coordinates Based Texture Mapping. 58
4.8 Texturized Rectum Colon using Two Different Texture Rendering Approaches. 59
4.9 Trajectories Estimated From SLAM Systems on Case 8 With Normal Camera Motion. 61
4.10 Maps Estimated From SLAM Systems on Case 8 with Normal Camera Motion. 61
4.11 Comparison of the Ground Truth Trajectory and Estimated Trajectories on Case 0 with Very Slow Camera Motion. 63
4.12 Reconstructed Maps on Case 0 with Very Slow Camera Motion. 64

4.13 Estimated Trajectory and Reconstructed Map on Case 0 with Very Slow Camera Motion using the Proposed Framework. 65
4.14 Flying Trajectory and Reconstructed 3D Map on The Simulator-planned Cases. 66
4.15 3D Reconstruction Errors on the Simulator-planned Datasets Case 1. 67
4.16 3D Reconstruction Results on the Simulator-planned Case 1 to Case 6. . . . 67
4.17 Examples of Texture Region Comparison. 68
4.18 Deficient Coverage Displaying on Case 1. 68
4.19 3D Reconstruction Results on Manually Flown Case 7, 8 and 9. 69
4.20 Deficient Coverage Displaying on Case 7, Case 8 and Case 9. 70
4.21 Designed Camera Flight Trajectories to Fully Inspect the Colon. 72
4.22 3D Reconstruction Results on the Fully Inspected Colon. 73
4.23 Mean Reconstruction Errors of Case 7 to Case 15. 74
4.24 Colon Segment Models Used in the In-vivo Experiments. 74
4.25 3D Reconstruction of a Real Colon Chunk. 74

5.1 The Framework of Reconstructing Deformable 3D Colon Surface with Detailed Textures. 79
5.2 Overview of Network Architecture of DenseDepth. 81
5.3 Domain Translation Transforms Simulated Images Into Real-like Representations. 82
5.4 Simulated and Real Colonoscopic Images with Predicted Depths. 83
5.5 Mean Absolute Errors Comparison with RNN-DP. 83
5.6 Reconstructed Scan Comparison With RNN-DP. 84
5.7 Pipeline for Initializing Scan and Extracting Sparse Key Correspondences. . 85
5.8 Reconstructed Colon Maps using Our approach On Simulation Datasets. . . 90
5.9 Reconstructed Colon Maps using RNNSLAM on Smulation Datasets. 92
5.10 The Reconstruction of Colon Chunks using Our Approach and RNNSLAM. 93

6.1 Overview of the Proposed Approach. 97
6.2 Visibility Determining Methods using Barycentric Ray-triangle Intersection Technique. 99
6.3 Direct BA with Random Colors of Vertices for Validation of Pose Estimation Independent of the Intensities of Vertices. 105
6.4 Reconstructed Colon Maps on the Three Simulated Datasets using the Proposed Method, RNNSLAM, DSO and COLMAP. 106
6.5 The Comparison of Ground Truth and Estimated Trajectories on Simulated Datasets using the Proposed Method, DSO and COLMAP. 107
6.6 Reconstructed Colon Maps on the Phantom Datasets using the Proposed Method, RNNSLAM, DSO and COLMAP, respectively. 108
6.7 Reconstructed Colon Maps on the In-vivo Datasets using DSO and COLMAP, respectively. 109
6.8 Reconstructed Colon Maps on the First In-vivo Dataset using the Proposed Method and RNNSLAM, respectively. 110

6.9 Reconstructed Colon Maps on the Second In-vivo Dataset using the Proposed Method and RNNSLAM, respectively. 110

List of Tables

2.1 Comparison of Colonoscopy Reconstruction Datasets 20

4.1 A Brief Summary of Data for Evaluating the Proposed Framework 60

6.1 Pose Evaluation Errors (mm) on the Simulated Colonoscopic Sequences. . . 107

Acronyms & Abbreviations

CT	Computed Tomography
MRI	Magnetic Resonance Imaging
SLAM	Simultaneous Localisation and Mapping
SfS	Shape from Shading
SfM	Shape from Motion
BA	Bundle Adjustment
SGM	Semi-Global Matching
RANSAC	Random Sample Consensus
P3P	Perspective-Three-Point
ED	Embedded Deformation
GN	Gauss-Newton
CNN	Convolutional Neural Network
DNN	Deep Neural Network
GAN	General Adversarial Network
SSIM	Structural Similarity
ICP	Iterative Closest Point
GPU	Graphics Processing Unit

SIFT	Scale-invariant Feature Transform
VO	Visual Odometry
UI	User Interface
FSM	Finite State Machine
FOV	Field of View

Nomenclature

General Notations

$\mathbb{R}^n$	The n-dimensional Euclidean space
$\mathbb{SO}(3)$	The special orthogonal group
$so(3)$	The Lie algebra corresponding correponding to $\mathbb{SO}(3)$
$\pi(\cdot)$	The camera projection function
E_G	The geometric term to solve scan to model rigid registration
E_F	The photometric term to solve scan to model rigid registration
$\mathbf{I}_n \in \mathbb{R}^{n \times n}$	The identity matrix
$\lVert \cdot \rVert$	Euclidean norm of a vector
$\mathbf{g}_j \in \mathbb{R}^3$	The position of ED node j
$\mathbf{A}_j \in \mathbb{R}^{3 \times 3}$	The affine matrix of ED node j
$\mathbf{t}_j \in \mathbb{R}^3$	The translation of ED node j
$\mathbf{P} \in \mathbb{R}^3$	The position of a 3D Point
$\phi(\cdot)$	The deformation function of ED graph
$w_j(\boldsymbol{v})$	The weight quantifying the influence of node j to a point $\boldsymbol{v}$
$\mathbb{N}(j)$	The set of all neighboring nodes to ED node j
E_{rot}	The rotation term to solve ED graph
E_{reg}	The regularisation term to solve ED graph
E_{con}	The constraint term to solve ED graph
E_{geo}	The geometric term to solve scan to model nonrigid registration
E_{pho}	The photometric term to solve scan to model nonrigid registration
E_{r}	The rigid rotation term measured by the variations of the rigid rotation R

E_{t}	The rigid translation term measured by the variations of the rigid translation $\boldsymbol{T}$
$\mathbf{v}_i$	The 3D position of vertex i from a polygonal mesh
$\widetilde{\mathbf{v}}_i$	The estimated new position of a vertex
$\mathrm{n}(\cdot)$	3D normal vector of a point.

Chapter 1

Introduction

Colorectal cancer is the second most commonly occurring cancer in women and the third most commonly occurring cancer in men all over the world. Colonoscopy is considered as the gold-standard method to detect changes and remove precancerous polyps in the large intestine (colon). During a standard colonoscopy procedure (Fig .1.1), a long, thin and flexible tube called colonoscope is inserted into the rectum, and a tiny video camera at the tip of the tube allows the endoscopist to view the inside of the entire colon and capture images inside. If suspected colorectal cancer lesions such as polyps are found, a snare device can be placed around a polyp for removal.

However, colonoscopy is not perfect, the flexures and colonic folds where polyps and adenomas hidden are not fully visualized during a standard forward-viewing colonoscopy. A good solution is to reconstruct the colonoscopic 2D images into a dense 3D textured colon map with displaying of unvisualized regions. Meanwhile, the reconstructed map can help the endoscopist to navigate the endoscope to cover the unseen surface and textures on the reconstructed map can further help the endoscopist to identify polyps and adenomas. Thus, the research in this thesis focuses on reconstructing a 3D map of the colon internal surface with detailed textures using colonoscopic videos of colonoscopy procedures.

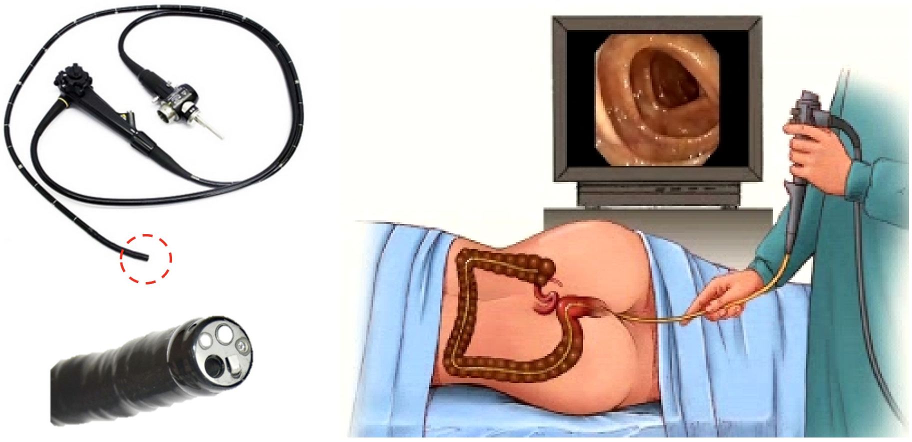


FIGURE 1.1: **An Example of the Colonoscope and Illustration of Standard Colonoscopy**. The left figure shows one Olympus colonoscope and the right figure shows the procedue of a standard optical colonoscopy [1]. During the procedure, the flexible colonoscope is inserted through the patient's rectum, and it is then carefully advanced through the entire length of the colon. The colonoscope is equipped with a camera at its tip, which allows the healthcare provider to view the colon's lining on a monitor in real-time.

1.1 Missed Abnormalities in Colonoscopy

Recent studies report that around 20% of the abnormalities (polyps, abnormal lesions and cancer) are missed [2, 3] and approximately 60% of colorectal cancer cases detected after optical colonoscopy are closely associated with missed polyps and lesions [4]. There are two main reasons for missed abnormalities: i) the areas where abnormalities reside are never detected by the colonoscopy; ii) these areas are inspected but the abnormalities are not recognized.

The human colon has long and narrow tubular structure with many colon folds and a lot of turns, which makes it difficult to observe the back side of colon folds during a colonoscopy screening. Thus, non-visualization results from the lack of getting around a curvature of the endoscope to the full circumference of parts of the colon [5] and the occlusion from the structural complexity of colon [6]. Non-recognition is due to the difficulty to detect abnormalities from video alone.

Although virtual colonoscopy is a non-invasive, radiographic method of visualizing the colon by flying through the segmented colon model, it has difficulty in detecting $5mm$ or less size lesions and flat lesions and meanwhile the patient will be exposed to a certain dose of radiation [7]. Furthermore, sometimes the standard optical colonoscopy will be ultimately needed to detect very small and flat colon lesions and remove polyps or any abnormalities identified from virtual colonoscopy.

1.2 3D Reconstruction for Colonoscopy

The conventional practice of colonoscopy involves the deployment of colonoscopes equipped with monocular miniature fish-eye cameras to capture images of colonic surfaces. However, due to the absence of direct depth information, endoscopists must rely on indirect cues, such as shading and motion parallax, to extrapolate the 3D configuration of the colon under examination. This approach demands significant training to become proficient and may contribute to clinician fatigue, reduced efficiency, and diminished accuracy. As a consequence, scholars have pursued the development of various 3D imaging technologies capable of recovering colon structures to enhance sensitivity, lesion resection, training, and automated lesion detection.

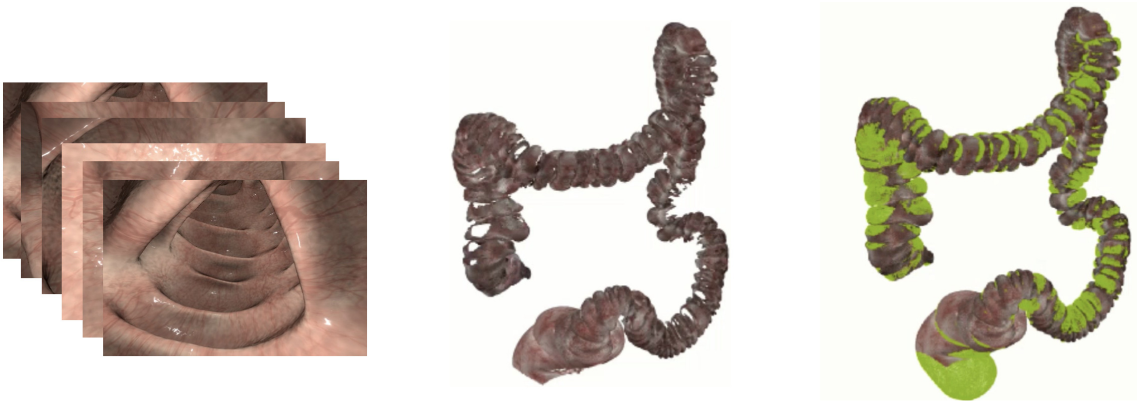


Figure 1.2: **3D Reconstruction for Colonoscopy**. The left shows colonoscopic images, the middle shows reconstructed the 3D colon map and the right shows the uninspected regions (in green color) on the colon map.

Normally, to reconstruct a 3D colon map from a sequence of 2D colonoscopic images, the sparse or dense 3D scans (point clouds with RGB information) should be obtained, and the relative frame poses would be optimized from initial values and used to register the scans to form a relatively large 3D scan in an incrementally or globally way. To achieve this goal, we have developed novel frameworks to reconstruct the 3D colon map from colonoscopic images and show unsurveyed regions on the reconstructed map. Fig 1.2 illustrates the research problem and our research aims.

1.3 Challenges of Textured 3D Colon Reconstruction

Reconstructing 3D dense colon maps from a sequence of colonic images has to deal with the following technical challenges:

- Special geometric structure. The human colon has long and narrow tubular structure with many colon folds and a lot of turns, which make it impossible to have large loop closures and difficult to observe the back side of colon folds during a colonoscopy screening. This is the main reason for deficient coverage in a normal colonoscopy;
- Camera motion estimation. During a standard colonoscopy, the tiny camera attached to the end of a colonsocope moves fast with significant view changes, which results in less overlaps between consecutive frames. Furthermore, the tubular environment of colon makes it impossible to have large loop closures, and this causes large drift in the camera motion estimation. How to improve the accuracy and robustness of camera motion estimation becomes critical;
- Reconstruction with detailed textures. Texture information on the reconstructed colon map is critical for the endoscopist to recognize polyps and adenomas. Accurate texture matching requires high accurate camera pose estimation, high accurate depth estimation and high accurate scan registration which is very difficult to achieve using information from images only;
- Depth estimation. Depth information is critical to reconstruct the 3D colon map, however it is difficult to attach a depth sensor to an endoscope. In addition, too close the distance between the colonoscope and the colon surface, complexities in tissue textures and less inter-frames overlapping, all make it difficult to predict the depth information using traditional computer vision techniques;

- Colon deformation. Practically the colon is deformable and does not hold a constant shape over long time periods. The deformation is caused by both physiological motion (peristaltic motion) and physical contact between the flexible colonoscope and the tissues. The shape of the shaft (body) of the colonoscope will deform the topological shape of the human colon and the bending of the colonoscope tip (distal end) will cause local deformation of its surrounding colonic surface. All kinds of deformations make the camera motion estimation and colon shape reconstruction challenging;

- Colonoscopy datasets with ground truth. The colonoscopic images with ground truth of depths and camera poses are critical to develop and validate the effectiveness of colon reconstruction algorithms. However, this is impractical to obtain in standard colonoscopy procedures.

1.4 Brief Outline of the Developed Colonoscopy Simulator and Proposed Textured Colon Reconstruction Methods

With all the aforementioned challenges in mind, in this dissertation we developed one colonoscopy simulator and proposed three frameworks for 3D textured colon map reconstruction from endoscopic videos based on a pre-operative model.

Due to the limited overlaps between consecutive frames and the nonexistence of large loop closures during a normal screening colonoscopy, the state-of-the-art simultaneous localization and mapping (SLAM) algorithms cannot be directly applied to the 3D reconstruction of colon. Thus, in our proposed three frameworks, the colon mesh model segmented from computed tomography (CT) scans is used together with the colonoscopic images to achieve the colon 3D reconstruction with high accuracy. The pre-operative colon mesh model is mainly used to reduce the camera pose estimation drift and improve the colon map reconstruction accuracy with the consistency of textures matching. We applied the state-of-the-art SLAM-based algorithms to reconstruct colon map without using a pre-operative model, the reconstructed map suffers from large drift and the textures on it are mismatched between consecutive scans, as seen in Section 4.3 of Chapter 4. One case of the advantage is that our proposed three frameworks have the potential to provide similar functions of a CT without exposing the patient to radiation. This is advantageous because virtual colonoscopy, a non-invasive radiographic method for visualizing the colon, has difficulty detecting small lesions and flat lesions and exposes the patient to radiation. With the proposed methods, the reconstructed colon map can display not only the structures of the patient's colon, such as polyps, but also texture information such as lesion regions.

1.4.1 Colonoscopy Simulator Development

To develop algorithms for recovering the 3D structures of the human colon in colonoscopy procedures or to train depth prediction networks for depth estimation of colonoscopic images, both synthetic and real clinical data are crucial. However, due to reasons of patient privacy, human and animal rights, guarantee of operation safety and conflicts of interest. There are hardly any public dataset with complete or segmental colonoscopic images with or without corresponding ground truth depth and camera poses. Therefore, we developed a realistic simulator to simulate colonoscopy procedures and generate complete colonoscopic images with ground truth dense depths and camera poses.

1.4.2 A Template-based 3D Reconstruction of Colon Structures and Textures from Stereo Colonoscopic Images

In this work, we aim to develop a pre-operative colon CT model based SLAM framework fusing stereo colonoscopic RGB images to recover a complete 3D map of the colon with detailed textures. Firstly, the corresponding depth of a monocular RGB frame is estimated from stereo matching on the pair of stereo images. Secondly, Scale-invariant Feature Transform (SIFT) features are used for matching between consecutive frames and then are lifted into 3D space for establishing sparse key correspondences between scans and the pre-operative colon model. Thirdly, Iterative-closest Points (ICP) algorithm is used for matching scans and the colon model for building dense correspondences. Fourth, based on the two sets of correspondences, the proposed joint photometric and geometric optimization pipeline of the framework is used to optimize the camera poses to address the inconsistency of texture matching problem. Last, using the estimated camera poses, the point correspondences between the reconstructed scans and the colon mesh model are extracted and used to map textures from the corresponding monocular image to the registered areas on the colon model.

The proposed framework mainly includes 3D scan reconstruction from stereo images, an visual odometry (VO)-based camera pose initialization module, a joint geometric and photometric registration scheme for matching textured scans to the segmented colon model, and a barycentric-based texture rendering module for mapping textures from colonoscopic images onto the reconstructed colonic surface. The developed realistic simulator is used to simulate the procedures of colonoscopy and to provide experimental datasets in different scenarios. Experimental results demonstrate the good performance of the proposed 3D colonic surface reconstruction method in terms of accuracy and robustness.

1.4.3 3D Reconstruction of Deformable Colon Structures based on Preoperative Model and Deep Neural Network

Due to the deformation of the colon in standard forward-viewing colonoscopies and most existing colonoscopes still use single-lens cameras, the proposed framework in Section 1.4.2 works poorly for the 3D reconstruction of deformable colon surfaces and is prone to severe drift. To improve the potential clinical value of the proposed first framework, the synthetic datasets generated using the developed colonoscopy simulator are utilized to train a supervised deep neural network for dense depth estimation of monocular colonoscopic images. Also, a generative adversarial network is used to transform the real colonoscopic images into their synthetic-like representations for more accurate depth estimation. Then, an embedded deformation-based non-rigid registration algorithm is proposed to transform and deform the 3D scans to the CT-segmented colon mesh model.

The proposed framework includes dense depth estimation from monocular colonoscopic images using a deep neural network (DNN), visual odometry (VO) based camera motion estimation and an embedded deformation (ED) graph based non-rigid registration algorithm for deforming 3D scans to the segmented colon model. The function of simulating different levels of colon deformation is developed and integrated into the realistic simulator. Simulation results demonstrate the good performance of the proposed 3D colonic deformable surface reconstruction method in terms of accuracy and robustness. In-vivo experiments are also conducted and the results show the practicality of the proposed framework for providing useful shape and texture information in colonoscopy applications.

1.4.4 Direct Camera-Only Bundle Adjustment for 3D Textured Colon Surface Reconstruction Based on Pre-operative Model

In both the previous proposed two frameworks, the data association operation (SIFT feature extraction and matching, sparse and dense correspondences establishment) is very computational and sometimes work poorly for some colonoscopic images with less texture.

The third work relies on maximizing the photometric consistency between the pre-operative colon model and multiple views of monocular colonoscopic images to optimize the camera motion parameters and the intensity of the pre-operative model vertices. Although the intensity of the pre-operative model vertices and all the colonoscopic frame poses are optimized together in the mathematical BA formulation of this problem, we prove that the optimization using the iterative Gauss-Newton (GN) method has the merit of optimizing camera pose only without optimizing the intensity of model vertices, which helps reduce the computational cost of the proposed algorithm. Thus, the direct camera-only BA algorithm is proposed and used to the scenario of 3D textured colon reconstruction from low-texture 2D colonoscopic images.

Specifically, we first estimate all the camera poses using the proposed camera-only BA algorithm. Then, we can obtain the intensities of mesh vertices by a closed-form formula. The optimal RGB colors of vertices can also be calculated by the closed-form formula using different channel of color images separately and used for texture rendering of the pre-operative colon model. The textured regions on the colon model are actually the visible maps viewed by all the frames. Meanwhile, we propose a method to automatically and accurately determine the 3D vertices' visibility from meshes under camera views. Furthermore, we pre-compute gridded intensity and gradient field for all the images to improve the efficiency and accuracy of the proposed camera-only BA algorithm. Validation using simulation, phantom and in-vivo datasets is performed to demonstrate the accuracy and feasibility of the proposed algorithm.

1.5 Thesis and contributions

The main contribution of this thesis is the three proposed frameworks for colon 3D reconstruction:

- Developing a template-based framework for 3D reconstruction of colon structures and textures from stereo colonoscopic RGB images, which mainly addresses the first three challenges listed above.
- Developing a framework for 3D reconstruction of deformable colonic surfaces from monocular colonoscopic images, which mainly addresses the 4th and 5th challenges listed above.
- Developing a direct camera-only BA framework for textured 3D colonic surface reconstruction, which optimizes all the frame poses simultaneously without requiring data association and image depth information.

Besides the above methodological contributions, I have also accomplished the following engineering contributions:

- A realistic colonoscopy simulator based on the colon model segmented from preoperative CT scans and the virtual reality platform Unity is used to provide colonoscopy datasets with ground truth of depths and camera poses, which addresses the 6th challenge listed above.
- A barycentric based texture rendering technique is used to map textures from colonoscopic images to the reconstructed colonic surface.
- The generated synthetic datasets are utilized to train a supervised deep neural network (DNN) for dense depth estimation of monocular colonoscopic images. Also a generative adversarial network (GAN) is used to transform the real colonoscopic images into their synthetic-like representations for depth estimation.
- An ED based non-rigid registration algorithm is proposed to transform and deform the 3D scans to the segmented colon mesh model, where the model is mainly used as

a global reference to increase the robustness and accuracy of the registration. In the non-rigid registration process, we use 2D SIFT-based algorithm to provide a set of pair-wise registering key points which can greatly overcome the texture misalignment caused by the deformation and smoothness of the colonic surface.

- Automatically determining visibility when the camera views are changing using barycentric ray-triangle intersection technique, as many scene points quickly go out of view, or become occluded.

1.6 Overview of Chapters

The rest of this dissertation is organized in the following chapters: Chapter 2 reviews mathematical backgrounds and related works for colon map reconstruction. Chapter 3 gives a brief overview of the process of developing the colonoscopy simulator. Chapter 4 describes the technical details and experimental results of our first framework. Chapter 5 describes the technical details and experimental results of our second framework. Chapter 6 describes the technical details and experimental results of our third framework. Chapter 7 concludes this thesis and outline our future work.

1.7 List of Publications

The first framework presented in Chapter 4 was published in *IEEE Transaction on Medical Robotics and Bionics.* The improved framework shown in Chapter 4 was published in *2021 IEEE Conference on Robotics and Automation.* The third framework presented in Chapter 6 is in prepration for submission to *IEEE Robotics and Automation Letters (RA-L).* Inspired by the colon reconstruction frameworks, we developed SLAM algorithms for precise and real-time intra-operative evaluation of the proximal tibial resection plane in conventional total knee replacement surgery. This work was published in *2022 International Conference on Medical Image Computing and Computer-assisted Intervention.* We are in the process of preparing an improved version of this paper for submission to *International Journal of Robotics Research (IJRR).*

The list of all papers is as follows[1]:

1. **Zhang, S.**, Zhao, L., Huang, S., Ye, M., Hao, Q. (2020). A Template-Based 3D Reconstruction of Colon Structures and Textures From Stereo Colonoscopic Images, *IEEE Transactions on Medical Robotics and Bionics (TMRB)*, 3(1), pp. 85–95.
2. **Zhang, S.**, Zhao, L., Huang, S., Ma, R., Hu, B., Hao, Q. (2021). Reconstruction of Deformable Colon Structures based on Preoperative Model and Deep Neural Network, *in proceedings of 2021 IEEE International Conference on Robotics and Automation (ICRA).* Springer, pp. 1875–1881.
3. **Zhang, S.**, Zhao, L., Huang, S., Wang, H., Luo, Q., Hao, Q. (2022). SLAM-TKA: Real-time intra-operative measurement of tibial resection plane in conventional total knee arthroplasty, *in proceedings of 25th International Conference on Medical Image Computing and Computer-assisted Intervention (MICCAI).* Springer, pp. 126–135.
4. Pan, K., **Zhang, S*.**, Zhao, L., Huang, S., Zhang, Y., Wang, H., Luo, Q. (2023). 3D Reconstruction of tibia and fibula using one general model and two X-ray images, *in proceedings of 2023 IEEE International Conference on Robotics and Automation (ICRA).* Springer, pp. 4732–4738.
5. **Zhang, S.**, Zhao, L., Huang, S., Hao, Q. (2023). Direct Camera-Only Bundle Adjustment for 3D Textured Colon Surface Reconstruction Based on Pre-operative Model, *in preparation (to be submitted to IEEE Robotics and Automation Letters (RA-L)).*
6. **Zhang, S.**, Zhao, L., Huang, S., Wang, H., Luo, Q., Hao, Q. (2023). SLAM-TKA: Simultaneous localising X-ray device and mapping contours of tibia and pins in conventional Total Knee Arthroplasty, *in preparation (to be submitted to International Journal of Robotics Research (IJRR).*

---

[1] It is noted that although the 3rd, 4th and 6th publications are for different SLAM problems, some techniques, e.g., the optimization method, are related to our thesis. $*$ The first two authors have equal contributions.

Chapter 2

Background and Related Works

This chapter presents the technical background and literature review related to the dissertation's focus on colon reconstructions. In Section 2.1, we provide an introduction to the technical aspects of rigid body motion in 3D space, which includes the geometry of perspective projection, the rotation matrix, translation vector, the relationship between Lie group and Lie algebra, the derivation model of Lie algebra, and the camera perspective projection of pin-hole model. Additionally, Section 2.2 describes some commonly used 3D non-rigid body transformation methods and mainly introduces the ED graph which is used to deal with the colonic surface deformation challenge in our work. In Section 2.4, we review relevant literature on colon reconstructions.

2.1 3D Rigid Body Transformations

2.1.1 Rotation Matrix and the Special Orthogonal Rotation Group $\mathbb{SO}(3)$

In our proposed 3D colon reconstruction frameworks, one main goal is to estimate the optimal pose of each 3D colonoscopic scan w.r.t. the coordinate space of a pre-opertive colon model. Here, the pose $[R, \mathbf{t}]$ is composed of a rotation matrix R and a translation vector $\mathbf{t} \in \mathbb{R}^3$, and they are used to describe the change of orientation and position of the 3D scan in the local frame space w.r.t. the coordinate frame of the pre-opertive colon model, respectively. Where the rotation matrix R is a member of the special orthogonal rotation group $\mathbb{SO}(3)$:

$$\mathbb{SO}(3) = \{R \in \mathbb{R}^{3\times 3} | RR^T = \mathbf{I}_{3\times 3}, det(R) = 1\} \tag{2.1}$$

Then, the estimated optimal frame pose is used to transform the reconstructed 3D scan to the coordinate frame of the pre-operative colon model as following:

$$\mathbf{P} = R^T \mathbf{P_C} - R^T \mathbf{t} \tag{2.2}$$

where $\mathbf{P}_C = [x^C, y^C, z^C]^T$ represents one 3D point of the 3D scan in its local camera space and $\mathbf{P}$ represents the transformed point into the pre-operaitve colon model space.

2.1.2 The Lie Algebra $so(3)$ Corresponding to the Special Orthogonal Rotation Group $\mathbb{SO}(3)$

Typically, the task of estimating frame poses is expressed mathematically as a non-linear square problem. Various optimization techniques, such as the GN algorithm and the Levenberg–Marquardt algorithm [8], are commonly utilized to determine the optimal frame poses by iteratively reducing the errors associated with the non-linear least squares problem. During each iteration, the optimization solver linearizes the problem at the current frame pose state to calculate the step change required to update the frame poses. However, directly adding the step change to the current frame poses is not feasible due to the additional constraints on the rotation matrices used as optimization variables. Specifically, rotation matrices must be orthogonal and possess a determinant of 1, which means that adding two rotation matrices no longer falls within the rotation group $\mathbb{SO}(3)$, and derivatives cannot be expressed in the form of a special orthogonal rotation group. However, by transforming the problem from the special orthogonal group $\mathbb{SO}(3)$ to its Lie algebra $so(3)$, it can convert the pose estimation problem into an unconstrained optimization problem.

In practice, each special orthogonal rotation matrix R corresponds to a unique vector $\phi = [\phi(1), \phi(2), \phi(3)]^T$ defined on $\mathbb{R}^3$:

$$R = exp(\phi^\wedge) \tag{2.3}$$

where the operator $\wedge$ is a skew-symmetric symbol that turns the vector ϕ into a unique anti-symmetric matrix as:

$$\phi^\wedge = \begin{bmatrix} 0 & -\phi(3) & \phi(2) \\ \phi(3) & 0 & -\phi(1) \\ -\phi(2) & \phi(1) & 0 \end{bmatrix} \tag{2.4}$$

Thus, the general definition of Lie algebra $so(3)$ is as following:

$$so(3) = \{\phi \in \mathbb{R}^3, \phi^\wedge \in \mathbb{R}^{3\times 3}\} \tag{2.5}$$

Meanwhile, the camera pose $[R, \mathbf{t}]$ can be represented by a six dimensional vector:

$$\xi = [\phi, \mathbf{t}] \tag{2.6}$$

To convert the vector $\xi = [\phi, \mathbf{t}]$ into a camera pose, the camera rotation matrix is an exponential map of $\phi^\wedge$ by using (2.3) and the translation $\mathbf{t}$ is still the same.

2.1.3 The Small Disturbance Model of Lie Algebra Derivation

In our work, the small left disturbance model of Lie algebra derivation is used to compute the derivation of a rotation matrix in the pose estimation problem, thus to turn the $\mathbb{SO}(3)$ property constrained optimization problem into an unconstrained optimization problem.

Suppose $\mathbf{P}$ is a 3D point in the world space and we use the rotation matrix R to rotate it and obtain the rotated point $R\mathbf{P}$. To calculate the derivative of the rotated point $R\mathbf{P}$ by the rotation matrix R, we multiply a small turbulance ΔR on the left of $R\mathbf{P}$ and define its Lie algebra as $\varphi(R)$. Then, taking the limit of the multiplication result relative to the small disturbance $\varphi(R)$ to compute the derivative as:

$$\begin{aligned}\frac{\partial R\mathbf{P}}{\partial\varphi(R)} &= \lim_{\varphi(R)\to 0}\frac{exp(\varphi(R)^{\wedge})exp(\phi^{\wedge})\mathbf{P} - exp(\phi^{\wedge})\mathbf{P}}{\varphi(R)} \\ &\approx \lim_{\varphi(R)\to 0}\frac{(1+\varphi(R)^{\wedge})exp(\phi^{\wedge})\mathbf{P} - exp(\phi^{\wedge})\mathbf{P}}{\varphi(R)} \\ &= \lim_{\varphi(R)\to 0}\frac{\varphi(R)^{\wedge}R\mathbf{P}}{\varphi(R)} = \lim_{\varphi(R)\to 0}\frac{-(R\mathbf{P})^{\wedge}\varphi(R)}{\varphi(R)} = -(R\mathbf{P})^{\wedge}\end{aligned} \tag{2.7}$$

2.1.4 The Geometry of Camera Perspective Projection

The pinhole camera model is used to describe the process of perspectively projecting a 3D scene point to a 2D image pixel plane. Suppose $\mathbf{p} = [u, v]^T$ is the ground truth coordinates of one observed 2D feature point $\bar{\mathbf{p}}$, its corresponding 3D point from the scene is $\mathbf{P} \in \mathbb{R}^3$, and the camera pose is $[R, \mathbf{t}]$. Then, the geometry of perspective observation model $\mathbf{p} = \pi(\mathbf{P}, \xi)$ of the feature point can be written as:

$$\begin{aligned}\bar{\mathbf{p}} &= \mathbf{p} + w \\ [\mathbf{p}^T, 1]^T &= \frac{1}{z_C}K\mathbf{P}_C \\ \mathbf{P}_C = [x_C, y_C, z_C]^T &= R\mathbf{P} + \mathbf{t}\end{aligned} \tag{2.8}$$

where w is the zero-mean Gaussian noise with covariance matrix Σ_p, and

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

is the camera intrinsic matrix, R and $\mathbf{t}$ represent the rotation matrix and translation vector of the camera pose w.r.t. the world coordinate system, respectively.

Thus, the derivative of observation point $\bar{\mathbf{p}}$ w.r.t. the camera pose is calculated as:

$$\frac{\partial\bar{\mathbf{p}}}{\partial\delta_\xi} = \frac{\partial\mathbf{p}}{\partial\mathbf{P}_C}\frac{\partial\mathbf{P}_C}{\partial\left[\delta_\phi, \delta_\mathbf{t}\right]^T} = \begin{bmatrix} \frac{f_y}{z_C} & 0 & -f_x\frac{x_C}{z_C^2} \\ 0 & \frac{f_x}{z_C} & -f_y\frac{y_C}{z_C^2} \end{bmatrix}\left[(-R\mathbf{P})^{\wedge}, \mathbf{I}_3\right] \tag{2.10}$$

2.2 3D Non-rigid Body Transformations

3D non-rigid body transformations refer to the deformation or change in shape of a 3D object or scene without changing its topology. In medical image registration, non-rigid body transformations are used to align images or 3D scans of the same patient acquired at different times or with different imaging modalities such as pre-operative models segmented from CT or MRI scans. In this section, we especially introduce the technical details of ED graph method which is used in our second proposed colon reconstruction work for dealing with the colonic surface deformation challenge. Similar to ED graph, some other commonly used non-rigid transformation methods including embedded deformation (ED) graph [9], finite element method [10], regularized kelvinlet functions [11], and thin plate splines [12] are briefly reviewed.

2.2.1 Embedded Deformation Graph

Our work for recovering 3D colonic surface deformation is based on ED graph, which is usually used for modeling the deformation of surfaces. ED graph represents a surface as a graph, which consists of ED nodes corresponding to sampled vertices or points on the surface, and edges denote connections between these nodes. Thus, the nodes connected by edges define a deformation skeleton. By manipulating the positions of the ED nodes, it can non-rigidly deform the original surface to another realistic and smooth surface.

Specically, each ED node is associated with a position $\mathbf{g}_j \in \mathbb{R}^3$, an affine matrix $A_j \in \mathbb{R}^{3\times3}$ and a translation vector $\mathbf{t}_j \in \mathbb{R}^3$. Given the parameter value for all the ED nodes, for each vertex $\mathbf{v}$ in the 3D surface space, it will be influenced by a set of neighbouring ED nodes in the ED graph, and the deformed position $\widetilde{\mathbf{v}}$ of the vertex $\mathbf{v}$ is given by (2.11):

$$\widetilde{\mathbf{v}} = \phi(\mathbf{v}) = \sum_{j=1}^{m} w_j(\mathbf{v})[A_j(\mathbf{v} - \mathbf{g}_j) + \mathbf{g}_j + \mathbf{t}_j] \tag{2.11}$$

where m denotes the number of neighboring ED nodes, $w_j(\mathbf{v})$ is the weight for vertex $\mathbf{v}$ and defined as:

$$w_j(\mathbf{v}_i) = (1 - \left\|\mathbf{v} - \mathbf{g}_j\right\| / d_{max}) \tag{2.12}$$

where d_{max} is the Euclidean distance of the vertex to the $m+1$ nearest ED nodes. Thus, the deformed vertex positions are a function of the ED deformation parameters.

Usually, we have the original and deformed vertex positions, and the ED affine transformations are unknown. Then, these vertices are used as the positional constraints for an optimization problem to estimate the deformation parameters:

$$E_{con} = \sum_{l=1}^{q} \|\widetilde{\mathbf{v}}_l - \mathbf{v}_l\|_2^2 \tag{2.13}$$

where $\mathbf{v}_l$ is the l-th original vertex and $\widetilde{\mathbf{v}}_l$ is deformed by the ED graph accroding to (2.11).

The deformation parameters are estimated by minimizing the following energy function:

$$\min_{A_1,\mathbf{t_1}\ldots A_k,\mathbf{t_k}} w_{rot}E_{rot} + w_{reg}E_{reg} + w_{con}E_{con} \tag{2.14}$$

where k is the number of ED nodes. The energy function has three components: rotation term, regularization term and the constraints term. The first and second terms are functions only defined over the ED graph, and the third term is enforced by the positional constraints.

The rotation term is for making the affine matrices close to rotations. E_{rot} sums all the rotation error:

$$E_{rot} = \sum_{j=1}^{M} Rot(A_j) \tag{2.15}$$

$$\begin{gathered} Rot(A_j) = (\mathbf{c_1}\cdot\mathbf{c_2})^2 + (\mathbf{c_1}\cdot\mathbf{c_3})^2 + (\mathbf{c_2}\cdot\mathbf{c_3})^2 + \\ (\mathbf{c_1}\cdot\mathbf{c_1}-1)^2 + (\mathbf{c_2}\cdot\mathbf{c_3}-1)^2 + (\mathbf{c_3}\cdot\mathbf{c_1}-1)^2 \end{gathered} \tag{2.16}$$

where $\mathbf{c_1}$, $\mathbf{c_2}$ and $\mathbf{c_3}$ are the column vectors of each affine matrix A_j, M is the number of ED nodes.

The regularization term is used to ensure a smooth deformation and prevent divergence of the neighbouring nodes:

$$E_{reg} = \sum_{j=1}^{M} \sum_{m\in\mathbb{N}(j)} \alpha_{jm} \left\| A_j(\mathbf{g}_m - \mathbf{g}_j) + \mathbf{g}_j + \mathbf{t}_j - (\mathbf{g}_m + \mathbf{t}_m)\right\|_2^2 \tag{2.17}$$

where α_{jm} is the weight computed by the Euclidean distance of the two ED nodes and it is set to 1 referring to [9]. $\mathbb{N}(j)$ is the set of all neighboring nodes to node j.

2.2.2 Finite Element Method

Finite element method (FEM) is a numerical method that can be used for non-rigid transformation of 3D objects with non-linear material properties. In FEM-based non-rigid transformation, a 3D object is discretized into small elements, and each element is represented by a set of nodes. These nodes can be moved in response to external forces or constraints, causing the element to deform. The deformation of each element is computed using a set of equations that describe the behavior of the material.

FEM has been used in the registration of brain MRI images [13, 14], lung CT images [15], and many other medical image registration applications. The advantages of FEM include its ability to model complex deformation behaviors and its compatibility with a wide range of image modalities. However, FEM is computationally expensive and requires high computational resources [16].

2.2.3 Regularized Kelvinlet Functions

Regularized Kelvinlet functions (RKF) [11] are a type of non-rigid method used for simulating the deformation of objects by displacing point sources within them. The technique is based on the concept of Kelvinlet functions, which are mathematical functions used to describe the deformation caused by a point force acting on an elastic material. In RKF, a regularization term is added to the Kelvinlet functions to ensure the smoothness of the deformation. To compute the deformation field using RKF, a set of control points is first selected on the object. For each control point, a Kelvinlet function is computed based on its position and a set of user-defined parameters that determine the deformation behavior. Finally, the Kelvinlet functions are combined to produce the overall deformation field.

