FROM PIXEL TO REGION TO TEMPORAL VOLUME: A ROBUST MOTION PROCESSING FRAMEWORK FOR VISUALLY-GUIDED NAVIGATION

by

Jianfei Liu

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2011

Approved by:

Dr. Kalpathi R. Subramanian

Dr. Terry S. Yoo

Dr. Aidong Lu

Dr. Richard Souvenir

Dr. Thomas P. Weldon

ABSTRACT

JIANFEI LIU. From pixel to region to temporal volume: a robust motion processing framework for visually-guided navigation.
(Under the direction of DR. KALPATHI R. SUBRAMANIAN)

The ability to view pre-operative CT colonoscopy images co-aligned with optical colonoscopy images from endoscopic procedures can provide useful information to the gastroenterologist and lead to improved polyp detection. Colonoscopy data presents significant challenges from an image processing perspective: colon deformation, insufficient visual cues, temporary loss of features due to blurry images, etc.

In this dissertation, advanced mathematical tools and computer vision techniques are used to tackle these challenges, resulting in an automatic and robust tracking algorithm capable of processing relatively long sequences of colonoscopy images. There are three specific contributions. (1) Multi-scale optical flow is used to identify relative image displacements between consecutive optical colonoscopy images, and egomotion estimation based on the Focus of Expansion is used to estimate camera motion parameters. Straight and curved phantoms were designed to quantitatively validate the accuracy of the method, and clinical colonoscopy sequences from multiple patients were used to qualitatively evaluate the algorithm's robustness. Phantom results validated that the error was less than $10mm$ of the $288mm$ displaced in tracking consecutive images. (2) A region-flow based method is used to measure large visual motion of pairs of images interrupted by a blurry image sequence, and an incremental egomotion estimation algorithm is developed to maintain accuracy. Large camera motion is computed by subdividing visual motion into a sequence of optical flow fields. Accuracy of the approach was statistically validated by excluding sequences of images, using phantom images. In the straight phantom, after 48 frames were excluded, error was less than $3mm$ of $16mm$ traveled. In the curved phantom, after 72 frames were

excluded, error was less than $4mm$ of $23.88mm$ traveled. The accuracy was also evaluated by visually inspecting co-aligned optical and virtual colonoscopy images. (3) Temporal volume flow improves on the region flow algorithm by comparing temporal volumes separated by blurry images, followed by selecting the best image pair for region flow computation. Results are demonstrated by comparing tracking results with and without temporal volume flow.

Based on these new techniques, we have been able to continuously track over 4000 images of colonoscopy sequences comprising multiple colon segments and multiple blurry sequences.

ACKNOWLEDGEMENTS

I would like to thank *Prof. Kalpathi. R Subramanian* for supervising my dissertation and giving me the opportunity to work in the Charlotte Visualization Center. *Prof. Subramanian* not only taught and guided my research, but also gave me plenty of freedom. He showed how to simplify a complicated problem into several subproblems. I also acknowledge *Dr. Terry S. Yoo* who, jointly with *Prof. Subramanian*, patiently mentored me through weekly teleconferences in working on this challenging and clinically important dissertation project. Special thanks also are given to *Prof. Aidong Lu*, *Prof. Richard Souvenir*, and *Prof. Tom Weldon* for agreeing to be the committee members.

Prof. William Ribarsky recruited me to work on a mobile emergency response system. I am grateful to him for the recruiting financial support and experience. *Prof. Taghi Mostafavi* and *Dr. John Merritt* greatly assisted me in setting up phantom experiments.

Also, I appreciate the assistance from *Ms. Doralyn Bradley, Ms. Dee Ellington, and Ms. Deborah Craig* in my Ph.D life. I thank *Dr. Ronald Summers, Dr. Robert Van Uitert, Dr. David Chen, Dr. Shijun Wang, Dr. Jiaming Liu, and Dr Jianhua Yao* from National Institute of Health for showing me the clinical importance of this dissertation. I acknowledge *Tom Polk*, *Mike Edwards*, *Julie Wright* and *Ed & Annette Conrad* for editing my dissertation.

Moreover, I want to thank all former and current members of Charlotte Visualization Center: *Jackie Guest*, *Onyewuchi Obirieze*, *Li Yu*, *Matthew Hawkins*. They created such a friendly atmosphere that made working with them a real pleasure.

