Feature Extraction and Feasibility Study on CT Image Guided Colonoscopy

Yuan Shen

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Committee Members:
Christopher L. Wyatt, Chair
William T. Baumann
Daniel J. Stilwell

April 22, 2010
Blacksburg, Virginia

Keywords: Feature Extraction, Image Guided Colonoscopy, SLAM, Video Feature Tracking

Feature Extraction and Feasibility Study on CT Image Guided Colonoscopy

Yuan Shen

(ABSTRACT)

Computed tomographic colonography(CTC), also called virtual colonoscopy, uses CT scanning and computer post-processing to create two dimensional images and three dimensional virtual views inside of the colon. Computer-aided polyp detection(CAPD) automatically detects colonic polyps and presents them to the user in either a first or second reader paradigm, with a goal reducing examination time while increasing the detection sensitivity. During colonoscopy, the endoscopists use the colonoscope inside of a patient's colon to target potential polyps and validate CAPD found ones. However, there is no direct information linking between CT images and the real-time optical colonoscopy(OC) video provided during the operation, thus endoscopists need to rely largely on their past experience to locate and remove polyps. The goal of this research project is to study the feasibility of developing an image guided colonoscopy(IGC) system that combines CTC images, real-time colonoscope position measurements, and video stream to validate and guide the removal of polyps found in CAPD. System would ease polyp level validation of CTC and improve the accuracy and efficiency of guiding the endoscopist to the target polyps. In this research project, a centerline based matching algorithm has been designed to estimate, in real time, the relative location of the colonoscope in the virtual colonoscopy environment. Furthermore, the feasibility of applying online simultaneous localization and mapping(SLAM) into CT image guided colonoscopy has been evaluated to further improve the performance of localizing and removing the pre-defined target polyps. A colon phantom is used to provide a testing setup to assess the performance of the proposed algorithms.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

In the United States, more than 50,000 people die of colorectal cancer each year. Yet advanced colorectal cancers are highly preventable since they start from a polyp, a growth of the colon lining. Although most polyps are harmless, over time larger ones may become cancer. Early detection and removal of polyps significantly reduces the incidence of colon cancer [1]. Computed tomographic colonography (CTC) is a non-invasive screening technique [2]. From CT images, the interior 3D rendering of the colonic wall structure can be generated, leading to an offline simulation of video colonoscopy. Currently, however, the interpretation of the entire CTC examination is time-consuming. In addition, it is prone to perceptual errors [3]. Computer-aided detection of polyps provides an efficient and automatic way to pre-detect colonic polyps in order to reduce examination time and cost. It analyzes the abdominal CT images to screen colonic polyps, a precursor of colon cancer. If a polyp is detected above the pre-defined threshold size, the patient is referred to optical colonoscopy (OC) for additional screening and polypectomy. The framework behind CTC typically consists of (1) segmentation of colonic wall, (2) feature extraction of polyps, and (3) classification for detection of final polyp candidates [4]. Within this area, there is a considerable amount of work that has been done already, continuously improving both the sensitivity and the specificity of polyp detection.

This work extends the concept of CTC to the guidance of polyp removal(polypectomy). Image guided surgery has revolutionized traditional surgical techniques by providing endoscopists with a way to navigate through the human body under the help of real-time generated three-dimensional images. Furthermore, those images can be changed, manipulated and merged to provide a level of detail not seen before in the operating room [5]. This research project aims to achieve the following goals:

1. Automatically identify the location of potential polyps.
2. Evaluate the feasibility of CT image guided colonoscopy system, which combines the information from CTC images and the diagnosis, which comes from sensor information of colonoscope position and the endoscopy video stream, to validate and guide the removal of polyps.

This system can benefit both CTC and OC. Firstly, by providing reliable correlation between CTC and OC, it eases the validation of polyp identified by CPAD. Secondly, it improves the accuracy and efficiency of the colonoscopy by visually guiding endoscopist to the target polyps.

Central to image guided colonoscopy is the real-time registration of radiologic CT image data and live endoscopic video. Before the colonoscopy, a full 3D CT colon scan is conducted. A computer-aided system generates a list of predefined polyp sites with additional size information. Then the system analyzes the live video stream from the colonoscopy for the data registration. The centerline matching algorithm, in coarse matching, is able to correct the colon deformation within a certain correction range. In fine matching, online simultaneous localization and mapping(SLAM) can further improve the quality of localizing pre-defined target polyps.

1.2 Related Work In Image Guided Procedure

The majority of prior work on endoscopic image guidance has been in applications where the structures of interest are relatively rigid, although some soft tissue structures have received some attention. Two common approaches are usually applied: processing the video stream alone or depending on additional information, such as binding sensors

to the colonoscope. In the former approach, either optical flow from monocular video [6] or stereo video reconstruction [7] have been applied to estimate the relative motion of endoscope camera relative to the anatomy. In the latter one, both electromagnetic and optical sensors have been used to measure camera position and orientation directly [8]. In this project, we combined both methods in the colon's special non-rigid environment.

Virtual Bronchoscopy is a similar technology to CTC that uses a CT exam of the thorax to construct a model of a patient's airway. Image guidance in virtual bronchoscopy for biopsy and related applications has been explored by several research groups. Solomon et al. [8] placed the sensor at the tip of bronchoscope and registered the CT scans with the patient during bronchoscopy by both multiple skin fiducial markers and inner structure of the trachea. Schwarz et al. [9] also used a bronchoscope modified with a sensor probe in fluoroscopy to measure the accuracy of the registration between the CT and the actual patient anatomy. Higgins et al. [10] relied only on video registration and fusion to map the endoscopic video with endoluminal rendering from CT scans.

The main difference between image guided bronchoscopy and the image guided colonoscopy is that the airway tree is a relatively rigid anatomical structure compared to the highly deformable colon lumen. Thus, while the equipment used to collect the real-time sensor measurements is almost identical, the registration problem is much more pronounced in CTC. Even re-positioning the patient can lead to large changes in the lumen shape [11]. Furthermore, insertion and traversing of the colonoscope exacerbates the problem to a large extent.

The only related work specifically in colonoscopy is that by Liu et al. [12]. Their approach is mainly based on constructing the optical flow between consecutive online images. Then the estimation of relative displacement between the colonoscopy camera and the colon lumen is acquired. The algorithm demonstrates tracking with a clear field of view for short periods of time. However optic-flow computation is quite susceptible to errors in estimating the motion field. Long sequence of blurry frames and abruptly changed field of view due to the fluid or camera collision to the wall can cause the failure of tracking. A robust method to recover from those failure has not been proposed yet.

In fact, a robust tracking over entire length of the colon requires multiple sources of complementary information. In this work, we add sensors to provide the real-time position and orientation measurement. This measurement helps us to build the online

centerline for the data registration between CT image data and the colonoscopy video stream. Furthermore simultaneous localization and mapping algorithm(SLAM) is also combined to provide a better accuracy to localize the target polyps.

Chapter 2

Computer Aided Polyp Detection

2.1 Overview

Colorectal cancer is caused by abnormal colon or rectum cells that grow without normal order and form a tumor later. These tumors usually start as polyps, most of which are benign and harmless. However small portion of them may still grow malignantly and eventually spread to other organs. Early detection and removal of these polyps (Figure 2.1) are essential for the prevention of colon cancer. Colonoscopy can perform this screening and is the follow-up procedure for any other screening test that indicates the presence of adenomatous polyps. Unlike the colonscopy, computed tomographic colonography(CTC) is minimally invasive way of detecting polyps and thus far it has matched the screening performance of colonscopy.

Figure 2.1: Polyp with diameter of 1.5cm has about 10% chance to be malignant

Computer-aided detection of polyps provides an efficient and automatic way to predetect colonic polyps in order to reduce examination time and cost. Given a study of CT images, the procedures in CTC consist of:

1. Segmentation of colonic wall
2. Detection and feature extraction of candidate polyps
3. Classification for the selection of final candidate polyps

Segmentation

The colon segmentation is based on the fact that there is a clear voxel intensity difference between colon tissue(40HU) and air-filled region(-1000HU). The improper patient preparation before the scan can lead segmentation into a challenging tasks, due to the leftover in the colon. Different algorithms have been used to solve these problems, such as the simple region growing, level set based method and anatomical knowledge based method [4].

Feature Extraction

A polyp candidate consists of a group of connected vertices, on which geometric features are calculated and applied to identify polyp candidates. Commonly used features include principal curvature, shape index, curvedness, radius and gradient concentration. Certain statistical operations mean, maximum, minimum, variance, skew, kurt and contrast can also help to characterize the features.

Classification

Support vector machine and several other algorithms have been used to tune the classification system. Given polyps identified by radiologist as training data, classification is able to improve the quality of automatic polyp identification in terms of sensitivity and specificity.

2.2 Polyp Feature

My major work in CAPD is focused on generating polyp features for their existence identification. A polyp candidate consists of a group of connected vertices or voxels, from which geometric features are calculated and classified to identify polyps. These candidates are initially detected using a subset of features and segmented roughly using various heuristic means to form candidate polyp regions, over which statistics can be computed per polyp. This section gives an introduction of the features we have collected for feature analysis and selection.

1. **Principal curvature, Gaussian curvature and Mean curvature** [13]: Principal curvature, measuring how the surface bends by different amounts in different directions at the specific point, can be computed from the first and second fundamental forms of differential geometry [14]. The equation to represent an iso-surface S at CT value of v in 3D space is given by $\{(x, y, z)\epsilon\Re^3; f(x, y, z) = v\}$. The implicit function theorem transforms the above iso-surface equation into the parametric form: $(x = u, y = v, z = \phi(u, v))$. Therefore, the iso-surface can also be expressed as:

$$S(u, v) = (u, v)\epsilon\Re^2; s(u, v, \phi(u, v) = v \tag{2.1}$$

Any point $S(u, v)$ on this surface, we have the first fundamental forms [14]

$$E = \parallel S_u \parallel, F = S_u \bullet S_v, G = \parallel S_v \parallel \tag{2.2}$$

The second fundamental forms [14] are defined as:

$$L = \frac{S_{uu} \bullet n}{A^{1/2}}, M = \frac{S_{uv} \bullet n}{A^{1/2}}, N = \frac{S_{vv} \bullet n}{A^{1/2}}, A = EG - F^2 \tag{2.3}$$

where $n(u, v)$ is a vector lying normal to the surface at $S(u, v)$. The following solutions can be derived by using implicit functions theorem and substitution of

derivatives of f for derivatives of S:

$$\begin{aligned} E = 1 + \frac{f_x^2}{f_z^2}, F = \frac{f_x f_y}{f_z^2}, G = 1 + \frac{f_y^2}{f_z^2} \\ L = \frac{2 f_x f_z f xz - f_x^2 f_{zz} - f_z^2 f_{xx}}{T} \\ M = \frac{f_x f_z f_{yz} + f_y f_z f_{xz} - f_x f_y f_{zz} - f_z^2 f_{x,y}}{T} \\ N = \frac{2 f_y f_z f yz - f_y^2 f_{zz} - f_z^2 f_{yy}}{T} \\ T = f_z^2 \sqrt{f_x^2 + f_y^2 + f_z^2} \end{aligned} \tag{2.4}$$

k_1, k_2 are respectively maximum and minimum principal curvatures, with associated direction $t1, t2$. $K = k_1 k_2$ is Gaussian curvature [13] and $H = \frac{k_1 + k_2}{2}$ is the mean curvature [13] of iso-surface.