RKF have been used in medical image registration applications, such as the registration of brain MRI images and liver CT images [11]. The advantages of regularized RKF include its computational efficiency and its ability to handle large deformations. However, it may not accurately capture complex deformation behaviors. Overall, RKF is a useful and widely-used technique for 3D non-rigid transformation that offers a good balance between computational efficiency and deformation quality.

2.2.4 Thin Plate Splines

Thin plate splines (TPS) is a widely used technique for 3D non-rigid deformation that is commonly used in medical imaging [14, 17], computer vision and computer graphics [18]. TPS can produce highly accurate and realistic results when combined with other techniques, such as FEM and ED graph. The fundamental idea behind TPS is to define a set of control points on the object to compute a series of deformation functions, which are then applied to the object to transform it from its original shape to its deformed form.

The deformation functions are derived from a thin plate energy function, which determines the amount of bending energy needed to deform the object. Initially, the control points are moved to their deformed positions using an initial deformation method, such as FEM or ED graph. Subsequently, the thin plate energy function is minimized subject to constraints that ensure the deformation functions are smooth and have the desired behavior at the control points. Once the deformation functions are computed, they can be applied to any point on the object to calculate its deformed position. This allows the object to be smoothly deformed in a way that precisely reflects the motion of its underlying structures. While TPS can handle complex deformations, it can be computationally expensive and require a large number of control points.

2.3 Colonoscopy Reconstruction Datasets

Complete colonoscopic video datasets with ground truth camera poses and image depth are crucial for developing and validating algorithms such as pose estimation, depth estimation, and 3D reconstruction methods. A summary of existing and the proposed colonoscopic 3D datasets referenced in relevant papers is reported in Table 2.1

TABLE 2.1: Comparison of Colonoscopy Reconstruction Datasets

Papers	R/V/P	Public	Depth	Poses	# Frames	3D Models
Armin et al. [19]	V	×	×	✓	30k	×
Rau et al. [20]	V	✓	✓	✓	18k	×
Bae et al. [21]	R	×	×	×	>34k	×
Freedman et al. [22]	V	×	✓	×	187k	×
Fulton et al. [23]	P	✓	×	✓	24k	×
Ozyoruk et al. [24]	V/R	partial	✓	✓	>30k	✓
Ma et al. [25]	R	partial	×	×	1.2m	×
Bobrow et al. [26]	P	✓	✓	✓	10k	✓
Proposed	V	⋆ ✓	✓	✓	infinite	✓

R (Real), V (Virtually simulated), P (Physical phantom)

⋆ The colonoscopy simulator and source code have been made publicly available.

Different works have used a variety of datasets and acquisition methods, but only a few published them. Mitchell et al. collected colonoscopic images inside a colon phantom model using an endoscope camera and estimated camera poses using an electromagnetic tracker attached to the endoscope [23]. However, the textureless colon phantom model is essentially a uniformly thick circular tube with repetitive colon folds, which is overly idealized and not suitable for feature-based SLAM or training depth estimation networks.

Later on, Talor et al. used a clonoscope and silicone colon models to generate a colonoscopy dataset with “ground truth” depth and camera poses [26]. A robotic arm was used to rigidly mount the colonoscope and measure the camera trajectory. Then, a GAN was used to predict depth for the captured optical video sequences, and GAN-estimated depth frames were compared with rendered predicted views of a 3D colon model along the measured camera trajectory for minimizing an geometric contours-based loss (Contours were extracted using Canny edge extraction and binarized). Thus, the camera trajectory and depth frames can be optimized together. However, the “ground truth” poses and depth still suffer from errors. Furthermore, the scope range and types of camera motion were limited, leading to most videos being captured from forward views along the centerline of colon models. Additionally, it is important to note that in this approach, the phantom models must remain static during video acquisition, and colon deformations are neither considered nor simulated.

Most recently, the game engine Unity has been used to render synthetic images from 3D anatomical models, such as CT-segmented 3D colon models [20, 24, 27]. Rendered data offers several advantages, primarily because of the availability of error-free, pixel-level ground truth labels like depth and surface normals from rendering primitives. Ozyoruk et al. introduced both ex-vivo and synthetically generated dataset for the stomach, small intestine, and colon [24]. Each video sequence in this dataset is paired with a ground truth camera trajectory and 3D surface model, and pixelwise depth ground truth was generated for synthetically generated endoscopic frames. Rau et al. generated synthetic data based on a CT-segmented human colon mesh model, and rendered endoscopic simulation images and corresponding depth maps using Unity. However, the color and texture of synthetic images are much different from those of actual colonoscopic images, and one important characteristic of colon deformation is not simulated.

In contrast, we developed a realistic colonoscopy simulator and it provides a dynamic and controlled environment for data generation. More importantly, we have made the simulator and source code publicly available. The simulator not only simulates colon deformations but also offers a user-friendly human-machine interface and reliable control capabilities. Users can use the simulator to rapidly generate substantial volumes of data with different textures, lighting conditions and levels of deformations.

2.4 Reconstruction of Colonic Surface Maps

Advancements in computer vision and image processing have led to the development of various techniques for reconstructing the colonic surface. Some of these methods aim to create a 2D visibility map of the internal colonic surface, while others focus on generating a portion of the 3D colonic surface.

Colorectal cancer mortality can be reduced by half through colonoscopy screening with a conventional 2D colonoscope. However, this procedure has limited protective value due to missed lesions. To enhance the sensitivity of colonoscopy to precancerous lesions, 3D imaging techniques can be utilized to highlight their distinctive morphology. Although 3D imaging has demonstrated benefits in laparoscopic procedures, further research is necessary to evaluate its efficacy in flexible endoscopy applications. In general, the methods utilized or developed can be categorized as follows.

2.4.1 Shape from Shading

The Shape from Shading (SfS) method [28] is a technique used for estimating the 3D structure of a scene from changes in illumination with respect to depth and surface orientation. SfS has been employed in various applications, including the reconstruction of colon structures from the brightness of the colon surface. However, one limitation of SfS is that it may incorrectly represent the colon lumen as a relatively far surface, rather than a tubular structure [29].

The reason behind this limitation lies in the assumptions and constraints of the SfS method. It assumes that the surface of the object being reconstructed is Lambertian, meaning it has a uniform diffuse reflectance and does not exhibit specular reflections. However, the colon lumen, which is the inner space of the colon, does not have a uniform diffuse reflectance as it contains air or gas and does not scatter light in the same way as a solid surface. As a result, the SfS method may not accurately estimate the depth and surface orientation of the colon lumen, leading to an incorrect representation of the tubular structure.

2.4.2 Structure from Motion

Structure from Motion (SfM) is a technique used to reconstruct the 3D structure of a scene or object from a series of 2D images. It relies on estimating camera poses and 3D points from the images, which are then used to reconstruct the 3D structure.

One limitation of SfM is that it typically requires slow camera motion to accurately estimate camera poses. This is because SfM algorithms rely on finding corresponding features in multiple images to triangulate and estimate the 3D points. Fast camera motion can result in motion blur and loss of feature correspondence, leading to inaccurate camera pose estimation and 3D reconstruction.

To reconstruct colonic surface maps from colonoscopic images, the requirement of slow camera motion can be a challenge. The colonoscope is typically advanced through the colon lumen by pushing and pulling motions, which can result in fast camera motion. This can make it difficult to obtain accurate camera poses and reconstruct the 3D structure of the colonic surface using traditional SfM algorithms.

Researchers have attempted to address this limitation by developing modified SfM algorithms that are specifically tailored for colonoscopic images. For example, Koppel et al. used sequential frames from colonoscopic videos to reconstruct a portion of the 3D colonic surface with textures, but it required slow camera motion and manual feature tracking [30]. Similarly, Chen et al. proposed a modified SfM approach that accounted for the fast camera motion in colonoscopic images by incorporating prior knowledge of the colon shape and camera motion constraints [31]. Despite these efforts, accurately reconstructing the 3D structure of the colonic surface with textures from colonoscopic images remains challenging due to the fast camera motion involved.

2.4.3 Combination of Shape from Shading and Structure from Motion

By combining SfM and SfS, Kaufman et al. were able to leverage the strengths of both techniques to reconstruct a relatively large colonic surface from several consecutive colonoscopic images . SfS provided local shape and shading cues, which helped to estimate the surface shape of individual frames. SfM, on the other hand, estimated the camera poses, which provided the necessary information for integrating the partially flattened surfaces from multiple frames into a larger surface map.

2.4.4 Approaches with Restrictive Assumptions

There are other advanced approaches with restrictive assumptions. Zhou et al. [32] adopted an optical flow-based method to reconstruct small colon segments with assumptions that the neighboring folds in an image are not occluded and that the colon fold contours are circular in nature. However, partial occlusion of folds is very common and the transverse, ascending and descending segments of the colon have no well circular characteristic. Hong et al. [33] took the advantage of the tubular nature of the colon to estimate colon folds and only reconstructed a colon segment from a single colonoscopic image. Armin et al. [34] fitted a cylinder model to the colon structure generated by 3D pseudo stereo vision and unrolled the fitted model to a 2D band image. Then the estimated camera poses were used as initial values to register these 2D band images together to build a large 2D visibility map, but the generated 2D map was less intuitive than a 3D dense reconstruction. Although remarkable progress has been made in this field, all of the research has focused on 3D or 2D surface reconstruction of very small parts of colon.

2.4.5 Deep Learning-based Depth Prediction

Deep learning networks have been explored for depth prediction in endoscopy. These can be divided into fully supervised depth prediction networks and self-supervised approaches. Since it is difficult to obtain the dense ground truth depth maps for the real endoscopic images, fully supervised networks are usually trained on synthetic dense depth maps generated from patient-specific CT data.

Mahmood et al [35] used simulated pairs of color images and dense depth maps from CT data to train a depth prediction network. To predict depth for the real endoscopic images, they used a GAN to transform real images to have the simulated-like appearance and then feed them to the trained depth estimation network. But their structure (i.e. real depth) information is not fully used, which can lead to decreased performance up to incorrect depth estimates. Liu et al. [36] trained a self-supervised network for depth prediction in sinus endoscopy, their work use monocular videos as training data and use sparse depth map estimated from structure from motion to supervise the training process. Rau et al. [20] have applied a variant of pix2pix called extended pix2pix to colonoscopy depth reconstruction. They first used phantom and virtual colonoscopy data to create paired depth and colon images, then included real colonoscopic images for the GAN loss to allows the network to partially train on real colon images while not needing the corresponding ground truth. Mathew et al. [37] took advantage of the texture information of optical colonoscopy (OC) and geometrical information of virtual colonoscopy (VC) to trained a CycleGAN for lossy unpaired image-to-image translation between the two modalities. Bae et al. [21] used sparse reconstruction obtained via SfM to develop a multi-view stereo reconstruction method that can produce a small segment of the colon from a short sequence of endoscopic images.

2.4.6 Stereo Shape Recovery

Stereo shape recovery is a process that involves using a pair of stereo images to recover the 3D shape of objects in a scene. Technically, the technique of estimating depth using a stereo camera involves triangulation and stereo matching. Triangulation requires accurate calibration and rectification to constrain the problem to a 2D plane, also known as the epipolar plane. Stereo matching, or disparity estimation, entails identifying the corresponding pixels in the different views that relate to the same 3D point in the scene. By computing the relative disparity, a depth map can be generated, which in turn can be utilized to reconstruct the 3D geometry of the scene. Recently, real-time 3D stereo shape recovery can be achieved by implementing traditional stereo vision algorithm on GPU [38].

Currently, studies have proved that stereoscopic imaging technology is widespread used and has the potential to improve sensitivity, lesion resection, training and automated lesion detection [39] for laparoscopic procedures. In addition, stereoscopic hardware is continuing to evolve to generate higher quality surgical vision [40, 41]. Although most existing endoscopic procedures especially a standard colonoscopy still use single-lens cameras, more research is needed to assess how stereoscopic imaging will improve applications of flexible endoscopy.

2.4.7 Visual SLAM Algorithms

Currently, according to the density of reconstructed maps, camera based visual SLAM algorithms can be classified into sparse [42, 43], semi-dense [44–47] and dense reconstruction [48–50]. These SLAM systems are template-free, adopt loop closure to reduce drift errors and able to process slow motion. Although promising results can be achieved, these algorithms are seldom directly applied in colon reconstruction scenarios mainly due to few or lost of overlaps and no loop closures. The small field view of the colonoscope, the tubular structure of the colon, difficulty to observe the back side of colon folds and especially when camera with motion of orientations only will cause few or lost of overlaps in the colonoscopy procedures and this will cause inaccurate or even failed camera pose estimations. Meanwhile, there is no loop closures in a normal colonoscopy procedure since the colonoscope is withdrawn from the cecum (the proximal end of colon) to the rectum (the proximal start of colon) and this will cause a large drift error for the camera pose estimation and scene reconstruction. All these will lead to misalignment in textures on the reconstructed colon map.

2.4.8 Combination of Deep Neural Network and Visual SLAM Algorithms

Recently, SLAM systems that incorporate depth predictions estimated by deep learning techniques have been applied to monocular colonoscopy sequences to reconstruct 3D colonic surfaces [25]. Depending on whether a SLAM system optimizes the photometric error maximizing the photometric consistency or not, it can be classified as a direct or in-direct method.

Chen et al. [51] trained a adversarial depth estimation neural network in a supervised approach where supervision from synthetic dataset of a phantom, then input monocular images paired with depth estimation to the ElasticFusion [48] to stitch depth images to reconstruct a dense surfel point cloud. However, the metric accuracy of estimated camera poses and reconstruction is not given. Also, it is not suitable to directly apply ElasticFusion on endoscopy since it requires slow camera motion and a as rigid as possible environment. Ma et al. [25, 52] used sparse depth estimated from the COLMAP [53] software as a ground truth proxy to train a recurrent neural network for depth and the camera pose estimation, then the bundle adjusted direct sparse odometry (DSO) [44] is used to jointly optimize the predicted poses and sparse point inverse depth by minimizing the intensity difference over a window of recent frames. After that, a fusion pipeline is used to reconstruct colon meshes for detecting missing regions. However, like most SLAM systems, their work requires slow camera motion and cannot handle the deformation of the colon surface, which makes the camera pose estimation suffering from large drift and further causes textures misalignment on the fused colon meshes. The main advantage of direct methods is that they do not require the feature extraction from the images, but they are susceptible to drastic illumination changes in the colonoscopy environment. Meanwhile, in-direct methods which optimize the reprojection error using tracked features highly depend on the successful extraction and tracking of sufficient distinct features from images, but the low-texture colonoscopic images have less salient features.

Our third proposed work has some relations to the research works on photometric BA for 3D mesh refinement which requires frequent remeshing (contributing to a high runtime) and a sufficiently good initialization [54], vision-based SLAM in which the inappropriate reference frame selection can result in accuracy degradation[55] and transesophageal echocardiography images registration in the 3D image domain [56]. [54] jointly refined the mesh shape and camera poses using the reprojection error between images of a mesh model and the observed images, which requires frequent remeshing (contributing to a high runtime) and a sufficiently good initialization. [55] jointly refined the camera and structure parameters by minimizing intensity difference between one reference frame and a few frames that are temporally close to the reference frame, and the inappropriate reference frame selection can result in accuracy degradation. [56] used photometric BA to minimize the intensity difference between multiple views of 3D heart ultrasound images and a 3D panaramic image such that the estimated camera poses are used to align the local 3D frames to enlarge the image FOV, which is directly minimized in the 3D image domain.

2.5 Chapter Summary

In this chapter, we provide the brief mathematical background and literature review of the colon reconstruction problem. In the optimization process of non-linear problems formulated in Chapters 5 and 6, the use of Lie algebra allows for the derivation of a rotation matrix to be computed efficiently, which in turn can transform the $\mathbb{SO}(3)$ property constrained optimization problem into an unconstrained optimization problem. The literature review summarizes previous techniques and works for reconstructing 2D or 3D colon maps, discussing the limitations and challenges of the previous methods.

Chapter 3

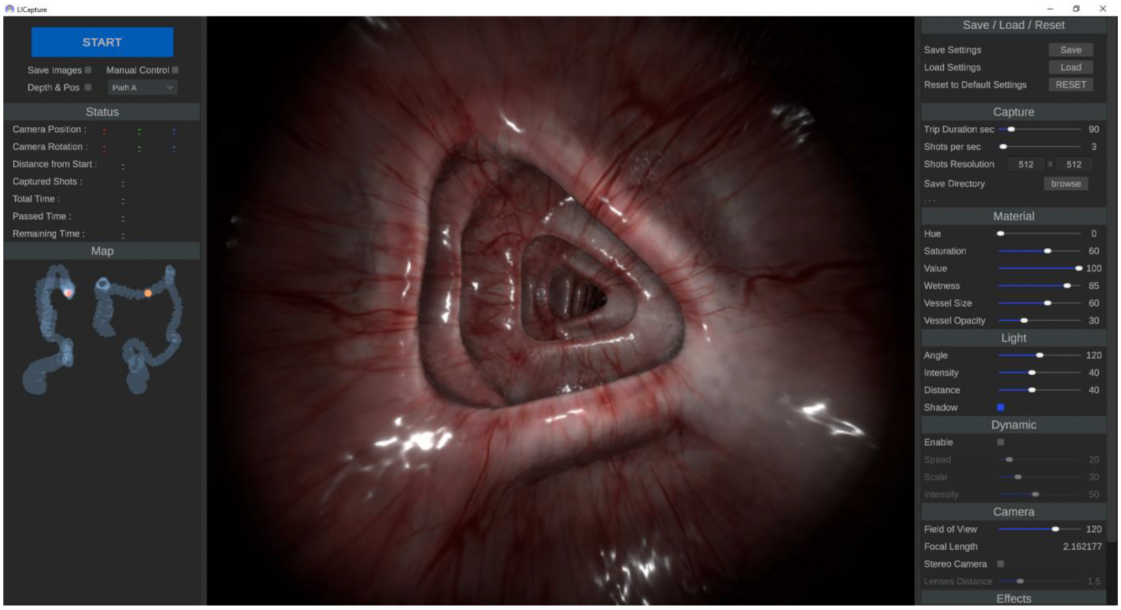
Colonoscopy Simulator Development

Obtaining colonoscopic images with accurate ground truth of depths and camera poses poses significant challenges in standard colonoscopy procedures. These challenges arise from various factors, including the invasiveness of the procedure, patient safety and comfort considerations, constraints in clinical workflow and time, ethical and legal considerations, and technical limitations in capturing precise data in real-time during the procedure. As a result, alternative methods, such as generating synthetic data from a realistic colonoscopy simulator, have emerged as viable options for developing and validating colon reconstruction algorithms when ground truth data from standard colonoscopy procedures is not available.

In addition to addressing the aforementioned challenges, the development of a virtual colonoscopy simulator offers compelling advantages. The simulator provides a controlled and accurate environment for generating benchmark data, which can be used to rigorously evaluate the accuracy and performance of computer vision algorithms employed in virtual colonoscopy, such as polyp detection and navigation assistance. Moreover, virtual colonoscopy simulators offer a non-invasive and safe environment for training and assessment of medical professionals. Within the simulator, practitioners can engage in practice sessions that involve various colonoscopy techniques, navigating the virtual colon without the need for real patients or invasive procedures. Furthermore, the simulator can provide real-time feedback on performance, facilitating the opportunity for practitioners to learn from mistakes, refine their skills, and enhance their proficiency in a controlled and low-risk environment.

These advantages contribute to the advancement of virtual colonoscopy techniques, augment the training and education of medical professionals, and ultimately result in improved patient care in the field of colonoscopy. As such, the development and utilization of virtual colonoscopy simulators hold significant promise in addressing the challenges associated with obtaining ground truth data in standard colonoscopy procedures, and offer valuable opportunities for advancing the field of colonoscopy through enhanced training and evaluation of computer vision algorithms.

In this chapter, we give a brief overview of the process of creating the realistic colonoscopy simulator. Fig. 3.1 shows a snapshot of the developed simulator.



Trip Duration sec 90
Shots per sec 3
Shots Resolution 512 × 512
Hue 0
Saturation 60
Value 100
Wetness 85
Vessel Size 60
Vessel Opacity 30
Angle 120
Intensity 40
Distance 40
Shadow ☑
Enable ☐
Speed 20
Scale 30
Intensity 50
Field of View 120
Focal Length 2.162177
Stereo Camera ☐
Lenses Distance 1.5

FIGURE 3.1: **The Snapshot of the Developed Colonoscopy Simulator:** on the top left panel of the UI, we can select the "Save Images", "Manual Control" and "Depth & Pos" buttons, then start the simulator by clicking the "START" menu, and move and rotate the virtual camera using the keyboard. The middle left "Status" panel will display the camera poses in real-time. The right panels on the UI mainly support the parameters settings of the virtual camera and colon environment. Once the "Manual Control" is selected, the control instruction panel will appear, providing the following options: To rotate the camera, use the W, S, A, D keys on the connected keyboard. To move the camera vertically, use the Up and Down arrow keys on the keyboard.

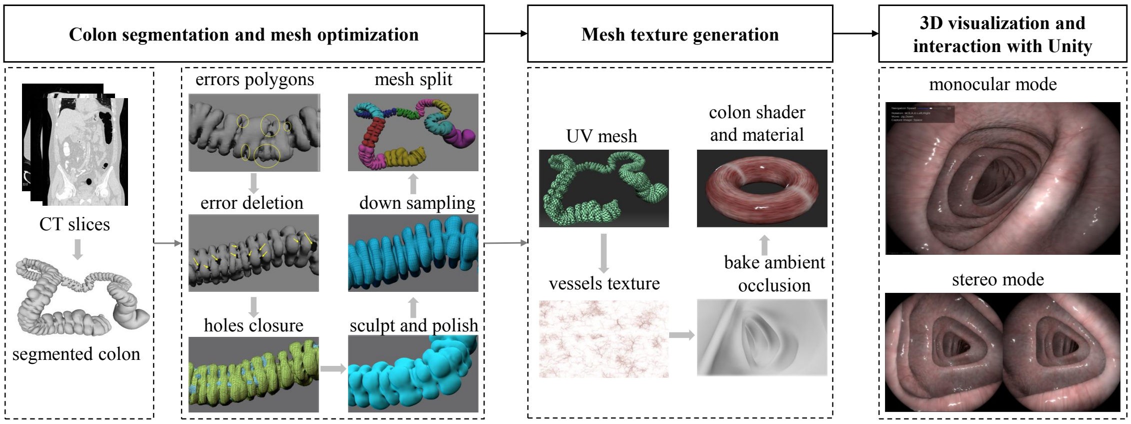


Figure 3.2: **Schematic Diagram of the Developed Colonoscopy Simulator Framework.** The length of CT-segmented colon is about 1.5 meters and its bounding box size is $36cm \times 26cm \times 14.9cm$. The framework primarily comprises four key components: 3D colon mesh model segmentation and optimization, the creation of a 2D image texture that envelops the segmented colon model, the implementation of a virtual visualization and interaction system, and the design of the virtual camera along with post-processing effects.

Fig. 3.2 shows the schematic diagram of the developed colonoscopy simulator framework. The framework mainly consists of 3D colon mesh model segmentation and optimization, creation of a 2D image texture that wraps around the segmented colon model, implementation of the virtual visualization and interaction system, virtual camera design and post processing effects.

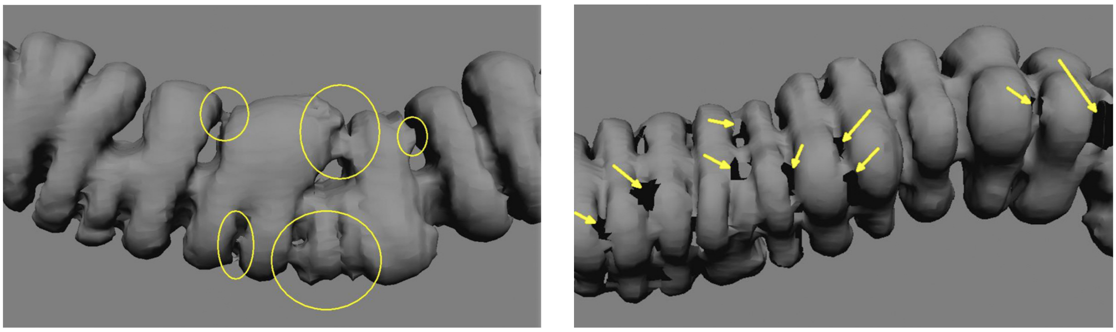
First, a colon mesh model is segmented from a set of human colon CT scans. Then, to render the colon model as realistic as possible, a 2D texture image with blood vessels, perlin noise and mucous is created by Photoshop and used to wrap around the segmented colon model. The blood vessels textures are extracted from the real colon images which are download from Google Images directly.

After that, the colon mesh model and the 2D texture image are loaded into the game engine Unity [57], and a wide-angle monocular virtual camera with two light sources is used to provide volume-based rendering of endoscopic views. The properties of the colon shader and material can be adjusted, such as the hue, saturation, colour, reflectiveness and wetness.

By using buttons on a keyboard, we can manually rotate and shift the camera inside the 3D colon model and capture images together with pixel-wise ground truth depths and camera poses. To prevent the camera from moving through the colonic surface, a mesh collider which roughly defines the shape of the colon mesh is built for the purposes of physical collisions.

3.1 Colon Segmentation and Mesh Optimization

The triangular 3D colon surface mesh is segmented from a set of 2D colon CT scans by using the free, open source software 3D Slicer [58]. Since the original CT-segmented colon model (in ".STL" format) has some errors in some parts that must be fixed, we import the CT-segmented colon mesh model into ZBrush software [59] and export it as ".OBJ" format, then the exported colon mesh model will be sculpted and polished using the software Softimage [60] and ZBrush. For example, as shown in Fig 3.3, some colon haustral folds on one contraction ring or on consecutive contraction rings are frequently segmented as one fold, the software Softimage is used to delete these polygon errors.

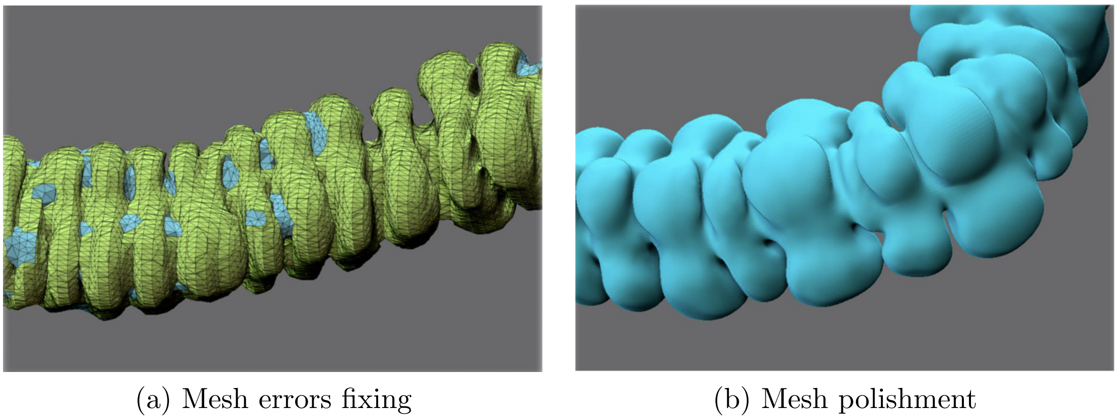


(a) Mesh errors

(b) Mesh errors deletion

Figure 3.3: **Errors Deletion from the CT-segmented Colon Mesh**. (a) shows some mesh errors of the CT-segmented colon .STL file; (b) shows the deletion of colon mesh errors.

After that, as shown in Fig 3.4, all holes of the colon mesh that are created by deleting these error polygons are closed and fixed by using the software ZBrush, and the sculpting and polish brushes of the ZBrush software are used to make the inside colonic surface of the segmented colon mesh as smooth as a real colon. Furthermore, to reduce the computational cost of the simulator in Unity, we downsample the colon mesh to make it has lower triangles (see Fig 3.5) and cut the downsampled mesh into several parts (see Fig. 3.6), thus to optimize the graphics performance of the developed simulator in Unity.


(a) Mesh errors fixing

(b) Mesh polishment

FIGURE 3.4: **Colon Mesh Errors Fixing and Further Polishment**. (a) shows the mesh errors fixing; (b) shows the polished colon mesh.

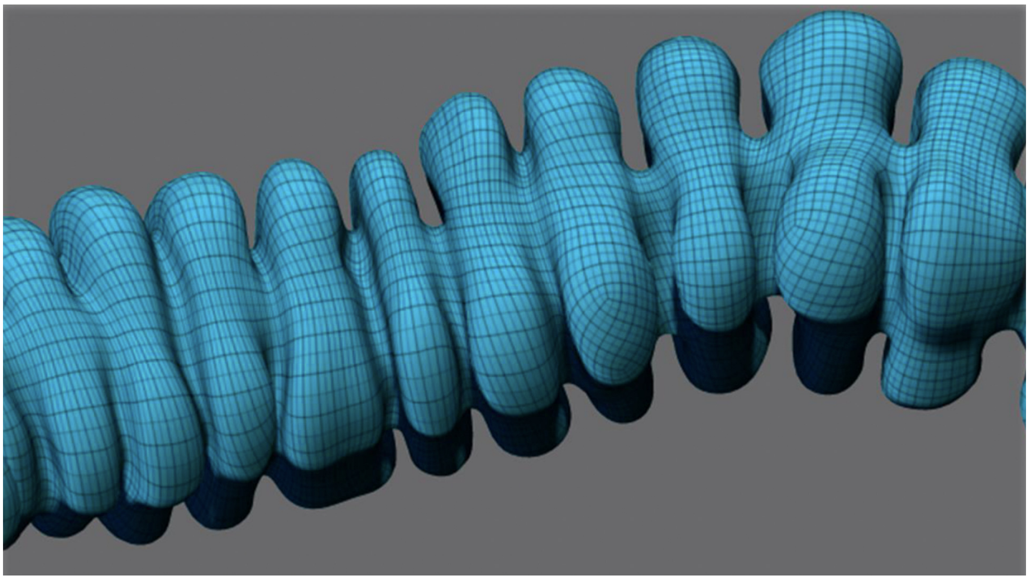


FIGURE 3.5: **Colon Mesh Down-sampling.** Reduce the complexity of the colon mesh model to reduce the computational cost of the simulator in Unity.

3.2 Mesh Texture and Colon Surface Material Generation

In this step, to texture the colon mesh and make the simulator as realistic as possible, the UV-mapping tool Unfold3D is used to create the UV ("U" and "V" denote the axes of the 2D texture image) map for the colon mesh and this UV map (see Fig 3.7) will be used in the Unity platform for applying the vessels texture over the mesh.

After that, as shown in Fig. 3.8, the created UV map is imported into the software Softimage [60] to bake ambient occlusion into the mesh vertex colors that can help mix some

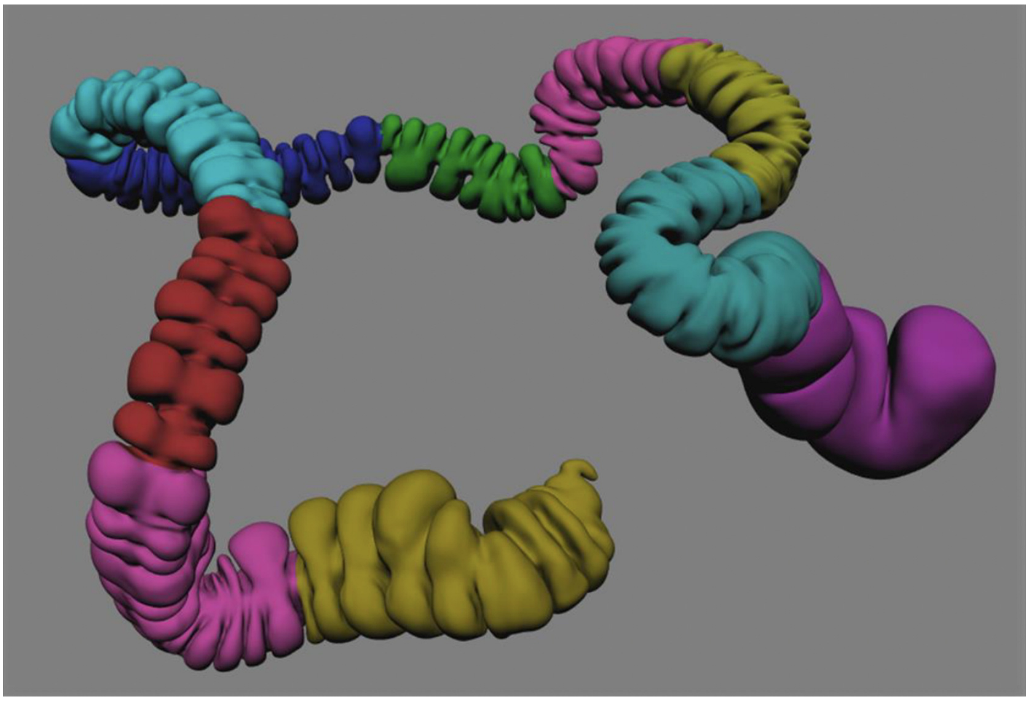


FIGURE 3.6: **Cutting the Mesh Into Several Parts for Better Performance in Unity**. Then mesh is cut into multiple parts for using in the simulator and also the split meshes are optimized again (the normals of the mesh vertices are adjusted to make the cut seams invisible in the simulator) to have better performance.

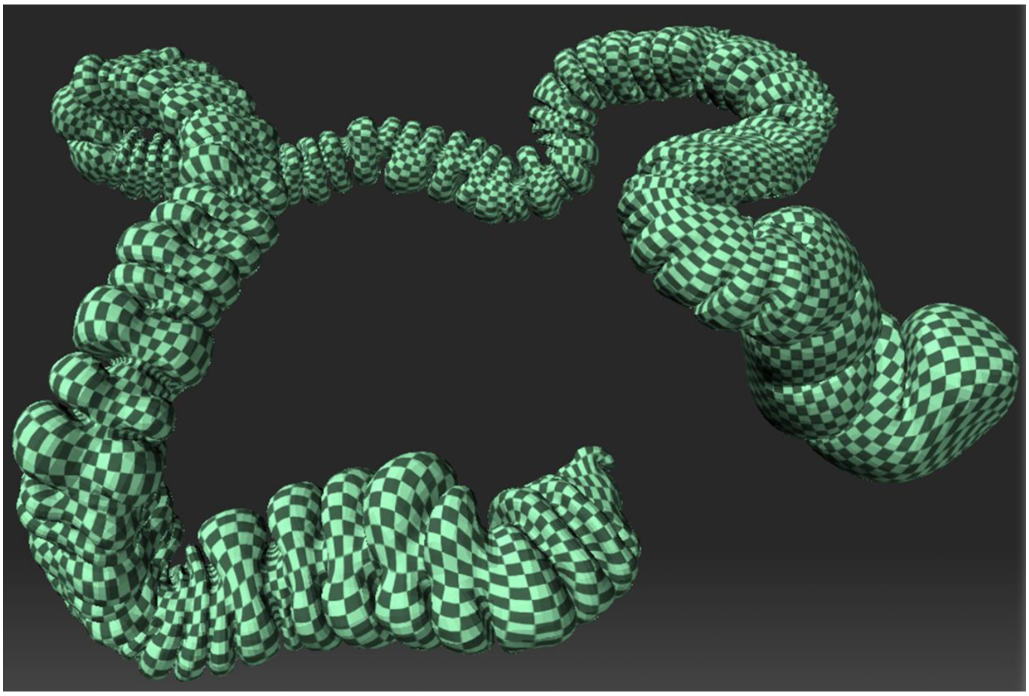


FIGURE 3.7: **UV Mesh Creation for the Colon Mesh Model.** "U" and "V" denote the axes of the 2D texture image.

deep shadow and realistic look to the shader and material of the colon.

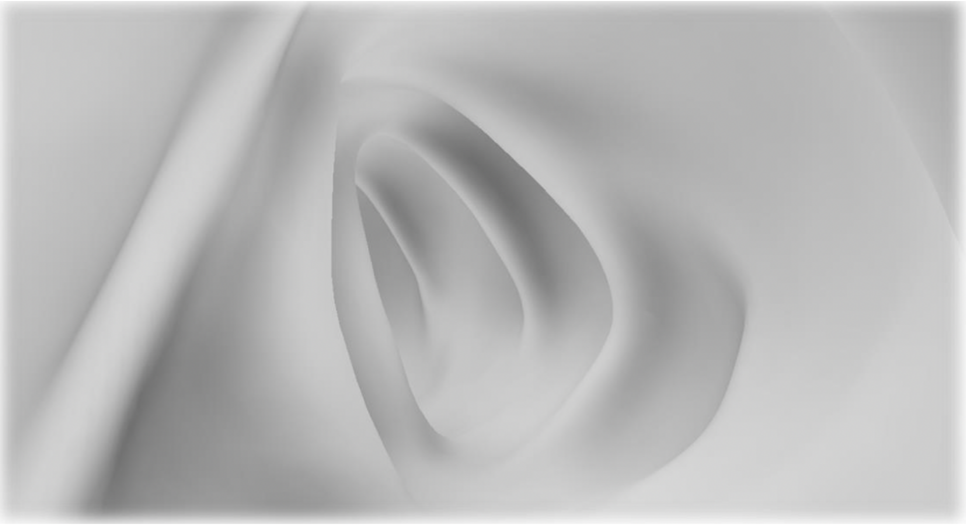
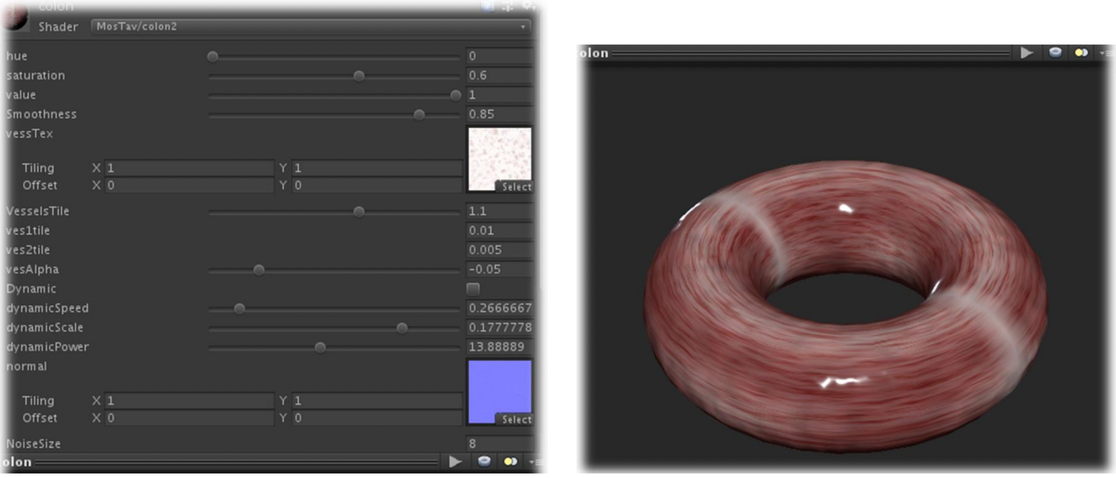


Figure 3.8: **Baked Ambient Occlusion Into Vertex Colors.** This process enhances the shader and material of the colon, creating a more realistic appearance with deeper shadows.

For the material of the colon surface, as shown in Fig. 3.9, the custom shader is created by mixing 2 layers of vessel textures with 2 different scales and adding some colors and other settings to make the colon surface have a realistic look.



Shader	MosTav/colon2		
hue	0		
saturation	0.6		
value	1		
Smoothness	0.85		
vessTex			
Tiling	X 1	Y 1	
Offset	X 0	Y 0	
VesselsTile	1.1		
ves1tile	0.01		
ves2tile	0.005		
vesAlpha	-0.05		
Dynamic	☐		
dynamicSpeed	0.2666667		
dynamicScale	0.1777778		
dynamicPower	13.88889		
normal			
Tiling	X 1	Y 1	
Offset	X 0	Y 0	
NoiseSize	8		

Figure 3.9: **Colon Model Shader and Material**. In Unity, we create a custom shader by blending two layers of vessel textures with distinct scales and introducing various colors and additional settings. This approach is employed to achieve a realistic appearance on the colon surface.

Finally, to texturize the UV mesh, seamless and tillable textures of the blood vessels, perlin noise and mucous are created using the software Photoshop [61] and added into the 2D texture image (see Fig. 3.10).