Finally, I want to thank my parents, *Xiangqian Liu* and *Ping Jiang*, for giving me constant support. And most of all, I deeply thank and appreciate my wife, *Bixiu Xiang*, for her love and reminder that there is more to life than science.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

VGN	**V**isually-**G**uided **N**avigation
MRI	**M**agnetic **R**esonance **I**maging
CT	**C**omputed **T**omography
GPS	**G**lobal **P**ositioning **S**ystem
FOE	**F**ocus **O**f **E**xpansion
VC	**V**irtual **C**olonoscopy
OC	**O**ptical **C**olonoscopy
DFS	**D**istance **F**rom **S**ource
DFB	**D**istance **F**rom **B**oundary
SIFT	**S**cale **I**nvariant **F**eature **T**ransform
LMS	**L**east **M**edian of **S**quares
LS	**L**east **S**um of squares
PDE	**P**artial **D**ifferential **E**quation
NCC	**N**ormalized **C**ross **C**orrelation
BP	**B**elief **P**ropagation
TVF	**T**emporal **V**olume **F**low
VTK	**V**isualization **T**ool**K**it
ITK	**I**nsight Segmentation and Registration **T**ool**K**it
GPU	**G**raphics **P**rocessing **U**nit

LIST OF NOTATIONS

$\mathbb{R}$	the set of real numbers
$\mathbb{Z}$	the set of integers
$\boldsymbol{p} = (x, y)$	spatial coordinates in an image
$\boldsymbol{p} = (x, y, t)$	spatial-temporal coordinates in a video
$I(\boldsymbol{p}) = I(x, y)$	an image
$I(\boldsymbol{p}) = I(x, y, t)$	a video stream
τ	a diffusion time in a diffusion-reaction system
σ	a scale parameter
σ_s	a scale parameter of the spatial coordinate
σ_t	a scale parameter of the temporal coordinat
σ_w	a scale parameter of a window function
$G(x, y; \sigma_s)$	an isotropic spatial Gaussian function
$L(x, y; \sigma_s)$	an isotropic Gaussian scale space in the spatial domain
$G(x, y, t; \sigma_s, \sigma_t)$	an anisotropic spatial-temporal Gaussian function
$L(x, y, t; \sigma_s, \sigma_t)$	an anisotropic Gaussian scale space in the spatial-temporal domain
$F'(x)$	abbreviation for $\frac{dF(x)}{dx}$
$F^{(n)}(x)$	abbreviation for $\frac{d^n F(x)}{dx^n}$
$\partial_x I$	abbreviation for $\frac{\partial I}{\partial x}$
$\partial_{xx} I$	abbreviation for $\frac{\partial^2 I}{\partial x^2}$
$\partial_{xy} I$	abbreviation for $\frac{\partial^2 I}{\partial x \partial y}$
∇I	abbreviation for spatial gradient of I, $(\partial_x I, \partial_y I)$
$\nabla_3 I$	abbreviation for spatial-temporal gradient of I, $(\partial_x I, \partial_y I, \partial_t I)$

$\nabla^2 I$	abbreviation for the spatial Laplacian of I, $\partial_{xx} I + \partial_{yy} I$
$\nabla_3^2 I$	abbreviation for the spatial-temporal Laplacian of I, $\partial_{xx} I + \partial_{yy} I + \partial_{tt} I$
$\mathrm{div}(I)$	abbreviation for the divergence operator, $\partial_x I + \partial_y I$
$\mathrm{div}_3(I)$	abbreviation for the divergence operator, $\partial_x I + \partial_y I + \partial_t I$
$\overrightarrow{u} = (u_x, u_y, u_t)$	optical flow vector
$\overrightarrow{v} = (v_x, v_y, v_t)$	visual motion flow vector
$\overrightarrow{w} = (w_x, w_y, w_t, w_\tau)$	temporal volume flow vector
$\overrightarrow{r} = (r_x, r_y)$	region flow vector
$\mathbf{A}$	a $n \times n$ matrix
$\mathbf{A}_{ij}$	a matrix entry at position (i, j) of $\mathbf{A}$
$\lambda_1, \ldots, \lambda_n$	the eigenvalues of $\mathbf{A}$
$\overrightarrow{e}_1, \ldots, \overrightarrow{e}_n$	the eigenvectors of $\mathbf{A}$
$\boldsymbol{P} = (X, Y, Z)$	spatial coordinates in the world coordinate
f	the focal length
$\overrightarrow{T} = (T_X, T_Y, T_Z)$	camera translation velocity
$\overrightarrow{R} = (R_X, R_Y, R_Z)$	camera rotation velocity
$\mathcal{N}(\boldsymbol{p})$	four-neighborhood of a pixel $\boldsymbol{p}$ or six-neighborhood of a voxel $\boldsymbol{p}$
$\mathcal{N}^+(\boldsymbol{p})$	the neighbors $\boldsymbol{q}$ of $\boldsymbol{p}$ with the indices of $\boldsymbol{q}$ are larger than those of $\boldsymbol{p}$
$\mathcal{N}^-(\boldsymbol{p})$	the neighbors $\boldsymbol{q}$ of $\boldsymbol{p}$ with the indices of $\boldsymbol{q}$ are smaller than those of $\boldsymbol{p}$
$\Psi_D(S^2), \Psi_S(S^2)$	non-quadratic penalise of the data