$$\begin{aligned} K = \frac{LN - M^2}{EG - F^2}, H = \frac{EN - 2FM + GL}{2(EG - F^2)} \\ k_1 = H + \sqrt{H^2 - K}, k_2 = H - \sqrt{H^2 - K} \end{aligned} \tag{2.5}$$

2. **Shape Index(SI) and Curvedness(CV)** [15]: Shape index characterizes the topological shape of the volume around a voxel and it classifies five major classes: cup, rut, saddle, ridge and cap [4]. On the other hand, curvedness represents both the size of polyp and the magnitude of the curvature.

$$\begin{aligned} SI = \frac{1}{2} - \frac{1}{\pi} \arctan \frac{k_1 + k_2}{k_1 - k_2} \\ CV = \sqrt{\frac{k_1^2 + k_2^2}{2}} \end{aligned} \tag{2.6}$$

3. **Maximum polyp radius(MAXPR) and minimum polyp radius(MINPR)** [14]

$$\begin{aligned} MAXPR = \frac{1}{Min\ Principal\ Curvature\ k_2} \\ MINPR = \frac{1}{Max\ Principal\ Curvature\ k_1} \end{aligned} \tag{2.7}$$

4. **Compactness(C_p)** [16]: This is a dimensionless geometric descriptor and it reaches maximum for a sphere.

$$C_p = \frac{\sqrt[3]{36\pi V^2}}{SA} \tag{2.8}$$

where SA is total area of triangles on the colonic wall that represent polyp candidates and V is the volume of extracted polyp.

5. **Sphericity** [17] This is another measure of how spherical an object is.

$$Sphericity = \frac{Max\ Principal\ Curvature\ k_1 - Min\ Principal\ Curvature\ k_2}{Mean\ Curvature\ H} \tag{2.9}$$

6. **Gradient concentration(GC) and directional gradient concentrate(DGC)** [18]: These two features characterize the degree of convergence of the gradient vectors at voxel p.

$$GC(p) = \frac{1}{D}\sum_{i=1}^{D} e_i^{max}(p)$$

$$DGC(p) = \frac{1}{2D}\sum_{i=1}^{D/2}\begin{cases} |e_i^{max}(p) - e_{i+D/2}^{max}(p)| & if\ e_i^{max}(p), e_{i+D/2}^{max}(p) > 0 \\ e_i^{max}(p) + e_{i+D/2}^{max}(p) & otherwise \end{cases} \tag{2.10}$$

$$e_i^{max}(p) = max\frac{1}{n+1}\sum_{i=1}^{n}\cos\varphi_{il}(p)$$

where D is the number of symmetrically 3D oriented direction vectors originating from. R_{max} and R_{min} are predefined the radius of vicinity for calculating gradient orientation. $R_{min} \leq n < R_{max}$. φ_{il} is the angle between direction vector $\overrightarrow{d^i}$ and gradient vector $\overrightarrow{g_{il}}$, which is at distance l away from voxel p. GC reaches maximum at the center of Gaussian sphere. In reality, polyps are usually presented as a hemisphere object. Based on the modification of GC , DGC has the maximum value of 1 at the center of hemisphere.

7. Other additional features include: polyp wall and interior density in Hounsfield unit, number of vertices along polyp surfaces and volume of the polyp candidate.

2.3 Polyp Extraction

At each colonic wall voxel, principal, Gaussian, mean curvatures, SI and CV values are computed respectively. Voxels that have SI and CV values within predefined range are extracted as seed voxels by defining threshold values ($SI : [0, 0.15], CV[0.075mm^{-1}, 0.2mm^{-1}]$). From those seed voxels, a region growing procedure then occurs to extract the major portion of a polyp. It is reasonable to relax the ranges ($SI : [0, 0.22], CV[0.05mm^{-1}, 0.25mm^{-1}]$) around polyps peripheral region, since that portion usually presents a dome like shape. Then a fuzzy c means clustering [4] is applied to remove some polyp candidates due to image noise and effectively group voxels belonging to the same polyp in a large cluster. The clustering is based on voxels feature values, which include SI, CV, magnitude of the gradient, CT intensity value and spatial coordinates. Further features such as GC, DGC, sphericity, compactness, wall/region density and polyp radius are computed after the clustering step. Figure 2.2 shows an extracted polyp candidate and its surrounding colonic wall. The cross of red, yellow and green lines is the center of polyp, which has a hemisphere-like shape.

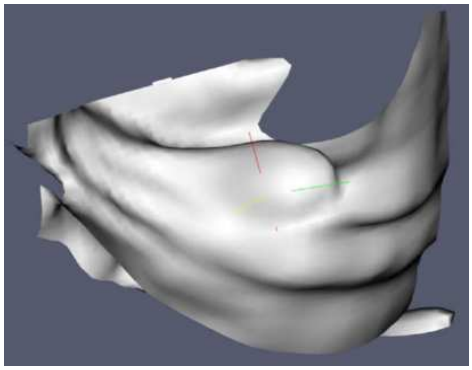


Figure 2.2: An extracted polyp candidate with surrounding colonic wall

2.4 Principle Component Analysis

We have collected 17 major features, 12 of which (shape index, curvedness, gradient magnitude, GC, DGC, wall density, interior density, polyp surface, maximum/minimum

principal curvature, surface Gaussian curvature, surface mean curvature and surface sphericity) are further characterized by 7 statistical operations: mean, maximum, minimum, variance, skew, kurt and contrast. Thus we consider a total number of 89 features in our test case. In order to improve the sensitivity and specificity of extracted polyps in CPAD, it is better to go through a feature reduction procedure to extract the dominant ones for later classification step.

Dimensionality reduction of the feature set is a common pre-processing step for pattern recognition. Methods applied in CTC include the wrapper [19] method to let the classifiers decide which features are useful, or progressive selection. Handling 89 features with the above methods could be computationally expensive. Instead, principal component analysis (PCA) is a good candidate on all the features described in last section in order to select the ones with most relevant and essential information. The method, proposed by I.T.Jolliffe [20] chooses the features corresponding to the highest coefficients of each of the first q principal components(PC) corresponding to q largest eigenvalues. Although this is a computationally efficient method, it considers each PC independently and features with similar information might be chosen. We used an optimized method by I. Cohen et al. [21] to select the important features without redundancy of information.

Let F be a N dimensional feature vector and its co-variance matrix is $\sum$. By eigen-decomposition:

$$\sum = VDV^T \tag{2.11}$$

where D is a diagonal matrix formed from eigenvalues of $\sum$, $\lambda_1 \geqslant \lambda_2 \geqslant ... \geqslant \lambda_n$ and columns of V are the corresponding eigen-vectors. Keep only the terms corresponding to the K largest eigenvalues by the following criterion:

$$Percentage\,of\,total\,variation = \frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{N} \lambda_i} \geqslant Threshold \tag{2.12}$$

Let V_K be the first K columns of V and $A_1, A_2, ...A_N \epsilon \Re^K$ be the rows of V_K. The feature reduction part would be summarized in the following steps:

1. Divide the feature data into true positive(TP) and false positive(FP) two sets

F_{TP} and F_{FP}. And feature reduction is applied to each of them.

2. Standardize the feature data prior to using PCA, since principal components are dependent on the units, as well as the range of values. The equation $\frac{f_i-\mu}{\sigma}$ (μ and σ are mean and standard deviation of feature f_i) makes sure all the feature data to have zero mean and unit standard deviation.

3. Compute the eigendecomposition according to Equation 2.11 and generate V_K with predefined threshold value 0.9 from Equation 2.12.

4. Cluster the vectors $A_1, A_2, ...A_N \epsilon \Re^K$ into $M \geqslant K$ using K-Means algorithm based on Euclidean distance between vectors. The number of clusters M should be no less than K in order to retain at least the same variability as the previous step.

5. From each cluster, pick up vector i A that is closest to the center of its cluster, since the corresponding feature f_i would best represent the cluster and other similar features in that cluster. This clustering method keeps the least redundant information of extracted essential features and large feature data spread in low dimensional space.

In the polyp feature extraction step, feature files are generated based on 20 CTC cases form Wake Forest University Baptist Medical Center and 20 from Walter Reed Army Medical Center. These data have difference in resolution, protocol, and bowel preparation quality that reflect the variety of CTC data encountered. Only polyp sizes, which are no less than 6mm and verified by a radiologist as either true positive (TP) or false positive (FP), are considered in the test. Using PCA method above, we are able to reduce features for TP set from 89 to 10, with the percentage of total variation equal to 0.9. The following table shows the dominant extracted features in TP set.

TPs
Maximum of Max principal curvature
Contrast of Max principal curvature
Variance of Gaussian curvature
Contrast of Gaussian curvature
Kurt of Gaussian curvature
Maximum of shape index
Maximum of gradient magnitude
Maximum of DGC
Variance of sphericity
Volume of polyp candidate

From the above 10 dominant features, it is easy to tell that most of them are based on geometric property of the polyp. In the TP class, Gaussian and principal curvature carry more information than others in identifying the poly candidates. Inside region density feature plays the least role in polyp feature extraction. Maximum, variance, contrast and kurt values of features outweigh minimum, mean and skew counterparts in this case.

The extracted features were then used by Xu et al. [22] to train, test and compare various classifiers, such as C4.5, Neural Network, Support Vector Machine and AdaBoost, in order to improve both the sensitivity and specificity on the identification of the polyp candidates.

Chapter 3

Centerline Matching

3.1 Overview

In the virtual environment, the efficiency of navigation capability, in terms of path planning, camera control model and real time surface rendering technique, plays an crucial role in the success of the whole system. H. Blum [23] defined "centerline" as "the locus of centers of maximal disks(2D) or balls(3D) contained in the shape". Centerline algorithms are often used during the path planning step of CTC review, mostly offline. Virtual bronchoscopy, a similar technology to CTC, uses CT exam of the thorax to generate multi-branching centerlines along the patients' airway. In colonoscopy, the single tubular anatomical structure of colon determines the extracted centerline as a single curvy without any additional branch. In this specific CT image guided colonoscopy environment, the centerline provides the guidance for online feature and object recognition, as well as curve matching. A robust algorithm should generate centerline that retains the same topology as the original 3D volume, especially around the some complicated territory, such as colonic folds. There have been extensive research activities involved in developing reliable centerline extraction algorithms, which can be generally classified into 3 categories. They are respectively 3D thinning algorithm, level set algorithm and distance transformation algorithm. 3D topological thinning algorithm [24] briefly relies on the "peel onion" technique, continuously peeling off the layers of volume of interest, until the last one is reached. This algorithm achieves its

better accuracy than other two at the cost of expensive computations. Bouix et al. [25] and Ge et al. [26] proposed different optimization methods with certain loss of the accuracy. Hassouna et al. [27] and Deschamps et al. [28] came up with different level-set based on algorithms, which both have fast running time and acceptable quality of the extracted centerline. The method we used in this project is the base on the last category: distance transformation proposed by Wan et al. [29]. This algorithm is based on an accurate *distance from boundary* field, which calculates the Euclidean distance from each voxel of colon lumen to the nearest object boundary. Different distance transformations can be used according to how differently people deal with the neighbor voxels. In 3D case, there are 26 neighbor voxels around each target voxel and their Euclidean distance to the central one vary by the applied metric. Zhou et al. [30] used "1-2-3" metric, Ge et al. [31] used "3-4-5" Chamfer metric and Chen et al. [32] "10-14-17" metric. One obvious downside of this distance transformation based methods is that within the certain tubular winding space, the centerline tends to move closer to the colonic wall, which causes certain level of inaccuracy, as shown in Figure 3.1. The introduction of *distance from boundary* [32] [33] can alleviate the condition.