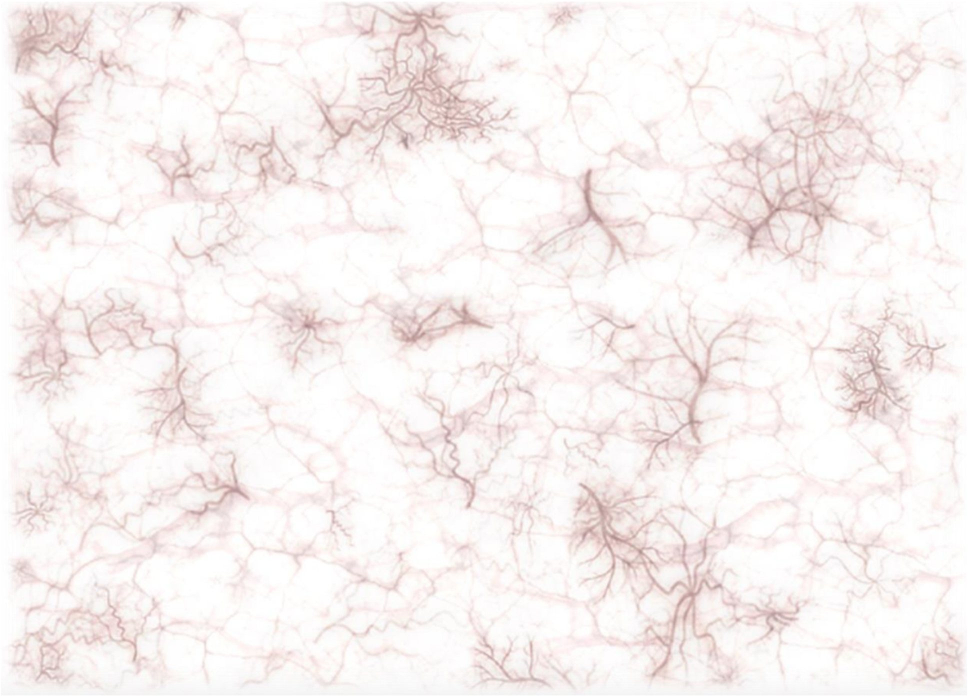


FIGURE 3.10: **Vessels Texture Map Creation.** Seamless and tillable texture of the vessels are created using Photoshop to creat customized and randomized brushes and mix some different layers together.

3.3 3D visualization and Interaction with Unity

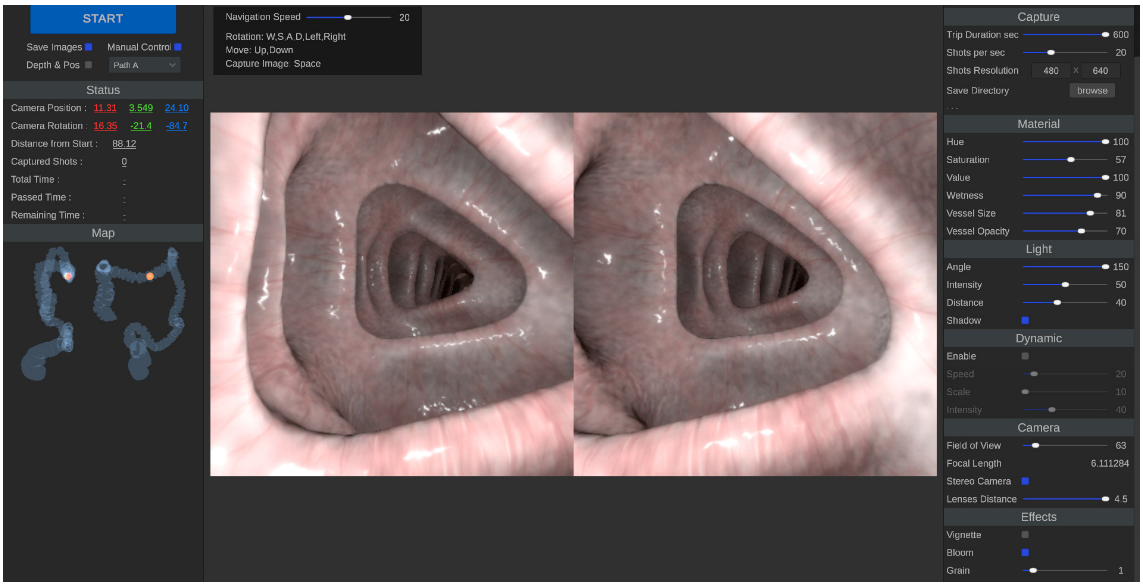
The visualization and interaction system is built with Unity. The visualization part can create 3D virtual visualization environment of the colon model and provide volume-based rendering of endoscopic views during the virtual camera's flight through the colon model. To prevent the camera from moving through the colonic surface, the mesh collider is used to build a collider based on the colon mesh. For the interaction part, it allows the player to interact with GameObjects (cameras, model, special effects, etc.) and output simulated colonoscopic images together with ground truth of camera poses and image depths.

The main functions of the developed simulator are created with visual scripting (node-based) and the main game objects that contains the most important FSMs (function nodes) are as follows:

- actionManager: this is the host of most important events
    - start path maker: creating a path for the virtual camera;
    - start cam mover: calling the cam mover to start moving the camera;
    - stop cam mover: calling the cam mover to stop moving the virtual camera;
    - load UI: updating all UI elements when we reset or load the settings;
    - start manual control: calling the manual control to start controlling the virtual camera manually.
- dataSaver: handling all data capture and saving processes;
- 3dCamPiv: containing monocular cameras, stereo cameras and also light sources;
- settingManager: containing all settings that users set, load or default simulator settings;
- statusUpdater: when "START" button is pressed, this FSM will update the status inside the simulator UI;
- camMover: when "START" button is pressed, this FSM will handle the camera movement through the generated path;
- manualControl: when "START" button is pressed, if the working mode is set to manual, this FSM will handle the manual camera movement;
- pathMaker: this FSM handles the processing of creating the camera path.

3.4 Virtual Camera Design and Configuration

The developed simulator can be set into monocular virtual camera mode and stereo virtual camera mode. With the UI of the simulator, users can adjust specific parameters of the virtual camera, such as the camera FOV, which can be set within $[50°, 150°]$, and the corresponding range of focal length, which can be set within $[1mm, 8mm]$. Additionally, the baseline of stereo camera can be set within $[0.5mm, 4.5mm]$. Fig. 3.11 shows the simulator when it is working in stereo mode.



Save Images ☑ Manual Control ☑
Depth & Pos ☐ Path A

Camera Position : 11.31 3.549 24.10
Camera Rotation : 16.35 -21.4 -84.7
Distance from Start : 88.12
Captured Shots : 0
Total Time : -
Passed Time : -
Remaining Time : -

Navigation Speed 20
Rotation: W,S,A,D,Left,Right
Move: Up,Down
Capture Image: Space

Trip Duration sec 600
Shots per sec 20
Shots Resolution 480 x 640
Save Directory

Hue 100
Saturation 57
Value 100
Wetness 90
Vessel Size 81
Vessel Opacity 70

Angle 150
Intensity 50
Distance 40
Shadow ☑

Enable ☐
Speed 20
Scale 10
Intensity 40

Field of View 63
Focal Length 6.111284
Stereo Camera ☑
Lenses Distance 4.5

Vignette ☐
Bloom ☑
Grain 1

FIGURE 3.11: **The Snapshot of the Developed Colonoscopy Simulator Working in Stereo Mode.** It supports users to adjust the field of view of the stereo camera and the baseline between the left and right cameras.

To simplify the camera parameters setting and fly the camera inside the colon mesh model in Unity, one default camera flying path is created by duplicating some empty GameObjects inside the colon mesh model (see Fig. 3.12). These empty GameObjects serve as the camera curve path points in Unity. Thus, to create some different paths, we just need to move these points a little to other positions. For a random path, these points will be move randomly by adding some random noises.

3.5 Colon Deformation Simulation and Post Processing Effects

To simulate the topological deformation of the colon, the centerline of the colon mesh model is extracted and represented by a set of points. Each point is associated with an orthogonal cross section [62]. By moving the points of the centerline and mapping the

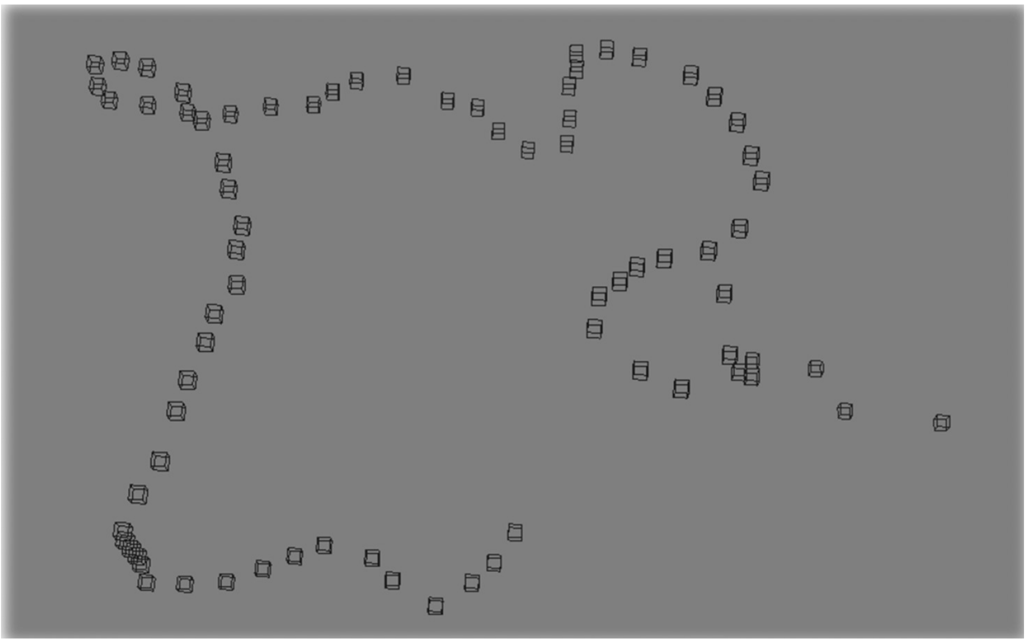


FIGURE 3.12: **A Default Camera Flying Path.** A default camera flying path created by duplicating some empty GameObjects inside the colon mesh model.

cross sections, the overall shape of the colon will be deformed. To simulate the local deformation of colon, the Vertex Manipulation model (a mesh deformer in Unity) is used and we adjust colon mesh vertex positions with different levels of force to simulate local deformations caused by muscle contractions or external forces.

To make the colon model in the simulator more close to the real colon inside environment and provide different scenarios of datasets, we add more properties to the colon shader. Then, users can change the parameters of hue, saturation, color, reflectiveness, wetness, vessel size, and vessel opacity. Fig. 3.13 shows a visual comparison between real colonoscopic images with clearly structure and the simulated images generated by the developed simulator. The motion blur and image distortion effects are currently not taken into consideration in the developed simulator.

3.6 Obtanning Synthetic Datasets from the Simulator

To use the developed simulator to collect synthetic colonoscopy datasets, we can follow the steps below:

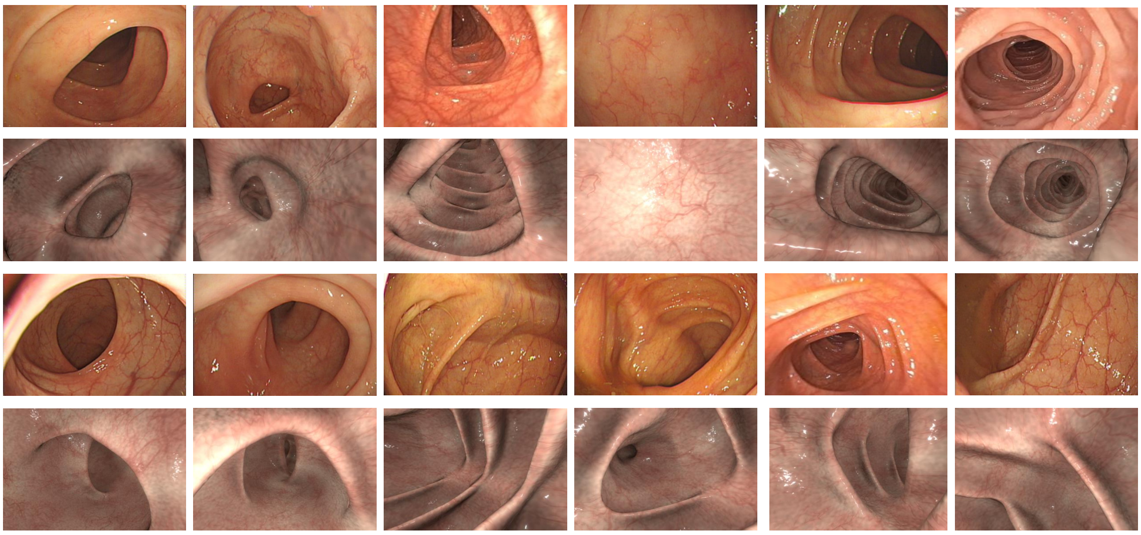
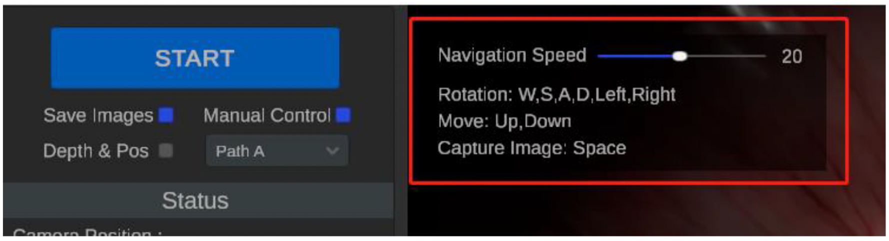


FIGURE 3.13: **Visual Comparison between Simulated and Real Colonoscopic Images.** The first and third rows show real images generated from a colonoscopy, the second and the last row shows simulated images generated from the simulator.

(1) Uisng the UI panel located in the top left corner of the simulator, the user can select the “Save images”, “Depth & Pos” menus to save captured images with corresponding ground truth of depth and poses (see Fig. 3.14);



Navigation Speed 20

Rotation: W,S,A,D,Left,Right
Move: Up,Down
Capture Image: Space

FIGURE 3.14: **Launch the Simulator and Control the Virtual Camera**. Press the “START” menu to launch the simulator. The move speed of the virtual camera can be adjusted to various values. To rotate the camera, use the W, S, A, D keys on the connected keyboard; to move the camera, use the Up and Down arrow keys on the keyboard.

(2) If the user wants to manually control the virtual camera inside the colon, just select the “Manual Control”, and the control instruction panel will appear and show the following information: Rotate the camera using the keys W,S,A,D on the connected keyboard; Move the camera using the Up and Down arrows on the keyboard. When these keys are pressed, the user can update the rotation and position of the virtual camera accordingly.

(3) In the top right panel named "Save/Load/Reset", the user can select the local directory on our computer to save and reload the settings of the configuration parameters;

(4) The "Capture" panel located in the top right corner allows the user to configure the capture settings for recording datasets. This panel include options to set the capture framerate, which determines the number of frames per second that are captured during the simulation. The user may be able to adjust this value based on their preferences or requirements.

Additionally, the "Capture" panel also include an option to set the directory or folder in the user's computer where the captured datasets will be saved. This allows the user to specify the location on their computer's file system where the captured data will be stored for later use or analysis.

The ability to set the capture framerate and directory in the colonoscopy simulator provides the user with flexibility and control over the simulation recording process, allowing them to customize the settings according to their needs and preferences.

(5) The "Material" panel located in the right of the simulator allows the user to adjust several parameters to change the appearance of the colon inside the environment. These parameters include:

- Hue: Hue refers to the color tone of the colon. By adjusting the hue parameter, you can change the overall color of the colon, ranging from warmer tones like red and orange to cooler tones like blue and green;
- Saturation: Saturation determines the intensity or purity of the color in the colon. Increasing the saturation parameter will result in more vibrant and vivid colors, while decreasing it will make the colors more muted and dull;
- Wetness: Wetness parameter controls the level of moisture or shininess on the surface of the colon. Higher wetness values will make the colon appear more glossy and reflective, while lower values will make it look drier and less reflective;
- Vessel size: Vessel size parameter determines the size of blood vessels or veins visible on the surface of the colon. Increasing the vessel size will make the vessels appear

larger and more prominent, while decreasing it will make them smaller and less noticeable;

- Opacity: Opacity parameter controls the transparency or opacity of the colon material. Higher opacity values will make the colon material more opaque and less transparent, while lower values will make it more translucent.

By adjusting these parameters in the “Material” panel, the user can customize the appearance of the colon in the simulator to suit desired visual aesthetics requirements.

(6) For the panel named “Light” located in the right of the simulator, it provides options for configuring the properties of the light source attache to the virtual colonsocope. These properties include:

- Light Intensity: This setting allows the user to adjust the brightness or intensity of the light source. Increasing the intensity will make the light brighter, while decreasing it will make it dimmer;
- Valid Light Distance: This parameter determines the maximum distance up to which the light will be effective. The user can adjust this setting to control how far the light reaches in the scene. Increasing the valid light distance will make the light cover a larger area, while decreasing it will limit the range of the light;
- Light Angle: This setting controls the angle of the light cone emitted by the light source. The user can adjust this parameter to change the spread of the light. A wider angle will result in a larger coverage area, while a narrower angle will create a more focused or spotlight effect;

  Light Shadow: This option allows the user to enable or disable shadows cast by the light source. Enabling shadows will create realistic lighting effects in the scene, with objects casting shadows based on the position and intensity of the light source. Disabling shadows will result in a flat or unrealistic lighting appearance;

These settings in the “Light” panel provide the user with control over various properties of the light source, allowing the user to fine-tune the lighting in your scene to achieve the desired visual effect.

(7) Using the “Dynamic” panel located in the right of the simulator, the user can set the colon deformation with different frequencies and scales;

(8) In the right panel named “Camera”, the user can set the “field of view”, “Focal length”, and “Stereo or Mono” working modes. If the camera is set to “Stereo”, the lenses distance can be set;

(9) Some notes: all the positions are defined in the world space, and the camera start position (first frame) is the center (0,0,0) of the world space. There is no lens distortion of the virtual cameras and all the captured images are without distortions.

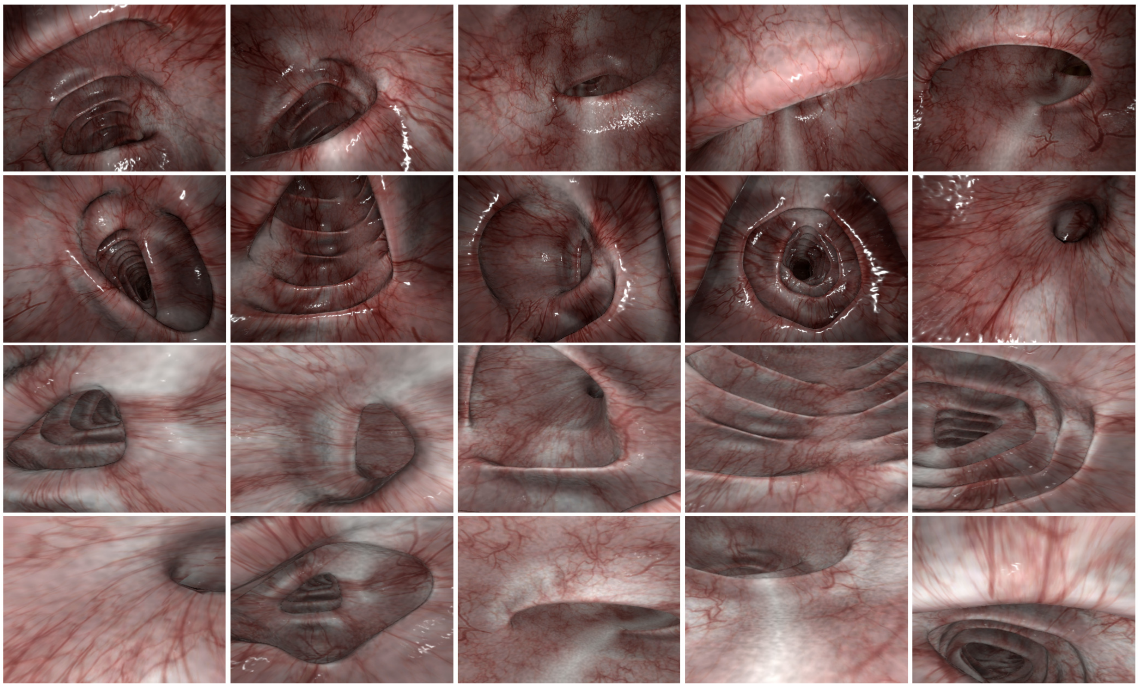


FIGURE 3.15: **Examples of Colonoscopic Images Obtained from the Simulator**. The virtual camera was repositioned to various anatomical regions of the colon in order to capture images from different perspectives and distances.

Fig. 3.15 show some examples of colonoscopic images obtained from the simulator. The parameters for configuring the simulator to generate the exampled colonoscopic images are as following:

- Capture: shots resolution - 320 × 240
- Material: hue - 0, saturation - 48, wetness - 100, vessel size - 84, vessel opacity - 41;

- Light: Light Angle - 150°, Light Intensity - 39, Valid Light Distance - 82 mm, Light Shadow - False
- Dynamic - False
- Camera: FOV - 74°, focal length - 4.969783 mm,

3.7 Chapter Summary

In this chapter, we introduce the brief development processing of the colonoscopy simulator. It can simulate the colonscopy procedures and provide experimental datasets in different scenarios. The main advantages of the simulator include: (1) the dataset with pose ground truth can be used to develop and test colonoscope camera estimation algorithms; (2) the dataset with dense depth ground truth can be used to train and test monocular colonoscopic image depth estimation networks; (3) the simulator can simulate different levels of colon deformation and help to develop colon reconstruction in deformable scenarios; (4) it can give researchers the freedom for generating customized datasets.

However, For this version of the developed simulator, there are certain texture and color differences between the simulated colonoscopic images and real images. This is primarily caused by the use of limited real images to generate the 2D texture map. In the near future, we plan to enhance the simulator by incorporating more real colonoscopic images to refine its color and texture. Meanwhile, we will also incorporate deformations caused by inflation/deflation. The new version of the developed colonoscopy simulator will be leased once we have completed its development and testing.

To encourage research in the field, we have made the developed colonoscopy simulator and datasets used in this thesis publicly available. The list of softwares used for the development of colonoscopy simulator includes:

- Unity (version 2018.2.7F1, 64-bit), which is used for programming and creating the main body and all functions of the application. The plugins used in Unity contains:
  - Playmaker, which is used for node-base programming and creating almost all functions of the simulator;

  - TextMesh Pro, which is used to have a nice and sharp texts;
  - Post Processing Stack, which is used for adding some visual effects such as Vignette, Bloom and Grain;
  - StandaloneFileBrowser, which is used for creating file open/save browser window;
  - AmplifyShaderEditor, which is used for creating the shader of the colonic urface;
  - Easy Save 3, which is used for adding save and load functions to the simulator.

- Softimage, which is used for editing the 3D model of the colon, modifying mesh normals, splitting mesh, baking ambient occlusion to mesh vertex colors, optimizing mesh triangles, and exporting meshes to ".obj" and ".fbx" formats.
- Zbrush, which is used for editing 3D model of the colon, sculpting over the mesh and also converting formats between ".stl" and ".obj" formats.
- Unfold 3D, which is used for creating the UV map of the 3D colon mesh.
- Photoshop, which is used for creating the vessels of the texture map.

Chapter 4

A Model-based 3D Reconstruction of Colon Structures and Textures from Stereo Colonoscopic Images

In this chapter, we introduce our first framework for reconstructing a 3D map of the internal surface of the colon using stereo colonoscopy, which is the main contribution of this chapter. The input of our framework is a sequence of stereo colonoscopic images and a corresponding colon mesh model segmented from pre-operative CT scans, and the final output of the framework is the reconstructed and texturized 3D colon maps. Specifically, this work will focus on resolving the following problems assuming no much deformation happens:

1. How to robustly estimate the motion of the camera inside a human colon during colonoscopy;
2. How to precisely reconstruct a complete 3D virtual colon map from stereo colonoscopic images;
3. How to map the texture from colonoscopic images to the reconstructed map.

The proposed framework is validated on datasets of different scenarios from the developed colonoscopy simulator and the accuracy of the reconstruction and texture rendering is within $[-0.04, 0.04]$ rad for Euler angles, and $[-0.5, 0.5]$ mm for translation. The rest of this chapter is organized as follows: Section 4.1 describes the proposed framework. Section 4.2 presents the technical details of the proposed framework. Section 4.3 provides validation and experimental results. Section 4.4 summarizes this chapter.

4.1 Overview of the Framework

Fig. 4.1 illustrates the proposed framework for reconstructing and texturing a 3D colon map from stereo colonoscopic images, which includes 3D scan reconstruction from stereo images, VO based camera initialization, geometric and photometric scan to colon model registration and barycentric-based texture rendering.

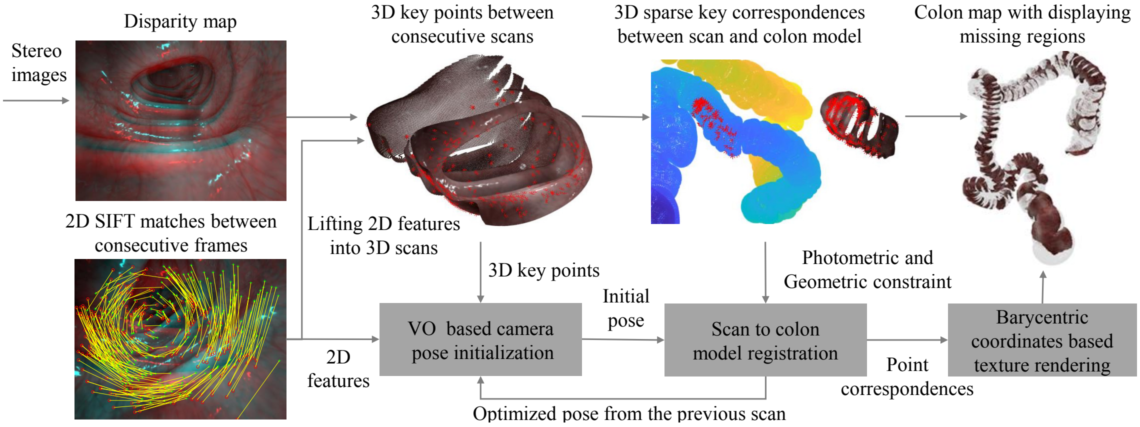

Stereo images
Disparity map
3D key points between consecutive scans
3D sparse key correspondences between scan and colon model
Colon map with displaying missing regions
2D SIFT matches between consecutive frames
Lifting 2D features into 3D scans
3D key points
Photometric and Geometric constraint
Initial pose
VO based camera pose initialization
Scan to colon model registration
Barycentric coordinates based texture rendering
2D features
Point correspondences
Optimized pose from the previous scan

FIGURE 4.1: **The Framework of Reconstructing and Texturing 3D Colon Structures From Stereo Colonoscopic Videos**

The developed colonoscopy simulator works in a way similar to a real colonoscopy, it starts to take images during the withdraw processing of the colonoscope, which means the reconstruction processing starts from the distal end of the human colon. Therefore, the 3D colon map is initially reconstructed by the geometric-only ICP registration between the first estimated scan and the colon model. Then, each time when a new frame is incorporated, the relative pose between the current scan and the previous scan is estimated by the VO module. As a result, this relative pose combined with the optimized pose between the previous scan and the colon model estimated in the last step is used as the initial guess of the relative pose between the current scan and the colon model.

This initial guess sets the current scan to a good initial position for registration between itself and the colon model. After that, the developed geometric and photometric based scan registration is applied between the current scan and the colon model. Hence, the pose of current scan is optimized and dense point correspondences between the scan and the vertices of the colon model are established from the proposed registration processing. Based on the established point correspondences, texture coordinates between 2D color images and the colon model are extracted using the barycentric-based mapping algorithm. Section 4.2 will explain all the modules in details.

4.2 Technical Details

The proposed framework includes 3D scan (point cloud with RGB information) reconstruction from stereo images, a visual odometry (VO) based camera pose initialization module, a 3D registration scheme for matching texture scans to the segmented colon model, and a barycentric-based texture rendering module for mapping textures from colonoscopic images to the reconstructed colonic surface.

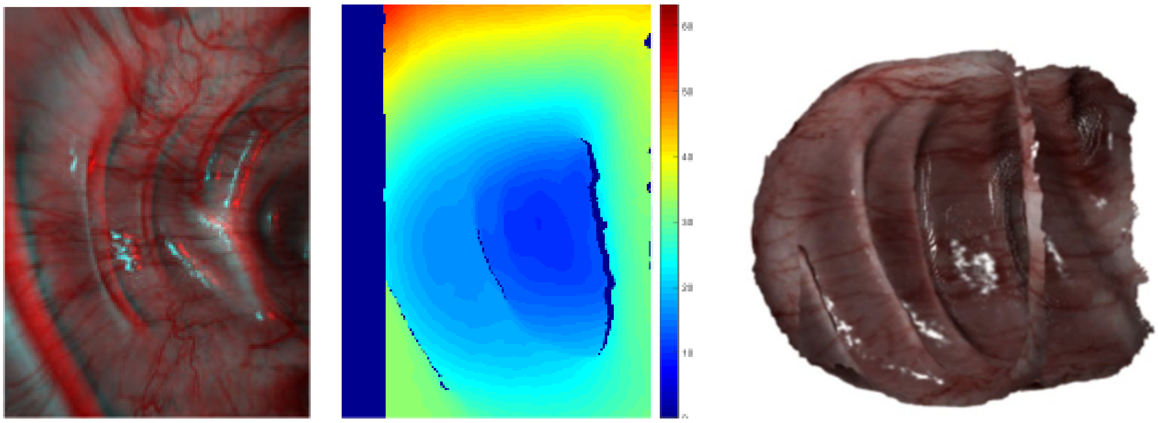


Figure 4.2: **3D Scan Reconstruction from Disparity Map Through SGM**. The left shows Red-Cyan composite view of the rectified stereo pair image; the middle shows the disparity map; the right shows reconstructed 3D scan.

4.2.1 3D Scan Reconstruction from Stereo Images

The SGM algorithm [63] is used as the scan reconstruction method. As shown in Fig. 4.2, first, create the stereo anaglyph of the rectified stereo pair images. Second, compute the disparity map from the pair of rectified stereo images using SGM algorithm. Then, the 3D coordinates of the pixel points in the camera coordinate frame are computed to reconstruct a 3D scan and each 3D scan has one to one correspondence to a corresponding 2D image.

Fig. 4.3 shows an example of ground truth scans and corresponding reconstructed 3D scans, respectively.

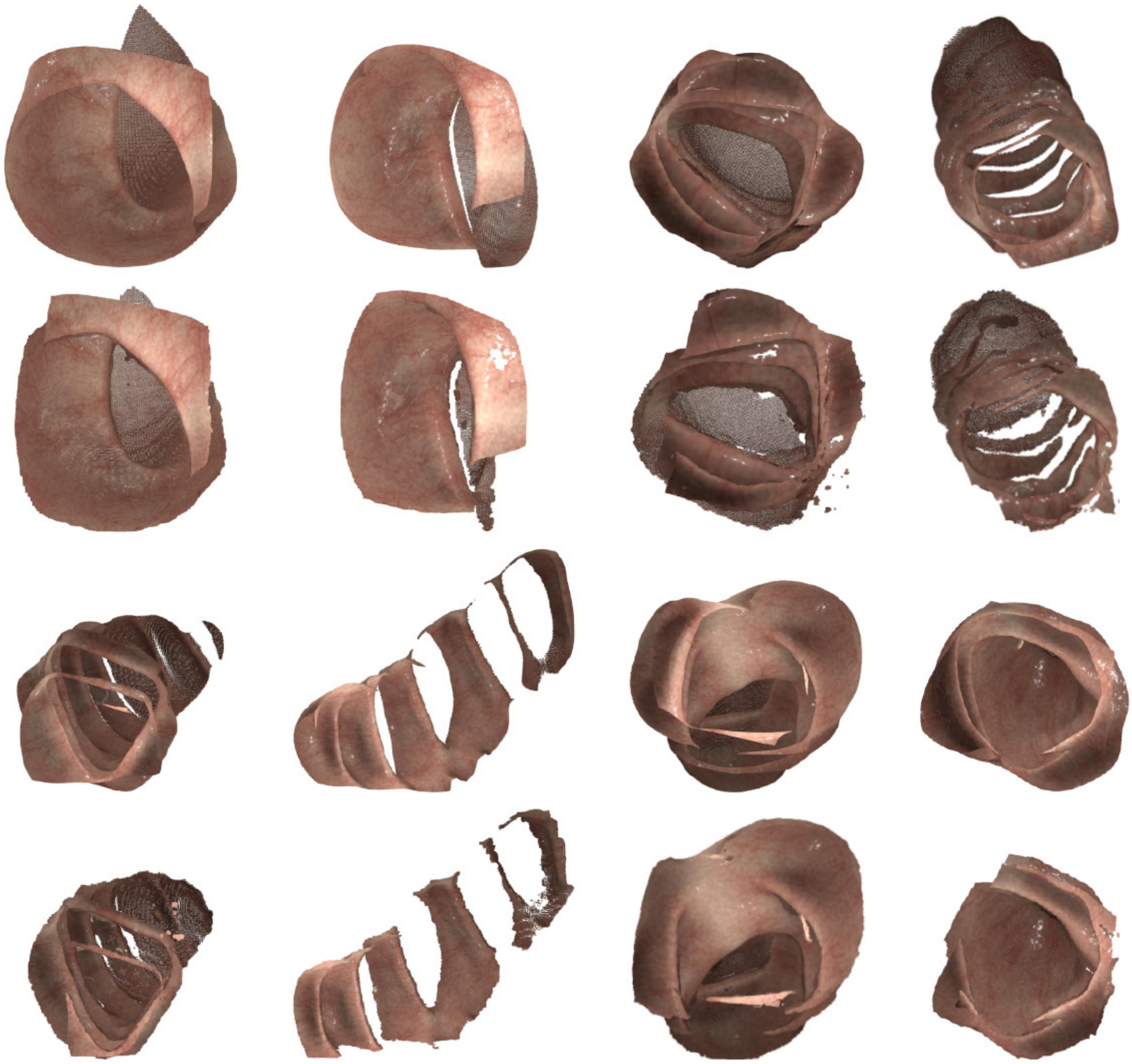


FIGURE 4.3: **Examples of reconstructed scans and ground truth.** The first and third rows show ground truth scans, the second and last rows show corresponding scans reconstructed from stereo images.

4.2.2 Sparse Key Correspondences and Camera Pose Initialization

In the VO based camera motion initialization module, as shown in the following Fig. 4.4, first, two disparity maps are computed from the current and the previous pairs of stereo images and the corresponding two 3D scans can be computed from the disparity maps. Then, SIFT features are extracted and matched between the consecutive left images. For an accurate motion estimation, the RANSAC algorithm is used to remove outliers from the set of 2D SIFT feature correspondences. After that, these 2D SIFT features are migrated into 3D scans by tracing the pixel indices of these 2D SIFT points in their corresponding 3D scans, and a set of 3D key point correspondences (anchor points) between the two scans are acquired.

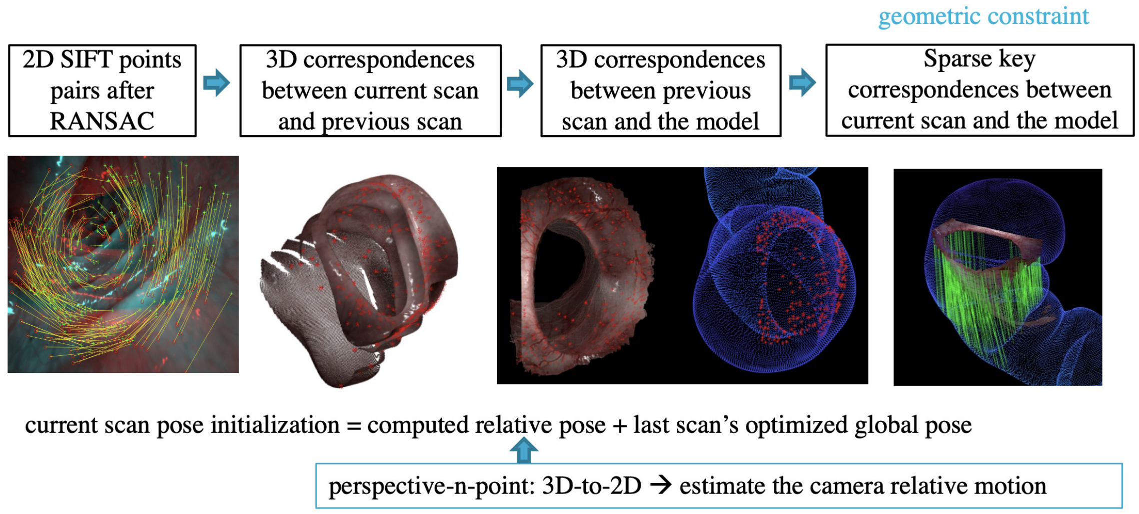
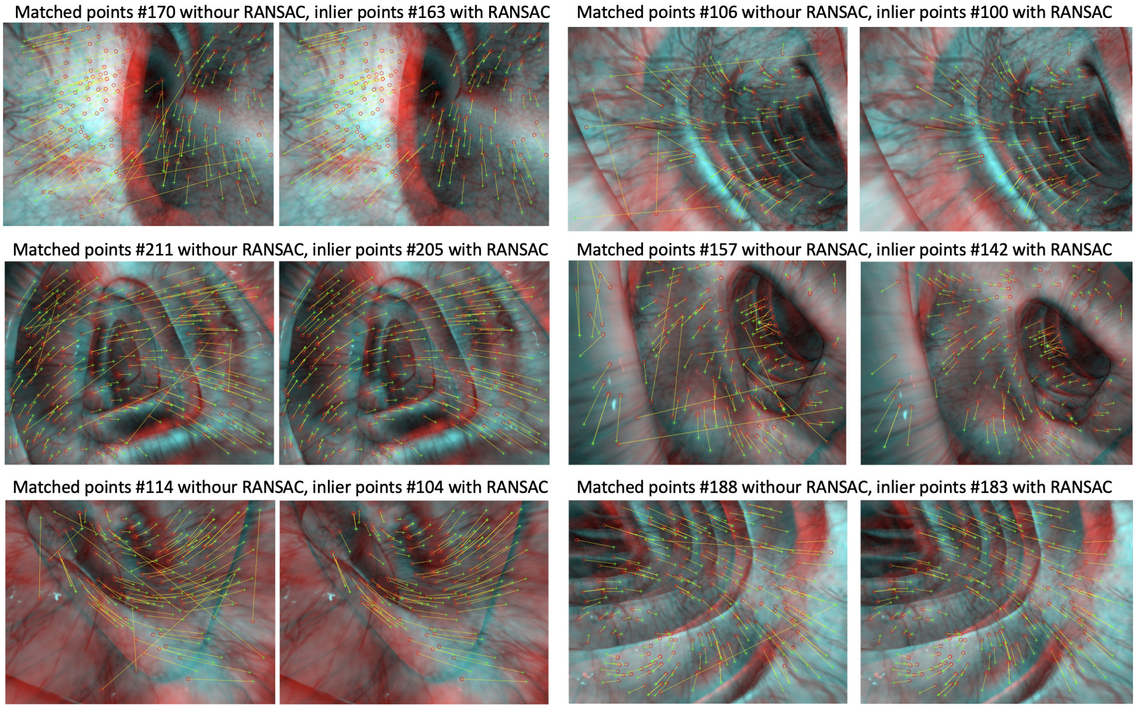

geometric constraint
2D SIFT points pairs after RANSAC
3D correspondences between current scan and previous scan
3D correspondences between previous scan and the model
Sparse key correspondences between current scan and the model
current scan pose initialization = computed relative pose + last scan's optimized global pose
perspective-n-point: 3D-to-2D → estimate the camera relative motion

Figure 4.4: **VO: Sparse Key Correspondences and Camera Pose Initialization.**

In our experiments, we extract and match SIFT features from the simulated datasets Case 1 to Case 15 (refer to Table 4.1), respectively. The average number of successfully matched SIFT features for consecutive images with resolution 640 × 480 is 129. After applying the RANSAC algorithm on the SIFT matches, the mean rate of outliers is 9.7%. Fig. 4.5 shows some examples of extracted SIFT features from simulated colonoscopy images.

We also calculate the rate of SIFT match outliers on real in-vivo colonoscopic images, the average number of successfully matched SIFT features between consecutive images with resolution 270 × 216 is 116. After applying the RANSAC algorithm on the SIFT matches,



Matched points #170 withour RANSAC, inlier points #163 with RANSAC

Matched points #106 withour RANSAC, inlier points #100 with RANSAC

Matched points #211 withour RANSAC, inlier points #205 with RANSAC

Matched points #157 withour RANSAC, inlier points #142 with RANSAC

Matched points #114 withour RANSAC, inlier points #104 with RANSAC

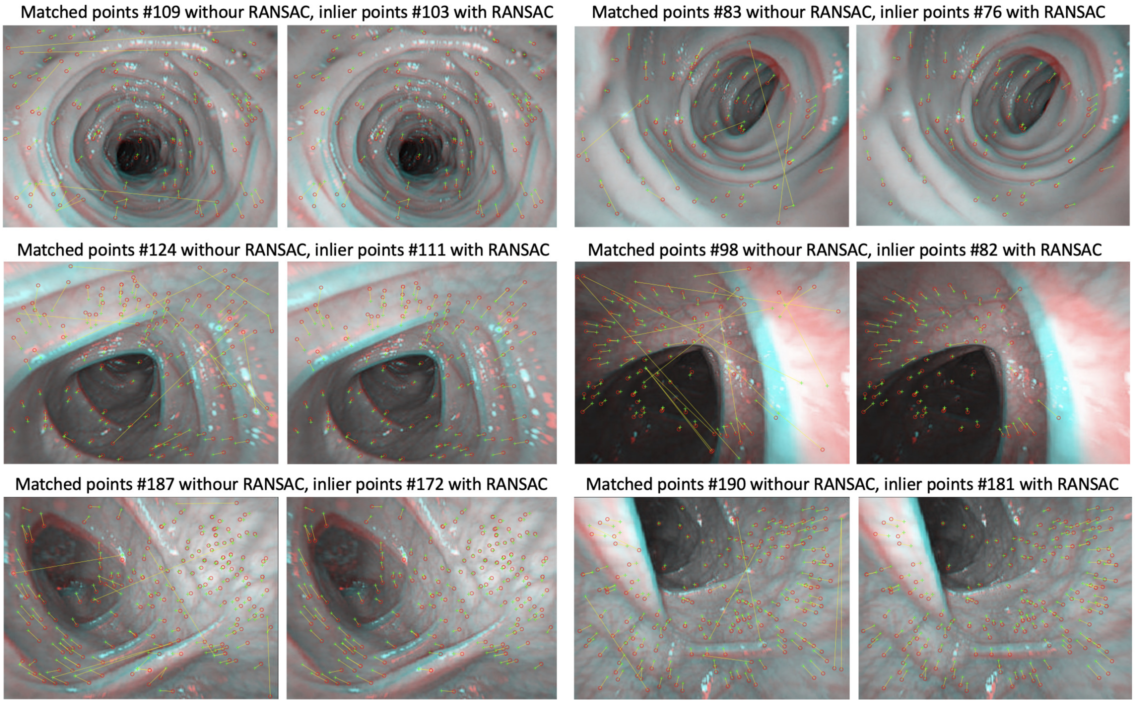
Matched points #188 withour RANSAC, inlier points #183 with RANSAC

FIGURE 4.5: **2D SIFT Matches Between Consecutive Simulated Colonoscopic Images:** the first column and the third column show SIFT feature matches with many outliers; the second and the fourth column show the corresponding SIFT feature matches after using RANSAC algorithm.

the mean rate of outliers is 8.2%. Fig. 4.6 shows some examples of extracted SIFT features from real colonoscopy images.

It should be noted that only clearly visible consecutive frames with specific overlaps have been tested. This is because fast camera motion can lead to motion blur and the loss of feature correspondence, which can result in inaccurate camera pose estimation and 3D reconstruction. If the camera speed is increased, the proposed colon reconstruction framework can still function effectively only when frames remain clearly visible and there are certain overlaps between consecutive frames.

Since the 3D-to-2D method is more accurate than 3D-to-3D methods [64] and the RANSAC algorithm can help to remove outliers. After acquiring 2D SIFT feature point correspondences and corresponding 3D anchor point correspondences from the SIFT approach. The P3P algorithm in conjunction with the RANSAC algorithm are applied [65] on 3D-to-2D point correspondences to estimate the camera motion robustly. Meanwhile, the RANSAC algorithm is used in conjunction with existing solutions to make the final solution for


Matched points #109 withour RANSAC, inlier points #103 with RANSAC

Matched points #83 withour RANSAC, inlier points #76 with RANSAC

Matched points #124 withour RANSAC, inlier points #111 with RANSAC

Matched points #98 withour RANSAC, inlier points #82 with RANSAC

Matched points #187 withour RANSAC, inlier points #172 with RANSAC

Matched points #190 withour RANSAC, inlier points #181 with RANSAC

FIGURE 4.6: **2D SIFT Matches Between Consecutive Real Colonoscopic Images:** the first column and the third column show SIFT feature matches with many outliers; the second and the fourth column show the corresponding SIFT feature matches after using RANSAC algorithm.

the camera pose more robust to outliers. In details, we recover the camera motion iteratively using the P3P algorithm and eliminate spurious point correspondences using the M-estimator sample consensus (MSAC) [66] algorithm which is a variant of RANSAC algorithm. In each iteration, a subset of four points correspondences are randomly selected and get up to 4 solutions for the pose using three pairs of points, then choose the best solution using the 4th point pair. After that, computing the reprojection errors in pixels for all the points using the estimated pose and finding outliers from the set of all points fit with a predefined threshold of reprojection error. If the fraction of inliers over the total number points in the set exceeds a predefined threshold, the model parameters are re-estimated using only the identified inliers, and the process is terminated. Otherwise, repeating the above steps for a prescribed maximum number of iterations. Finally, this relative pose between the current scan and the previous scan is then combined with the optimized pose of the previous scan and used as the initial pose of the current scan in the scan-to-model registration processing described in Section 4.2.3.

The proposed approach involves determining the relative pose between a current scan and a previous scan in the scan-to-model registration process. This is achieved by first estimating the pose using all available point correspondences between the two scans, followed by computing the reprojection errors in pixels for all points using the estimated pose. Outliers are identified using a predefined threshold of reprojection error, and if the fraction of inliers exceeds a predefined threshold, the model parameters are re-estimated using only the identified inliers, and the process is terminated. Otherwise, the aforementioned steps are repeated for a prescribed maximum number of iterations. Finally, the relative pose between the current and previous scans is combined with the optimized pose of the previous scan to obtain the initial pose of the current scan for the scan-to-model registration process, as described in Section 4.2.3.

4.2.3 Scan to Colon Model Registration

One can build the 3D colon map by incrementally registering all the scans together, but the errors of poses estimation accumulate during scan to scan registration. Also, only the geometric constraint applied on the registration causes inconsistency of texture matching in the overlapping region of two scans. To address these problems, we formulate an objective function by combining the geometric constraint and the photometric feature constraint:

$$E(T) = (1 - \sigma)E_G(T) + \sigma E_F(T), \tag{4.1}$$

where $E_G(T)$ is the geometric term of the objective function and the $E_F(T)$ is the photometric feature term provided by the pair-wise 3D sparse anchor points generated from 2D SIFT features described in Section 4.2.2, $\sigma \in [0, 1]$ is the weight that balances the two terms. Here "*photometric*" is used to express that these constraints are from the texture information instead of the geometric structure. Our goal is to find the optimal transformation T that best aligns the reconstructed scan to the colon model.

The geometric term $E_G(T)$ sums all the squared distances between each source point $\mathbf{s}_i = [s_{ix}, s_{iy}, s_{iz}, 1]^T$ in a scan and the tangent plane at its closest point $\mathbf{d}_i = [d_{ix}, d_{iy}, d_{iz}, 1]^T$ in the colon model:

$$E_G(T) = \sum_i ((T \cdot \mathbf{s}_i - \mathbf{d}_i) \bullet \mathbf{n}_i)^2 \tag{4.2}$$

where $\mathbf{n}_i = [n_{ix}, n_{iy}, n_{iz}, 0]^T$ is the unit normal vector at $\mathbf{d}_i$, and "$\bullet$" denotes the dot product.

Similarly, the photometric term $E_F(T)$ sums all the point-to-point distances between the 3D anchor point $\mathbf{s}_j^f = [s_{jx}^f, s_{jy}^f, s_{jz}^f, 1]^T$ in a scan and its corresponding 3D anchor point $\mathbf{d}_j^f = [d_{jx}^f, d_{jy}^f, d_{jz}^f, 1]^T$ in the colon mesh, provided in Section 4.2.2:

$$E_F(T) = \sum_j (T \cdot \mathbf{s}_j^f - \mathbf{d}_j^f) \bullet (T \cdot \mathbf{s}_j^f - \mathbf{d}_j^f). \tag{4.3}$$

4.2.4 Optimization Details

We minimize the objective function $E(T)$ of the non-linear least-squares problem by linear approximation to the rotation matrix [67]. At the k^{th} iteration, T can be expressed as following:

$$T = \Delta T \cdot T^k \tag{4.4}$$

where T^k is the global transformation estimated in the last iteration and ΔT is the incremental 3D rigid-body transformation which is composed of a rotation matrix $R(\alpha, \beta, \gamma)$ and a translation matrix $\mathrm{t}(t_x, t_y, t_z)$:

$$\Delta T = \mathrm{t}(t_x, t_y, t_z) \cdot R(\alpha, \beta, \gamma) \tag{4.5}$$

where

$$\mathrm{t}(t_x, t_y, t_z) = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

and

$$R(\alpha,\beta,\gamma) = R_z(\gamma)\cdot R_y(\beta)\cdot R_x(\alpha)$$
$$= \begin{bmatrix} \cos\gamma\cos\beta & -\sin\gamma\cos\alpha + \cos\gamma\sin\beta\sin\alpha & \sin\gamma\sin\alpha + \cos\gamma\sin\beta\cos\alpha & 0 \\ \sin\gamma\cos\beta & \cos\gamma\cos\alpha + \sin\gamma\sin\beta\sin\alpha & -\cos\gamma\sin\alpha + \sin\gamma\sin\beta\cos\alpha & 0 \\ -\sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.7}$$

$R_x(\alpha)$, $R_y(\beta)$, $R_z(\gamma)$ are rotations of the angles α, β and γ around the x-axis, y-axis and z-axis, respectively. When the incremental rotations of each iteration are small, it can be approximated as following:

$$\mathbf{R}(\alpha,\beta,\gamma) \approx \begin{bmatrix} 1 & \alpha\beta-\gamma & \alpha\gamma+\beta & 0 \\ \gamma & \alpha\beta\gamma+1 & \beta\gamma-\alpha & 0 \\ -\beta & \alpha & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \approx \begin{bmatrix} 1 & -\gamma & \beta & 0 \\ \gamma & 1 & -\alpha & 0 \\ -\beta & \alpha & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.8}$$

Then, T is approximated by:

$$T \approx \begin{bmatrix} 1 & -\gamma & \beta & t_x \\ \gamma & 1 & -\alpha & t_y \\ -\beta & \alpha & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot T^k \tag{4.9}$$

Each $(T\cdot \mathrm{s}_i - \mathrm{d}_i)\bullet \mathrm{n}_i$ in (4.2) can be written as a linear expression of the six parameters α, β, γ, t_x, t_y and t_z:

$$(T\cdot \mathrm{s}_i - \mathrm{d}_i)\bullet \mathrm{n}_i = [\bar{\mathrm{s}}_i \times \mathrm{n}_i, \mathrm{n}_i^T]\cdot[\alpha,\beta,\gamma,t_x,t_y,t_z]^T - [\mathrm{d}_i - \bar{\mathrm{s}}_i]\bullet \mathrm{n}_i \tag{4.10}$$

where $\bar{\mathrm{s}}_i = T^k\cdot \mathrm{s}_i$. Given N_1 pairs of point correspondences in term $E_G(T)$, we can arrange all $(T\cdot \mathrm{s}_i - \mathrm{d}_i)\bullet \mathrm{n}_i$, $1\le i\le N_1$, into a matrix expression:

$$A_1\mathrm{x} - \mathrm{b}_1 \tag{4.11}$$

where A_1 is a N_1 by 6 matrix, b_1 is a N_1 by 1 vector and $\mathrm{x} = [\alpha, \beta, \gamma, t_x, t_y, t_z]^T$ is a 6 by 1 vector:

$$A_1 = \begin{bmatrix} \bar{\mathrm{s}}_1 \times \mathrm{n}_1,\, \mathrm{n}_1^T \\ ... \\ \bar{\mathrm{s}}_i \times \mathrm{n}_i,\, \mathrm{n}_i^T \\ ... \\ \bar{\mathrm{s}}_{N_1} \times \mathrm{n}_{N_1},\, \mathrm{n}_{N_1}^T \end{bmatrix},\ \mathrm{b}_1 = \begin{bmatrix} [\mathrm{d}_1 - \bar{\mathrm{s}}_1] \bullet \mathrm{n}_1 \\ ... \\ [\mathrm{d}_i - \bar{\mathrm{s}}_i] \bullet \mathrm{n}_i \\ ... \\ [\mathrm{d}_{N_1} - \bar{\mathrm{s}}_{N_1}] \bullet \mathrm{n}_{N_1} \end{bmatrix} \tag{4.12}$$

Similarly, each $(T \cdot \mathrm{s}_j^f - \mathrm{d}_j^f)$ in (4.3) can also be written as a linear expression group of x:

$$T \cdot \mathrm{s}_j^f - \mathrm{d}_j^f = \begin{bmatrix} [\bar{s}_{jz}^f \cdot \beta - \bar{s}_{jy}^f \cdot \gamma + t_x] - [d_{jx}^f - \bar{s}_{jx}^f] \\ [\bar{s}_{jx}^f \cdot \gamma - \bar{s}_{jz}^f \cdot \alpha + t_y] - [d_{jy}^f - \bar{s}_{jy}^f] \\ [\bar{s}_{jy}^f \cdot \alpha - \bar{s}_{jx}^f \cdot \beta + t_z] - [d_{jz}^f - \bar{s}_{jz}^f] \end{bmatrix} \tag{4.13}$$

where $\bar{\mathrm{s}}_j^f = T^k \cdot \mathrm{s}_j^f$. Given N_2 pairs of anchor point correspondences in term $E_F(T)$, we can arrange all $T \cdot \mathrm{s}_j^f - \mathrm{d}_j^f$, $1 \leq j \leq N_2$, into a matrix expression:

$$A_2\mathrm{x} - \mathrm{b}_2 \tag{4.14}$$

where A_2 is a $N_2 \times 3$ by 6 matrix and b_2 is $N_2 \times 3$ by 1 vector:

$$A_2 = \begin{bmatrix} A_{21}^T & ... & A_{2j}^T & ... & A_{2N_2}^T \end{bmatrix}^T,\ \mathrm{b}_2 = \begin{bmatrix} \mathrm{b}_{21}^T & ... & \mathrm{b}_{2j}^T & ... & \mathrm{b}_{2N_2}^T \end{bmatrix}^T \tag{4.15}$$

$$\text{with } A_{2j} = \begin{bmatrix} 0 & \bar{s}_{jz}^f & -\bar{s}_{jy}^f & 1 & 0 & 0 \\ -\bar{s}_{jz}^f & 0 & \bar{s}_{jx}^f & 0 & 1 & 0 \\ \bar{s}_{jy}^f & -\bar{s}_{jx}^f & 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } \mathrm{b}_{2j} = \begin{bmatrix} d_{jx}^f - \bar{s}_{jx}^f \\ d_{jy}^f - \bar{s}_{jy}^f \\ d_{jz}^f - \bar{s}_{jz}^f \end{bmatrix}.$$

Therefore, we can obtain the optimal x by solving for:

$$\min_{\mathrm{x}} (1-\sigma)|A_1\mathrm{x} - \mathrm{b}_1|^2 + \sigma|A_2\mathrm{x} - \mathrm{b}_2|^2, \tag{4.16}$$

which is a linear least-squares problem, and can be solved by setting the derivative of the objective function with respect to the x to zero. Then, the solution is:

$$\mathrm{x}_{opt} = ((1-\sigma) \cdot A_1^T \cdot A_1 + \sigma \cdot A_2^T \cdot A_2)^{-1} \cdot ((1-\sigma) \cdot A_1^T \cdot \mathrm{b}_1 + \sigma \cdot A_2^T \cdot \mathrm{b}_2) \tag{4.17}$$

Since the obtained solution is an approximation, we will apply it to (4.5) to map the estimated transformation into $\mathbb{SE}(3)$. In each iteration, we solve the linear system in (4.16), and update T by applying the incremental transformation ΔT to T^k using (4.4). In the next iteration, we rep-linearize T around T^{k+1} and repeat.

In (4.17), We use the parameter σ to balance the geometric and the photometric term. If the value of σ is too large, the optimization objective will be more close to the point-to-point objective in the proposed registration algorithm and the optimal solution will mainly depend on the relatively small number of the 3D anchor correspondences provided by the 2D SIFT approach which represent texture features, but the optimal solution may not be reliable when some anchor correspondences are not correct or less accurate. However, if the value of σ is too small, the optimization objective will become close to the geometric term and the feature based regulation term becomes less effect on the optimal solution, and this causes inconsistency of texture matching in the overlapping region of two scans. In this work, we set σ to 0.5.

Once the optimization processing is finished, the optimal pose is estimated and point correspondences between the scan and vertices of the colon model are established for texture rendering described in Section 4.2.5.

4.2.5 Texture Mapping using Barycentric Coordinates

One can assign RGB color data from points in each scan to the corresponding vertices in the colon mesh, then color each pixel of a triangle face by interpolating between the colors of the three vertices in the colon mesh model. However, the texture in triangle faces will be blurry since the vertices in the colon mesh are much sparser than the point cloud in the scans and one vertex in the colon mesh may correspond to multiple points in a scan.

Thus, in this work, we use a barycentric based texture rendering technique to map textures from colonoscopic images to the reconstructed colonic surface. As we can see from Fig. 4.7, for three vertices A, B, C of one triangular $\triangle$ABC face in the colon mesh, we can extract their matched points in the 3D reconstructed scan by referring to the established point correspondences between the scan and the vertices of colon mesh. Furthermore, as each

3D point in a scan corresponds to a 2D pixel in a 2D image when reconstructs the scan, we can extract a triangular texture region $\triangle$abc (where a, b, and c are the 2D location of three vertices of the triangle) in 2D images corresponding to each triangle $\triangle$ABC face in the colon mesh.

After that, we use barycentric mapping technique [68] to map pixel color from the 2D texture region $\triangle$abc to the 3D triangle $\triangle$ABC face. For an arbitrary 3D point $\mathrm{P}(x_{\mathrm{P}}, y_{\mathrm{P}}, z_{\mathrm{P}})$ inside the triangle $\triangle$ABC, there is a unique sequence of three numbers, $\lambda_1 \geq 0, \lambda_2 \geq 0, \lambda_3 \geq 0$ to represent it:

$$\begin{cases} x_{\mathrm{P}} = \lambda_1 x_{\mathrm{A}} + \lambda_2 x_{\mathrm{B}} + \lambda_3 x_{\mathrm{C}} \\ y_{\mathrm{P}} = \lambda_1 y_{\mathrm{A}} + \lambda_2 y_{\mathrm{B}} + \lambda_3 y_{\mathrm{C}} \\ z_{\mathrm{P}} = \lambda_1 z_{\mathrm{A}} + \lambda_2 z_{\mathrm{B}} + \lambda_3 z_{\mathrm{C}} \\ 1 = \lambda_1 + \lambda_2 + \lambda_3 \end{cases} \tag{4.18}$$

where λ_1, λ_2, λ_3 indicate the barycentric coordinates of the point p with respect to the triangle. Once we have the barycentric coordinates, the texture coordinates of P can be determined by interpolating the texture values at the vertices using the barycentric coordinates as weights:

$$\begin{cases} u_{\mathrm{p}} = \lambda_1 u_{\mathrm{a}} + \lambda_2 u_{\mathrm{b}} + \lambda_3 u_{\mathrm{c}} \\ v_{\mathrm{p}} = \lambda_1 v_{\mathrm{a}} + \lambda_2 v_{\mathrm{b}} + \lambda_3 v_{\mathrm{c}} \end{cases} \tag{4.19}$$

Overall, it takes the following steps to texturize the reconstructed colonic surface from multiple colonoscopic 2D images:

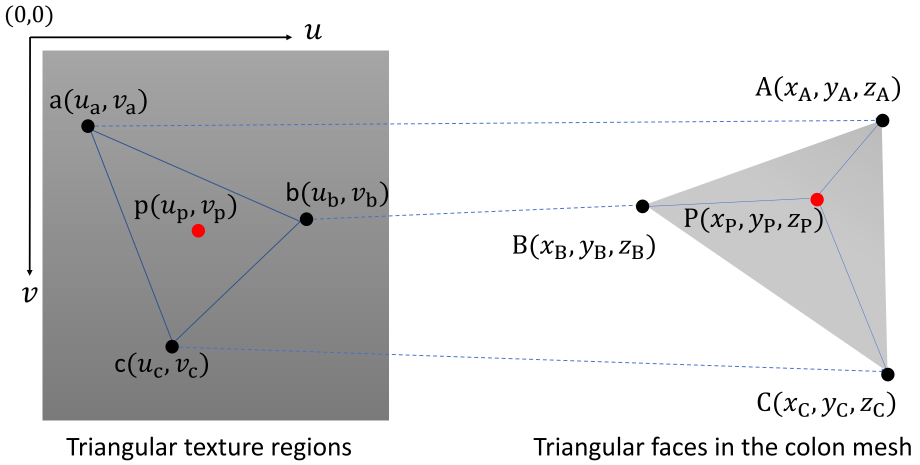
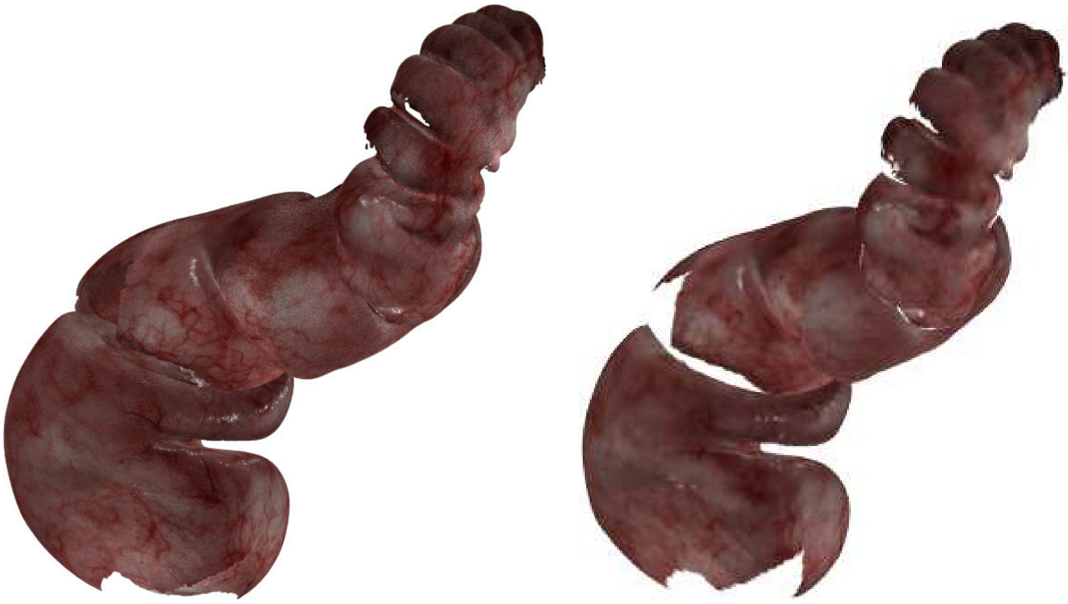

(0,0)
u
v
$\mathrm{a}(u_{\mathrm{a}}, v_{\mathrm{a}})$
$\mathrm{p}(u_{\mathrm{p}}, v_{\mathrm{p}})$
$\mathrm{b}(u_{\mathrm{b}}, v_{\mathrm{b}})$
$\mathrm{c}(u_{\mathrm{c}}, v_{\mathrm{c}})$
$\mathrm{A}(x_{\mathrm{A}}, y_{\mathrm{A}}, z_{\mathrm{A}})$
$\mathrm{B}(x_{\mathrm{B}}, y_{\mathrm{B}}, z_{\mathrm{B}})$
$\mathrm{P}(x_{\mathrm{P}}, y_{\mathrm{P}}, z_{\mathrm{P}})$
$\mathrm{C}(x_{\mathrm{C}}, y_{\mathrm{C}}, z_{\mathrm{C}})$
Triangular texture regions
Triangular faces in the colon mesh

FIGURE 4.7: **Barycentric Coordinates Based Texture Mapping.**

- Establishing triangular texture region in 2D texture images for each triangle face in the colon model;
- Using a set of barycentric coordinates to interpolate arbitrary points inside each triangle face in the colon model;
- Calculating each interpolated point's texture coordinates in its corresponding triangular texture region based on its barycentric weights;
- Mapping textures from the triangular texture region to the triangle in the colon model.

Fig. 4.8 shows the texture quality comparison between the proposed approach and patch coloring approach. We can find that the texture quality from the proposed texture rendering method is more clear and accurate.



(a) Barycentric-based texture rendering (b) Patch rendering

FIGURE 4.8: **Texturized Rectum Colon using Two Different Texture Rendering Approaches.** (a) shows barycentric coordinates based texture rendering; (b) shows texture rendering from patch rendering.

4.3 Experiments and Results

In the experiments, we begin with showing the limitations of the state-of-art SLAM algorithms Kintinuous [49], ElasticFusion [48], KinectFusion [69], ORB-SLAM2 [42] and StereoDSO [47] to colonoscopic datasets captured in scenarios simulating the real normal colonoscopy screening as well as in scenarios where the camera is operated with very slow camera motion. Then, we validate the robustness and accuracy of the proposed framework using 15 different datasets collected in different scenarios using the developed colonoscopy simulator. Finally, an in-vivo video sequence is used to demonstrate the practicality of the proposed framework. Note that the experiments with state-of-the-art RGB-D SLAM algorithms are not trying to make comparisons, but to show the limitations of these methods when applied to colonoscopic images.

Table 4.1: A Brief Summary of Data for Evaluating the Proposed Framework

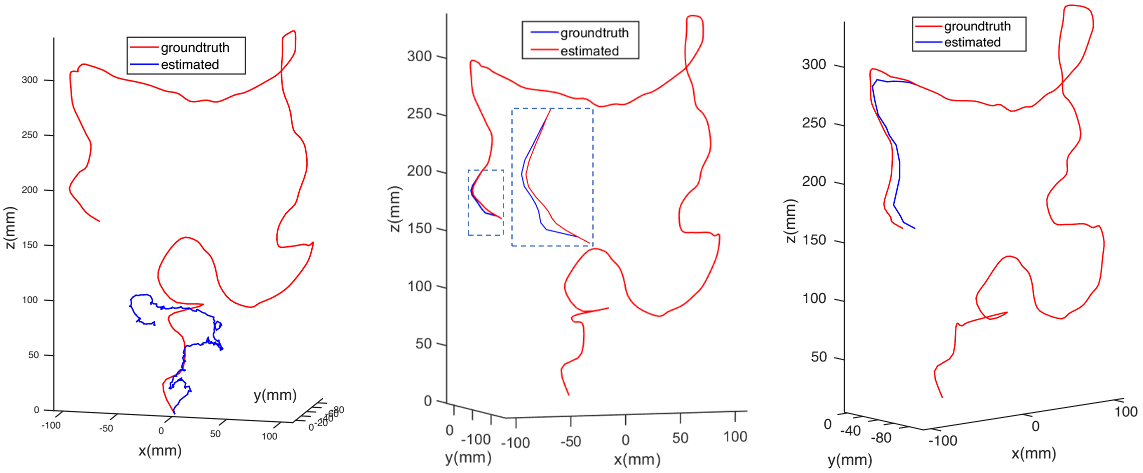
Case	**Frames**	**Path**	**Case**	**Frames**	**Path**	**Case**	**Frames**	**Path**
0	6000	manual	1	259	auto	2	260	auto
3	260	auto	4	260	auto	5	260	auto
6	260	auto	7	845	manual	8	362	manual
9	279	manual	10	192	fully	11	618	fully
12	859	fully	13	679	fully	14	339	fully
15	150	fully						

The resolution of all collected colonoscopic images is 640 × 480, the baseline of stereo camera is set to $4.5mm$ and the camera field of view is set to 74° with corresponding focal length $4.969mm$. "*auto*" represents that datasets were auto captured on the simulator-planned camera flight paths; "*manual*" represents that datasets were manually captured by three people with different clinical skills; "$fully$" represents that datasets were captured on the designed camera flight paths that aim to fully recover the internal surface of the colon.

The summary of experimental datasets captured using our developed simulator is shown in Table 4.1. For the colonoscopy simulator, the virtual camera can work in two modes, one is automatic flying mode and the other is manually controlling mode. In the automatic mode, we can set the camera's total flight time to fly through the whole colon and set the number of captured images per second, in the experiments, the framerate is set to 4. In the manually controlling mode, we rotate and move the camera using a keyboard and capture images in different camera poses. Therefore, the actual frame rate of the video is not specified in this mode.

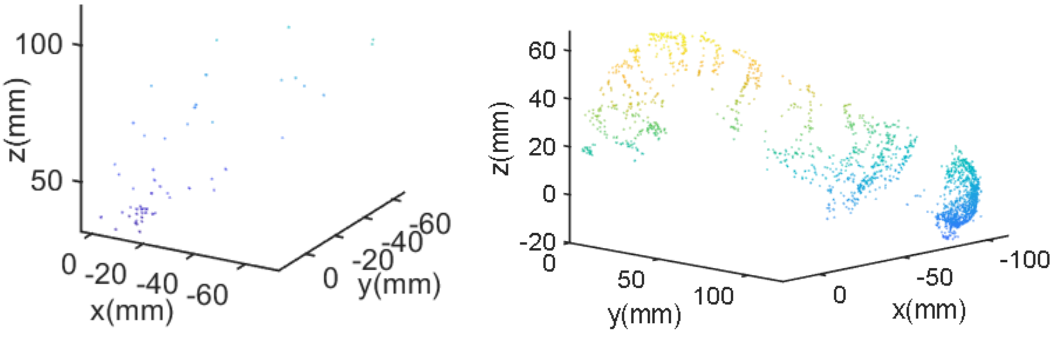
4.3.1 Evaluation of RGB-D and Stereo SLAM Systems on Colonoscopic Images

We run all the SLAM algorithms in offline mode. For RGB-D SLAM algorithms Kintinuous, ElasticFusion and KinectFusion, the images from the left camera together with the corresponding ground truth depth are used. The paired stereo color image sequences are input into ORB-SLAM2 and StereoDSO to reconstruct maps.



(a) ElasticFusion (b) StereoDSO (c) ORB-SLAM2

Figure 4.9: **Trajectories and Maps Estimated From SLAM Systems on Case 8 with Normal Camera Motion:** (a) The trajectory estimated from ElasticFusion suffers from large errors; (b) StereoDSO only obtains the trajectory of the last part of the colon; (c) ORB-SLAM2 only obtains the trajectory of the last part of the colon.



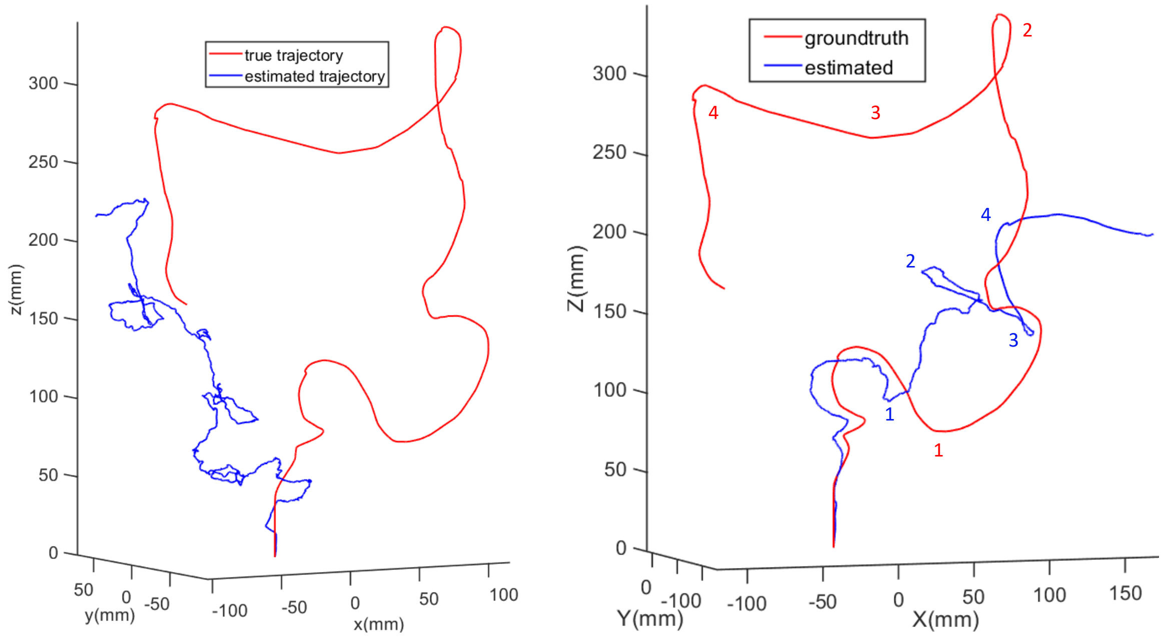
(a) StereoDSO (b) ORB-SLAM2

Figure 4.10: **Maps Estimated From SLAM Systems on Case 8 With Normal Camera Motion:** (a) StereoDSO obtains sparse point clouds; (b) ORB-SLAM2 obtains a small segment map corresponds to its trajectory.

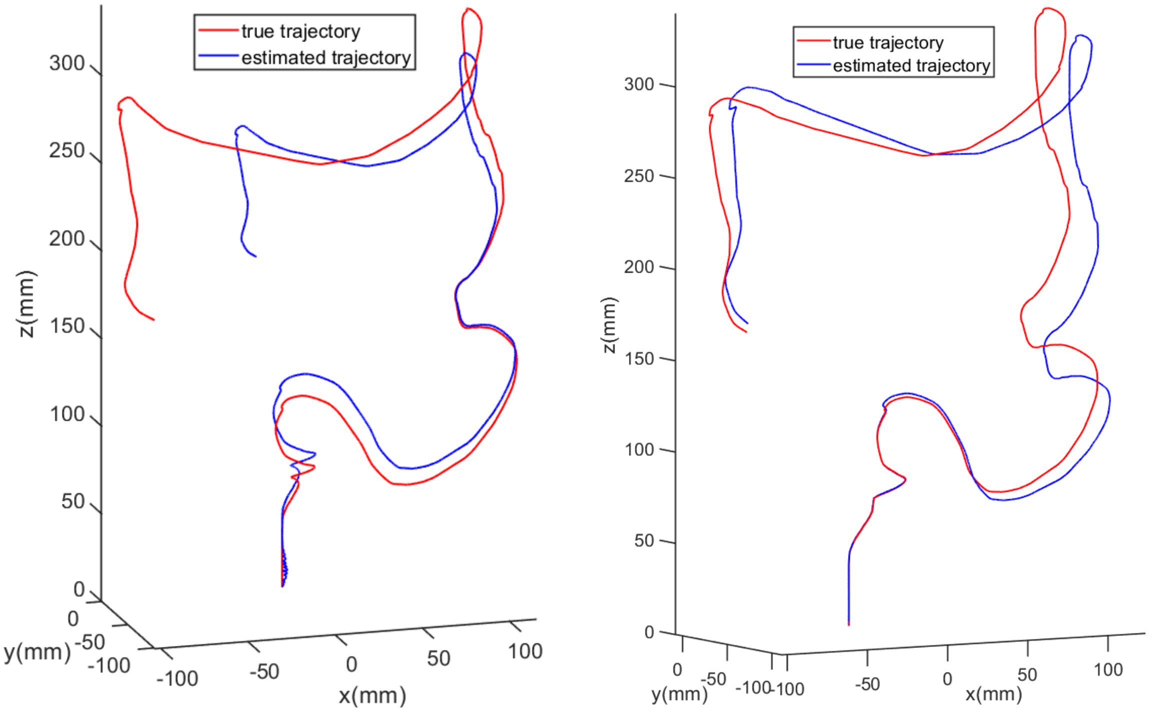
The datasets captured from a normal colonoscopy scenario (Cases 7 to 9) are first used and all the SLAM algorithms fail. Fig. 4.9 illustrates the failures using Case 8. Since the major working principle of KinectFusion relies heavily on feature matching step using ICP, it fails when the camera moves fast during the normal colonoscopy procedures. For the voxel-based Kintinuous and surfel-based ElasticFusion, fast camera motion violates the assumption behind projective data association and hinders tracking performance, so their estimated trajectories suffer from very large errors which create many outlier points and no map is generated. StereoDSO extracts candidate points from the first frame in initialization and fails to track them in the following key frames. It keeps resetting until the last segment of the colon, and thus only generates a very short trajectory with sparse point clouds (see Fig. 4.10 (a)). This also happens to ORB-SLAM2, it only obtains a small segment of sparse map (see Fig. 4.10 (b)). Therefore, the experiment results show that these SLAM systems are not suitable for map reconstruction using images from normal colonoscopy procedures.

By constrast, as shown in Fig. 4.19, our proposed method can estimate camera poses with high accuracy and reconstruct clear colon maps on Case 7, Case 8 and Case 9. Our proposed method can deal with large inter frame motions and small overlaps during normal colonoscopy, while the state-of-the-art stereo or RGB-D SLAM methods cannot. This is mainly because of the use of preoperative CT model and optimization with both geometric and photometric constraints. If there is no overlap between two frames, the optimization will be degenerated to a pure geometric registration problem. But this is not suggested since the accuracy (especially the texture accuracy) will decrease a lot because of the lack of photometric constraints.