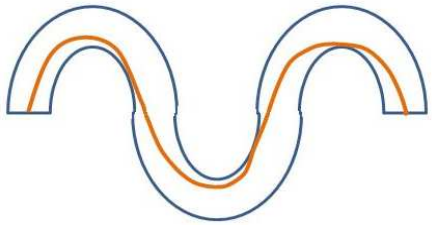


Figure 3.1: Example of centerline "touching" the colonic wall

Wan et al. [29] proposed a three-stage method, firstly converting the volumetric DFB fild into a 3D directed weighted graph, secondly building a minimum cost spanning(MCS) tree from the weighted graph and lastly, extracting the centerlines from MCS tree.

We used the centerline extraction algorithm implemented by Wake Forest University. Figure 3.2 shows the centerline extracted from the colon phantom.

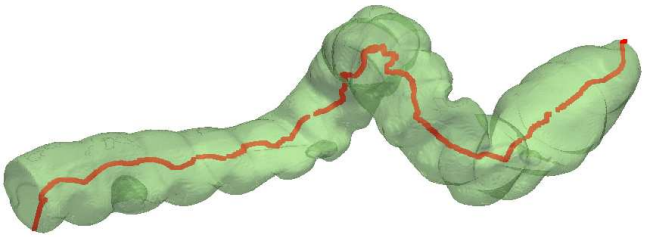


Figure 3.2: Colon Phantom Centerline

3.2 Centerline Matching Method

3.2.1 Rigid Transformation

Since the sensor measurement data is using the coordinate system defined by Aurora device and CT images are acquired in the DICOM coordinate system, an initial rigid transformation between above two coordinates is applied before the centerline matching. Given several feature points $P(c)$ such as known polyps in Aurora tracking coordinate system, the goal is to find the affine transformation parameter A and the translation T such that $P_{dicom}(c) = AP(c) + T$, where A is a $3X3$ rotation matrix, T is a $3X1$ translation matrix and $P_{dicom}(c)$ is the corresponding point in the DICOM coordinate system.

3.2.2 Cubic Spline Interpolation

A total number of eight sensors are evenly placed along the colonoscope shaft. During the real-time operation, the colonoscope traverses inside the colon lumen and the colon is liable to experience a certain amount of deformation. Accordingly, its real-time shape, as well as centerline, appears differently from the initial condition. Eight sensors provide accurate online 3D location information for those eight points on the shaft, and we use cubic spline interpolation to interpolate the other points along the shaft.

Subject to the boundary condition of $n+1$ data points, the spline routine fits n equations over n intervals. The form for the cubic polynomial curve fit for each interval

is:

$$y_i = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \quad i = 1, 2, 3..., n - 1 \tag{3.1}$$

In order to make the curve smooth, the cubic spline constrains the derivatives, ensuring that $y(x)$, $y'(x)$ and $y''(x)$ are equal at the data points for adjacent intervals. That is,

$$y_i(x_i) = y_{i+1}(x_i) \quad y_i'(x_i) = y_{i+1}'(x_i) \quad y_i''(x_i) = y_{i+1}''(x_i) \tag{3.2}$$

After this interpolation, the real-time shape of shaft can be numerically presented by Equation 3.1 for later use of the centerline matching.

3.2.3 Piecewise Centerline Matching

Eventually, this is a registration problem between two curving centerlines. Our approach is based on determining the matching points along the axis of centerline path. The path projections on x, y and z axis are shown in Figure 3.3.

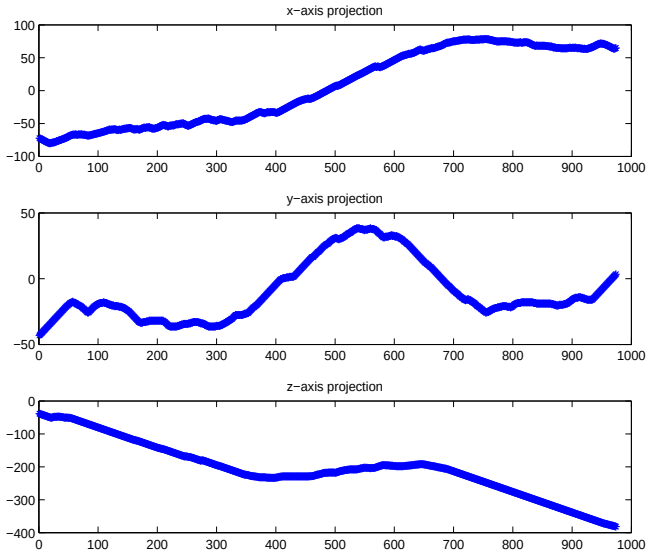


Figure 3.3: Colon centerline projection on x, y and z axis

The matching process fundamentally relies on finding the extrema along each axis. Two extrema constrain a local region to be matched and if the correlation of two

corresponding local regions on two centerlines is above a pre-defined threshold value, the matched extrema are treated as feature points to evaluate the deformation vector. Acar et al. [34] focused on the z axis for matching, however we find in our setting that y axis is more likely to form the extrema in Figure 3.3. From the matched feature points, a difference vector or deformation vector along two centerline pieces is computed as the Euclidean distance between those matched points.

$$V_{difference}(x, y, z) = (P_{matched}(x) - P(x), P_{matched}(y) - P(y), P_{matched}(z) - P(z)) \quad (3.3)$$

where P is the point in virtual colonoscopy and $P_{matched}$ in real-time environment.

These difference vectors can then be applied to evaluate the deformation at the tip of the scope. It can also be extended to other non-matched regions. By doing such at the tip of colonoscope where the camera locates, we are able to correct the real-time deformed camera view in the virtual colonoscopy(VC) environment. Since all the polyp candidates are identified offline ahead in VC, real-time correspondence of colonscope tip in VC environment can tell us how far away it is from those known polyps and plan the later path to reach those points. For the simplicity, the linear interpolation is used based on two difference vectors. Let P_1 be the starting point of a matched centerline piece in VC and P_2 be the ending point, and $V(P_1)$ be P_1's difference vector and $V(P_2)$ P_2's. In the real time environment, any point P_k's difference or deformation vector is linearly calculated as:

$$V_{difference}(P_k) = \frac{V(P_1) * d_{2-k} + V(P_2) * d_{1-k}}{d_{1-k} + d_{2-k}} \quad (3.4)$$

where d_{1-k} is Euclidean distance from P_1 to P_k and d_{2-k} from P_2 to P_k.

During the real-time operation, the tip of the camera is continuously corrected to counter the colon deformation and its corrected VC view, as well as the uncorrected one are shown in the next section.

3.3 Result and Analysis

3.3.1 System Overview

System Hardware

Figure 3.4 is the illustration of CT image guided colonoscopy system hardware. A quad core computer with 4 GB of DDR2 memory and Linux operation system installed serves as the main data processing workstation. The PCI Pinnacle frame grabber is the interface to Olympus Colonoscopy Video System, which receives video stream from the fish-eye lens equipped at the tip of colonoscope. The video capturing rate is 30 frames per sec and the image size is 640 pixels X 480 pixels in bitmap format. We have used the Aurora electromagnetic spatial measurement system (by NDI, Ontario, Canada) to augment a standard colonoscope with an accurate image guided localization device. The Aurora system consists of small sensor coils mounted on the colonoscope and an external magnetic field coil. The sensor coil mounted on the tip of flexible metal cable has a diameter of $1.8mm$ and length $9mm$. When the sensor coil is placed in the controlled magnetic field, its 3D location (x, y and z) as well as its orientation (roll, pitch and yaw) are calculated by measuring the induced voltages in the sensor coil. Up to eight sensor coils can be simultaneously placed along the colonoscope and tracked to acquire real-time scope position. Figure 3.4 shows the Olympus colonoscopy video system and the modified colonoscope with eight sensors used in the experiments. The software is written in C++ and third party libraries include VTK, ITK, KLT, Bayesian particle filter. The implemented program covers multiple data processing: sampling the colonoscope sensors, video sampling, feature tracking, correspondence identification, and SLAM particle filter.

Software Graphical User Interface

The visualization tool in this system is based on 3D Slicer, open source software package for scientific visualization and image analysis. 3D Slicer is coded based on VTK, a graphical library that provides a high-level interface to OpenGL and a pipeline mechanism to the graphical filters. Figure 3.5 gives a demonstration of this viewing tool.

Figure 3.4: (Left) Olympus Colonoscopy Video System. (LowerRight) the colonoscope tip with sensors mounted. (UpperRight) A colon phantom and the external Electromagnetic tracking system.

The list below summarizes some windows within the tool.

1. 3D rendering window(top-right): it displays what a virtual view is in real-time. The camera position depends on the real-time scope position in the magnetic field and view angle on the orientation of 6DOF sensors. During the operation, the Aurora tracking system continuously feed the sensor data through serial cable into workstation under the supervision of Open IGT Link protocol. Once Slicer 3D captures the coming packets, it extracts the sensors' position and orientation and automatically updates the view in this window.

2. 2D projection window(bottom-right): axial, sagittal and coronal projection of CT image provides a global view for interacting with the predefined central axes. If a user wants to track the position of colonoscope in these gray-scale data, this window can be updated in real-time and the position of colonoscope is displayed in these three 2D projection sub-windows. Meanwhile, 2D slices can also be displayed within the 3D rendering window along with a virtual view to provide a good context to interpret the gray-scale data.

3. Real camera window(top-left): it displays the online video captured from a frame

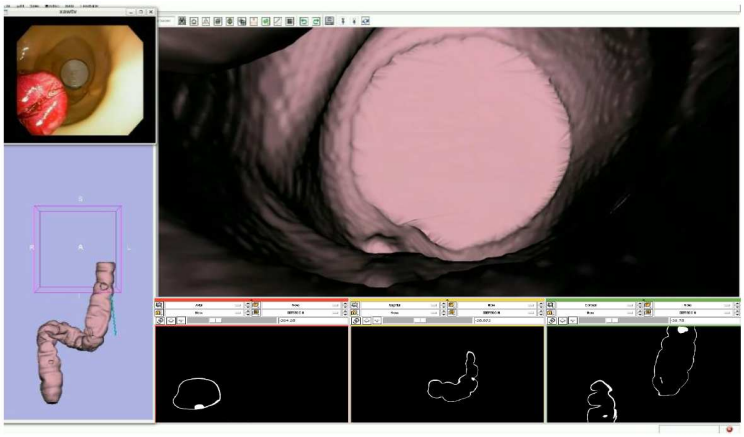


Figure 3.5: Software GUI

grabber.