Then, we collect a large complete set of colonoscopic image sequences with very unrealistically slow camera motions (Case 0). It contains 6k pairs of stereo color and depth images. Fig. 4.11 shows comparison of the ground truth trajectory and estimated trajectories from the different SLAM algorithms, and the reconstructed maps are shown in Fig. 4.12. It shows that Kintinuous performs poorly because the trajectory is long and has a lot of turns as well as the camera is forward facing. ElasticFusion recovers the main topological structures but the estimated trajectory is very wrong. KinectFusion is very easy to lose tracking and only able to reconstruct a small segment of the colon map. The initialization of StereDSO is slow and unstable if there are only little rotations without relatively large translations. The estimated trajectory of StereoDSO has large drift and the obtained

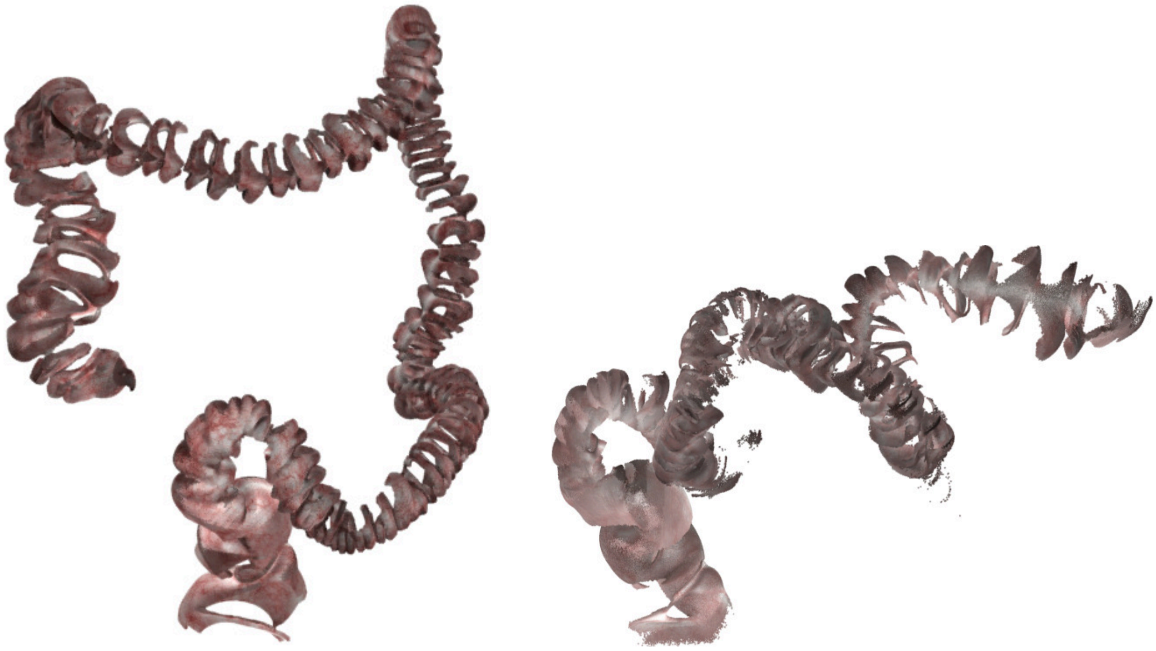


(a) Kintinuous (b) ElasticFusion

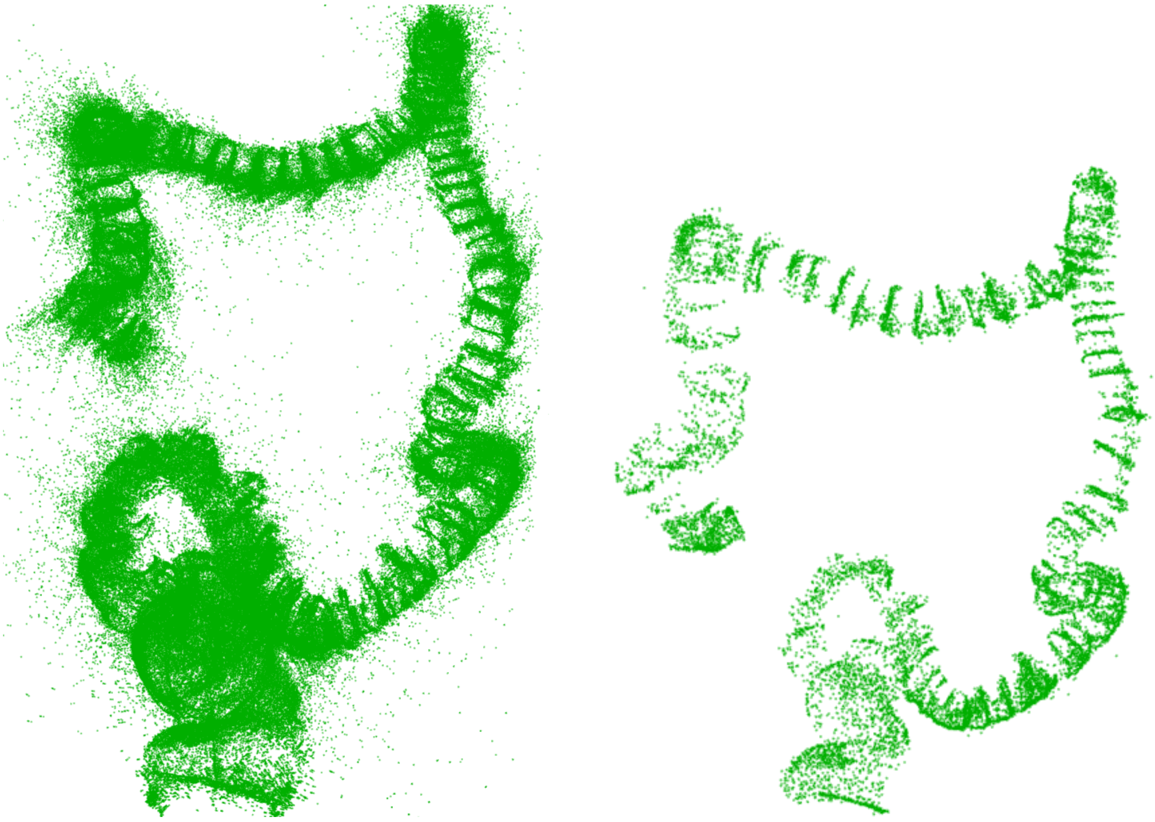


(c) StereoDSO (d) ORB-SLAM2

FIGURE 4.11: **Comparison of the Ground Truth Trajectory and Estimated Trajectories on Case** 0 **with Very Slow Camera Motion:** (a) Kintinuous suffers from large errors; (b) ElasticFusion recovers the main topological structures, the turns numbered 1, 2, 3 and 4 in estimated trajectory correspond to the turns numbered 1, 2, 3 and 4 in the ground truth trajectory, respectively; (c) The initialization of StereoDSO is unstable and it recovers a complete trajectory with large drift; (d) The initialization of ORB-SLAM2 is more stable than StereoDSO and it obtains a relatively good trajectory with drift.



(a) The ground truth (b) ElasticFusion

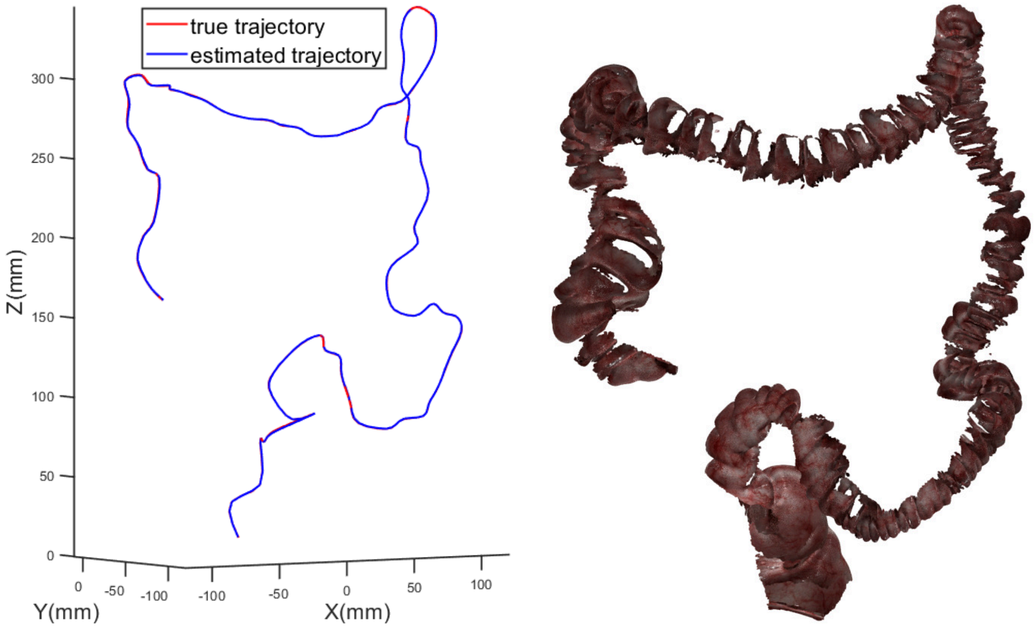


(c) StereoDSO (d) ORB-SLAM2

FIGURE 4.12: **Reconstructed Maps on Case** 0 **with Very Slow Camera Motion:** (a) The ground truth map; (b) ElasticFusion recovers the main topological structure of the colon; (c) StereoDSO recovers a complete semi-dense map with large drift; (d) ORB-SLAM2 obtains sparse map.

map is unacceptable. ORB-SLAM2 can obtain a reasonable trajectory with drift but it only built a sparse map. The evaluation results show that these stereo or RGB-D SLAM algorithms are not directly suitable for 3D reconstruction in colonoscopy even with the unrealistic very slow camera motion.

Fig. 4.13 shows the estimated trajectory and reconstructed map using the proposed framework on Case 0 with very slow camera motion. Our estimated trajectory is very close to the ground truth trajectory, which proves the high accurate pose estimation of the proposed algorithm. Also, the reconstructed colon map using our method is accurate and shows clear textures, and we can find a lot of missing detected regions on the reconstructed map.



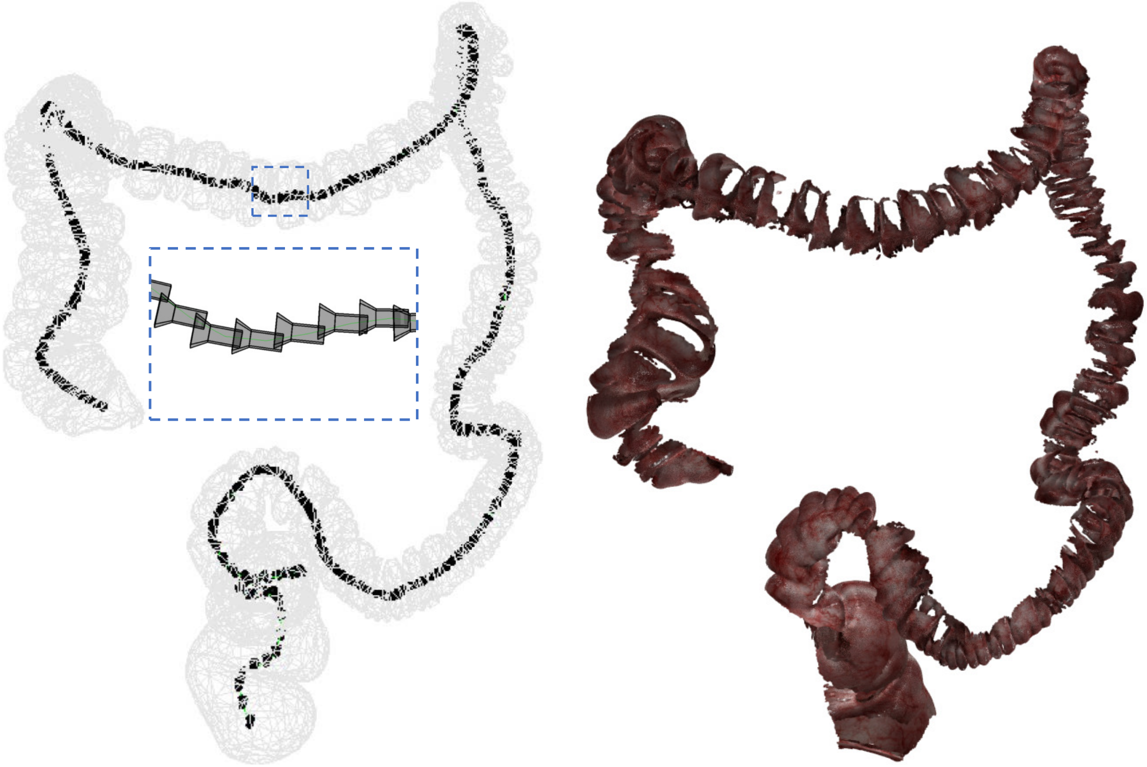
(a) Trajectories (b) Our reconstructed map

FIGURE 4.13: **Estimated Trajectory and Reconstructed Map on Case 0 with Very Slow Camera Motion using the Proposed Framework.** The left figure shows that our method can achieve very accurate pose estimation. The right figure shows that our reconstructed map is close to the ground truth.

4.3.2 Colon 3D reconstruction on Simulator-planned Camera Flight Paths

In this and the following two subsections, we evaluate our algorithm using datasets collected in different scenarios. Six planned camera flight paths are generated by the simulator to automatically guide the camera through the entire colon lumen and we record Case 1 to 6 of the experimental datasets.

Fig. 4.14 (a) shows the trajectory of camera flight path in Case 1 and Fig. 4.14 (b) shows the corresponding reconstructed complete colon map with detailed textures.



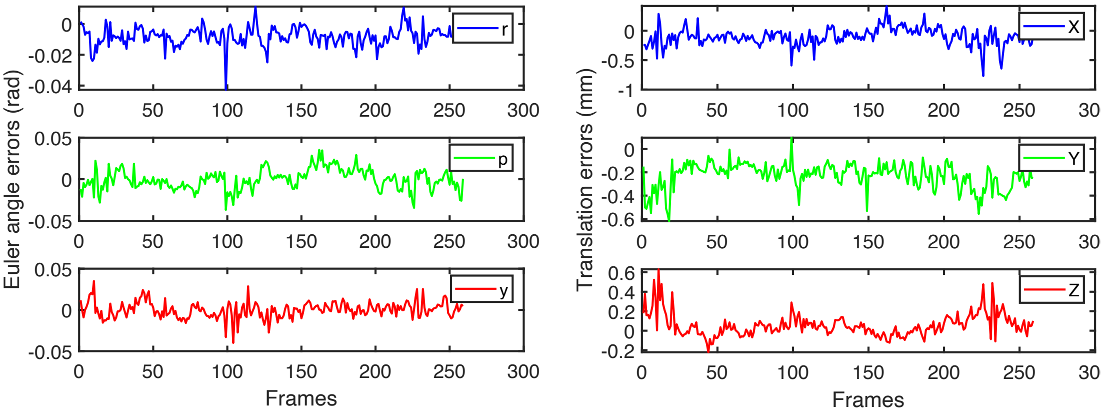
(a) Camera path on Case 1

(b) Reconstructed map on Case 1

FIGURE 4.14: **Flying Trajectory and Reconstructed 3D Map on the Simulator-planned Cases:** (a) shows the trajectory of camera flight path in Case 1 and (b) shows the reconstructed complete colon map with detailed textures.

Fig. 4.15 (a) and Fig. 4.15 (b) illustrate the registration errors (on datasets Case 1) between scans from stereo images and colon model using the proposed joint optimization algorithm. The Euler angle errors along X, Y and Z axis are within $[-0.04, 0.04]$ rad and the translation errors along X, Y, Z are within $[-0.5, 0.5]$ mm, respectively.

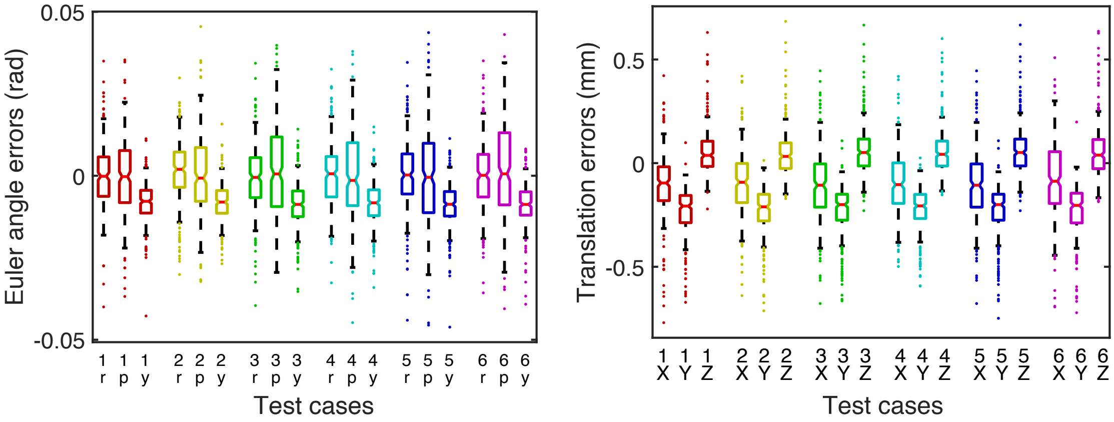
Fig. 4.16 (a) and Fig. 4.16 (b) show the Euler and translation error distributions on datasets Case 1 to Case 6, respectively, which validates the robustness and accuracy of the proposed method. For each scan to colon model registration, the algorithm takes 50 iterations on average to converge.



(a) Euler angle error on Case 1

(b) Translation error on Case 1

FIGURE 4.15: **3D Reconstruction Errors on The Simulator-planned Datasets Case** 1**.** r, p and y represent roll, pitch and yaw angles along axis X, Y and Z axis respectively.



(a) Euler angle error statistic

(b) Translation error statistic

FIGURE 4.16: **3D Reconstruction Results on The Simulator-planned Case** 1 **to Case** 6**.**

Fig. 4.17 shows the comparison between several textured regions which are reconstructed by the proposed method and the actually seen regions, their textures are slightly different as the field of view of a scan is smaller than the corresponding pair of stereo images.

The reconstructed colon map is compared to the CT-segmented colon model and the uninspected regions are shown in Fig. 4.18 in green color. The endoscopist in a local hospital identify that there are around $25 - 40\%$ of the colon internal surface are missed in the colonoscopy procedure on Case 1, especially the opposite sides of the colon wall, since the camera always keep forward moving during its flight.

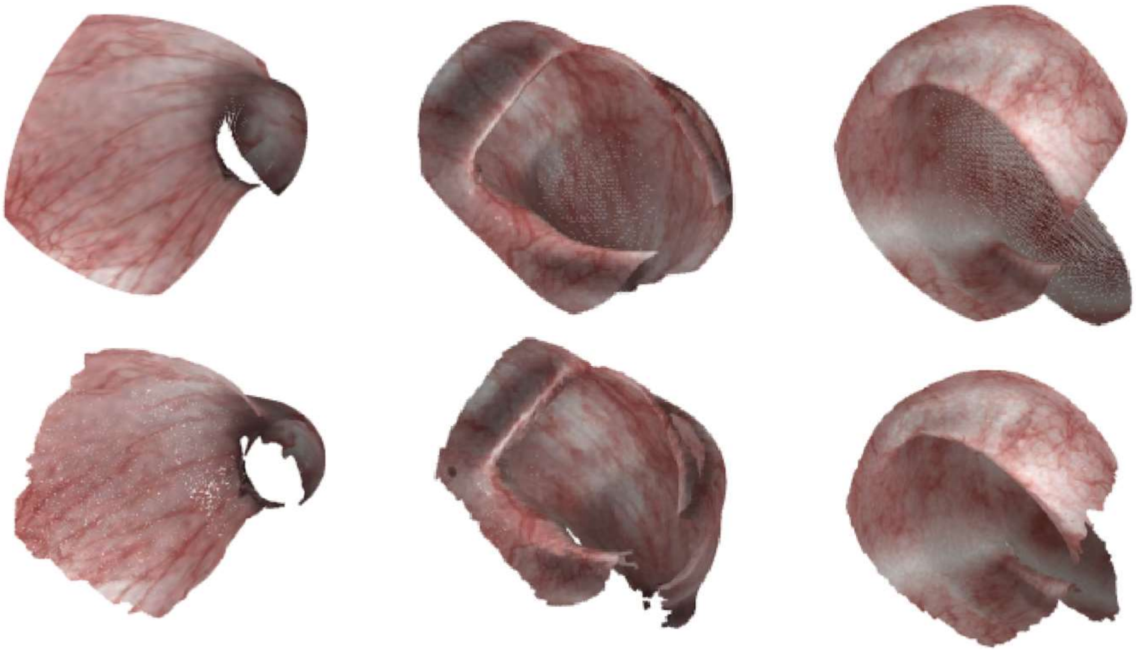


FIGURE 4.17: **Examples of Texture Region Comparison.** The first row shows the ground truth texture regions and the second row shows the corresponding reconstructed texture regions.

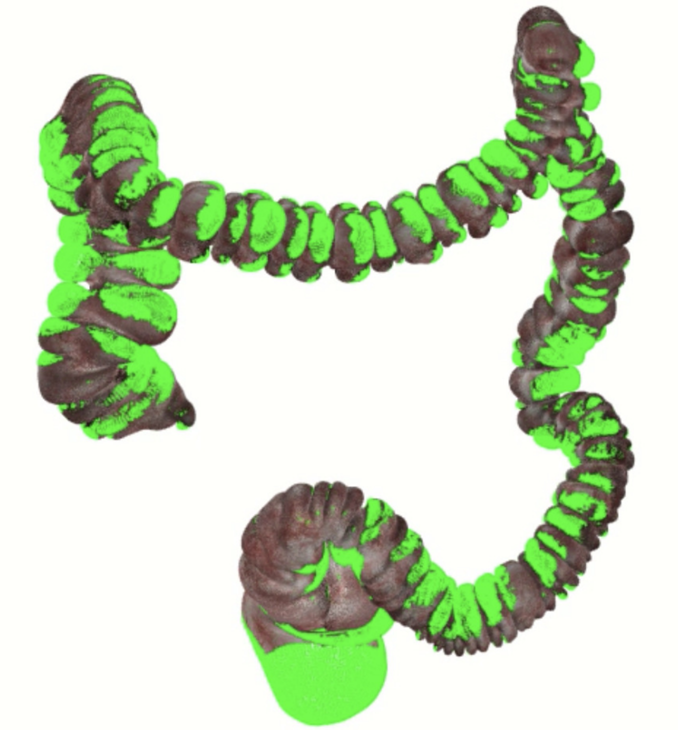


FIGURE 4.18: **Examples of Texture Region Comparison.** 25 – 40% of the colon internal surface are missed in the colonoscopy procedures.

4.3.3 Colon 3D Reconstruction on Manually Flown Paths

To simulate the real colonoscopy procedures by clinicians with different skills, datasets of Case 7, 8 and 9 are manually collected by three people with different level of clinical skills after training.

Fig. 4.19 shows the estimated camera trajectories and reconstructed colon maps of Case 7, 8 and 9, respectively. The camera in Case 7 is flown through the entire colon lumen and the images are taken from the forward, side and opposite view of the colon. For the camera in Case 8, it took images from the forward views and some side views of the colon. Very similar to the real colonoscopy procedures, the camera in Case 7 and Case 8 are operated with sudden changes of rotation and translation. By contrast, the trajectory of the camera in Case 9 is smooth and the least number of images are taken.

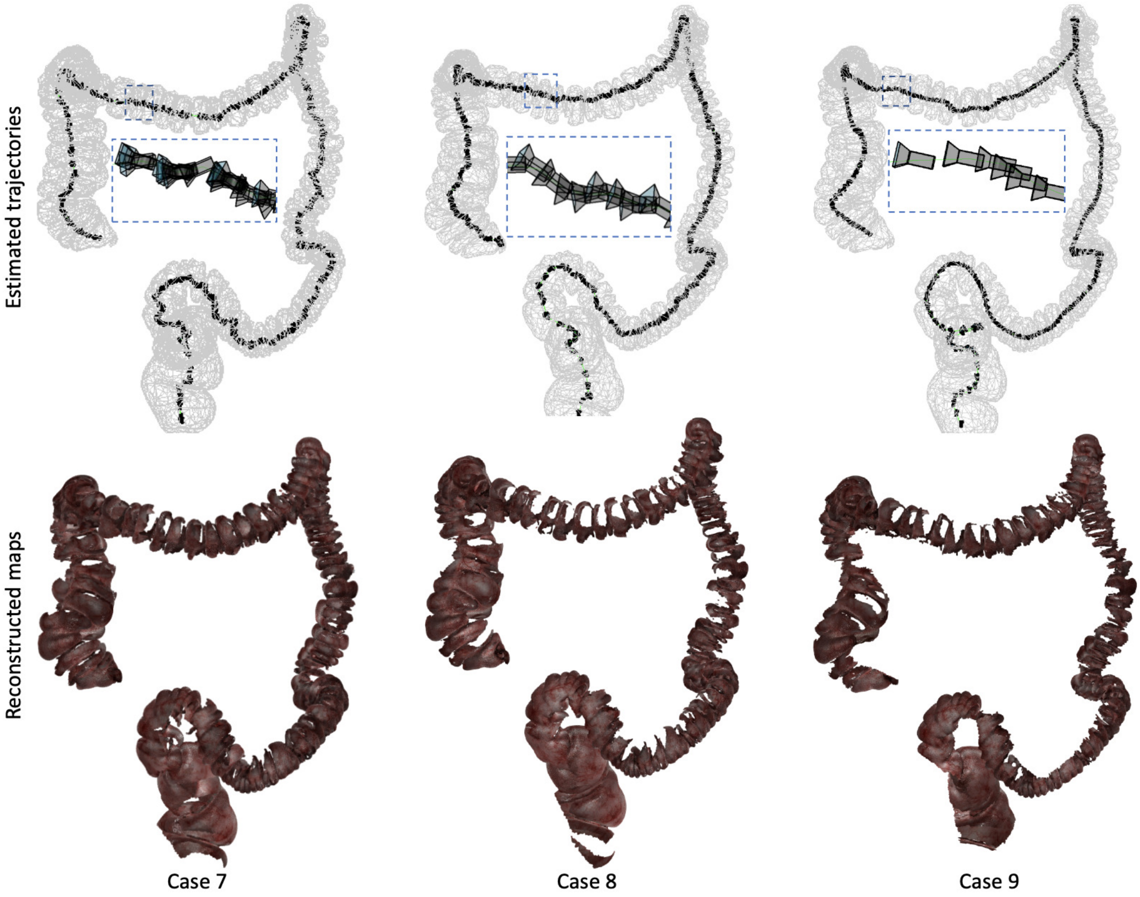
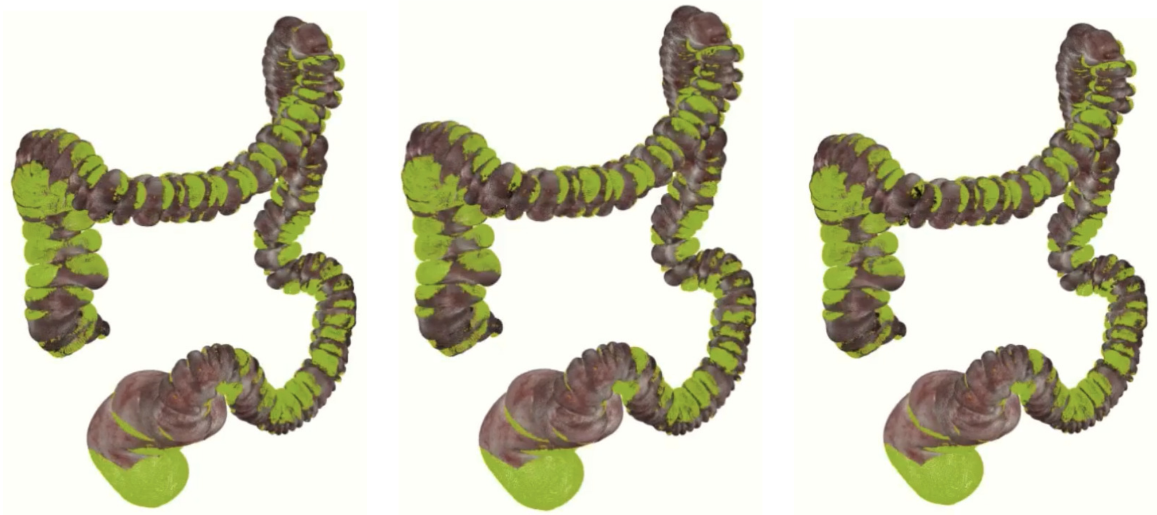


FIGURE 4.19: **3D Reconstruction Results on Manually Flown Case 7, 8 and 9.** The first row shows the camera flight paths, the second row shows the corresponding reconstructed colon maps.

For the reconstructed and texturized colon maps. We can find that the reconstructed map from Case 7 is more complete than Case 8 and Case 9 because a large amount of colon internal surface is covered. Although the map from Case 8 is slightly more complete than Case 9, there are still many areas that are uninspected, especially the opposite sides of colon folds. After that, the uninspected regions on Case 7, 8 and 9 are shown in Fig. 4.20.



(a) Case 7 (b) Case 8 Case 9

FIGURE 4.20: **Deficient Coverage Displaying on Case 7, Case 8 and Case 9.**

The registration error distributions on Case 7, 8 and 9 are shown in Fig. 4.23. The errors in Case 9 are relative small compared to Case 7 and 8 because its camera motion is smooth and there are certain overlapped areas between each pair of consecutive frames. Overall, all the Euler angle errors and translation errors are relatively small.

4.3.4 Colon 3D reconstruction on fully inspected colon

The last evaluation is conducted on datasets of Case 10 to 15 which are manually collected and aimed to validate the ability of the proposed 3D reconstruction framework to fully recover the internal colon surface.

As shown in Fig. 4.21, six segments of camera flight paths (from Case 10 to Case 15) are designed to fully inspect the internal surface of anatomical segments (Rectum, Sigmoid, Descending, Transverse, Ascending and Cecum) of the human colon, respectively.

To inspect as much area as possible of the internal surface of the colon and simulate the real colonoscopy procedures, the camera is manually flown to inspect from the forward, side and opposite views of the colon segments with challenging conditions including large changes of viewing angles and close distance to the colon surface. Fig. 4.22 shows very complete colon maps with detailed textures. Fig. 4.23 shows the mean registration errors of X, Y, Z axis, which demonstrates the capability and high accuracy of 3D reconstruction with fully recovery of internal colon surface.

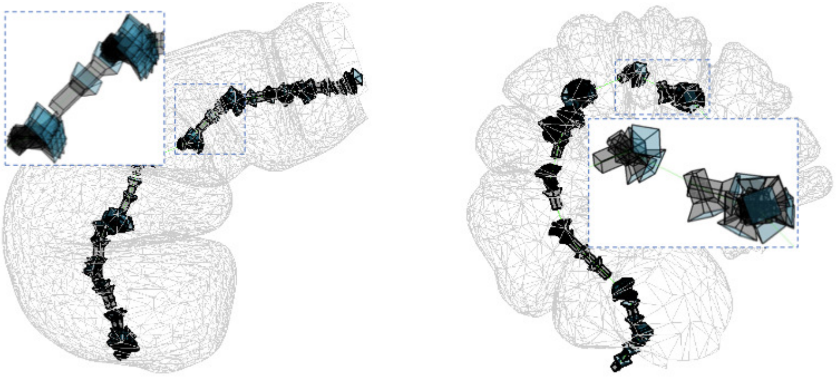
4.3.5 In-Vivo Experiments

We also show some preliminary results using two in-vivo datasets to demonstrate the practicality of the proposed framework. The synthetic colonoscopy images with ground truth of depths are used to train a supervised convolutional neural network for monocular depth estimation, then the trained network is used to predict depth for the real colonoscopy images. The predicted depth images are dense and we can reconstruct 3D scan for each real monocular colonoscopy image. The impementation details of the depth estimation neural network can be found in Chapter 5.

Fig. 4.24 and Fig. 4.25 show the used colon chunk models and corresponding reconstructed map of the colon chunks with structures and textures. However, the quality of the reconstructed map is not as good as that in the simulation experiments. The degradation is mainly caused by errors of predicted depth images and the deformation of the real colon. In our next work of Chapter 5, we will show the improved framework to better handle in-vivo data.

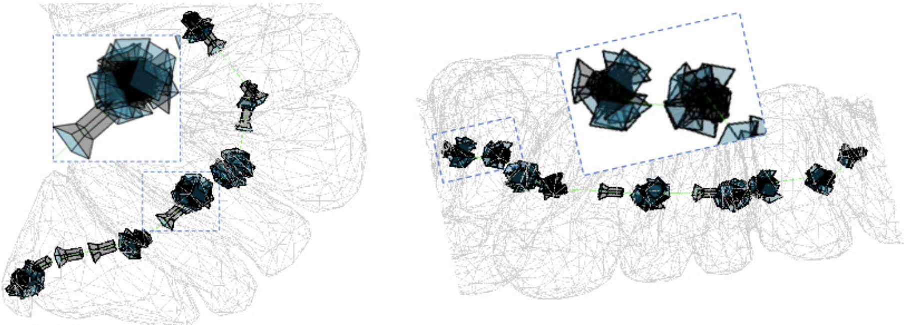
4.4 Chapter Summary

This chapter presents our first framework for 3D reconstruction of colon structures and detailed textures from stereo colonoscopic images. A colon model segmented from CT is used together with the colonoscopic images to achieve high quality reconstruction results. The developed realistic colonoscopy simulator is used for providing experimental datasets under different scenarios. Indeed, the proposed framework is validated using 15 different



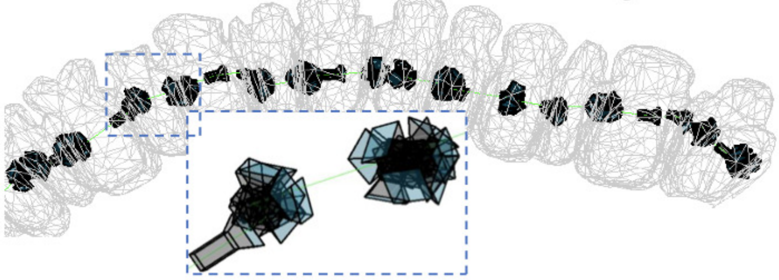
Rectum colon

Sigmoid colon

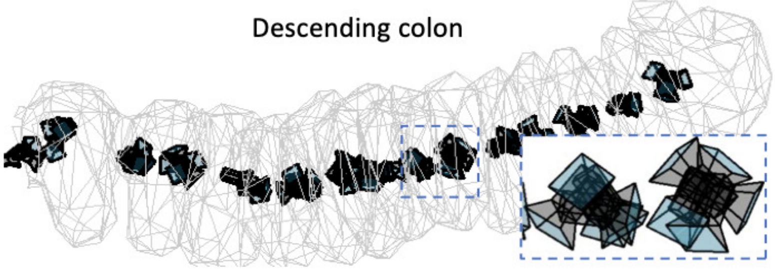


Cecum colon

Ascending colon

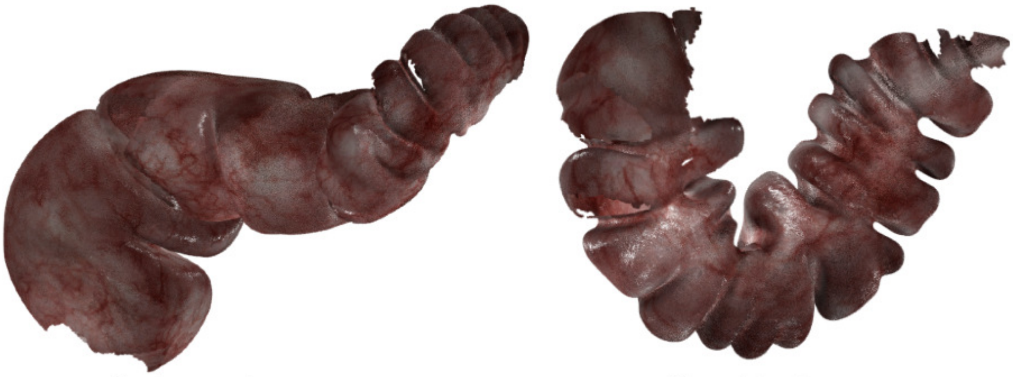


Descending colon

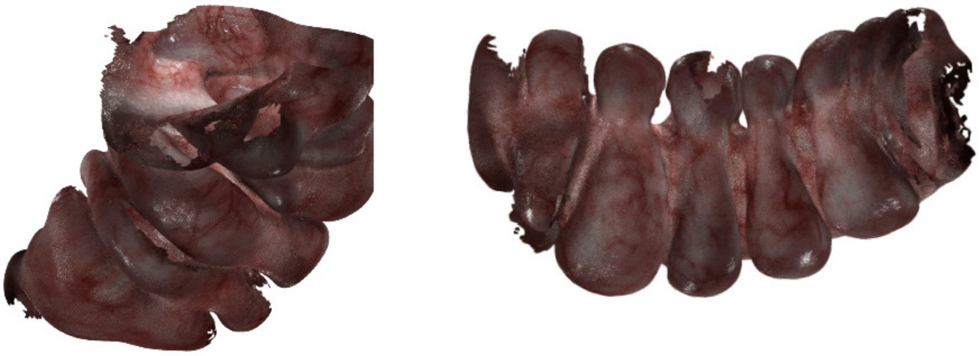


Transverse colon

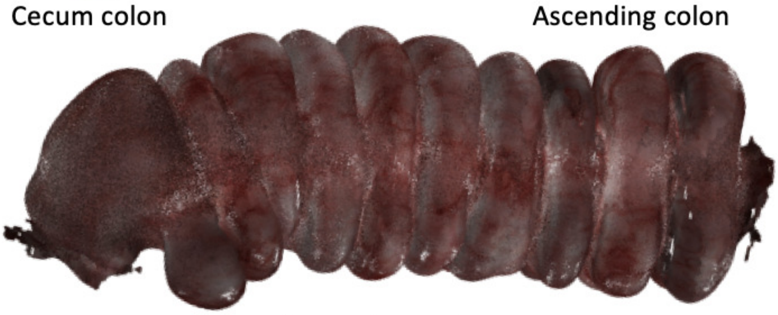
FIGURE 4.21: **Designed Camera Flight Trajectories to Fully Inspect the Colon.**



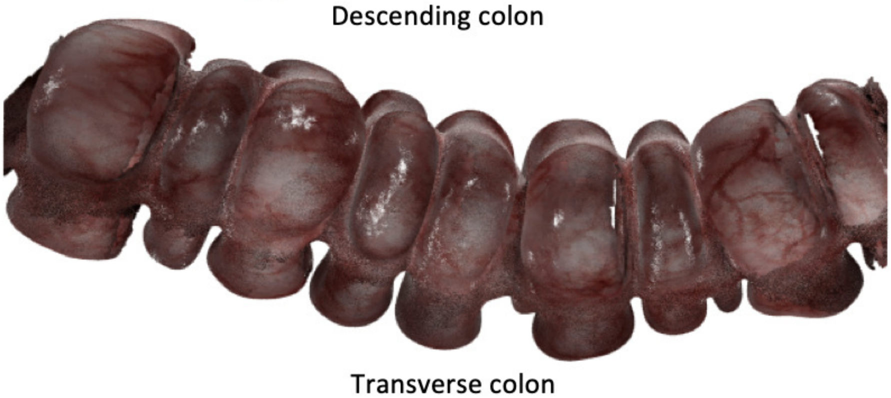
Rectum colon Sigmoid colon



Cecum colon Ascending colon

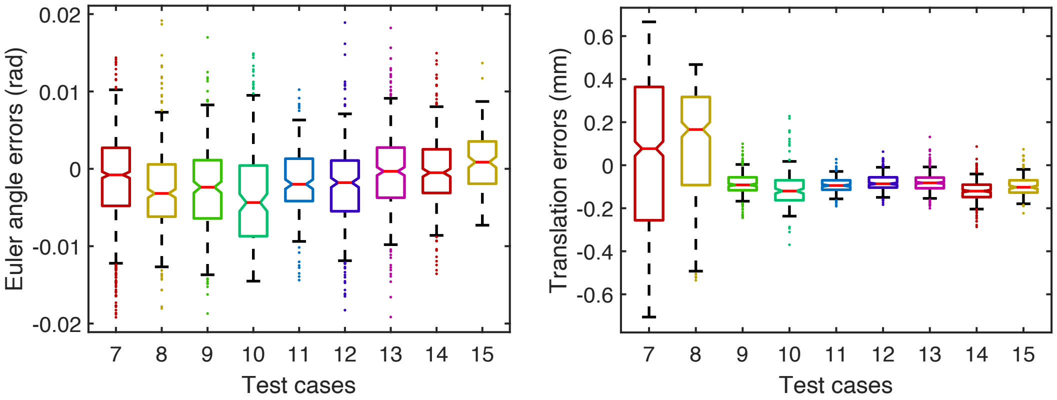


Descending colon



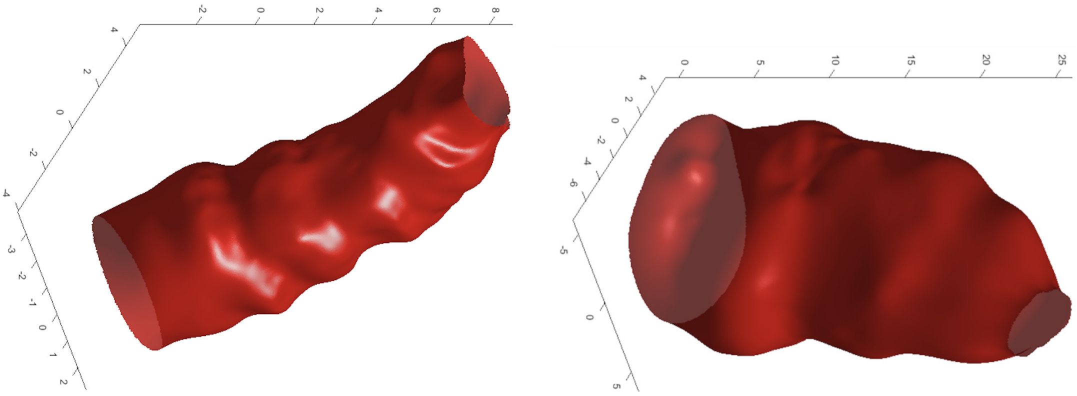
Transverse colon

Figure 4.22: **3D Reconstruction Results on the Fully Inspected Colon.**



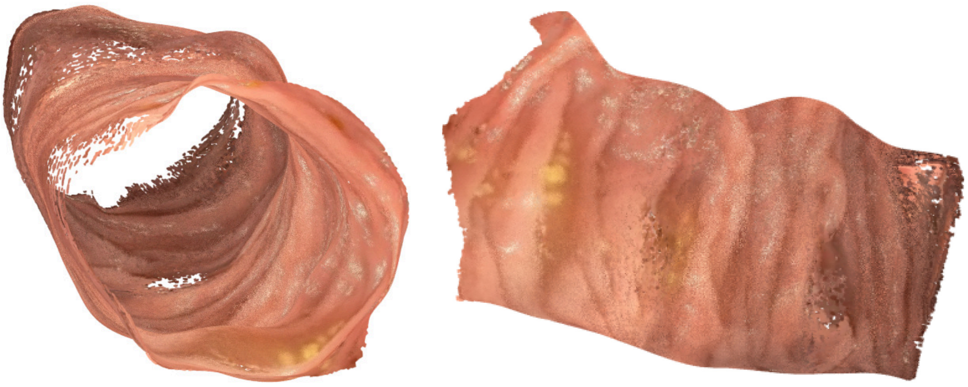
(a) Euler angle error statistic (b) Translation error statistic

Figure 4.23: **Mean Reconstruction Errors of Case 7 to Case 15.**



(a) front view (b) side view

Figure 4.24: **Colon Segment Models Used in the In-vivo Experiments.**



(a) front view (b) side view

Figure 4.25: **3D Reconstruction of A Real Colon Chunk:** (a) and (b) show the reconstructed colon chunk from the front view and the side view, respectively.

datasets generated from the simulator. Experimental results have demonstrated the high accuracy and robustness of the proposed framework. Also, an in-vivo dataset is used to show the potential clinical applications in colonoscopy procedures.

The proposed framework helps to overcome the limitations of other SLAM methods in the context of colonoscopy, primarily by addressing three key aspects:

- Dense and textured 3D mapping: Unlike feature-based SLAM algorithms and semi-dense methods that reconstruct sparse or semi-dense point clouds, the proposed framework generates a dense and textured 3D colon map. This is advantageous as it provides a more detailed and visually informative representation of the colon's interior. Dense maps can be particularly valuable in medical applications where fine details may be clinically significant.
- Loop closure handling: In a normal colonoscopy procedure, there are no large loop closures, which can lead to significant drift errors in camera pose estimation and scene reconstruction over time. Many SLAM algorithms rely on accurate feature tracking and large loop closures to mitigate this drift. However, the proposed method takes a different approach. It avoids the need for large loop closures by using a pre-operative colon CT-segmented model as a global map for its SLAM framework. This strategy improves the stability and reduces the drift of successive frame reconstructions. By leveraging the prior knowledge from the CT model, it compensates for the lack of loop closures in colonoscopy procedures.
- Fusing photometric and geometric Information: Other SLAM algorithms often rely on either photometric constraints or geometric terms to estimate camera poses, which can make it challenging to ensure both geometric accuracy and texture consistency in the reconstructed maps. In contrast, the proposed method combines photometric and geometric optimization pipelines within its SLAM framework. By doing so, it accurately estimates camera poses while simultaneously addressing issues related to texture matching inconsistencies. This fusion of information likely results in more robust and visually consistent reconstructions.

While the current work has yielded promising results, it is essential to acknowledge certain limitations. First, our proposed framework relies on stereo images due to the necessity of depth information derived from a stereo matching method. To apply our framework to 3D reconstruction using monocular colonoscopic images, one way is to predict the depth in monocular images using deep learning based method (as in the in-vivo experimental result). For example, the developed colonoscopy simulator can generate complete datasets of colonoscopic images with ground truth of camera poses and depths, then the dataset can be used to train a supervised network for monocular depth estimation.

Second, due to the texture difference between simulated images and real in-vivo images, an image to image translation network for domain adaptation [70] can be used to transform the real colonoscopy images into their synthetic-like representations for depth estimation. However, if we use monocular images for depth estimation, the achievable reconstruction accuracy is expected to be reduced.

Third, the non-rigid characteristic of the real colon will cause some degradations such as inaccuracy in estimating image depth and recovering camera motion. Especially, colon deformation can impact camera pose estimation from the following aspects:

- Loss of visual features: Deformation of the colon can cause the loss of distinctive visual features that are used for pose estimation. These features might include landmarks, blood vessels, or anatomical structures that the camera relies on to determine its position and orientation.
- Distorted geometry: Deformation can introduce non-linear distortions in the colon's shape and geometry. This can make it challenging to accurately estimate camera poses, as the geometry that the camera "sees" may not match the expected geometric model used for pose estimation.
- Tracking errors: Camera pose estimation often relies on tracking specific points or features in consecutive frames. Deformation can lead to the erroneous tracking of features or result in the loss of tracking altogether, making it difficult to compute accurate camera poses.

- Drift and inaccuracies: Deformation-induced tracking errors and distorted geometry can lead to pose estimation drift over time. This means that as the camera moves through the deformed colon, its estimated position and orientation may become progressively less accurate.

In the next chapter, firstly, we will develop and train deep neural networks for the depth estimation of monocular colonoscopic images. Secondly, we will improve the proposed framework with the capability of overcoming colon deformation using a general model and non-rigid SfM-based approaches. Our goal is to develop robust reconstruction algorithms for clinical colonoscopic images, and we believe that effective handling of colon deformation will be an important step towards achieving this goal.

Chapter 5

3D Reconstruction of Deformable Colon Structures based on Preoperative Model and Deep Neural Network

In this chapter, we provide a more robust framework for 3D reconstruction of deformable colonic surfaces with high accuracy. The input of the framework is a sequence of monocular colonoscopic images and a corresponding colon mesh model segmented from pre-operative CT or MRI scans. The output is a reconstructed and texturized 3D colon map. The proposed framework includes dense depth estimation from monocular colonoscopic images using a DNN, VO based camera motion estimation and an ED graph based non-rigid registration algorithm for deforming 3D scans to the segmented colon model. The developed realistic colonoscopy simulator is used to generate simulation datasets with different levels of deformation. Simulation results demonstrate the good performance of the proposed 3D deformable colonic surface reconstruction method in terms of accuracy and robustness. In-vivo experiments are also conducted and the results show the practicality of the proposed framework for providing useful shape and texture information in colonoscopy applications.

Compared with our first framework introduced in Chapter 4, the main contributions of this chapter are as follow.

- A novel framework that can reconstruct 3D deformable colon structures and textures from monocular colonoscopic videos;
- A ED graph-based non-rigid registration algorithm. Which is used to non-rigidly register (transform and deform) the 3D scans to the segmented colon model;
- A DNN neural network for depth estimation of monocular colonoscopic images. Compared with the first framework in Chapter 4, the proposed framework is able to reconstruct colon map from monocular colonoscopic videos;
- A GAN is used to transform the real colonoscopic images into their synthetic-like representations for depth estimation.

5.1 Framework Overview

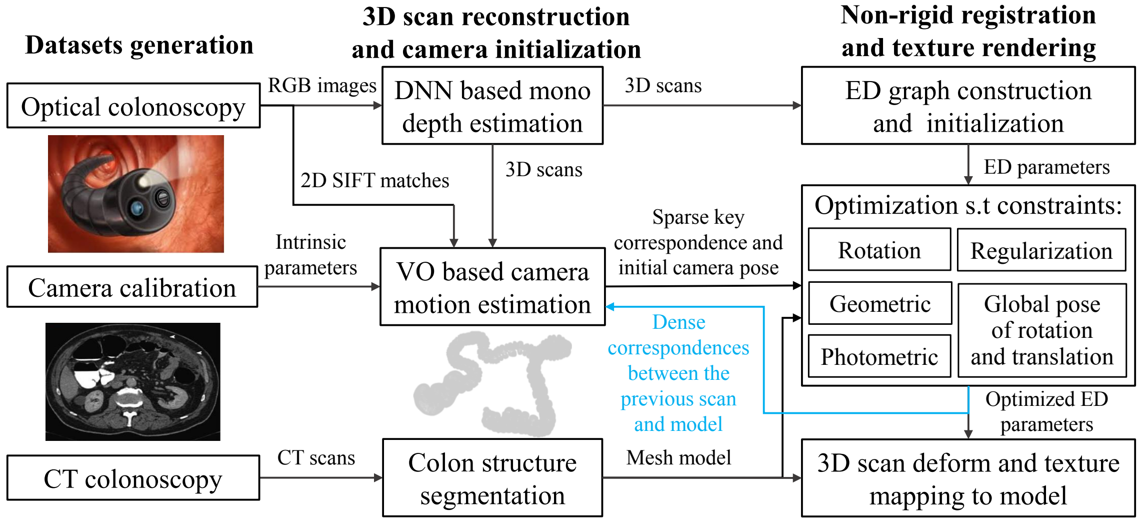

Datasets generation
3D scan reconstruction and camera initialization
Non-rigid registration and texture rendering
Optical colonoscopy
RGB images
DNN based mono depth estimation
3D scans
ED graph construction and initialization
ED parameters
2D SIFT matches
3D scans
Optimization s.t constraints:
Rotation
Regularization
Geometric
Global pose of rotation and translation
Photometric
Sparse key correspondence and initial camera pose
Intrinsic parameters
Camera calibration
VO based camera motion estimation
Dense correspondences between the previous scan and model
Optimized ED parameters
CT scans
CT colonoscopy
Colon structure segmentation
Mesh model
3D scan deform and texture mapping to model

FIGURE 5.1: **The Framework of Reconstructing Deformable 3D Colon Surface with Detailed Textures**.

Fig. 5.1 shows the proposed framework for reconstructing and texturing the deformable colon surface. It mainly includes the following modules: 1) Dense 3D scan reconstruction using DNN: reconstructing 3D scan using predicted depths from the depth estimation

DNN; 2) VO based camera pose initialization: estimating the initial pose of the scan relative to the CT segmented colon model; 3) Non-rigid registration and texture rendering: using an ED graph to represent the scan deformation and optimizing the ED parameters under observed constraints, then using the optimized ED parameters to deform the scan to the colon model and mapping the textures from the colonoscopy images to the registered regions on the colon mesh model.

5.2 Technical Details

5.2.1 Generating Ground Truth Dataset

The availability of colonoscopyic images with ground truth of image depths and camera poses is critical to develop and evaluate colon reconstruction methods. The colonoscopy simulator described in Chapter 3 is used to generate simulated datasets with different levels of deformation for traning depth estimation neural nertworks and validating the proposed colon reconstruction framework.

5.2.2 Dense 3D Scan Reconstruction using DNN

Accurate depth estimation from colonoscopic images is a fundamental task in colon structure reconstruction. In our project of reconstructing 3D colon map, we prefer to directly use or develop upon existing front end algorithms to predict image depths. Thus, in our work of estimating the depth of colonoscopy images, we use the same encoder-decoder network architecture as DenseDepth [71] which is a high quality monocular depth estimation network using a simple encoder-decoder architecture via transfer learning.

Fig. 5.2 shows an overview of architecture of the DenseDepth network. It mainly contains two parts which are encoder and decoder. The encoder is used to learn deep features from the input images and the decoder is used to build the mapping between the extracted deep features and ground truth depths. For the encoder which consists of multi-convolutional layers and multi-pooling layers, the top layers that are related to the original ImageNet classification task is removed and the left network is used as the encoder network. For

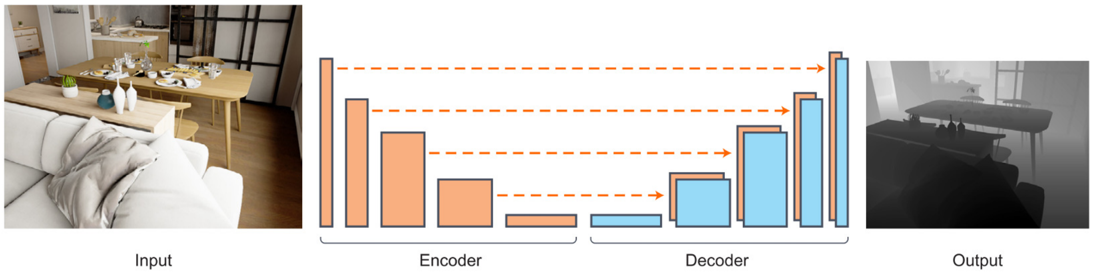


FIGURE 5.2: **Overview of Network Architecture of DenseDepth [71].**

the decoder, it starts with a convolutional layer with the same number of output channels as the output of our truncated encoder, then add four up-sampling blocks, each block composed of a bilinear up-sampling followed by two convolutional layers.

The loss function of the network is defined as the weighted sum of three terms as following:

$$L(y,\hat{y}) = \lambda L_{depth}(y,\hat{y}) + L_{grad}(y,\hat{y}) + L_{SSIM}(y,\hat{y}) \tag{5.1}$$

The first loss term represents the depth difference of each pixel in the depth image y and $\hat{y}$:

$$L_{depth}(y,\hat{y}) = \frac{1}{n}\sum_{p}^{n} |y_p - \hat{y}_p| \tag{5.2}$$

The second term represents the differences in the x and y components for the depth image gradients of y and $\hat{y}$:

$$L_{grad}(y,\hat{y}) = \frac{1}{n}\sum_{p}^{n} |\mathbf{g_x}(y_p,\hat{y}_p)| + |\mathbf{g_y}(y_p,\hat{y}_p)| \tag{5.3}$$

The third term uses the SSIM metric [72] which is a commonly-used metric for image reconstruction tasks. However, to make the network compatible with different input data size, we replace the structural similarity loss function with the multi-scale structural similarity loss. Whose structure is smoother and depth gradient is smaller, a smoothness loss function is added for the in-vivo depth prediction network.

The synthetic colonoscopy dataset is used to train the deep network for monocular colonoscopy depth estimation. We manually move and rotate the camera inside the colon model and capture 10 colonoscopy videos, each containing about 1K frames with ground truth of depths and camera poses. Then, the datasets are used to train, validate and test the depth estimation network.

Furthermore, there is texture difference between simulated images and real images, to make the depth prediction model trained with simulation data perform well on in-vivo colonoscopic images [37], a GAN is used to transform the real colonoscopy images into their synthetic-like representations for depth estimation [70]. Therefore, 1K frames of real colonoscopy images are used together with synthetic frames to train the image domain transformation network.

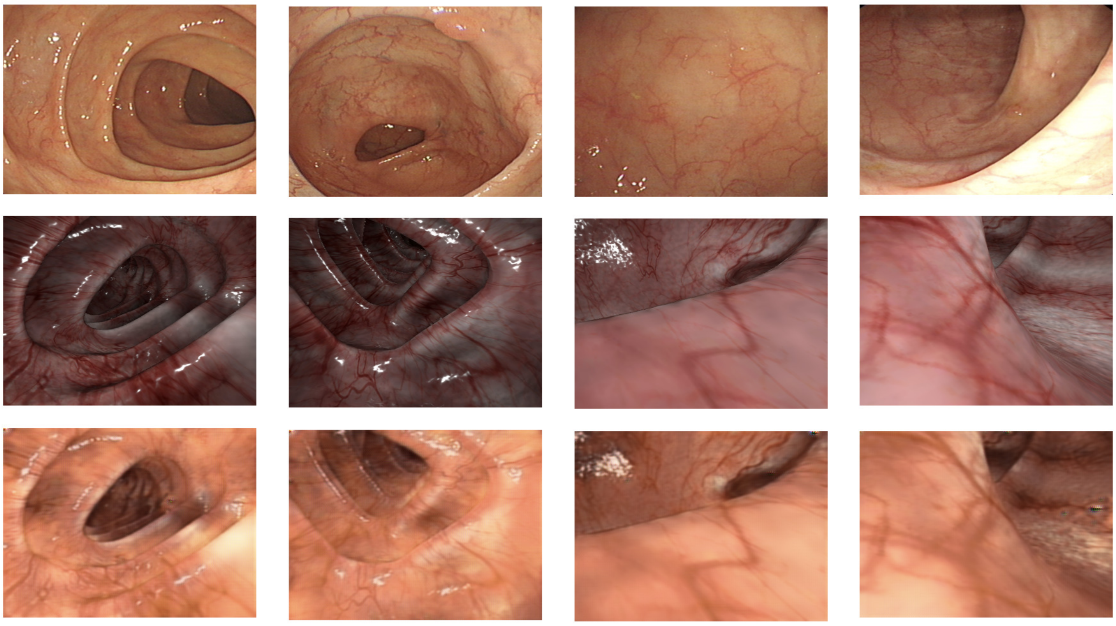


Figure 5.3: **Domain Translation Transforms Simulated Images into Real-like Representations.** The first and second rows show real images and the simulated images, the third rows show the real-like representations for simulated images.

Fig. 5.3 shows some examples of real in-vivo colonoscopic images and simulated images with their corresponding real-like representations.

Fig. 5.4 shows some examples of simulated and real in-vivo colonoscopic images and the corresponding estimated depth images.

To evaluate the accuracy of the trained depth estimation network, we compare the simulation depth estimation accuracy between a recurrent neural network for depth and pose

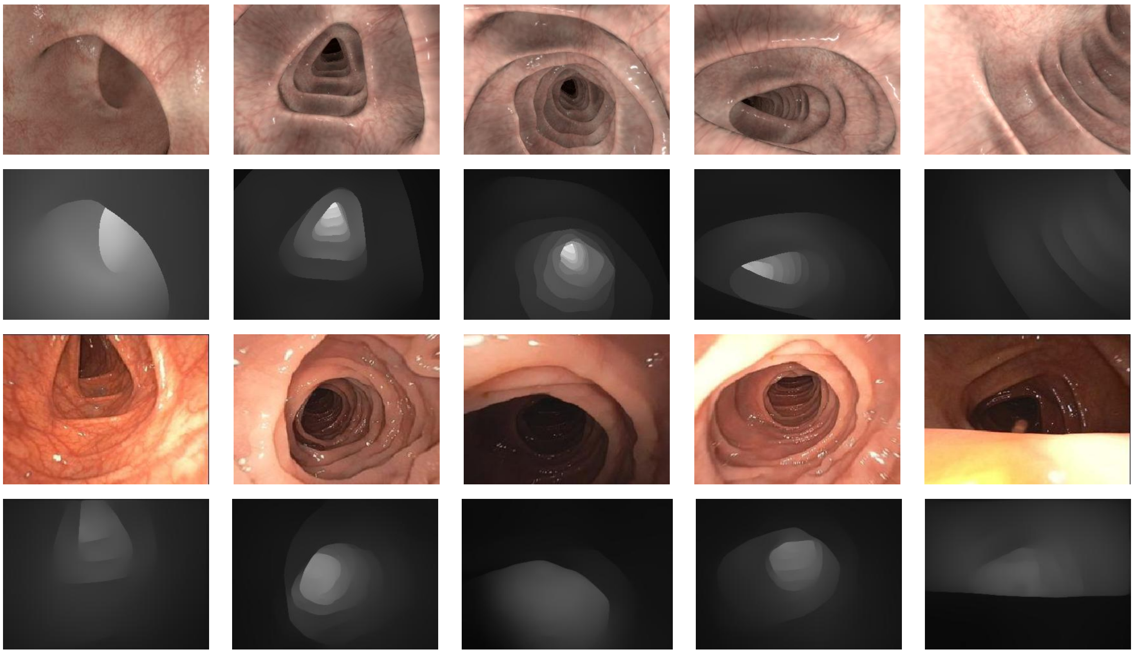


FIGURE 5.4: **Simulated and Real Colonoscopic Images with Predicted Depths.** The first and second rows show simulated images and the corresponding estimated depth images, the third and the fourth rows show real in-vivo colonoscopic images and the corresponding estimated depth images.

estimation (RNN-DP) in [25, 52] and our network on 10K frames, the mean absolute errors of ours and RNN-DP's are 0.45 mm and 5.70 mm, and the corresponding root mean square errors are 1.28mm and 7.16mm, respectively. Fig. 5.5 shows the mean absolute errors comparison from four group of data, the each odd column represent our's mean depth error from 250 images and the even column represent corresponding RNN-DP's.

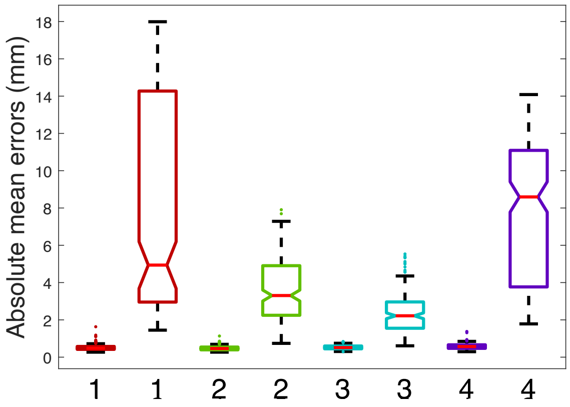
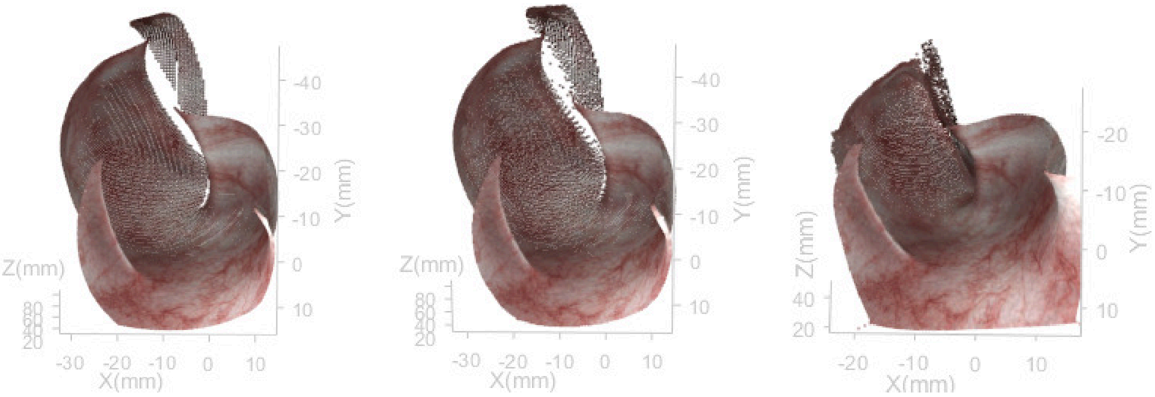


FIGURE 5.5: **Mean Absolute Errors Comparison with RNN-DP.** Each odd column represents our's mean depth error from 250 images and the even column represents corresponding RNN-DP's.

Finally, the estimated dense depth images together with corresponding colonoscopy images are used to reconstruct 3D scans. Fig. 5.6 shows examples of reconstructed scan on one frame and it shows that RNN-DP's depth prediction scale is larger in the fore-end than the rear end and also its depth accuracy become worse in rear end.



(a) (b) (c)

FIGURE 5.6: **Reconstructed Scan Comparison With RNN-DP:** (a) Ground truth scan; (b) Scan using our depth; (c) Scan using RNN-DP's depth.

5.2.3 Sparse Key Correspondences and VO Based Camera Pose Initialization

The VO module is similar to the one in Section 4.2.2, which is used to initialize the global poses of 3D scans relative to the colon model and provide sparse key correspondences to enhance the accuracy of the non-rigid registration between scans and the colon mesh model. The main differences are 3D scans reconstruction using the trained depth estimation DNN and 3D scan non-rigid registration.

To make this chapter complete and easy to follow, we will briefly illustrate the main steps of sparse key correspondences extraction and camera initialization. Fig. 5.7 shows the steps to initialize one scan and extract the set of sparse key correspondences between the scan and the colon model.

First, SIFT and RANSAC algorithms are used to extract spatially scattered SIFT features between the current RGB frame and the previous RGB frame. Secondly, we trace the pixel indices of these 2D SIFT features in the current and the previous 3D scans, thus a set

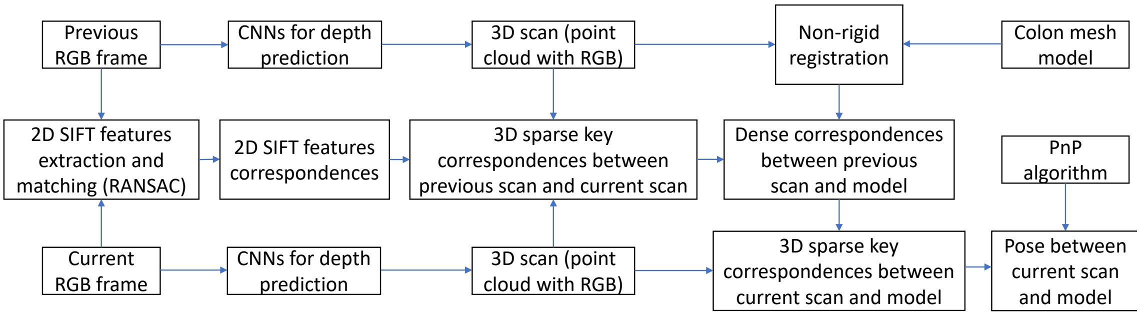

Previous RGB frame
CNNs for depth prediction
3D scan (point cloud with RGB)
Non-rigid registration
Colon mesh model
2D SIFT features extraction and matching (RANSAC)
2D SIFT features correspondences
3D sparse key correspondences between previous scan and current scan
Dense correspondences between previous scan and model
PnP algorithm
Current RGB frame
CNNs for depth prediction
3D scan (point cloud with RGB)
3D sparse key correspondences between current scan and model
Pose between current scan and model

FIGURE 5.7: **Pipeline for Initializing Scan and Extracting Sparse Key Correspondences.**

of sparse 3D point correspondences between the two scans are obtained. If the previous scan was well registered to the colon model using the proposed non-rigid registration algorithm, then dense geometrical 3D point correspondences between the previous scan and the vertices of the colon mesh model can be extracted using the nearest neighbor search method. Based on the set of sparse correspondences between the current scan and the previous scan and the set of dense correspondences between the previous scan and the colon model, we can infer the sparse key correspondences between the current scan and the colon model.

After that, the P3P algorithm conjunction with RANSAC algorithm are applied to the sparse key correspondences to calculate the global pose of the current scan relative to the colon model. Finally, this computed global pose provides a good initial input for later deformation parameter estimation. The extracted sparse key correspondence will also be used as photometric constraint to enhance the accuracy of the non-rigid registration in Section 5.2.4.

5.2.4 Non-rigid Registration using ED graph

Because of the deformation nature, the region of the colon from the same field view of the camera does not hold constant shape over long time period, which causes difficulty to accurately estimate the camera motion and reconstruct smooth 3D colon map with aligned texture. Furthermore, there will be topological difference between the colon model segmented from CT scans and the actual colon during the colonoscopy.

Our non-rigid registration module assumes that the colon model segmented from CT scans is one moment shape of the dynamic colon and uses this shape as a template of the colon. Borrowing the basic idea of ED [9], we uniformly sample scattered ED nodes from a scan and build a deformation graph to facilitate space deformation of the scan.

Different from the classical ED nodes, a rigid pose is added to ED nodes of the deformation graph since we alternately take two steps to transform and deform the scan to the colon model. Specifically, the first step is to locally deform the scan using the non-rigid parameters of the ED nodes to register it onto the surface of the colon model. The second step is to use the rigid pose to rotate and translate the scan to ensure its topological structure is aligned with the colon model.

Thus, each ED node is associated with a position $\mathbf{g}_j \in \mathbb{R}^3$, an affine matrix $A_j \in \mathbb{R}^{3\times3}$, a translation vector $\mathbf{t}_j \in \mathbb{R}^3$ together with the rigid rotation matrix $R \in \mathbb{SO}(3)$ and rigid translation vector $\mathbf{t} \in \mathbb{R}^3$. Each vertex $\mathbf{v}$ in the scan has a set of neighbouring ED nodes in the deformation graph and the deformed position of the vertex $\widetilde{\mathbf{v}}$ is calculated as:

$$\widetilde{\mathbf{v}} = \phi(\mathbf{v}) = R\sum_{j=1}^{m} w_j(\mathbf{v})[A_j(\mathbf{v}-\mathbf{g}_j)+\mathbf{g}_j+\mathbf{t}_j]+\mathbf{t} \tag{5.4}$$

where m representing the number of neighboring ED nodes is set to 6.

The proposed non-rigid registration problem is to obtain the optimal ED parameters of the deformation graph by minimizing the energy function:

$$\min_{R,\mathbf{T},A_1,\mathbf{t_1}\ldots A_k,\mathbf{t_k}} w_{rot}E_{rot} + w_{reg}E_{reg} + w_{geo}E_{geo} + w_{pho}E_{pho} + w_rE_r + w_tE_t \tag{5.5}$$

where k is the number of ED nodes. The energy function has six components: rotation term, regularization term, geometric term, photometric term, global rigid rotation term and rigid translation term. In all our experiments, we use the weights as $w_{rot} = 1$, $w_{reg} = 100$, $w_{pho} = 1000$, $w_{pho} = 1$, $w_r = 1000$, $w_t = 1000$. Referring to Section 2.2.1, the first and second terms are functions only defined over the ED graph. Meanwhile, the other four terms are constrained by data observations and defined as following:

The geometric term is the sum of point-to-point Euclidean distance on a set of closest point correspondences between the scan and the vertices of the colon mesh model:

$$E_{geo} = \sum_{g} \|\phi(\mathbf{v}_g) - \mathbf{v}\|^2 \tag{5.6}$$

where $\mathbf{v}_g$ is one source point in a scan and $\mathbf{v}$ is its corresponding closest point in the colon model, $\phi(\mathbf{v}_g)$ is the result of applying (5.4) to $\mathbf{v}_g$.

The photometric term is used to enhance the alignment of texture in the overlapping regions from the registration of consecutive scans to the colon model. This term is the error on the sum of Euclidean distance between the set of pair-wise sparse key correspondences provided in Section 5.2.3 in the following form:

$$E_{pho} = \sum_{p} \|\phi(\mathbf{v}_p) - \mathbf{v}\|^2 \tag{5.7}$$

where $\mathbf{v}_p$ and $\mathbf{v}$ are one pair of photometric correspondences between a scan and the colon model, $\phi(\mathbf{v}_p)$ is the result of applying (5.4) to $\mathbf{v}_p$.

The rigid rotation and translation terms are measured by the variations of the rigid rotation R and translation $\mathbf{t}$:

$$E_r = \|\mathbf{r} - \bar{\mathbf{r}}\|^2 \qquad E_t = \left\|\mathbf{t} - \bar{\mathbf{t}}\right\|^2 \tag{5.8}$$

where E_r measures the Euler angles difference, and $\mathbf{r}$ and $\bar{\mathbf{r}}$ are the Euler angles of the rigid rotation R to be estimated and the initial rigid rotation $\overline{R}$ obtained from VO in Section 5.2.3, respectively. Similarly, E_t measures the Euclidean distance between the rigid translation $\mathbf{t}$ and the initial rigid translation $\bar{\mathbf{t}}$ obtained in Section 5.2.3.

5.2.5 Optimization Details

We minimise the energy function in (5.5) using the iterative GN algorithm. Here, we rewrite the energy function as $F(\mathbf{X}) = \mathbf{f}(\mathbf{X})^T\Sigma^{-1}\mathbf{f}(\mathbf{X})$, the vector $\mathbf{f}(\mathbf{X})$ is defined by stacking the six constraint functions, the vector $\mathbf{X}$ is defined by stacking the embedded deformation parameters of all the ED nodes together with the rigid transformation, and $\Sigma^{-1} = diag(w_{rot}, ..., w_{reg}, ..., w_{geo}, ..., w_{pho}, ..., w_r, ..., w_t)$ with corresponding dimensions.

The GN algorithm first linearizes $\mathbf{f}$ in the neighborhood of $\mathbf{X}$ with Taylor expansion:

$$\mathbf{f}(\mathbf{X}+\delta) \approx \mathbf{f}(\mathbf{X}) + \mathbf{J}\delta \tag{5.9}$$

where $\mathbf{J}$ is the Jacobian of $\mathbf{f}(\mathbf{X})$, in which the Jacobian part of the geometric and photometric terms w.r.t. the rigid camera pose R and $\mathbf{t}$ can be calculated by referring to (2.10). Thus, in each iteration k, an incremental step δ_k is computed to minimize the linearized least squares problem $(\mathbf{f}(\mathbf{X}_k) + \mathbf{J}\delta)^T \Sigma^{-1} (\mathbf{f}(\mathbf{X}_k) + \mathbf{J}\delta)$ by solving the following equation:

$$\mathbf{J}(\mathbf{X}_k)^T \Sigma^{-1} \mathbf{J}(\mathbf{X}_k)\delta_k = -\mathbf{J}(\mathbf{X}_k)^T \Sigma^{-1} \mathbf{f}(\mathbf{X}_k) \tag{5.10}$$

Before the next iteration, the updated $\mathbf{X}_{k+1}$ will be used to deform and transform the original scan to generate a temporary optimized scan. We apply the nearest neighbor search algorithm to the temporary optimized scan and the colon mesh model to establish new dense correspondences for the geometric constraint term in (4.2). The sparse key correspondences for the photometric constraint kept fixed during the whole optimization. The process repeats until the GN algorithm is converged. The detailed optimization procedure is listed in Algorithm 1.

5.3 Experiments and Results

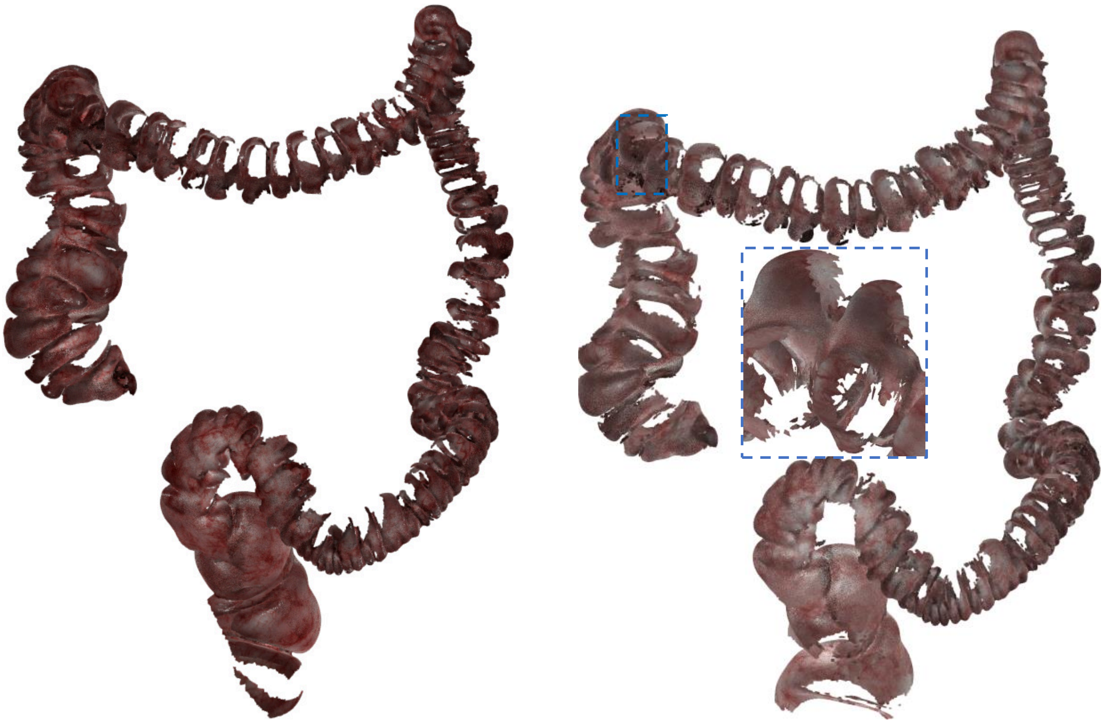
The proposed reconstruction framework is validated by simulations and in-vivo experiments. In the simulations, the robustness and accuracy of deformable colon reconstruction is first assessed via one dataset captured in the scenario simulating the real normal colonoscopy screening where the camera moves fast with significant view changes. Then, the framework is validated using other three simulated datasets captured in scenarios where the camera is operated with slow camera motion and different levels of deformations. Afterwards, in-vivo experiments with two colonoscopy videos are performed. Currently, the proposed algorithm is able to reconstruct a colon in chunks when the colon structure is clearly visible. In the in-vivo experiments, there are some deformation caused by peristaltic motion.

Algorithm 1: Optimization of the ED deformation

```
Input: Model, scan, photometric set $(\mathbf{v}_p, \mathbf{v})$
Output: ED parameters, geometric set $(\mathbf{v}_g, \mathbf{v})$
Initialization: extract ED nodes, $A_j = I_3$, $\mathbf{t}_j = \mathbf{0}$, $R = \overline{R}$, $\mathbf{t} = \overline{\mathbf{t}}$
$k := 1, \epsilon_1 = \epsilon_3 = 10^{-12}, \epsilon_2 = 0, k_{max} = 50, stop := false$
while (not stop) and ($k < k_{max}$) do
    Step 1: Extract dense geometric correspondences $(\mathbf{v}_g, \mathbf{v})$ between scan and model
    for each vertex $\mathbf{v}_g$ in scan do
        $\phi(\mathbf{v}_g) \leftarrow$ Applying (2.11) to $\mathbf{v}_g$
        $\mathbf{v} \leftarrow$ Nearest neighbor find in model for $\phi(\mathbf{v}_g)$
    end
    Step 2: One iteration of GN iteration
    Solve $\mathbf{J}(\mathbf{X}_k)^T \Sigma^{-1} \mathbf{J}(\mathbf{X}_k) \delta_k = -\mathbf{J}(\mathbf{X}_k)^T \Sigma^{-1} \mathbf{f}(\mathbf{X}_k)$
    If $||\delta_k|| \leq \epsilon_3$ then stop:=true else $\mathbf{X}_{k+1} = \mathbf{X}_k + \delta_k$
    $k := k + 1$
    stop :=
      $||\mathbf{f}(\mathbf{X}_k)^T \Sigma^{-1} \mathbf{f}(\mathbf{X}_k)|| \leq \epsilon_1 \vee ||\mathbf{f}(\mathbf{X}_k)^T \Sigma^{-1} \mathbf{f}(\mathbf{X}_k) - \mathbf{f}(\mathbf{X}_{k-1})^T \Sigma^{-1} \mathbf{f}(\mathbf{X}_{k-1}|| \leq \epsilon_2$
end
Deforming scan by applying $\mathbf{X}_k$ to the original scan
for each vertex $\mathbf{v}$ in scan do
    $\phi(\mathbf{v}) \leftarrow$ Applying (2.11) to $\mathbf{v}$
end
```

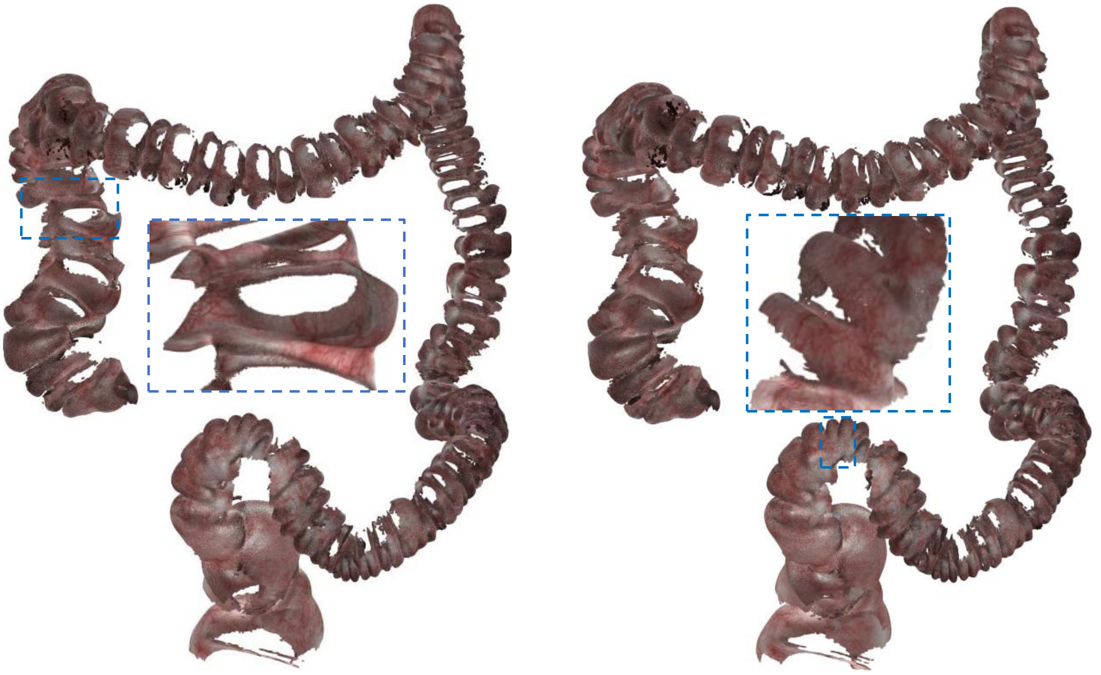
5.3.1 Validation using Simulation Datasets

The first dataset numbered 1 contains 272 frames and there is no deformation force applied to the mesh deformer. The other three datasets numbered $2, 3, 4$ contain around 800 frames and small, medium and large levels of force are applied to the mesh deformer respectively to make the colon model has different levels of deformation. Fig. 5.8 shows the reconstructed colon maps using our approach on all the 4 groups of simulation datasets. The result from dataset 1 shows that in a normal colonoscopy procedure the camera moves fast and there is less overlaps between consecutive frames, and our framework can still robustly reconstruct 3D colon structures. In the simulations, around 25-40 % of the colon internal surface are missed in the colonoscopy procedures, especially the opposite sides of the colon wall.