4. Global position window(bottom-left): this view helps the surgeon to know where exactly the real-time colonoscope is in virtual colonoscopy environment, as well as its orientation. The camera view can be adjusted for the different angles.

3.3.2 System Operation

Given the hardware set described above, we first setup Aurora tracking system and connected colonoscope video stream to the frame grabber in the workstation. 3D Slicer has been loaded with segmented colon model, as well as the corresponding 2D CT images. We then inserted the colonoscope all the way to the other end of colon phantom as shown in the upper-right part of Figure 3.4. Figure 3.5 gives a good view before providing live colonoscopic guidance. Before proceeding, the initial transformation between CT DICOM coordinates and Aurora tracking coordinates was calculated. Then we carefully picked multiple fiducial points on polyps from CT images, record their 3D coordinates, moved the sensors to the corresponding locations inside of physical colon phantom and recorded their sensor measured coordinates. Afterward, we used open source "Coherent Point Drift" point-set registration program to get the initial rigid transformation parameters: rotation matrix A and the translation T.

At the beginning of the operation, the eight sensors covered the entire colon phantom.

We did the first time sensor data acquisition, applied the transformation and then cubic spline interpolation on them. The interpolated centerline is shown in Figure 3.6. As a reference, centerline calculated from CT images is also included.

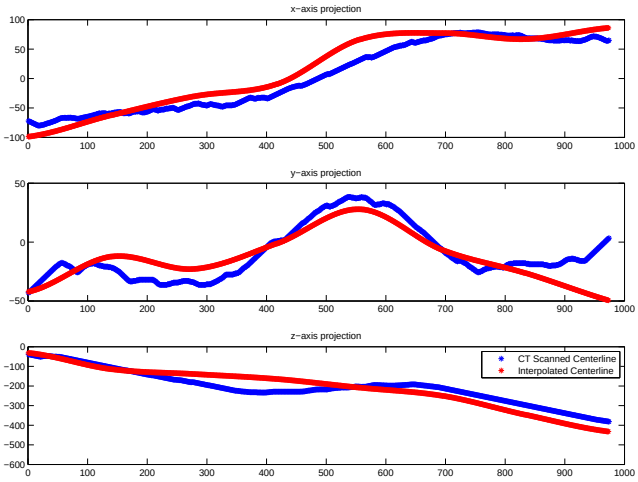


Figure 3.6: Interpolated(Red) and CT scanned(Blue) centerline projection on x, y and z axis

In Figure 3.6, the projections of x, y and z of interpolated centerline C_i have a similar shape as one calculated from CT images C_t, when the colonoscope covers the entire colon phantom. The morphological similarity between C_i and C_t, to some extent, can be explained in terms of the similarity between their local extrema and the corresponding values. In other word, extrema of C_i and C_t with close 3D coordinates are more likely to correspond to the same anatomical points along the centerline. At this moment, the extrema based centerline algorithm has the solid matching result. We then calculated the deformation vector as in Equation 3.3 and corrected the camera tip placement in a virtual view environment as in Equation 3.4. The corrected view is shown in Figure 3.7

However, as the colonoscope controlled by technician traverses all the way back toward anus, the knowledge of real-time centerline decreases since we are only be able to capture part of its overall shape and some extrema disappear. Since centerline matching algorithm requires at least two extrema to operate on, some pathway could fail to

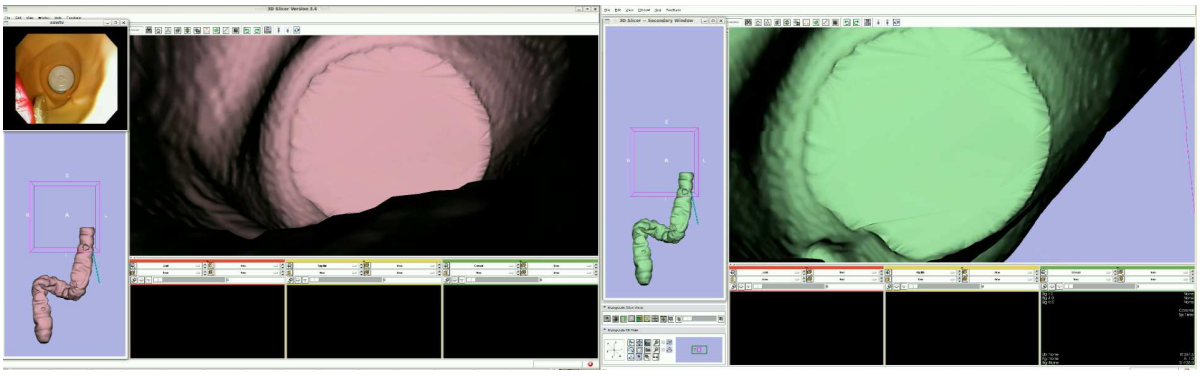


Figure 3.7: Left is the uncorrected virtual view and Right is the corrected one

provide the deformation estimation and correction. Figure 3.8 shows x, y and z axis data projection at the failure moment of the algorithm.

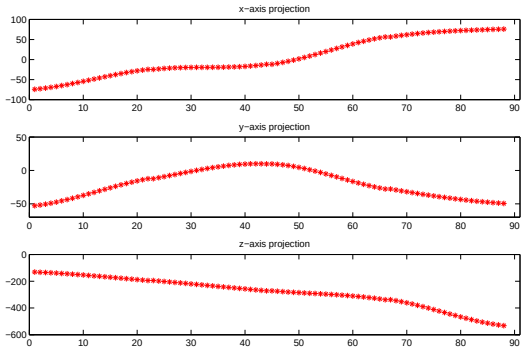


Figure 3.8: Interpolated centerline projection on x, y and z axis on failing the matching

We have conducted the test three times and calculate the length along the centerline from failure point to the end and its ratio to the entire centerline length.

$$Ratio = \frac{Length\ from\ failure\ point\ to\ ending : point(FL)}{whole\ centerline\ length(WL)} \tag{3.5}$$

TestID	FL	WL	Ratio
1	429.15mm	521.28mm	82.33%
2	443.91mm	535.21mm	82.94%
3	439.21mm	554.91mm	79.15%

A major cause of matching failure is that the colon phantom used in the test is not close to the real human colon shape, and its morphological structure can't generate enough number of extrema points. The x, y, z projections of real human colon contain at least 7 or 8 extrema points [34] and losing some of them can not completely tear down the algorithm. We are planning to do additional tests on human colon like phantom. Meanwhile, improving the robustness of centerline matching is also a key to success of the whole system. Frechet Distance based partial curve matching algorithm is a good candidate of making use the information of all the points along the curve, rather than only extrema points.

Chapter 4

Simultaneous Localization and Mapping in Image Guided Colonoscopy

4.1 Fundamental of Simultaneous Localization and Mapping

The problem of Simultaneous Localization and Mapping(SLAM) has attracted an enormous amount of attention in robotics. SLAM addresses the problem of building a map of unknown environment from a sequence of landmark measurements obtained from a moving robot. A robot can be equipped with laser range finders, sonar sensors or cameras to make those measurements. The local environment might be dynamic with people and objects moving around. The core of SLAM is about building a model leading to a new map and repetitively improving an existing map while at the same time localizing the robot within that map [35].

4.2 SLAM in CT Imaged Guided Colonography

Consider a colonoscope traversing the inside of colon, with the camera on its tip taking the observations of many features. CT Image guided colonoscopy is an online SLAM problem. The fundamental difference between SLAM in IGC and traditional SLAM are:

1. The traditional SLAM problem makes the assumption that the environment that robotics are exploring is stationary and consistent. However, during colonoscopy, the colon is always under a certain level of deformation.
2. There are uncertainties about the measurement of location and orientation of the vehicles in the traditional robotic SLAM problem. On the other end, with the presence of global position information, the pose of colonoscope can be accurately measured by the electromagnetic field tracker.

In the case of image guided colonoscopy, the a-priori map is the lumen model segmented from the CT colonography images. The pose of the robot is the position and orientation of the colonoscope camera. The sensor measurement includes the position and orientation of the sensors along the colonoscope shaft. 3D feature observations are identified from the video stream. The largest class of algorithms is based on probability using Markov constraints and is most applicable to the CT image guided colonoscopy problem [35].

At a time-stamp t, the following variables are defined:

- $x(t)$: The current estimate of virtual camera pose (position and orientation)
- $m(t)$: The map of the lumen, which can be represented as an occupancy grid or landmark based map. $m(0)$ is the segmented lumen from CTC data.
- $c(t)$: The vector describing the location of captured features on current map m.
- $U(1:t) = u_1, u_2, \ldots, u_t$: The history of colonoscope position and orientation measured from sensors. And it also represents the control unit of the colonoscope motion.

- $Z(1:t) = z_1, z_2, \ldots, z_t$: The set of all landmark observations, which are estimated from vision system. It represents the present and past landmark positions.

The variables x, m and c represent the unknowns to be estimated at each time step. While Z and U are known parameters throughout the whole process. The task aims to estimate x, m and c given Z and U from the joint posterior probability density $p(x, m, c|U, Z)$. This is called the online SLAM problem because we only estimate the virtual camera pose and map at current time-stamp, rather than the entire period.

Probabilistically, SLAM in IGC is requires that the probability distribution

$$P(x_t, m_t, c_t|U_{1:t}, Z_{1:t}, m_0) \tag{4.1}$$

to be computed for all the time-stamps t throughout the entire operation. In general, a recursive solution to SLAM is desirable. Starting with an estimate for the distribution $p(x_{t-1}, m_{t-1}, c_{t-1}|U_{1:t-1}, Z_{1:t-1})$ at time $(t-1)$, the joint posterior can be computed using Bayes theorem with a control u_t and observation z_t. This computation requires that a state transition model and an observation model to be defined to describe the effect of the control input and observation respectively.

- The **observation model** describes the probability of making an observation z when the virtual camera pose, current map and feature locations in the map are known and is generally in the form of

  $$P(z_t|x_t, m_t, c_t) \tag{4.2}$$

  The assumption here is that once x_t, m_t and c_t are known, the observation is conditionally independent given the virtual camera pose, current map and feature locations.

- The **state transition model** can be described in terms of a probability distribution of the virtual camera state transition:

  $$P(x_t|x_{t-1}, u_t) \tag{4.3}$$

  The state transition is a Markov process in which the next state x_t depends only on the immediately preceding state x_{t-1} and control unit u_t and is independent of the observation z_t, map m_t and feature locations c_t.

Probabilistic SLAM represents *belief* through conditional probability distributions. *Belief* distribution is a joint posterior conditioned on the available state variables data.

$$belief(x_t, m_t, c_t) = P(x_t, m_t, c_t | U_{1:t}, Z_{1:t}) \tag{4.4}$$

Belief is evaluated after incorporating the latest landmark observation z_t. And its form before incorporating z_t and right after executing the control u_t can be denoted as:

$$\overline{belief}(x_t, m_t, c_t) = P(x_t, m_t, c_t | U_{1:t}, Z_{1:t-1}) \tag{4.5}$$

The SLAM in IGC can now be formulated in the standard two-step recursive predication and correction process [35]. The pseudo-algorithmic form is provided as below:

1. **Bayes filter algorithm**
2. for all time-stamp t do
3. $\quad \overline{belief}(x_t, m_t, c_t) = \int P(x_t | x_{t-1}, u_t) belief(x_{t-1}, m_{t-1}, c_{t-1}) dx_{t-1}$
4. $\quad belief(x_t, m_t, c_t) = \eta P(z_t | x_t, m_t, c_t) \overline{belief}(x_t, m_t, c_t)$
5. endfor
6. return $belief(x_t, m_t, c_t)$

where η is the normalization constant.

In practice, computing the full posterior is unfeasible given the high dimensionality of the parameter space involved, thus an approximation is required and forms the basis of most practical online SLAM algorithms. The Extended Kalman Filter (EKF) has been widely used in the early days; however its key limitation is the computational complexity. Especially in IGC application, hundred thousand surface points or voxels need to be processed, leading to a complexity that is quadratic in terms of that large number of features. On the other hand, particle filters provide an alternative way to estimate Bayesian model. FastSLAM [36], with its basis in recursive Monte Carlo sampling, can represent a nonlinear process model. Furthermore, this algorithm can reduce the sample space by apply Rao-Blackwellization, thus provide efficiently

computational feasibility. Its complexity is logarithmic in the number of features, an attractive property for our IGC application. Other advantages include being more robust to correspondence error and that it solves both the full and online SLAM problem simultaneously. A potential solution to IGC problem is to use FastSLAM algorithm to simultaneously estimate the pose of the virtual camera and the deformation of the prior CTC lumen surface, so that the features in the colonoscopy video identified by video tracking are aligned with the corresponding ones on the deformed CTC lumen. Features and correspondences could be used identified using Kanade-Lucas-Tomasi feature tracking. Since SLAM estimates confidence distribution for the pose state, if the video frame is obscured for certain period of time, it is possible to reinitialize the tracking process as error grows large.

First of all, we did a particle filter based simulation in 2D environment to evaluate its validity and performance.

4.3 2D Demonstration

The traditional SLAM problem is focused on building a stationary map of unknown environment from a sequence of landmark measurements obtained from a moving robot. The uncertainties come from robot control system and landmark measurement system. In the setting of IGC, the colon undergoes stretching and shrinking, thus the lumen environment is not stationary at all. Meanwhile, the pose of colonoscope can be accurately measured by Electromagnetic field tracker and a-priori map is known ahead after the lumen model is segmented from the CT Colonography images. Given the above setting, we want to estimate the deformation that occurs throughout the operation. The following simulation localizes a moving robot inside of a known environment, which has two walls perpendicular to each other (Figure 4.1). The robot is guided to move in the direction that is 45 degree to each wall. This robot is equipped with a laser range sensor, which is able to calculate its absolute distance to both walls. And during the movement, two walls stretch and shrink at pre-defined time-stamp in order to simulate the deforming colon walls during colonoscopy. We applied the SLAM's observation model and state transition model to compare robot's real and believed trajectory. This difference leads to an estimation of the deformation of the colonic

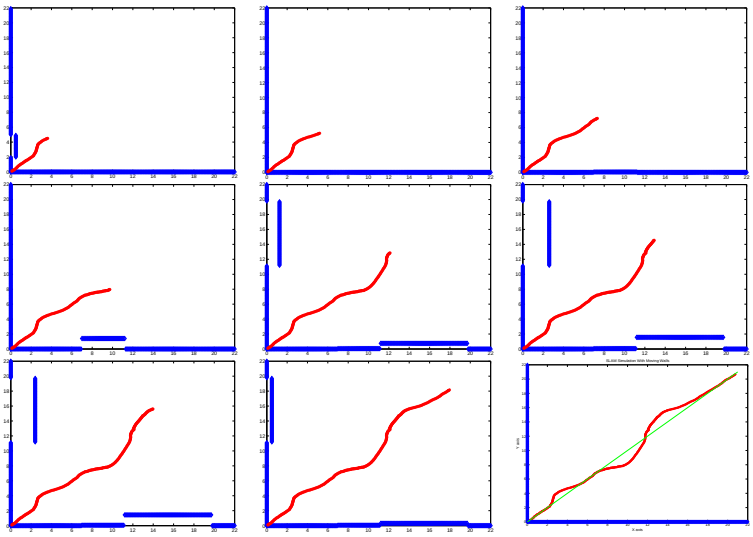


Figure 4.1: SLAM Simulation with Deformed Walls

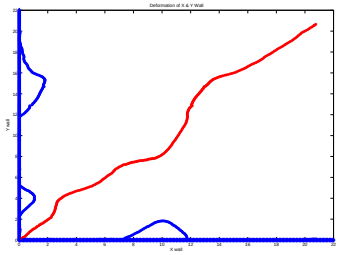
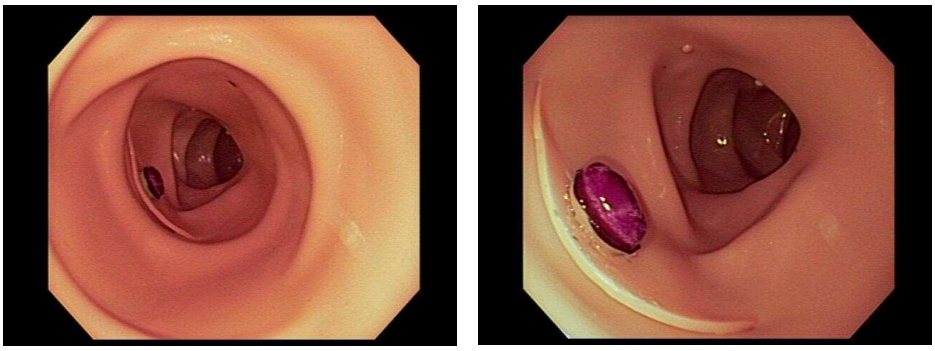


Figure 4.2: Estimation of the Deformation of the Wall

wall.

In Figure 4.1, the actual robot trajectory is depicted as a diagonal green line; however its belief is a red line due to the deformation of the wall. The robot does not know its relative location inside the map and the only information it can rely on to locate itself is the measurement of distance to each wall. Since wall at some time-stamps shrinks to the left, distance becomes less and robot thinks it is moving closer to the left wall. By calculating robot's deviation from the planned path, it can estimate the deformation of the wall as shown in Figure 4.2.



(a) (b)

Figure 4.3: Illustration of fish-eye lens. (a) Image taken inside of colon phantom (b) Close look of polyp sample

4.4 Fish-Eye Camera and the Calibration in SLAM Vision System

4.4.1 Fundamental of Fish-eye Lens

Fish-eye lens use a wide angle lens to acquire wide, hemispherical image. The fundamental difference between a fish-eye lens and conventional lens is that projection from 3D ray to a 2D image position in the fish-eye lens is intrinsically non perspective [37]. Its focal length is usually shorter than conventional camera lens. The disadvantages of this type of lens are (1) the image resolution is non-uniform and (2) radial and tangential distortions are severe in the marginal region. Despite the obvious downside, fish-eye lens are the favorite in endoscopic medical devices due to their wide field of view. Figure 4.3 illustrates the appearance of fish-eye lens distortion.

4.4.2 Camera Calibration

Calibration is a process of estimating the intrinsic parameters of a camera system, such as focal length, skew coefficients and distortion coefficients. Since we eventually acquire scene information from pixels in 2D images and there is transformation between pixel

coordinates and world coordinates, camera calibration is an indispensable step in 3D point reconstruction from the images.

Background

According to the prior knowledge of imaging scene, calibration techniques can be categorized as follows [38]:

- Calibration with exact prior knowledge. The imaging objects' 3D spacial knowledge is completely known before calibration, such as coordinates and orientations. This knowledge can improve the accuracy of calibration results to some extent. A common imaging object in this category is the checkerboard.
- Calibration with partial or no prior knowledge. In case of there is no calibration object or only partial spatial knowledge is known, the estimation of the parameters depend on feature correspondences in the image. The partial knowledge can be used to as constraint during the estimation. In most cases, this type of calibration generates poor results compared to the former one due to the noise in both the image and camera displacement.

Camera Projection Model

The following describes the camera model used and the calibration procedure we currently employed to reconstruct points on the lumen surface. The camera shown in Figure 4.4 has three transformations and one projection involved.

1. The transformation from world coordinates to camera coordinates is F_1. A point P_w in world coordinates can be mapped to the point P_c in camera coordinates by using rotation matrix R and translation matrix T. And transformation is defined as:
$$P_c = RP_w + T \tag{4.6}$$
2. The projection of point $P_c(x_c, y_c, z_c)$ in camera coordinates to point $C'_p(x'_i, y'_i)$ in the image plane is F_2.
$$x'_i = f\frac{x_c}{z_c} \quad y'_i = f\frac{y_c}{z_c} \tag{4.7}$$

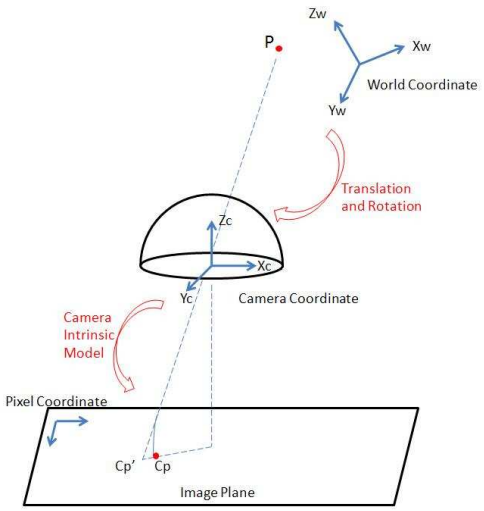


Figure 4.4: Camera projection model

where f is the focal length of fish-eye lens

3. The Transformation from the non-distorted image point $C'_p(x'_i, y'_i)$ to the distorted one $C_p(x_i, y_i)$ is F_3 due to lens radial and tangential distortion.

$$C_p = (1 + kc(1)r^2 + kc(2)r^4 + kc(5)r^6)C'_p + dx \tag{4.8}$$

where kc is a $1x5$ vector containing both radial and tangential distortion coefficients and dx is the tangential distortion vector with equation

$$dx = \begin{bmatrix} 2kc(3)x'_i y'_i + kc(4)(r^2 + 2x'^2_i) \\ 2kc(4)x'_i y'_i + kc(3)(r^2 + 2y'^2_i) \end{bmatrix} \qquad r^2 = x'^2_i + y'^2_i \tag{4.9}$$

4. The transformation from point $C_p(x_i, y_i)$ image plane coordinates to pixel coordinates $C_{pixel}(u, v)$ is F_4. Assume that the image center is located at pixel $C(u_0, v_0)$, scaling factors along u and v axes are k_u and k_v and skew parameter s.

$$C_{pixel} = KC_p + C, \quad K = \begin{pmatrix} k_u & s \\ 0 & k_v \end{pmatrix} \tag{4.10}$$

5. Give any point P_w in the world coordinates, its pixel representation in the image plane is

$$C_{pixel} = F4 \circ F3 \circ F2 \circ F1(P_w) \tag{4.11}$$

The whole calibration procedures measure the following parameters:

- Camera intrinsic parameters: u_0,v_0,k_u,k_v,f,s and kc
- Camera extrinsic parameters: rotation matrix R and translation matrix T

Calibration Procedures

We used a checkerboard Figure 4.5 as a known calibration object. Within the world coordinate system, this checkerboard lies in the plane of $Z = 0$ and two perpendicular sidelines along $X = 0$ and $Y = 0$ two axes respectively. Due to the fact that dimension of this checkerboard is pre-defined, it is easy to know the coordinates of any point on the board.