(a) Ours using dataset 1

(b) Ours using dataset 2



(c) Ours using dataset 3

(d) Ours using dataset 4

FIGURE 5.8: **Reconstructed Colon Maps using Our approach On Simulation Datasets:** The first row shows the 3D reconstruction of colon using our approach on dataset 1 and 2, respectively. The second row shows the 3D reconstruction of colon using our approach on dataset 3 and 4, respectively. There is nearly no drift of our approach since the colon template is used and 3D scans can be well registered to the template.

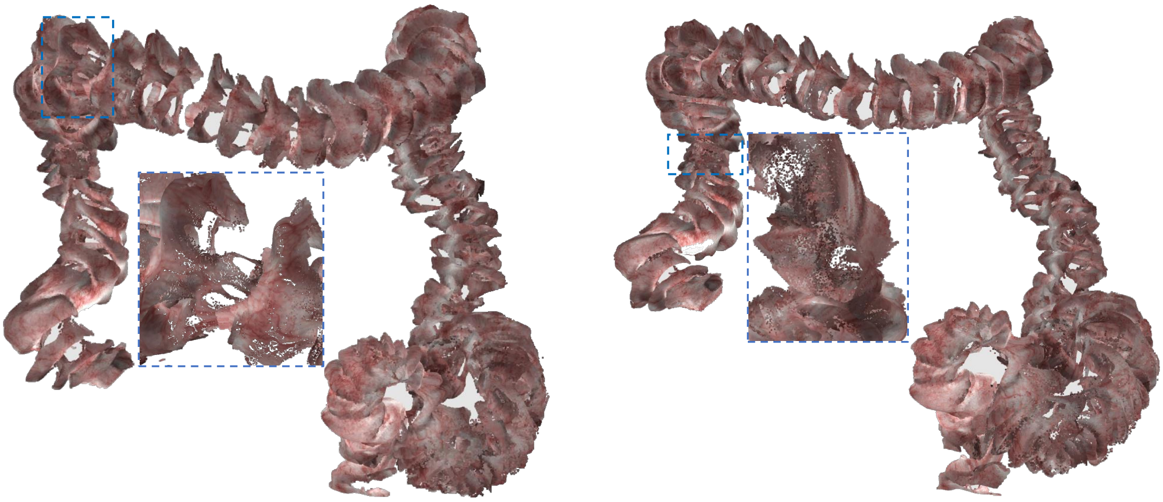
5.3.2 Comparison Between Our approach and RNNSLAM on Simulation Datasets

We compare the proposed approach with the colon reconstruction method RNNSLAM in [25, 52] and the maps reconstructed using RNNSLAM are shown in Fig. 5.9.

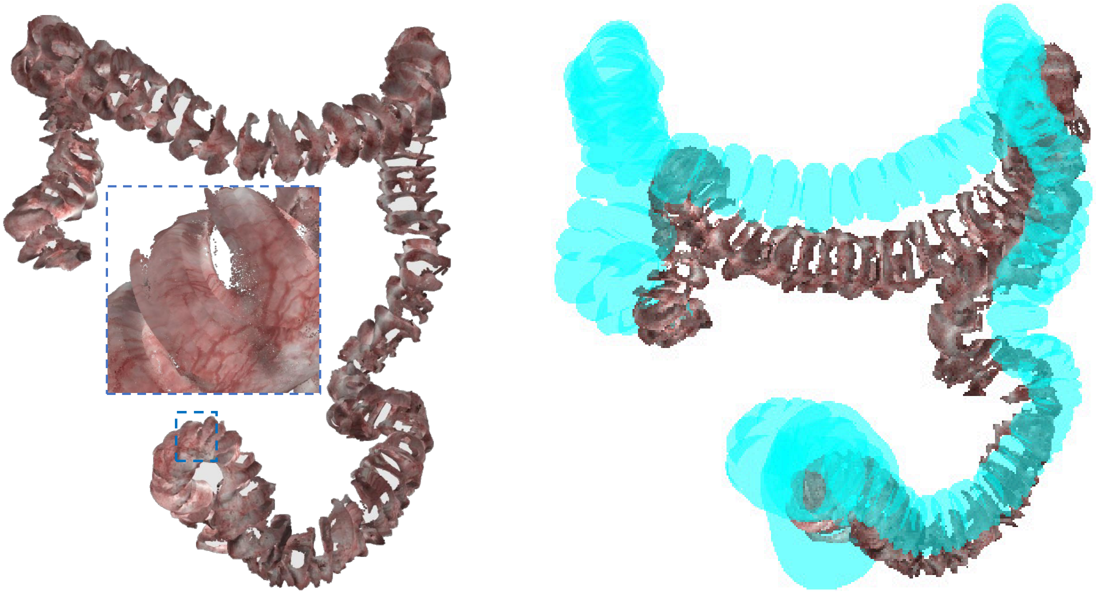
The result from dataset 1 shows that in a normal colonoscopy procedure the camera moves fast and there is less overlaps between consecutive frames, and our framework can still robustly reconstruct 3D colon structures. While RNNSLAM cannot work because it requires DSO to refine the camera pose and DSO fails when camera moves relatively fast. For dataset 2, 3 and 4, although RNNSLAM can obtain results with slow camera motion, the results show that the approach in RNNSLAM is prone to large drift especially when the colon datasets have deformation, while our approach has much less drift. Also, textures from our framework are clearer and missing regions are shown more clearly in the maps, this is mainly because our approach handles deformation while RNNSLAM does not consider that.

5.3.3 Validation using In-vivo Datasets and Compared to RNNSLAM

We use the same in-vivo videos as used in RNNSLAM in our in-vivo experiments. All the consecutive frames in the same video have certain overlaps and frames which cannot be clearly visible are removed. The first video contains 96 informative frames and the second video contains 129 informative frames with the resolution of 320×240. Fig. 5.10 shows the reconstructed 3D maps of colon chunks using our approach and RNNSLAM. Although there is no ground truth of colon maps, it can be seen that the reconstructed colon chunks from our method can recover clearer colon structures including colon folds and topological shape, while the reconstructed 3D colon meshes by RNNSLAM are flat and lose most of the geometrical structures. Meanwhile, the textures on our reconstructed map looks clearer compared to RNNSLAM.

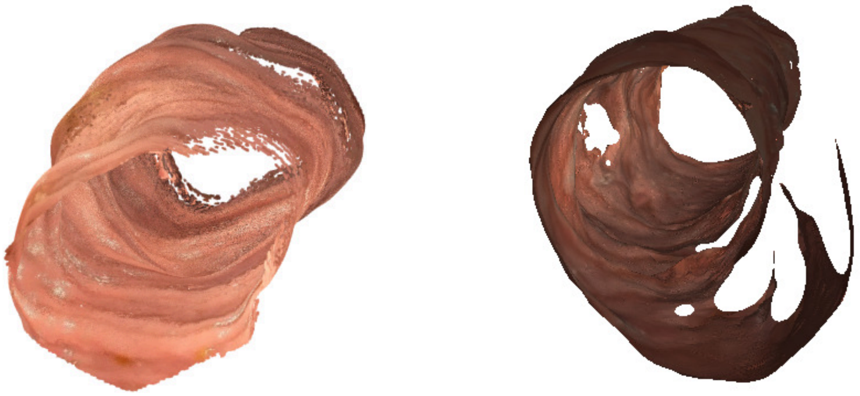


(a) RNNSLAM using dataset 2 (b) RNNSLAM using dataset 3

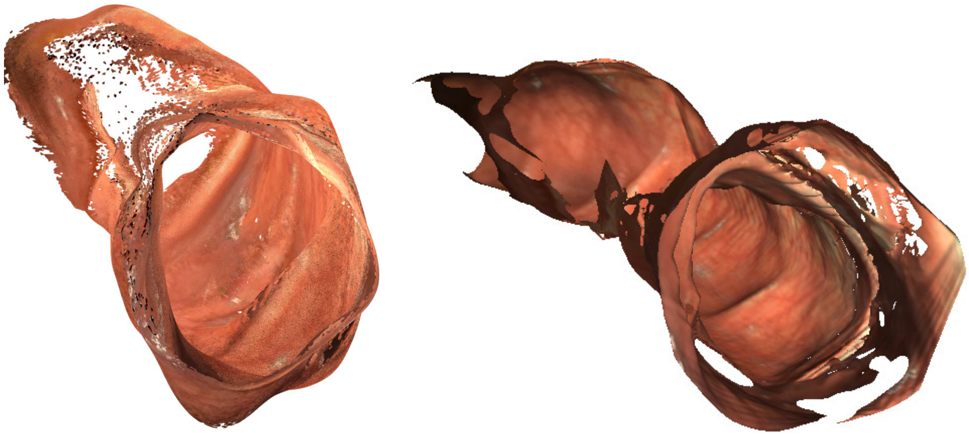


(c) RNNSLAM using dataset 4 (d) RNNSLAM using dataset 4

Figure 5.9: **Reconstructed Colon Maps using RNNSLAM on Smulation Datasets:** The first row shows the results using the method in RNNSLAM on dataset 2 and 3, respectively. No reconstructed colon map generated from dataset 1 by RNNSLAM when the camera moves fast. The left figure in second shows the reconstructed colon maps on datasets 4 by RNNSLAM. The right figure in the seond row shows the ground truth (in blue) and the reconstructed 3D colon from dataset 4 using RNNSLAM which shows the large drift.



(a) Ours using video 1 (b) RNNSLAM using video 1



(c) Ours using video 2 (d) RNNSLAM using video 2

FIGURE 5.10: **The Reconstruction of Colon Chunks using Our Approach and RNNSLAM:** (a) and (b) show the reconstructed maps on the first in-vivo dataset using our approach and RNNSLAM, respectively; (c) and (d) show the reconstructed maps on the second in-vivo dataset using our approach and RNNSLAM.

5.4 Chapter Summary

This chapter presents a robust framework that recovers 3D shape of deformable colon structures with textures from monocular colonoscopic videos. The proposed reconstruction framework uses a segmented colon CT model as the template to help deform and register 3D scans into a large 3D colon map while keeping the alignment of textures. The joint geometric and SIFT based photometric constraints are used to formulate a nonlinear least squares problem based on ED graph. The reconstructed map is obtained after solving the optimization problem. Validation by simulation and in-vivo experiments is conducted and the results demonstrate the practicality of the non-rigid 3D reconstruction framework.

Compared to the first framework, this framework uses deep learning techniques to estimate 3D scans and reconstruct 3D colon maps from monocular colonoscopic videos, and can deal with the colon deformation problem by using the proposed non-rigid registration method.

However, both the first and second framework have the following shortcomings that need further improved. First, the robustness and accuracy of the two frameworks highly rely on accurate SIFT feature-based sparse key correspondences extraction and camera initialization. However, the extraction of SIFT is cost computational and work poorly on some colonoscopic images with less texture. Second, the ICP algorithm is used to establish dense correspondences between 3D scans and the pre-operative colon model, this is also time-consuming and computationally costly. Third, the VO module for frame pose estimation works in an incrementally way, which is slow and suffers from drift. To solve the mentioned disadvantages, in the next chapter, a batch optimization-based framework is used to optimize all the frame poses together without extracting any sparse or dense correspondences.

Chapter 6

Direct Camera-Only Bundle Adjustment for 3D Textured Colon Surface Reconstruction Based on Pre-operative Model

In this chapter, we provide the third framework for 3D reconstruction of colon maps from monocular colonoscopic images. Different from the first and second frameworks that rely on VO module to estimate the camera poses w.r.t. the pre-operative colon mesh model, the colon 3D reconstruction problem in this chapter is formulated as a BA problem, which estimates all the frame poses simultaneously by maximizing the intensity consistency between the colon model vertices and multiple views of monocular colonoscopic images. The key novelty is that the proposed algorithm can avoid the extraction of sparse photometric and dense geometric 3D correspondences, which is significantly different from the state-of-the-arts where pair-wise correspondences are required which pose a great challenge for low-textured colonoscopic images.

Although the intensities of the pre-operative model vertices and all the colonoscopic camera poses are optimized together in the bundle adjusted formulation of this problem, the proposed method with GN iterations has the merit of optimizing camera poses only without optimizing the intensities of the model vertices, which significantly reduce the computational cost of the proposed algorithm. Building on this finding, we propose the camera-only BA to solve the camera poses only and show that the algorithm generates exactly the same camera poses in each iteration as the GN method. The optimal intensity of each vertex can be easily recovered using a closed-form formula after the optimal camera poses are obtained. Simulation results demonstrate the good performance of the proposed 3D colon surface reconstruction method in terms of accuracy and robustness. Phantom and in-vivo experimental results show the practicality of the proposed frameworks for providing useful shape and texture information in colonoscopy applications.

6.1 Methodology

6.1.1 Problem statement and Mathematical Formulation

In our proposed framework for solving the problem of textured colon surface reconstruction, the input is a pre-operative CT-segmented 3D colon mesh model M which consists of 3D vertices $\{\mathbf{v}_j\}$, $j \in \{1, ..., N_\mathbf{v}\}$ and triangles $\mathbf{F}_f$, $f \in \{1, ..., N_F\}$, and a sequence of observed 2D colonosopic images $\{I_i\}$, $i \in \{1, ..., K_I\}$, and the output is a textured 3D colonic surface map. Given an initial estimation of the camera poses and intensity values of model vertices, the direct BA jointly optimizes the camera poses $\{\xi_i\}_{i=1}^{K_I}$, $\xi_i \in \mathbb{R}^{6\times 1}$ as described in (2.6), and intensity values of model vertices $\{M(\mathbf{v}_j)\}_{j=1}^{N_\mathbf{v}}$, $M(\mathbf{v}_j) \in \mathbb{R}^1$, by minimizing the photometric reprojection errors between the mesh model M and the observed images $\{I_i\}_{i=1}^{K_I}$. Fig. 6.1 gives an overview the proposed approach.

Suppose the state to be estimated is defined as:

$$\mathbf{X} = \{\{\xi_i\}_{i=1}^{K_I}, \{M(\mathbf{v}_j)\}\}_{j=1}^{N_\mathbf{v}}\}^T \tag{6.1}$$

where the variable ξ_i is the camera pose for the i-th image and $M(\mathbf{v}_j)$ represents the intensity for the j-th vertex of the mesh model. Then, the photometric error between

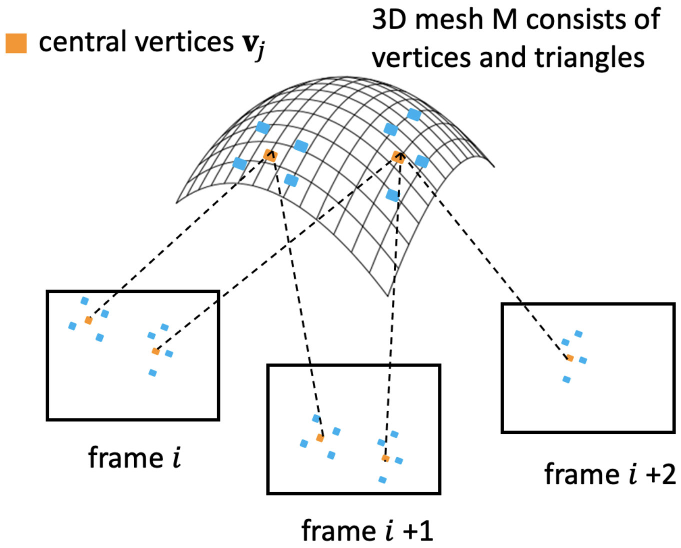

central vertices $\mathbf{v}_j$
3D mesh M consists of vertices and triangles
frame *i*
frame *i* +1
frame *i* +2

Figure 6.1: **Overview of the Proposed Approach.** The intensity differences between the mesh vertices and their re-projections onto the observed images are minimized to estimate the camera poses.

vertices $\mathbf{v}_j$ and its reprojection onto the i-th frame is defined as:

$$\begin{gathered} e_{ij}(\xi_i, M(\mathbf{v}_j)) = I_i(\mathbf{p}_{ij}) - M(\mathbf{v}_j)), \\ \mathbf{p}_{ij} = \pi(\mathbf{v}_j, \xi_i) \end{gathered} \tag{6.2}$$

where $I_i(\mathbf{p}_{ij})$ represents the intensity observation of vertices $\mathbf{v}_j$ projected onto the i-th image I_i, $\pi(\cdot)$ is the camera projection function in as shown in (2.8).

Overall, the proposed BA problem can be mathematically formulated as a nonlinear optimisation problem minimising:

$$\min_{\mathbf{X}} \sum_{i=1}^{K_I} \sum_{j=1}^{N_V} \rho(p_{ij}) \left\| e_{ij}(\xi_i, M(\mathbf{v}_j)) \right\|^2 \tag{6.3}$$

where $\rho(\mathbf{p}_{ij})$ is equal to 0 or 1, if $\mathbf{p}_{ij}$ is outside or inside the pixel plane of I_i.

It is noted that data association (feature extract and matching, nearest neighbour searching and loop closure detections) is not required in the proposed formulation described in (6.3). By contrast, data association is required and necessary in classical BA which refers to jointly optimize camera motion parameters (intrinsic and extrinsic parameters) and scene structures (3D landmarks) to ensure the 3D landmark projections match the detected 2D features.

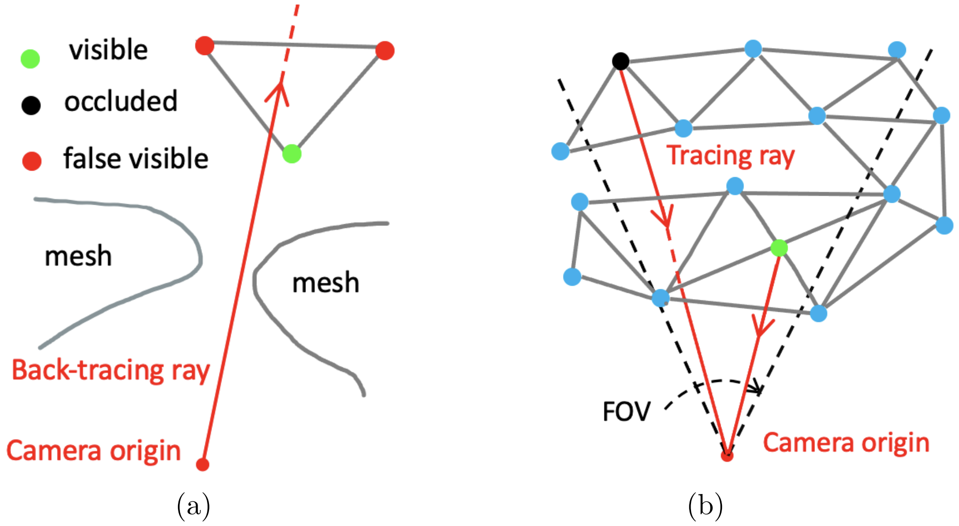
6.1.2 Determining Visibility

Visibility of vertices on the model needs to be considered in the proposed direct BA problem. As the camera view changes, some viewed 3D vertices will move out of the camera FOV and some new 3D vertices will become viewed, more importantly some vertices (in the camera FOV) will be occluded from the mesh structures. To automatically determine the 3D vertices visibility information is a challenging task.

Typically, the methods are either developed to determine the visible points in a point cloud or from a surface mesh. For the visibility estimation in point clouds (without reconstructing a surface mesh), the hidden point removal [73] method is a simple and fast method, it extracts the points that reside on the convex hull of a transformed point cloud, which amounts to determine the visible points. However, its accuracy strongly relies on the tuning of its sphere radius (a global parameter). For the method in [74], it estimates the visibility of each point by considering its screen-space neighborhood from a given view points, but this method relies on the parameter tuning of the estimated visibilities.

Ray-casting methods using barycentric-based ray-triangle intersection algorithm are commonly used for polygon mesh texture rendering and visibility determination. In detail, a ray shooting from the camera origin, goes through from every pixel center and into the scene space. If one casting ray hit more than one triangle, the ray-triangle intersected depth will be compared and the nearest intersected triangle will be selected as visible and be used for color rendering of the pixel. However, this ray-casting method cannot be used directly for the visibility determining of mesh vertices because it will introduce some occluded vertices and treat them as visible. As shown in Fig. 6.2 (a), if the back-tracing ray intersects with one triangle from the nearest distance, then the three vertices on the triangle are considered visible. In fact, two vertices of this triangle are occluded by the mesh structure. Meanwhile, ray-casting method is very time-consuming as each pixels are needs to tested against every single triangle in the mesh.

In contrast, as shown in Fig. 6.2 (b), our proposed method casts rays starting from each 3D vertex (within the camera FOV) and ending at the camera origin, if one casting ray does not intersect with any triangles, then this vertex is visible, otherwise, it is occluded and invisible. Our method is also much more efficient than classical ray-casting methods, since only the vertices within the camera FOV are tested.



(a) (b)

Figure 6.2: **Visibility Determining Methods using Barycentric Ray-triangle Intersection Technique:** (a) Occluded vertices are treated as visible using the classical ray-casting method; (b) our proposed method for visible vertices detection from a mesh.

6.2 Camera-Only BA Solution

6.2.1 Solving BA using Iterative GN Method

The iterative GN algorithm is used to minimise the objective function (6.3). We rewrite the objective function as:

$$F(\mathbf{X}) = \mathbf{f}(\mathbf{X})^T \Sigma^{-1} \mathbf{f}(\mathbf{X}) \tag{6.4}$$

where the vector $\mathbf{f}(\mathbf{X}) = [..., e_{ij}, ...]^T$ is defined by stacking all the valid ($\rho(p_{ij}) = 1$) error term functions and $\sum$ is the covariance matrix which is assigned to an identity matrix.

In each iteration, the GN solver first linearizes $\mathbf{f}$ in the neighborhood of current $\mathbf{X}$ with Taylor expansion:

$$\mathbf{f}(\mathbf{X} + \Delta\mathbf{X}) \approx \mathbf{f}(\mathbf{X}) + J(\mathbf{X})\Delta\mathbf{X} \tag{6.5}$$

where $J(\mathbf{X})$ is the Jacobian matrix of $\mathbf{f}(\mathbf{X})$ w.r.t. state vector $\mathbf{X}$ and the Jacobian matrix corresponding to the error term has the following form:

$$J_{ij}(\mathbf{X}) = \left[\mathbf{0}_{1\times6}, .., .\mathbf{0}_{1\times6}, \frac{\partial e_{ij}}{\partial \xi_i}, \mathbf{0}_{1\times6}, ..., \mathbf{0}_{1\times6}, 0, ..., 0, \frac{\partial e_{ij}}{\partial M(\mathbf{v}_j)}, 0, ..., 0\right], \tag{6.6}$$

where

$$\begin{aligned} \frac{\partial e_{ij}(\xi_i, M(\mathbf{v}_j))}{\partial \xi_i} &= \frac{\partial I_i}{\partial \mathbf{p}_{ij}} \frac{\partial \mathbf{p}_{ij}}{\partial \xi_i}, \\ \frac{\partial e_{ij}(\xi_i, M(\mathbf{v}_j))}{\partial M(\mathbf{v}_j)} &= -1, \end{aligned} \tag{6.7}$$

$\partial e_{ij} \,/\partial \xi_{ij}$ is the partial derivative of the error term w.r.t. the camera pose and can be calculated using (2.10), which has a dimension of 1 × 6. $\partial e_{ij} \,/\partial M(\mathbf{v}_j)$ is the partial derivative of the error term w.r.t. the vertex intensity, which has dimension of 1 × 1. $\partial I_i \,/\partial \mathbf{p}_{ij}$ is the image intensity gradient on pixel $\mathbf{p}_{ij}$.

Then, the step change $\Delta\mathbf{X}$ can be obtained by solving the following linear equation:

$$(J(\mathbf{X})^T J(\mathbf{X}))\boldsymbol{\Delta}\mathbf{X} = -J(\mathbf{X})^T \mathbf{f}(\mathbf{X}) \tag{6.8}$$

We define the coefficients $J^T J$ on the left as the approximation of the second-order Hessian matrix H and the coefficient on the right as g, then the (6.8) becomes:

$$H\boldsymbol{\Delta}\mathbf{X} = g \tag{6.9}$$

If $\boldsymbol{\Delta}\mathbf{X}$ is small enough that less than a threshold or the maximal iteration is reached, stop the algorithm. Otherwise update $\mathbf{X} = \mathbf{X} + \boldsymbol{\Delta}\mathbf{X}$ and repeat the iterative processing.

6.2.2 BA Sparsity and Schur Trick

To solve the linear equation in (6.8), we need to compute the inverse of matrix H, however, due to the high dimensions of the matrix $H \in \mathbb{R}^{(N_I\times6+N_V)\times(N_I\times6+N_V)}$, such an inversion operation is computationally cost.

An important property of BA is the sparsity pattern of the Gaussin Hessian matrix H given by the Jacobian matrix J [75, 76]. As we can find that, in (6.6), $J_{i,j}$ only has non-zero blocks in column i and j, this indicates that the error term $e_{i,j}$ is only relared to for ξ_i and $M(\mathbf{v}_j)$, and independent of other frame poses and intensities of vertices. Correspondly, $J_{i,j}$ will add four non-zero blocks into the overall Hessian matrix H at the positions $[i,i]$, $[i,j]$, $[j,i]$ and $[j,j]$. Thus H can be formulated as the folllowing form:

$$H = \sum_{i=1}^{K_I} \sum_{j=1}^{N_{\mathbf{v}}} J_{i,j}^T J_{i,j} \tag{6.10}$$

The Schur elimination [77] is used to make use of the sparsity of matrix H to speed up the solution process. Specifically, if we categorize the state $\mathbf{X}$ into image poses $\mathbf{X}_c = \{\xi_i\}_{i=1}^{K_I}$ and intensity values $\mathbf{X}_M = \{M(\mathbf{v}_j)\}_{j=1}^{N_{\mathbf{v}}}$ of mesh vertices vetice intensity, then the Jacobian matrix can be divided into two parts:

$$\begin{aligned} J &= [F, E] \\ J_{ij}(\mathbf{X}) &= \Bigg[\underbrace{\mathbf{0}_{1\times6}, ..., \mathbf{0}_{1\times6}, \frac{\partial e_{ij}}{\partial \xi_i}, \mathbf{0}_{1\times6}, ..., \mathbf{0}_{1\times6},}_{F_{ij}} \underbrace{0, ...0, \frac{\partial e_{ij}}{\partial M(\mathbf{v}_j)}, 0, ..., 0}_{E_{ij}}\Bigg] \end{aligned} \tag{6.11}$$

where $F = \partial \mathbf{f}(\mathbf{X}) \,/ \partial \mathbf{X}_c$ which is the partial deriavative of the entire cost function $\mathbf{f}(\mathbf{X})$ w.r.t. all frame poses and $E = \partial \mathbf{f}(\mathbf{X}) \,/ \partial \mathbf{X}_M$ which is the partial deriavative of the entire cost function $\mathbf{f}(\mathbf{X})$ w.r.t. all intensities of mesh vertices.

Then, So the Gaussian Hessian matrix can be formulated as the following form:

$$H = J^T J = \begin{bmatrix} F^T F & F^T E \\ E^T F & E^T E \end{bmatrix} \tag{6.12}$$

The linear equation in (6.9) can be rewritten as:

$$\begin{bmatrix} F^T F & F^T E \\ E^T F & E^T E \end{bmatrix} \begin{bmatrix} \Delta \mathbf{X}_c \\ \Delta \mathbf{X}_M \end{bmatrix} = \begin{bmatrix} -F^T \mathbf{f}(\mathbf{X})) \\ -E^T \mathbf{f}(\mathbf{X})) \end{bmatrix} \tag{6.13}$$

where $F^T F$ is a block-diagonal matrix and the dimension of each diagonal block is the same as the dimension of the camera poses. And $E^T E$ is also a block-diagonal matrix and each diagonal block is just a scalar. To compute the inverse of a block-diagonal matrix, we just need to invert the non-zero diagonal blocks separately. Thus, it takes less computational cost to compute the inverse of a block-diagonal matrix compared to a general matrix.

Schur elimination is applied to (6.9) as following:

$$\begin{bmatrix} I & -F^T E(E^T E)^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} F^T F & F^T E \\ E^T F & E^T E \end{bmatrix} \begin{bmatrix} \Delta \mathbf{X}_c \\ \Delta \mathbf{X}_M \end{bmatrix} = \begin{bmatrix} I & -F^T E(E^T E)^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} -F^T \mathbf{f}(\mathbf{X}) \\ -E^T \mathbf{f}(\mathbf{X}) \end{bmatrix} \tag{6.14}$$

By rearranging (6.14), we can obtain:

$$\begin{aligned} (F^T F - F^T E(E^T E)^{-1} E^T F)\Delta \mathbf{X}_c &= -F^T \mathbf{f}(\mathbf{X}) + F^T E(E^T E)^{-1} \mathbf{f}(\mathbf{X}) \\ (E^T F)\Delta \mathbf{X}_c + (E^T E)\mathbf{X}_M &= -E^T \mathbf{f}(X) \end{aligned} \tag{6.15}$$

After the schur elimination, the first equation in (6.15) becomes independent of $\Delta \mathbf{X}_M$. It is easily to solve $\Delta \mathbf{X}_c$ first and substitute the solved $\Delta \mathbf{X}_c$ into the second equation in (6.15) to solve $\Delta \mathbf{X}_M$.

6.2.3 Pose Only Optimization without Intensity

By analyzing the special structure of the matrix E, we can derive that in each iteration of the GN optimization, the estimation of all frame poses is independent of the intensity of mesh vertices. Even if the intensities of mesh vertices are set to random values before each iteration, the estimated intensities after the iteration never change. Thus, we can propose an algorithm to optimize the frame poses only, and the intensities of mesh vertices can be obtained by a closed-form formula after the optimal frame poses are estimated.

Let $\mathbf{Z} = [..., I_i(\mathbf{p}_{ij}), ...]^T$ represent the measurement vector which contains all the intensities of vertices observed in (6.3) and $\mathbf{X}_M^Z = [..., M(\mathbf{v}_j), ...]^T$ containing all the corresponding intensities of vertices in the state $\mathbf{X}_M$ w.r.t. the measurement vector $\mathbf{Z}$. Then, the entire cost error function vector $\mathbf{f}(\mathbf{X})$ in (6.5) is formulated as $\mathbf{Z} - \mathbf{X}_M^Z$, and the squares optimization problem in (6.3) is to seek $\mathbf{X}$ which minimizes:

$$\left\|\mathbf{Z} - \mathbf{X}_M^Z\right\|^2 \tag{6.16}$$

For the matrix E which is the Jacobian of $\mathbf{f}(\mathbf{X})$ w.r.t. $\mathbf{X}_M$, in each row, there is only one non-zero element whose value is equal to -1, and its row index and column index represent the index of observation term and the index of observed vertex in $\mathbf{X}_M$, respectively. Thus, the Jacobian matrix E is a constant matrix and $\mathbf{X}_M^Z = E\mathbf{X}_M$.

Now we prove that solving the update of frame poses in (6.15) is independent of the intensities $\mathbf{X}_M$ of vertices. Back substitute $\mathbf{f}(\mathbf{X}) = \mathbf{Z} - \mathbf{X}_M^Z$ and $\mathbf{X}_M^Z = E\mathbf{X}_M$ into the first row of (6.15), we have:

$$(F^T F - F^T E(E^T E)^{-1} E^T F)\Delta\mathbf{X}_c = -F^T\mathbf{Z} + F^T E(E^T E)^{-1} E^T\mathbf{Z} \tag{6.17}$$

where the update of frame poses $\mathbf{X}_c$ is independent of intensities $\mathbf{X}_M$ of vertices.

After the optimal camera poses $\hat{\mathbf{X}}_c$ are obtained, the proposed direct BA problem in (6.4) becomes a linear least squares problem and the optimal intensities $\hat{\mathbf{X}}_M$ of vertices can be easily recovered using a closed-form formula:

$$(F^T F)\hat{\mathbf{X}}_M = -F^T\mathbf{Z} \tag{6.18}$$

The optimal RGB values of vertices can also be calculated by the closed-form formula using different channels of color images in (6.18) separately and used for texture rendering the pre-operative colon model, and the textured regions on the colon model are actually the visible maps viewed by all the frames.

6.2.4 Pre-computaion of Gridded Intensity and Gradient Field

Besides using the sparsity of H to perform Schur elimination for improving the computation efficiency of the algorithm. To further improve the efficiency and accuracy of the proposed algorithm, first, we pre-compute the gridded gradient field of intensity for all the images , then in (6.7), the intensity gradient of any pixel in one frame can be obtained directly. Second, we also pre-compute the gridded intensity field for all the images since the pixels projected from mesh vertices commonly are not integers, then the intensity value of the projected pixels can be interpolated directly with high accuracy.

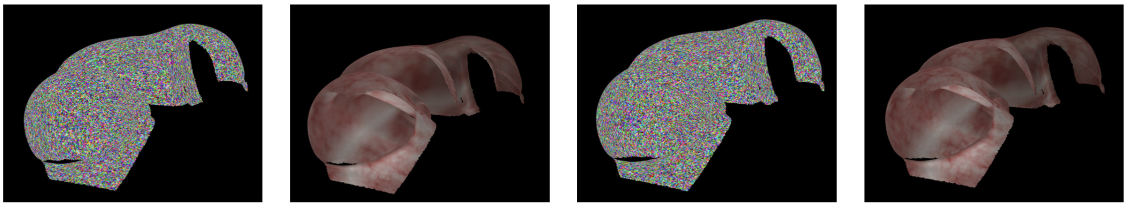
6.3 Experiments and Results

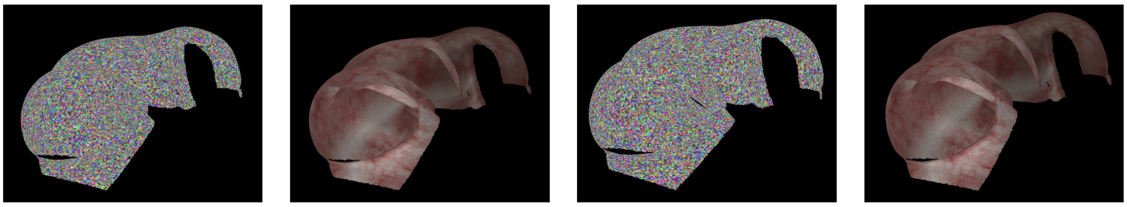
In this section, the proven theorem of independence of poses and intensities in Section 6.2 will be demonstrated. Then, the proposed direct camera-only BA framework for 3D textured colon reconstruction is validated using synthetic data collected from our developed colonoscopy simulator, phantom dataset from high-fidelity silicone colon models [26], and in-vivo datasets [52]. Since the human colon has a long tubular shape and the point light is moving with the colonoscope camera, the overall intensity in the closer part of the image is brighter than the farther part, which violates the lighting consistency assumption. Thus, in the proposed method, we truncate the valid depth range when projecting 3D mesh vertices onto 2D local frames (the far part of model vertices in camera FOV will not project on the local frames, only the near part model vertices in FOV works).

6.3.1 Proposed Theorem Validation

To visually validate the theorem that pose estimation is independent of the intensities of vertices, we use the three channels of RGB colors instead of pixel intensities as the observation in the proposed direct BA algorithm. Before each GN iteration, the RGB colors of vertices are set to random values. After each iteration, the optimized frame poses are used to calculate the RGB colors by the closed-form (6.18) separately and used for texture rendering the pre-operative colon mesh model.

As shown in Fig. 6.3, the colon map has mixed RGB colors before each iteration, and it is not possible to recognize any clear textures. After each iteration, the optimal RGB colors can be calculated directly, resulting in high-quality textures. Furthermore, the textures on the colon map become clearer with each iteration, which demonstrates the independence of the poses, intensities, or RGB colors of the model vertices.


Before iteration 1 After iteration 1 Before iteration 2 After iteration 2


Before iteration 3 After iteration 3 Before iteration 4 After iteration 4

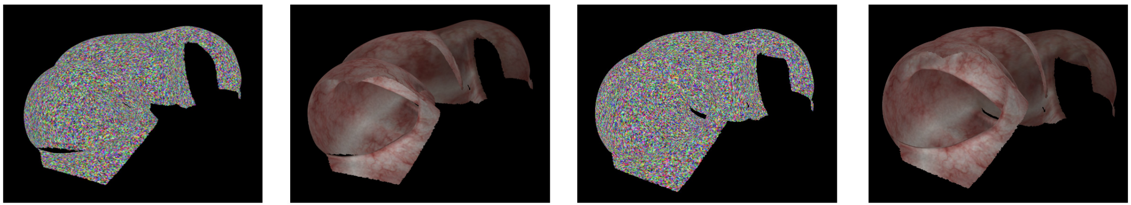

Before iteration 5 After iteration 5 Before iteration 6 After iteration 6

FIGURE 6.3: **Direct BA with Random Colors of Vertices for Validation of Pose Estimation Independent of the Intensities of Vertices.**

6.3.2 Simulation Experiments

The simulated dataset consists of three sequences (rectum colon segment, sigmoid colon segment and cecum colon segment) of 2D colonoscopic images and a pre-operative CT-segmented colon mesh model. The reconstruction results of the proposed method are compared to those of RNNSLAM [25, 52], DSO [44] and COLMAP [53] (DSO and COLMAP do not use a pre-operative model). For the proposed camera-only BA, the poses estimated by DSO are used to initialize the camera poses in state $\mathbf{X}$. After the optimization, the RGB information for observed model vertices can be easily calculated by the one step closed-form solution (6.18) using the optimized camera poses.

Fig. 6.4 shows the comparison of map reconstructions on three simulated colonoscopy sequences. Compared to the proposed method, RNNSLAM can also recover the overall topological shapes of colon structures, but the reconstructed maps are not complete as those from the proposed method. For example, the rear part of RNNSLAM's sigmoid colon map exhibits a structural collapse. This also happens to the outer edges of RNNSLAM's cucum colon map. The structural collapse or missing problem is mainly caused by the inaccuracy of depth and pose estimation. For the colon maps reconstructed by COLMAP and DSO, they are sparse and semi-dense 3D points, and it is not easy to recognize the main topological structures of the colon segments from the reconstructed maps. Meanwhile, the map points reconstructed from DSO suffer from large noise due to the large errors in estimating inverse depth of points. In contrast, the maps reconstructed using the proposed method show high quality structures and textures.