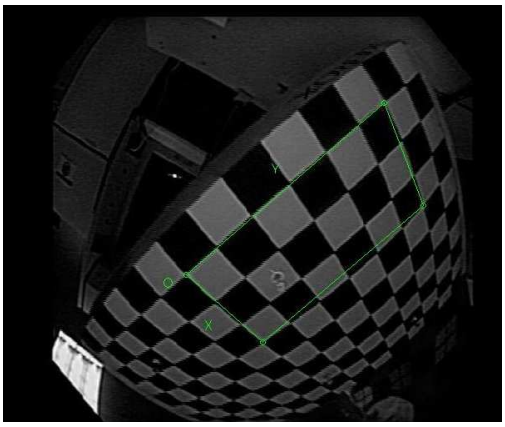


Figure 4.5: A image taken by fish-eye lens camera in colon phantom

The cost function of is the sum of square of the distance between known image feature locations and those calculated ones. In this case, the feature points are the intersections of white and black grids.

Given a set of m images, each of which has n feature points, the calibration is to calculate the set of camera parameters $S = \{u_0, v_0, k_u, k_v, f, s, kc\}$ with the cost function formulated as:

$$\underset{S}{\operatorname{argmin}} \sum_{i=1...m} \sum_{j=1...j} \|P_{i,j} - P'_{i,j}\| \tag{4.12}$$

where $P_{i,j}$ is the feature point with known 2D pixel coordinates and $P'_{i,j}$ with calculated corresponding position by Equation (4.11).

The calibration procedures can be summarized as:

1. Print out a planar checkerboard pattern with predefined parameters
2. Take several pictures of it from different view angles by the camera located at the tip of the colonoscope.
3. Extract the grid corner points from the images and manually point out the origin in world coordinate system.
4. Given the initial guess of intrinsic parameters and iteratively reduce the cost function by Levenberg-Marquardt algorithm. [39]

The calibration parameters are estimated only once for the given fish-eye lens. The camera calibration result is calculated based on the calibration MATLAB program provided by Compute Vision Group at Caltech and the followings are acquired camera model parameters.

Focal Length	[321.48 325.51] +/− [2.14 2.22]
Principal Point	[324.57 241.11] +/− [2.21 2.36]
Distortion	[-0.3911 0.11599 -0.00017 -0.00246 0.00004]
Pixel Error	[0.30599 0.25822]

Figure 4.6 shows that during the calibration procedure, the checkerboard pictures are first taken from six different view angles. Or relatively a fixed camera captures the images of checkerboard lying in six different planes.

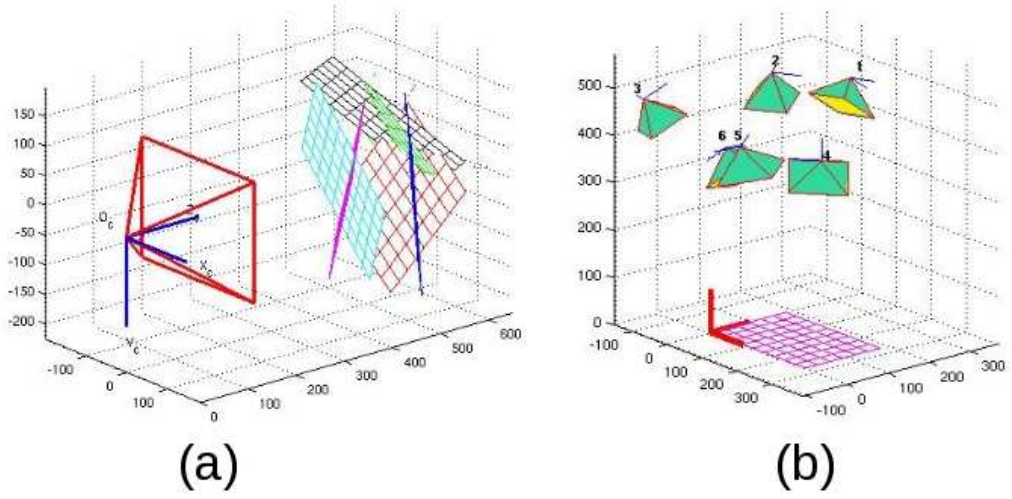


Figure 4.6: (a) Fixed checkerboard against moving camera (b) Fixed checkerboard against moving camera

4.5 Image Processing and Tracking in SLAM Vision System

Within the framework of CT imaged guided colonoscopy, the vision system plays an essentially important role. Later 3D scene reconstruction and estimation of the real-time shape of the deformed colon all depend on the quality of how vision system is processed online. Four sub-sections will be discussed sequentially in the order of feature extraction, feature tracking, outlier rejection and 3D scene reconstruction.

4.5.1 Feature Extraction

The major goal in detecting the feature is to reduce the information of the entire image into a discrete set of representative regions, such as special points, edges and other apparent patterns. Rather than inspecting the entire image to identify motion, it is easy to focus only on the parts of the image which are of interest. Due to the fact that this real time feature extraction step not only should be computationally inexpensive, but also easy to track to detect the their motion, we decide to select point wise features. The drawback of this type feature is it is vulnerable to noise inside of

imaging environment, such as abrupt change of illumination, low resolution around image marginal region. The other type of line and plane features can also be used but these will be the future work for feature extraction performance evaluation. The method adopted here is based on "good features to track algorithm" by J. Shi[40].

Within a image I and tuned feature window size, matrix Z is defined as:

$$Z = \begin{bmatrix} \sum P_u P_u & \sum P_u P_v \\ \sum P_v P_u & \sum P_v P_v \end{bmatrix} \tag{4.13}$$

where P is a pixel in the image, P_u and P_v is the respective image intensity derivative along u and v direction. Inside summation is assigned within the feature window.

If the two eigenvalues of Z are λ_1 and λ_2, this feature window can be accepted as a feature to track if

$$min(\lambda_1, \lambda_2) > \lambda \tag{4.14}$$

where λ is a predefined threshold value.

4.5.2 Feature tracking

Two popular feature extractions nowadays are Kanade-Lucas-Tomasi Feature Tracker (KLT) [41] and Scale Invariant Feature Transform (SIFT) [42]. The KLT algorithm for point feature tracking is widely used because it is fast, performs quite well in many situations and mathematical model behind it is quite simple. However, non-rigid objects sometimes fail to be tracked. On the end hand, SIFT has more robust performance that KLT, it is computationally expensive especially performing the tracking among many frames. Due to the fact that most point-wise movement in the scene is rotation and translation and we need tracking process not consuming too much computing resource in the whole system, it is reasonable to use KLT for the concern of efficiency.

The way how KLT tracker works is quite straightforward. Given two images I taken at time t and J at time $t + \tau$, and a pixel $c(u, v)$, feature tracking finds the affine transformation parameter A and the translation T such that $J(c) = I(Ac + T)$, where A is a $2X2$ coefficient matrix and T is a $2X1$ vector. Feature tracking is to minimize

the matching residual defined as

$$R = \sum_{c \epsilon w} [I(Ac + T) - J(c)]^2 \tag{4.15}$$

where w is the feature window size. By using Newton-Raphson method, it can easily find out the transformation. The quality of this estimation depends on the size of feature window, the texture of the image within it and the amount of camera motion between frames. According to J.Shi [40], the small window size can not contain enough variation of the motion within it for the successful track and less reliable. On the other end, big window is more like to miss depth discontinuity.

Other robust feature tracking algorithm SIFT [42] focuses on finding features that are invariant to image translation, rotation, scaling, even partially to illumination change and local geometric distortion. Features are detected through a staged filtering that identifies stable point in the scale space.

4.5.3 Feature Outlier Rejection

In the vision system of colonoscopy, due to the change of illumination, occlusion, fluid specular reflection and lens distortion, there exists a large number of poorly tracked features. The outlier rejection is based on the robust estimation of matching residual model. Traditional statistical estimators have breakdown points that are no more than 50%. Other relatively recent ones such as RESC, MINPRAN, etc claim to have tolerance of more than 50% outliers, but require users to tune many parameters or behave less effective under multiple distribution structures.

In this part, I adopted the Two-Step Scale estimator proposed by Wang [43]. The algorithm is described as follows:

1. Given samples drawn from unknown distribution. The multivariate kernel density estimator with kernel K and window radius h is defined as

$$f(x) = \frac{1}{nh^d} \sum_{i=1...n} K(\frac{x - X_i}{h}) \tag{4.16}$$

   for a set of n feature data points $X_i(i = 1\dot{n}$ in d-dimensional Euclidean space.

Epanechnikov kernel is used here to yield the minimum mean integrated square error:

$$K_e(x) = \begin{cases} \frac{1}{2}c_d(d+2)(1-x^Tx) & if\ x^Tx < 1 \\ 0 & otherwise \end{cases} \tag{4.17}$$

where c_d is the volume of the unit d-dimensional sphere, e.g. $c_1 = 2$, $c_2 = \pi$, $c_3 = \frac{4\pi}{3}$.

2. Starting from any sample point, find local density maximum and minimum by mean shift vector:

$$M_h(x) = \frac{1}{n_x} \sum_{x_i \epsilon S_h(x)} [X_i - x] \tag{4.18}$$

where $S_h(x)$ is a hyper sphere of radius h. Since the mean shift vector always points towards the direction of maximum increase in the density, it can locate the peak of density distribution. Let $\{y_k\}\, k = 1, 2$ denote the sequence of successive locations of the mean shift procedure.

$$y_{k+1} = y_k + pM_h(y_k) \quad 0 < p < 1 \tag{4.19}$$

Similarly, we can define the mean shift valley vector in the opposite direction of peak, $MV_h(x) = -M_h(x)$, in this way, it can easily locate a valley or local minimum as well.

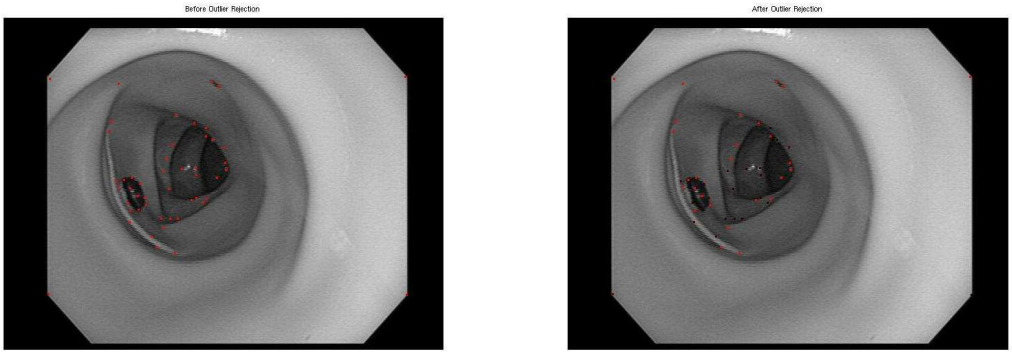
3. Estimate the scale of the fit by median scale estimator using the feature matching residual points that are less than above computed valley value.

$$M = 1.4826(1 + \frac{5}{n-p})\sqrt{medR_i^2} \tag{4.20}$$

where n is the number of sample points and p is the dimension of the parameter space.

4. If feature i with matching residual R_i is larger than M, simple get it rejected.

Figures 4.7 show the extracted features before and after outlier rejection.



(a) (b)

Figure 4.7: Illustration of the outlier rejection. (a) Extracted features before outlier rejection (b) Inliers in red color and outliers in dark color

4.5.4 3D Scene Reconstruction

To reconstruct lumen points in the world coordinate system; at least two images are required. We used an N-ocular stereo setup Figure 4.8 where two sequential images and the associated camera motion are measured. Camera movement is required before taking the next image. Given a pixel in the first image, we find its correspondence in the second image and use ocular stereo (stereo from motion) to reconstruct the point in 3D. These tracked points are a sparse subset of the image pixels and may be obtained from a point-wise feature tracking algorithm, KLT [41] in this case. These reconstructed 3D points can be used to decide the relative location of colonoscope inside the colon lumen.

Figure 4.8 demonstrates the setup of two cameras at two different positions capturing the images of the same object. Given a point $C_{pixel_1}(u_1, v_1)$ in the first image and its correspondence $C_{pixel_2}(u_2, v_2)$ in the second one from KLT feature tracking, the reconstruction of the same point in 3D world coordinate system is an inverse mapping process. The major steps are listed as follows:

1. Transform the points C_{pixel_1} and C_{pixel_2} in the pixel coordinates to C_{p_1} C_{p_2} in

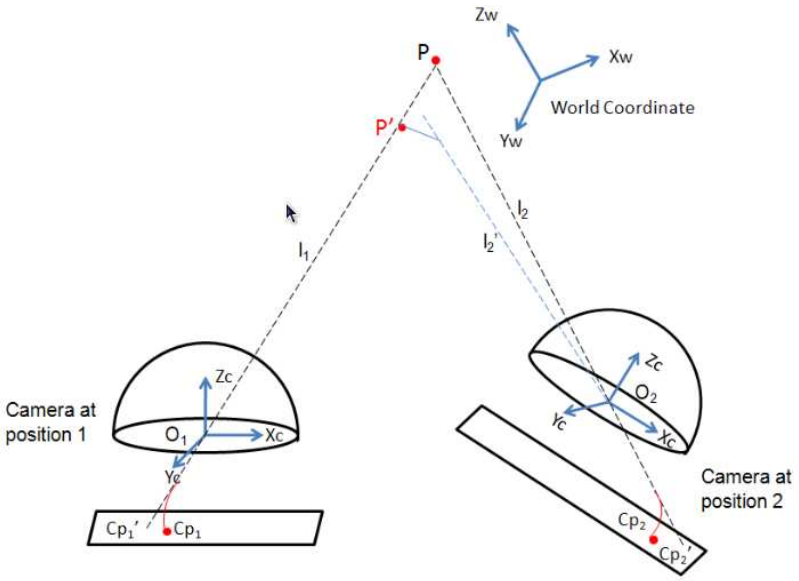


Figure 4.8: N-ocular stereo setup

image plane coordinates.

$$C_p = K^{-1}(C_{pixel} - C) \tag{4.21}$$

where K is a 2X2 matrix with camera intrinsic parameters defined in last chapter and $C(u_0, v_0)$ is the image principal point.

2. Recover the fish-lens' distortion effect by mapping distorted points C_{p_1} and C_{p_2} to the non-distorted ones C'_{p_1} and C'_{p_2}. However, due to the high degree distortion model mentioned in calibration section, there is no general algebraic expression for this inverting recovery step. We use the method of giving initial guess, iteratively updating the result until the pre-defined cost function reaches the threshold value.

3. With sensor-measured camera positions and orientations O_1, O_2 at those two time-stamps, calculated the corresponding points C'_{p_1} and C'_{p_2} can be mathematically mapped back into 3D space point P by finding the intersection of two co-planar lines $\overrightarrow{C'_{p_1}O_1}$ and $\overrightarrow{C'_{p_2}O_2}$ in noise free environment. However, in the vision system, noise is inevitable due to error from feature tracking, camera configuration or occasional interference to EM field during the position measurement. In

such case, lines $\overrightarrow{C'_{p_1}O_1}$ and $\overrightarrow{C'_{p_2}O_2}$ are likely to lie n two different planes and there is no single common point. To solve this issue, we use the existing method of finding the shortest line segment(solid blue line in above figure) that joins both $\overrightarrow{C'_{p_1}O_1}$ and $\overrightarrow{C'_{p_2}O_2}$ and define the intersection P' as that of $\overrightarrow{C'_{p_1}O_1}$ and added line segment as show in Figure 4.8.

$$P' = O_1 + a\frac{(c \times b) \cdot (a \times b)}{|a \times b|^2} \tag{4.22}$$

where $a = C_{p_1} - O_1, b = C_{p_2} - O_2$ and $c = O_1 - O_2$

Figure 4.9 shows the result of reconstruction of corner points(blue) in the checkerboard, as a comparison to the actual ones(red).

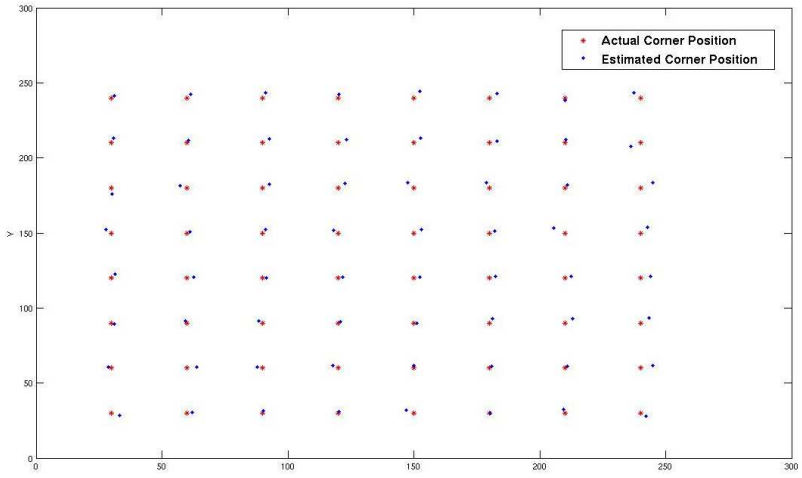


Figure 4.9: Reconstructed checkerboard corner point

Figure 4.10 shows the extracted feature from colonoscopy video stream and its corresponding reconstructed 3D points in virtual view. Due to the certain vision system errors discussed above, the reconstructed polyp location is slightly away from the actual one (Figure 4.10(c)).

Sensitivity Analysis

The system of 3D point reconstruction involves many variables such as camera extrinsic and intrinsic parameters as well as feature tracking algorithm. These variables play

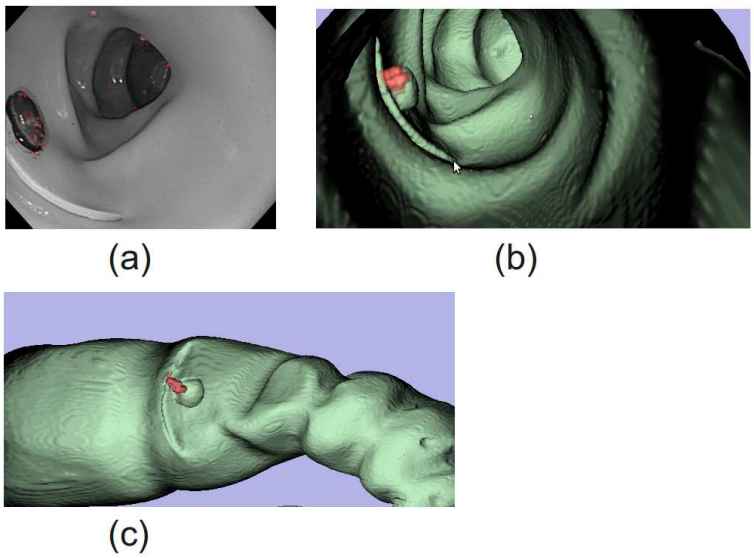


Figure 4.10: Illustration extracted features from video stream and their 3D reconstruction. (a) Extracted features (b) Reconstructed 3D feature points on polyp in virtual view (c) Another view angle of the scene

an important role in affecting the performance of point reconstruction. The overall equation can be formulated as:

$$P_w = F(C_{pixel} : p_1, p_2, p_3, \dots) \tag{4.23}$$

where P_w is the point in world coordinate system, C_{pixel} is the corresponding feature pixel in the image and $p_1, p_2, p_3, \dots$ are system variables discussed above.

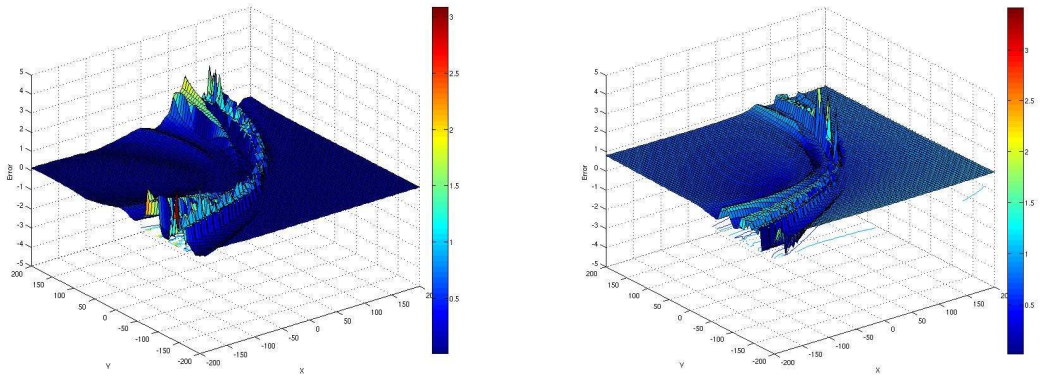
The sensitivity within Equation 4.23 with respect to the change of any parameter p_i can be formulated continuously as:

$$\triangle P_w = \frac{\partial F}{\partial p_i} \triangle p_i \tag{4.24}$$

or discretely as:

$$\triangle P_w = F(C_{pixel} : p_1, p_2, \dots, p_i + \triangle p_i, p_{i+1}, \dots) - F(C_{pixel} : p_1, p_2, \dots, p_i, p_{i+1}, \dots) \tag{4.25}$$

In this section,we evaluate the sensitivity with respect to only camera extrinsic parameters: translation T and rotation R. During the evaluation, only one parameter is



(a) (b)

Figure 4.11: Reconstruction error distribution (a)w.r.t camera translation (b)w.r.t camera rotation

changed at a time in order to reflect the influence of this parameter to the whole system. Multiple parameters change can also be a good choice for further system performance evaluation.