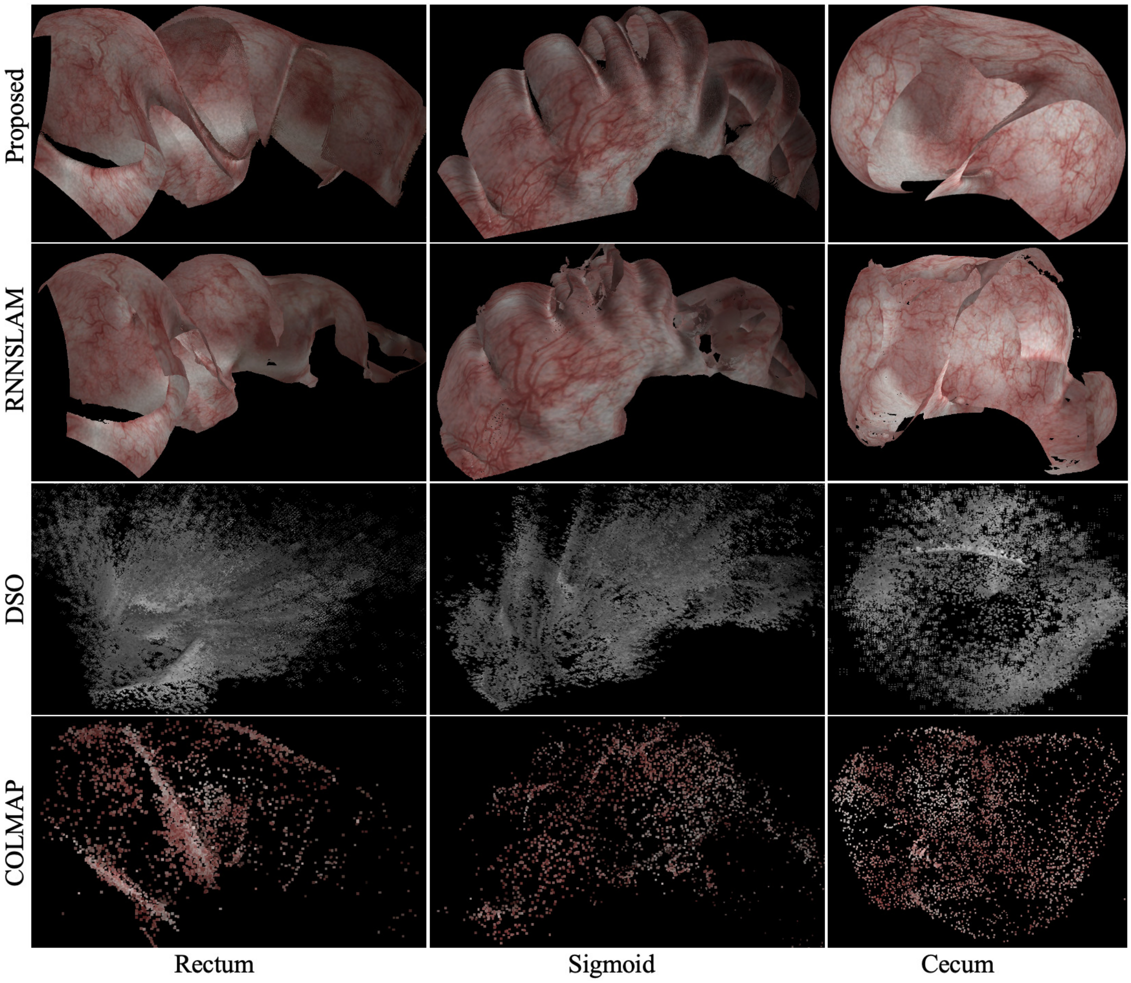
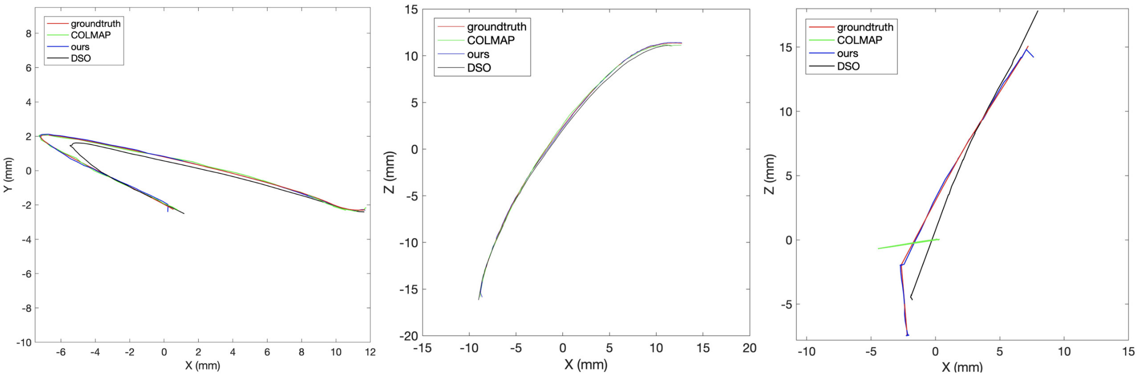


Figure 6.4: **Reconstructed Colon Maps on the Three Simulated Datasets using the Proposed Method, RNNSLAM, DSO and COLMAP.**

Fig. 6.5 shows a comparison of ground truth and the estimated trajectories on simulated datasets using the proposed method, DSO and COLMAP. Note that RNNSLAM takes the output poses from DSO as its input poses. The trajectory estimated by the proposed method is closest to the ground truth. The pose evaluation errors for the three methods, as compared with the ground truth, are shown in Table 6.1. This table shows that our method achieves the best result on all the metrics.



Rectum Sigmoid Cecum

FIGURE 6.5: **The Comparison of Ground Truth and Estimated Trajectories on Simulated Datasets using the Proposed Method, DSO and COLMAP.**

TABLE 6.1: Pose Evaluation Errors (mm) on the Simulated Colonoscopic Sequences.

Dataset	Rectum				Sigmoid				Cecum			
Metrics	rmse	mean	median	std	rmse	mean	median	std	rmse	mean	median	std
Proposed	**0.117**	**0.104**	**0.089**	**0.054**	**0.094**	**0.084**	**0.074**	**0.042**	**0.155**	**0.105**	**0.081**	**0.115**
DSO	0.761	0.694	0.652	0.315	0.119	0.103	0.092	0.059	2.054	1.793	1.832	1.009
COLMAP	0.686	0.423	0.230	0.545	0.411	0.308	0.254	0.274	9.190	8.227	8.237	4.120

6.3.3 Phantom Experiments

The phantom datasets [26] used consists of three sequences (cecum, descending and transcending colon segments) of colonoscopy images with corresponding surface mesh models. The coarse camera poses (with pose errors mainly caused by colon model dynamics, hand-eye calibration) provided by the electromagnetic sensors are used as the initial values for the proposed method. After optimization, the vertices observed by all the frames are textured using the proposed closed-form solution (6.18).

Fig. 6.6 shows a comparison of maps reconstructed by the proposed direct camera-only BA, RNNSLAM, DSO and COLMAP, respectively. From the results we can find that our approach can reconstruct colon maps with clear structures and consistent textures. The consistency of textures, calculated using the one step of closed-form, proves the high accuracy of the estimated camera poses. In the maps reconstructed by RNNSLAM, there are still instances of missing or collapsed regions. Moreover, the overall quality of the

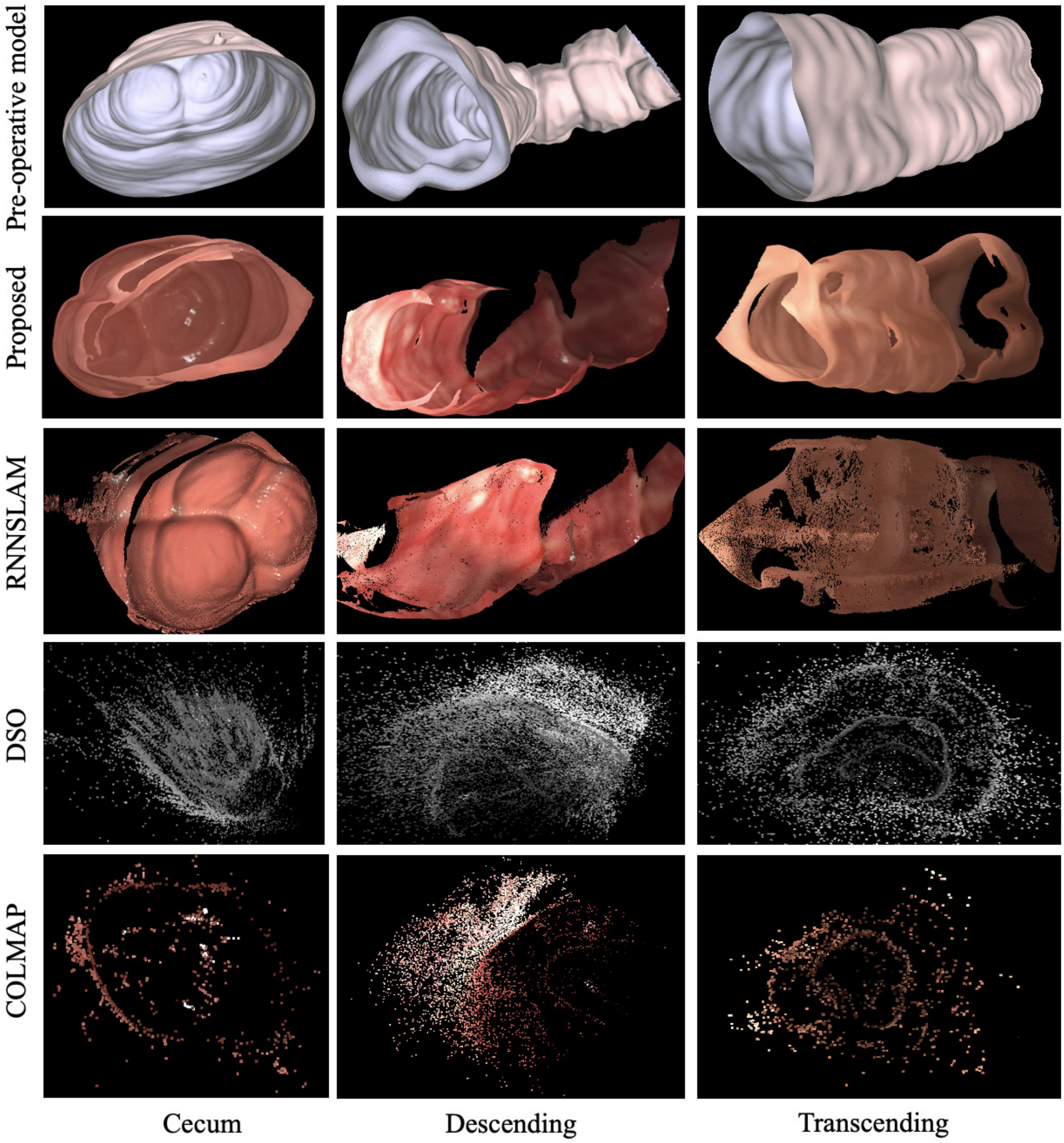


FIGURE 6.6: **Reconstructed Colon Maps on the Phantom Datasets using the Proposed Method, RNNSLAM, DSO and COLMAP, respectively.**

RNNSLAM maps reconstructed from phantom datasets is inferior to those from simulation datasets. This discrepancy arises from errors present in the "ground truth" depth dataset used to train RNNSLAM's depth estimation network.

6.3.4 In-vivo Experiments

The in-vivo dataset with two real colonoscopy videos [25, 52] are used to validate the practicality of the proposed method. The first video contains 53 frames, and the second contains 115 frames, all of which are clearly visible and undistorted. The poses estimated by DSO are used to initialize the poses in our proposed method. The proposed method is compared to the colon reconstruction method RNNSLAM, as well as DSO and COLMAP.

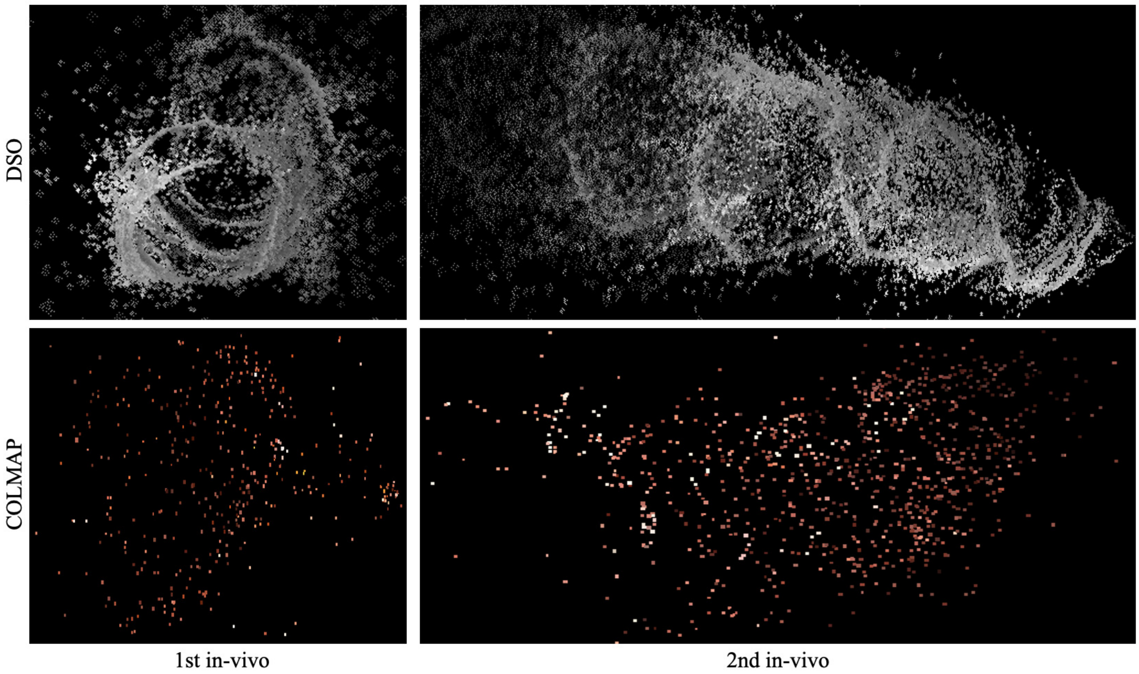


FIGURE 6.7: **Reconstructed Colon Maps on the In-vivo Datasets using DSO and COLMAP, respectively.**

The reconstructed 3D colon maps using DSO and COLMAP are shown in Fig. 6.7. The results show that the maps reconstructed from DSO and COLMAP are semi-dense points with large noise and very sparse points, respectively. DSO even fails when there is less overlap between consecutive frames.

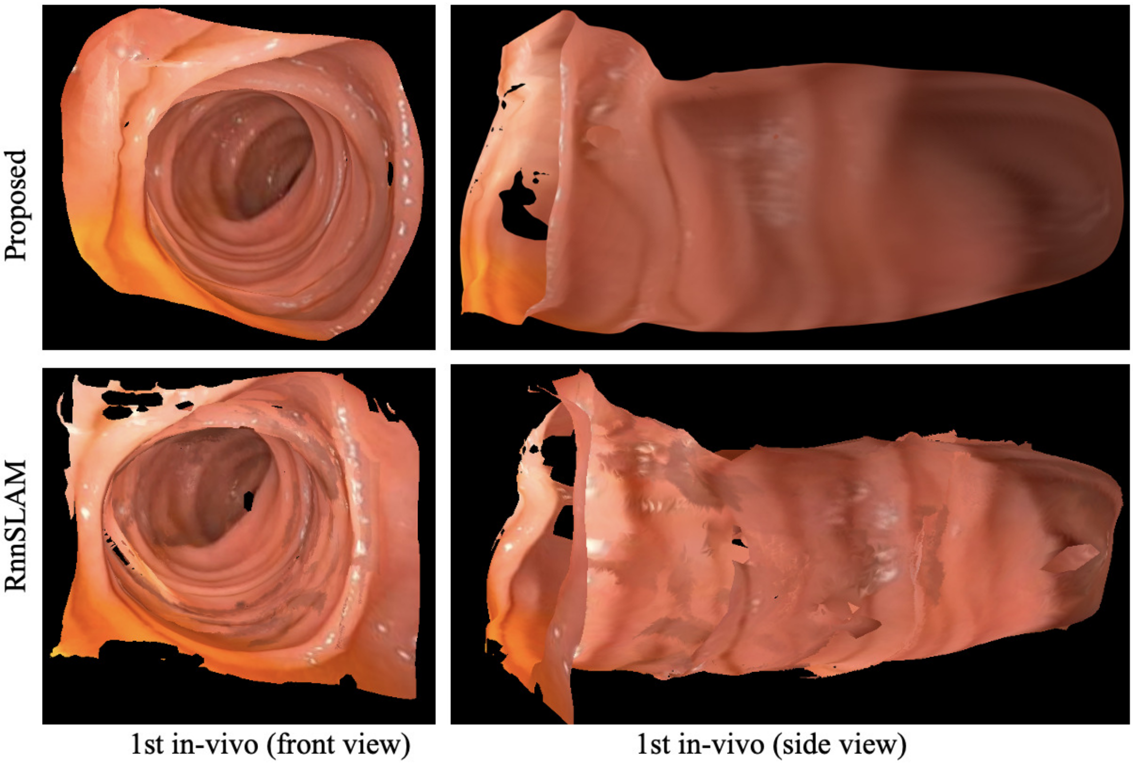


FIGURE 6.8: Reconstructed Colon Maps on the First In-vivo Dataset using the Proposed Method and RNNSLAM, respectively.

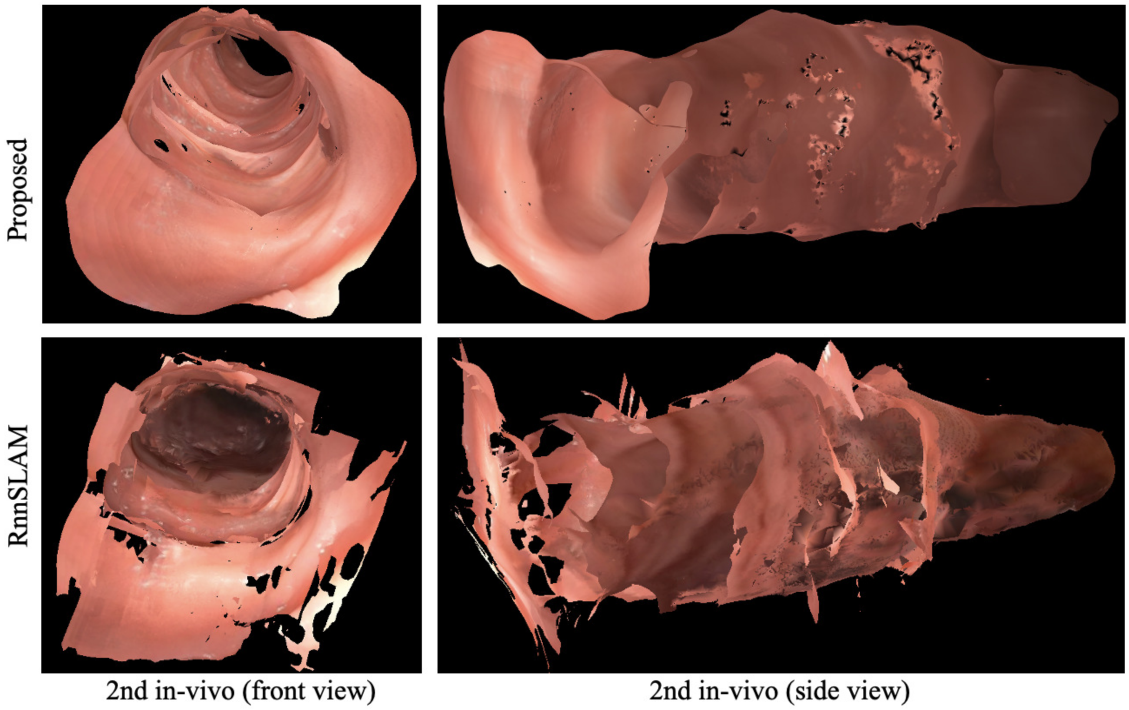


FIGURE 6.9: **Reconstructed Colon Maps on the Second In-vivo Dataset using the Proposed Method and RNNSLAM, respectively.**

Fig. 6.8 and Fig. 6.9 show the reconstructed 3D colon maps on the first and second in-vivo datasets using our approach and RNNSLAM, respectively. Both the front and side views of the reconstructed colon maps are shown. It can be seen that both the two methods can recover the main topological shape of the colon segments. However, the reconstructed colon maps from our method show clearer colon structures and textures.

6.4 Chaper Summary

This chapter presents a direct camera-only BA framework for 3D colon reconstruction, which optimizes all the frame poses simultaneously without requiring data assocation and image depth information. We prove that when solving the formulated direct BA problem using GN iterations, the pose estimation is completely independent of the intensities of vertices, which is more efficient than its traditonal BA formulation. Validations using simulation, phantom and in-vivo datasets have demonstrated the accuracy and feasibility of the proposed algorithm.

However, when there are relatively large intensity variations (lighting inconsistency) across multiple frames, it can result in pose accuracy degradation or failure. Thus, in the near future, we aim to further improve the proposed method by improving the lighting consistency of colonoscopy videos [78]. In addition, this proposed method currently works offline as our current focus is more on accuracy than efficiency. Specifically, the algorithm runs with each Gauss-Newton iteration typically taking around 5 minutes, and it requires approximately 30 iterations to converge. The most time-consuming step is the visibility determination procedure. Additionally, the number of mesh vertices or the density of the pre-operative mesh can significantly impact the computational cost of the algorithm. Thus, our future work will also focus on the efficient implementation of the proposed method, such as the time-consuming visibility determining procedure can be parallel implemented on GPU to achieve the fast implementation.

Chapter 7

Conclusion and Future Work

7.1 Summary of Contributions

In this thesis, we study colon 3D reconstruction techniques, provide three frameworks for reconstructing 3D textured colon maps by fusing a pre-operative 3D colon mesh model and a sequence of colonoscopic images, and develop a realistic colonoscopy simulator which can simulate the colonoscopy screening procedures inside the human colon and output colonoscopic images with ground truth poses and depths.

In summary, the contributions of this thesis are:

- Our first framework can reconstruct 3D colon maps with detailed textures from stereo colonoscopic images. It can robustly estimate the poses of 3D scans (reconstructed from paired stereo colonoscopic images using SGM algorithm) w.r.t. the pre-operative model and map textures from colonoscopic images to the registered regions on the pre-operative colon model. The experimental results show the feasibility and high accuracy of the proposed algorithm;
- In the second work, we improve the first work mainly in two apsects. First, we train a depth estimation network for monocular colonoscopic images. Second, we deal with the colon deformation challenge by proposing an ED-based non-rigid registration algorithm. The non-rigid registration algorithm can nonrigidly register 3D

colon scans to the pre-operative colon model, thus reducing the scale and structrure differences between scans, camera pose drift, and improving the texture consistency on the reconstructed colon map;

- In the third work, we formulate the camera pose estimation problem as a BA problem. The intensity difference between the model vertices and their projection onto all the colonoscopic images are minimized to jointly optimize the camera poses and intensities of the model vertices. The proposed framework has advantages over the previous two frameworks in that it can avoid the exhaustive extraction and tracking of features, does not use image depth information, and is more applicable to the colon 3D reconstruction from low-textured colonoscopic images;

- In the formulated BA problem of third framework, we prove that camera pose estimation is independent of the intensities of model vertices in each iteration of the GN optimization. Thus, we propose the direct camera-only BA algorithm that only optimizes camera poses, which helps reduce the computational cost of the optimization. Then, the estimated camera poses are used to calculate the optimal intensities of mesh vertices using a closed-form;

- The developed colonoscopy simulator is used to provide different scenarios of colonoscopy datasets to validate 3D colon reconstruction algorithms. Simulation and phantom experiments are performed to demonstrate the good performance of the proposed frameworks, and in-vivo experiments are conducted to validate the potential clinical value of the proposed frameworks. To promote the research of colon reconstructions, the developed colonoscopy simulator together with source code have been made publicly available.

7.2 Future Work

There are some future directions that are natural extensions of this work. For the sake of clarity, we itemize them as follows:

- Our proposed frameworks require pre-oeprative CT-segmented colon mesh models (corresponding to colonoscopic videos) as the global map for colon reconstructions. Since the colonoscope moves very fast during the normal inspection procedures, pre-operative colon models are mainly used to reduce pose estimation drift and improve map reconstruction accuracy with the consistency of textures matching. However, usually a CT colonoscopy is not always performed before normal colonoscopy procedures, and therefore, no corresponding CT model can be provided. In the future, we will investigate colon reconstruction using one general model for different patients in cases when the pre-operative datasets are not available;

- The third proposed framework aims to avoid feature extraction and matching by using the intensity consistency assumption, and some good results have been achieved. However, when there are relatively large intensity variations (i.e., lighting inconsistency) across multiple frames, it can result in degraded pose accuracy or failure. Thus, in the near future, we aim to further improve the proposed method by enhancing the lighting consistency of colonoscopy videos [78]. Moreover, the proposed method currently works offline as our current focus is more on accuracy than efficiency. Our future work will also focus on the efficient implementation of the proposed method. For instance, the time-consuming visibility determining procedure can be implemented in parallel on a GPU to achieve faster implementation;

- The developed colonoscopy simulator is crucial for developing and validating deformable colon reconstruction algorithms. However, in the current iteration of our simulator, we have observed notable disparities in texture and color when comparing the simulated colonoscopic images to actual images. This discrepancy primarily arises due to the utilization of a limited set of authentic images for generating the 2D texture map. In the near future, we are committed to enhancing the simulator by integrating a more extensive collection of genuine colonoscopic images to refine its color and texture representation. Simultaneously, we will incorporate deformations resulting from inflation and deflation to further enhance its fidelity to real-world scenarios. Upon the successful completion of the development and rigorous testing phases, we will proceed to offer the new version of the developed colonoscopy simulator for lease.

- Reconstructing 3D colon maps from colonoscopic videos is a very challenging task. In addition to the challenges listed in Section 1.3, there are many other challenges need to be addressed, such as the blurring of images. The main factors that cause image blurring include insufficient air inflation, colonoscopic lens fogging, or the lens being stained with fecal matter or opaque water in the lumen. In our study, only clear and visible colonoscopic images are used to reconstruct 3D colon maps. With the aim of reconstructing colon maps in real-time during the colonoscopy procedure, it will be worthwhile to investigate how to automatically detect blurry images or even remove blur from the blurry images.

Bibliography

[1] Colonoscopy, the worlds of david darling. `https://www.daviddarling.info/encyclopedia/C/colonoscopy.html`, 2016. Accessed: 2016.

[2] AM Leufkens, MGH Van Oijen, FP Vleggaar, and PD Siersema. Factors influencing the miss rate of polyps in a back-to-back colonoscopy study. *Endoscopy*, 44(05): 470–475, 2012.

[3] Jeroen C Van Rijn, Johannes B Reitsma, Jaap Stoker, Patrick M Bossuyt, Sander J Van Deventer, and Evelien Dekker. Polyp miss rate determined by tandem colonoscopy: a systematic review. *Official journal of the American College of Gastroenterology— ACG*, 101(2):343–350, 2006.

[4] Chantal MC le Clercq, Mariëlle WE Bouwens, Eveline JA Rondagh, C Minke Bakker, Eric TP Keulen, Rogier J de Ridder, Bjorn Winkens, Ad AM Masclee, and Silvia Sanduleanu. Postcolonoscopy colorectal cancers are preventable: a population-based study. *Gut*, 63(6):957–963, 2014.

[5] Jerome D Waye. Difficult colonoscopy. *Gastroenterology & Hepatology*, 9(10):676, 2013.

[6] Hongbin Zhu, Matthew Barish, Perry Pickhardt, and Zhengrong Liang. Haustral fold segmentation with curvature-guided level set evolution. *IEEE Transactions on Biomedical Engineering*, 60(2):321–331, 2012.

[7] Andrew Guinigundo. Is the virtual colonoscopy a replacement for optical colonoscopy? In *Seminars in oncology nursing*, volume 34, pages 132–136. Elsevier, 2018.

[8] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[9] Robert W Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. *ACM Transactions on Graphics (TOG)*, 26(3):80, 2007.

[10] Olek C Zienkiewicz, Robert Leroy Taylor, and Jian Z Zhu. *The finite element method: its basis and fundamentals*. Elsevier, 2005.

[11] Fernando De Goes and Doug L James. Regularized kelvinlets: sculpting brushes based on fundamental solutions of elasticity. *ACM Transactions on Graphics (TOG)*, 36(4):1–11, 2017.

[12] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11 (6):567–585, 1989.

[13] Marco Riva, Patrick Hiepe, Mona Frommert, Ignazio Divenuto, Lorenzo G Gay, Tommaso Sciortino, Marco Conti Nibali, Marco Rossi, Federico Pessina, and Lorenzo Bello. Intraoperative computed tomography and finite element modelling for multimodal image fusion in brain surgery. *Operative Neurosurgery*, 18(5):531–541, 2020.

[14] Sarah Frisken, Ma Luo, Parikshit Juvekar, Adomas Bunevicius, Ines Machado, Prashin Unadkat, Melina M Bertotti, Matt Toews, William M Wells, Michael I Miga, et al. A comparison of thin-plate spline deformation and finite element modeling to compensate for brain shift during tumor resection. *International journal of computer assisted radiology and surgery*, 15:75–85, 2020.

[15] Daniel E Hurtado, Nicolás Villarroel, Jaime Retamal, Guillermo Bugedo, and Alejandro Bruhn. Improving the accuracy of registration-based biomechanical analysis: a finite element approach to lung regional strain quantification. *IEEE Transactions on Medical Imaging*, 35(2):580–588, 2015.

[16] Hongjian Shi. *Finite element modeling of soft tissue deformation*. University of Louisville, 2007.

[17] Zhiyu Qiu, Huihui Tang, and Dongsheng Tian. Non-rigid medical image registration based on the thin-plate spline algorithm. In *2009 WRI World Congress on Computer Science and Information Engineering*, volume 2, pages 522–527. IEEE, 2009.

[18] Jian Zhao and Hui Zhang. Thin-plate spline motion model for image animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3657–3666, 2022.

[19] Mohammad Ali Armin, Nick Barnes, Jose Alvarez, Hongdong Li, Florian Grimpen, and Olivier Salvado. Learning camera pose from optical colonoscopy frames through deep convolutional neural network (cnn). In *Computer Assisted and Robotic Endoscopy and Clinical Image-Based Procedures: 4th International Workshop, CARE 2017, and 6th International Workshop, CLIP 2017, Held in Conjunction with MICCAI 2017, Quėbec City, QC, Canada, September 14, 2017, Proceedings 4*, pages 50–59. Springer, 2017.

[20] Anita Rau, PJ Eddie Edwards, Omer F Ahmad, Paul Riordan, Mirek Janatka, Laurence B Lovat, and Danail Stoyanov. Implicit domain adaptation with conditional generative adversarial networks for depth prediction in endoscopy. *International journal of computer assisted radiology and surgery*, 14:1167–1176, 2019.

[21] Gwangbin Bae, Ignas Budvytis, Chung-Kwong Yeung, and Roberto Cipolla. Deep multi-view stereo for dense 3d reconstruction from monocular endoscopic video. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 774–783. Springer, 2020.

[22] Daniel Freedman, Yochai Blau, Liran Katzir, Amit Aides, Ilan Shimshoni, Danny Veikherman, Tomer Golany, Ariel Gordon, Greg Corrado, Yossi Matias, et al. Detecting deficient coverage in colonoscopies. *IEEE Transactions on Medical Imaging*, 39(11):3451–3462, 2020.

[23] Mitchell J Fulton, J Micah Prendergast, Emily R DiTommaso, and Mark E Rentschler. Comparing visual odometry systems in actively deforming simulated colon environments. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4988–4995. IEEE, 2020.

[24] Kutsev Bengisu Ozyoruk, Guliz Irem Gokceler, Taylor L Bobrow, Gulfize Coskun, Kagan Incetan, Yasin Almalioglu, Faisal Mahmood, Eva Curto, Luis Perdigoto, Marina Oliveira, et al. Endoslam dataset and an unsupervised monocular visual odometry and depth estimation approach for endoscopic videos. *Medical image analysis*, 71: 102058, 2021.

[25] Ruibin Ma, Rui Wang, Yubo Zhang, Stephen Pizer, Sarah K McGill, Julian Rosenman, and Jan-Michael Frahm. Rnnslam: Reconstructing the 3d colon to visualize missing regions during a colonoscopy. *Medical image analysis*, 72:102100, 2021.

[26] Taylor L Bobrow, Mayank Golhar, Rohan Vijayan, Venkata S Akshintala, Juan R Garcia, and Nicholas J Durr. Colonoscopy 3d video dataset with paired depth from 2d-3d registration. *arXiv preprint arXiv:2206.08903*, 2022.

[27] Faisal Mahmood, Richard Chen, and Nicholas J Durr. Unsupervised reverse domain adaptation for synthetic medical images via adversarial training. *IEEE transactions on medical imaging*, 37(12):2572–2581, 2018.

[28] Berthold KP Horn. A method for obtaining the shape of a smooth opaque object from one view. *PhD thesis, Massachusetts Institute of Technology, Cambridge, 1970*, 1970.

[29] Alexandros Karargyris and Nikolaos Bourbakis. Three-dimensional reconstruction of the digestive wall in capsule endoscopy videos using elastic video interpolation. *IEEE transactions on Medical Imaging*, 30(4):957–971, 2010.

[30] Dan Koppel, Chao-I Chen, Yuan-Fang Wang, Hua Lee, Jia Gu, Allen Poirson, and Rolf Wolters. Toward automated model building from video in computer-assisted diagnoses in colonoscopy. In *Medical Imaging 2007: Visualization and Image-Guided Procedures*, volume 6509, page 65091L. International Society for Optics and Photonics, 2007.

[31] Chao-I Chen, Dusty Sargent, and Yuan-Fang Wang. Modeling tumor/polyp/lesion structure in 3d for computer-aided diagnosis in colonoscopy. In *Medical Imaging 2010: Visualization, Image-Guided Procedures, and Modeling*, volume 7625, page 76252F. International Society for Optics and Photonics, 2010.

[32] Jin Zhou, Ananya Das, Feng Li, and Baoxin Li. Circular generalized cylinder fitting for 3d reconstruction in endoscopic imaging based on mrf. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, 2008.

[33] DongHo Hong, Wallapak Tavanapong, Johnny Wong, JungHwan Oh, and Piet C De Groen. 3d reconstruction of virtual colon structures from colonoscopy images. *Computerized Medical Imaging and Graphics*, 38(1):22–33, 2014.

[34] Mohammad Ali Armin, Girija Chetty, Hans De Visser, Cedric Dumas, Florian Grimpen, and Olivier Salvado. Automated visibility map of the internal colon surface from colonoscopy video. *International Journal of Computer Assisted Radiology and Surgery*, 11(9):1599–1610, 2016.

[35] Faisal Mahmood and Nicholas J Durr. Deep learning and conditional random fields-based depth estimation and topographical reconstruction from conventional endoscopy. *Medical image analysis*, 48:230–243, 2018.

[36] Xingtong Liu, Ayushi Sinha, Masaru Ishii, Gregory D Hager, Austin Reiter, Russell H Taylor, and Mathias Unberath. Dense depth estimation in monocular endoscopy with self-supervised learning methods. *IEEE transactions on medical imaging*, 39(5):1438–1447, 2019.

[37] Shawn Mathew, Saad Nadeem, Sruti Kumari, and Arie Kaufman. Augmenting colonoscopy using extended and directional cyclegan for lossy image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4696–4705, 2020.

[38] J Kowalczuk, A Meyer, J Carlson, ET Psota, S Buettner, LC Pérez, SM Farritor, and D Oleynikov. Real-time three-dimensional soft tissue reconstruction for laparoscopic surgery. *Surgical Endoscopy*, 26(12):3413–3417, 2012.

[39] Nicholas J Durr, Germán González, and Vicente Parot. 3d imaging techniques for improved colonoscopy, 2014.

[40] Alberto Arezzo, Nereo Vettoretto, Nader K Francis, Marco Augusto Bonino, Nathan J Curtis, Daniele Amparore, Simone Arolfo, Manuel Barberio, Luigi Boni, Ronit Brodie,

et al. The use of 3d laparoscopic imaging systems in surgery: Eaes consensus development conference 2018. *Surgical endoscopy*, 33(10):3251–3274, 2019.

[41] Tobias Bergen and Thomas Wittenberg. Stitching and surface reconstruction from endoscopic image sequences: a review of applications and methods. *IEEE journal of biomedical and health informatics*, 20(1):304–321, 2014.

[42] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[43] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007.

[44] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017.

[45] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[46] Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza. Svo: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2):249–265, 2016.

[47] Rui Wang, Martin Schworer, and Daniel Cremers. Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3903–3911, 2017.

[48] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016.

[49] M Kaess, M Fallon, H Johannsson, and JJ Leonard. Kintinuous: Spatially extended kinectfusion. In *Proceedings of the RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras, Sydney, Australia*, pages 9–10, 2012.

[50] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136. IEEE, 2011.

[51] Richard J Chen, Taylor L Bobrow, Thomas Athey, Faisal Mahmood, and Nicholas J Durr. Slam endoscopy enhanced by adversarial depth prediction. *arXiv preprint arXiv:1907.00283*, 2019.

[52] Ruibin Ma, Rui Wang, Stephen Pizer, Julian Rosenman, Sarah K McGill, and Jan-Michael Frahm. Real-time 3d reconstruction of colonoscopic surfaces for determining missing regions. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 573–582. Springer, 2019.

[53] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[54] Amaël Delaunoy and Marc Pollefeys. Photometric bundle adjustment for dense multi-view 3d modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1486–1493, 2014.

[55] Hatem Alismail, Brett Browning, and Simon Lucey. Photometric bundle adjustment for vision-based slam. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part IV 13*, pages 324–341. Springer, 2017.

[56] Zhehua Mao, Liang Zhao, Shoudong Huang, Yiting Fan, and Alex PW Lee. Dsr: Direct simultaneous registration for multiple 3d images. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VI*, pages 98–107. Springer, 2022.

[57] John K Haas. A history of the unity game engine. *Diss. Worcester Polytechnic Institute*, 483:484, 2014.

[58] Ron Kikinis, Steve D Pieper, and Kirby G Vosburgh. 3d slicer: a platform for subject-specific image analysis, visualization, and clinical support. In *Intraoperative imaging and image-guided therapy*, pages 277–289. Springer, 2013.

[59] Eric Keller. *Introducing ZBrush*. John Wiley & Sons, 2011.

[60] Ingrid Hoelzl and Rémi Marie. *Softimage: Towards a new theory of the digital image*. Intellect Books, 2015.

[61] Wikipedia contributors. Adobe photoshop — Wikipedia, the free encyclopedia, 2023. URL `https://en.wikipedia.org/w/index.php?title=Adobe_Photoshop&oldid=1146911454`. [Online; accessed 31-March-2023].

[62] Ruibin Ma, Qingyu Zhao, Rui Wang, James Damon, Julian Rosenman, and Stephen Pizer. Deforming generalized cylinders without self-intersection by means of a parametric center curve. *Computational Visual Media*, 4(4):305–321, 2018.

[63] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 807–814. IEEE, 2005.

[64] Friedrich Fraundorfer and Davide Scaramuzza. Visual odometry: Part i: The first 30 years and fundamentals. *IEEE Robotics and Automation Magazine*, 18(4):80–92, 2011.

[65] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003.

[66] Philip HS Torr and Andrew Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer vision and image understanding*, 78 (1):138–156, 2000.

[67] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proceedings Third International Conference on 3D Digital Imaging and Modeling*, pages 145–152. IEEE, 2001.

[68] Ofir Weber, Mirela Ben-Chen, Craig Gotsman, and Kai Hormann. A complex view of barycentric mappings. In *Computer Graphics Forum*, volume 30, pages 1533–1542. Wiley Online Library, 2011.

[69] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011.

[70] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[71] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*, 2018.

[72] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[73] S Katz. Direct visibility of point. sets. *ACM Trans Graphics*, 26(3):24–1, 2007.

[74] Pierre Biasutti, Aurélie Bugeau, Jean-François Aujol, and Mathieu Brédif. Visibility estimation in point clouds with variable density. In *VISIGRAPP (4: VISAPP)*, pages 27–35, 2019.

[75] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.

[76] Lukas Polok, Viorela Ila, Marek Solony, Pavel Smrz, and Pavel Zemcik. Incremental block cholesky factorization for nonlinear least squares in robotics. In *Robotics: Science and Systems*, pages 328–336, 2013.

[77] Fuzhen Zhang. *The Schur complement and its applications*, volume 4. Springer Science & Business Media, 2006.

[78] Yubo Zhang, Shuxian Wang, Ruibin Ma, Sarah K McGill, Julian G Rosenman, and Stephen M Pizer. Lighting enhancement aids reconstruction of colonoscopic surfaces. In *Information Processing in Medical Imaging: 27th International Conference, IPMI 2021, Virtual Event, June 28–June 30, 2021, Proceedings 27*, pages 559–570. Springer, 2021.