1. Camera Translation The translation error is $\triangle T = [\triangle T_x, \triangle T_y, \triangle T_z]$. Figure 4.11(a) shows the error distribution of all the points within the same plane $Z = 0$. Specifically, camera translation is $T = [30, 30, 0]$mm, and translation error $\triangle T = [5, 5, 0]$mm. Error is formulated as $\frac{|\triangle P_w|}{|P_w|}$. It is easy to tell from Figure 4.11(a) that the error is relatively high in the direction of translation.

2. Camera Rotation The rotation error in Euler angle representation is $\triangle \Theta = [\triangle \alpha, \triangle \beta, \triangle \gamma]$. The parameter setting is camera translation is $T = [30, 30, 0]$mm, rotation $\theta = [\frac{\pi}{6}, \frac{\pi}{6}, \frac{\pi}{6}]$ and $\triangle \theta = [\frac{\pi}{18}, \frac{\pi}{18}, 0]$. Figure 4.11(b) shows that error increases as points are closer to the camera translation direction.

Chapter 5

Future Work and Conclusion

In this project, we implemented an automatic polyp detection algorithm and designed an image guided colonoscopy system, which combines the information from CT colonography images, sensors providing colonoscope position measurements and the live colonoscopy video stream, to guide the removal of polyps. We used electromagnetic field to measure the real-time sensor position to interpolate the online shape of colonoscope, then did the centerline matching for data registration between CT colonography and optical colonoscopy in order to estimate and validate the target position of pre-defined polyps. Camera calibration, image feature tracking and 3D scene reconstruction have been researched to investigate the feasibility of applying Simultaneous Localization and Mapping to this application.

The centerline based matching algorithm relies on finding extrema during colonoscope's traversal throughout the colon. Due to the fact that colon is stretching and shrinking during the operation, sometimes those extrema do not always exist throughout the operation, which reduces the robustness of the algorithm. We are currently developing Frechet Distanced based partial curve matching algorithm in order to make use of the information of all the points along the centerline. Furthermore, SLAM could help locating the relative position between the colonoscope and polyps in a virtual colonoscopy view. 2D simulation has already shown the promising result, estimating the environment deformation in a numerical way and we will move the framework to 3D environment with the vision system incorporated for the next milestone. Specifi-

cally, Monte Carlo sampling based FastSLAM algorithm will be used to simultaneously estimate the pose of the virtual camera and the deformation of the prior CTC lumen surface, so that the features identified by the video tracking in colonoscopy video stream can be aligned with the corresponding ones in CT images.

Publications

Conference Proceedings

Y. Shen and C. L. Wyatt, "Open Implementation of Feature Extraction Methods for Computer Aided Polyp Detection with Principal Component Analysis". Proceedings of the eleventh MICCAI workshop on Computational and Visualization Challenges in the New Era of Virtual Colonoscopy, Sep 2008, New York, NY.

Bibliography

[1] M. J. O'Brien, S. J. Winawer, A. G. Zauber, L. S. Gottlieb, S. S. Sternberg, B. Diaz, G. R. Dickersin, S. Ewing, S. Geller, and D. Kasimian, "The National Polyp Study. Patient and polyp characteristics associated with high-grade dysplasia in colorectal adenomas," *Gastroenterology*, vol. 98, pp. 371–379, Feb 1990.

[2] C. D. Johnson and A. H. Dachman, "CT colonography: the next colon screening examination?," *Radiology*, vol. 216, pp. 331–341, Aug 2000.

[3] J. C. Saurin and F. Pilleul, "[Computed tomographic virtual colonoscopy to screen for colorectal neoplasia in asymptomatic adults]," *Gastroenterol. Clin. Biol.*, vol. 28, pp. 641–643, 2004.

[4] H. Yoshida and J. Nappi, "Three-dimensional computer-aided diagnosis scheme for detection of colonic polyps," *IEEE Trans Med Imaging*, vol. 20, pp. 1261–1274, Dec 2001.

[5] Medtronic, "Image-guided surgery overview," 2010.

[6] D. Burschka, M. Li, M. Ishii, R. H. Taylor, and G. D. Hager, "Scale-invariant registration of monocular endoscopic images to CT-scans for sinus surgery," *Med Image Anal*, vol. 9, pp. 413–426, Oct 2005.

[7] C. H. Wu, Y. N. Sun, and C. C. Chang, "Three-dimensional modeling from endoscopic video using geometric constraints via feature positioning," *IEEE Trans Biomed Eng*, vol. 54, pp. 1199–1211, Jul 2007.

[8] S. B. Solomon, P. White, C. M. Wiener, J. B. Orens, and K. P. Wang, "Three-dimensional CT-guided bronchoscopy with a real-time electromagnetic position

sensor: a comparison of two image registration methods," *Chest*, vol. 118, pp. 1783–1787, Dec 2000.

[9] Y. Schwarz, J. Greif, H. D. Becker, A. Ernst, and A. Mehta, "Real-time electromagnetic navigation bronchoscopy to peripheral lung lesions using overlaid CT images: the first human study," *Chest*, vol. 129, pp. 988–994, Apr 2006.

[10] W. E. Higgins, J. P. Helferty, K. Lu, S. A. Merritt, L. Rai, and K. C. Yu, "3D CT-video fusion for image-guided bronchoscopy," *Comput Med Imaging Graph*, vol. 32, pp. 159–173, Apr 2008.

[11] J. W. Suh, A. L. Abbott, P. D, A. A. Beex, P. D, T. M. P. D, D. Y. Gao, P. D, and J. W. Suh, "Deformable registration of supine and prone colons for ct colonography," 2007.

[12] J. Liu, K. Subramanian, T. Yoo, and R. Van Uitert, "A stable optic-flow based method for tracking colonoscopy images," *IEEE Computer Vision*, pp. 1–8, July 2008.

[13] J.-P. Thirion and A. Gourdon, "Computing the differential characteristics of isointensity surface," *Comput. Vis. Image Underst.*, vol. 61, no. 2, pp. 190–202, 1995.

[14] S. Kobayashi and K. Nomizu, *Foundations of Differential Geometry*. Massachusetts: Addison-Wesley, 1996.

[15] J. J. Koenderink, *Solid Shape*. Cambridge, Massachusetts: MIT Press, 1990.

[16] N. Sladoje, "On Analysis of Discrete Spatial Fuzzy Sets in 2 and 3 Dimensions," *PhD.thesis*, 2005.

[17] R. M. Summers, C. F. Beaulieu, L. M. Pusanik, J. D. Malley, R. B. Jeffrey, D. I. Glazer, and S. Napel, "Automated polyp detector for CT colonography: feasibility study," *Radiology*, vol. 216, pp. 284–290, Jul 2000.

[18] J. Nappi and H. Yoshida, "Automated detection of polyps with CT colonography: evaluation of volumetric features for reduction of false-positive findings," *Acad Radiol*, vol. 9, pp. 386–397, Apr 2002.

[19] P. Cathier, S. Periaswamy, A. Jerebko, M. Dundar, J. Liang, G. Fung, J. Stoeckel, T. Venkata, R. Amara, A. Krishnan, B. Rao, A. Gupta, E. Vega, S. Laks, A. Megibow, M. Macari, and L. Bogoni, "Cad for polyp detection: an invaluable tool to meet the increasing need for colon-cancer screening," *International Congress Series*, vol. 1268, pp. 978 – 982, 2004. CARS 2004 - Computer Assisted Radiology and Surgery. Proceedings of the 18th International Congress and Exhibition.

[20] I. T. Jolliffe, *Principal Component Analysis*. New York: Springer-Verlag, 1986.

[21] Y. Lu, I. Cohen, X. S. Zhou, and Q. Tian, "Feature selection using principal feature analysis," in *ACM Multimedia Conference*, 2007.

[22] H. Xu, D. Gage, C. Wyatt, and P. Santago, "Design Classification System for Computer-Aided Diagnosis in CT Colonography," *The Society for Imaging Informatics in Medicine*, 2009.

[23] H. Blum, *A transformation for extracting new parameter of shape*. Cambridge, Massachusetts: MIT Press, 1967.

[24] L. Hong, A. Kaufman, Y.-C. Wei, A. Viswambharan, M. Wax, and Z. Liang, "3d virtual colonoscopy," in *BIOMEDVIS '95: Proceedings of the 1995 Biomedical Visualization (BioMedVis '95)*, (Washington, DC, USA), p. 26, IEEE Computer Society, 1995.

[25] S. Bouix, K. Siddiqi, and A. Tannenbaum, "Flux driven fly throughs," 2003.

[26] Y. Ge, D. R. Stelts, J. Wang, and D. J. Vining, "Computing the centerline of a colon: a robust and efficient method based on 3D skeletons," *J Comput Assist Tomogr*, vol. 23, pp. 786–794, 1999.

[27] M. S. Hassouna and A. A. Farag, "PDE-based three dimensional path planning for virtual endoscopy," *Inf Process Med Imaging*, vol. 19, pp. 529–540, 2005.

[28] T. Deschamps and L. D. Cohen, "Fast extraction of minimal paths in 3D images and applications to virtual endoscopy," *Med Image Anal*, vol. 5, pp. 281–299, Dec 2001.

[29] M. Wan, Z. Liang, Q. Ke, L. Hong, I. Bitter, and A. Kaufman, "Automatic centerline extraction for virtual colonoscopy," *IEEE Trans Med Imaging*, vol. 21, pp. 1450–1460, Dec 2002.

[30] Y. Zhou and A. W. Toga, "Efficient skeletonization of volumetric objects," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, pp. 196–209, 1999.

[31] Y. Ge, D. R. Stelts, and D. J. Vining, "3d skeleton for virtual colonoscopy," in *VBC '96: Proceedings of the 4th International Conference on Visualization in Biomedical Computing*, (London, UK), pp. 449–454, Springer-Verlag, 1996.

[32] T. S. For, D. Chen, B. Li, Z. Liang, M. Wan, A. Kaufman, and M. Wax, "A tree-branch searching, multiresolution approach," in *in Image Processing, Kenneth M. Hanson, Editor, Proc. SPIE*, pp. 726–734, 2000.

[33] I. Bitter, A. E. Kaufman, and M. Sato, "Penalized-distance volumetric skeleton algorithm," *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, no. 3, pp. 195–206, 2001.

[34] B. Acar, S. Napel, D. Paik, P. Li, J. Yee, R. Jeffrey, and C. Beaulieu, "Medial axis registration of supine and prone ct colonography data," vol. 3, pp. 2433 – 2436 vol.3, 2001.

[35] S. Thrun, "Probabilistic robotics," *Commun. ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[36] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *In Proceedings of the AAAI National Conference on Artificial Intelligence*, pp. 593–598, AAAI, 2002.

[37] M. Kedzierski, "Precise determination of fisheye lens resolution," p. B5: 761 ff, 2008.

[38] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2003.

[39] K. Levenberg, "A method for the solution of certain problems in least squares," *Quart. Applied Math.*, vol. 2, pp. 164–168, 1944.

[40] J. Shi and C. Tomasi, "Good features to track," pp. 593–600, 1994.

[41] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," pp. 674–679, 1981.

[42] D. Lowe, "Object recognition from local scale-invariant features," pp. 1150–1157, 1999.

[43] H. Wang and D. Suter, "Robust adaptive-scale parametric model estimation for computer vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1459–1474, 2004.