	and the smoothness term
$\Psi'_D(S^2), \Psi'_S(S^2)$	derivatives of $\Psi_D(S^2)$ and $\Psi_S(S^2)$ with respect to s^2
$(\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}$	the diffusivity between points $\boldsymbol{p}$ and $\boldsymbol{q}$

CHAPTER 1: INTRODUCTION

"Imagination is more important than knowledge."

– Albert Einstein

1.1 Motivation

Visually-guided navigation(VGN)[116] is a technique used by robots and autonomous vehicles to navigate reliably. It systematically integrates all motion or location information and develops optimized strategies to guide moving agents. VGN involves several sub-problems depending on the extent to which the exterior environment is known. Given a known model, a typical problem is motion planning, which attempts to control the robot to move smoothly along a planned route. It also attempts to address inaccuracies in sensing and deviation between the model and the actual environment. Compared to off-line motion planning, simultaneous localization and mapping[199, 200] handles real-time navigation to reconstruct an unknown environment or update a known map. At the same time, the robot's position is determined on the map.

A critical component in VGN is to collect the agent's motion data and to compare it with a current set of perceptions. This aspect is called *sensing* because data are commonly acquired by external sensors – laser rangefinders, 2D or 3D sonar sensors, and quasi-optical wireless sensors. However, the use of these devices brings extra cost, and they might be unstable under some conditions.

Computer-vision-based approaches[94, 93, 104, 205, 191] are another means to achieve the same purpose. Computer vision[80, 65, 66] is a research field that enables

computers to analyze and understand the underlying information in digital images. Like humans perceiving 3D structure in their environment while moving through it, cameras are mounted in robotic agents, and computer vision algorithms reconstruct 3D scene structures and estimate camera motion from the video stream. Together, 3D scene reconstruction and camera motion estimation are called *Structure from Motion*[217] because they both attempt to reconstruct three-dimensional structures by analyzing visual signals over time. Structure from Motion is a key step for visually-guided navigation based on computer vision, because it computes the agent's location and motion, and it is functionally equivalent to sensors.

In this dissertation, I concentrate on developing a VGN system that uses only the colonoscopy video stream to co-align optical colonoscopy (OC) and virtual colonoscopy (VC) images. Clinical study[177] has demonstrated that OC or VC, when used alone, is prone to missing precancerous lesions; they are complementary for detection. Here, *co-align* means aligning orientations and positions of OC and VC cameras. The goal is to keep the co-alignment error within $25mm$ for the entire OC procedure, because a colon fold is about $25mm$ long[50]. This ensures pre-detected polyps are simultaneously visible in both modalities. Anatomical features such as folds or polyps will appear within the same fold of the co-aligned OC and VC images. As a result, the polyp-miss rate will be reduced. Fig. 1.1 illustrates the VGN problem: given an OC video stream (left image), to determine the actual colonoscope's movements to drive the VC camera (shown in the right image) so as to achieve accurate OC and VC co-alignment, and permit the camera to be located in the external view (bottom right image).

However, OC and VC co-alignment challenges traditional VGN technologies based on computer vision, due to difficulties with OC images: insufficient visual cues, colon deformation, visual variance from VC images, and blurry image interruptions. Accordingly, very few people have attempted to work on this problem, although it is

clinically significant. In this dissertation, I reformulate visual motion computation and egomotion estimation to develop a robust VGN system. The fundamental contribution of the proposed VGN system is presented as three levels of visual motion, from pixel to region to temporal volume. Multi-scale optical flow accurately represents pixel shifts between two consecutive OC images. FOE-based egomotion estimation is then developed to precisely compute camera motion by employing multi-scale optical flow. Region flow computation densely matches all region pairs to calculate large visual motion between image pairs interrupted by blurry frames. Incremental egomotion estimation recovers large camera motion by subdividing visual motion into a sequence of optical flow fields. Temporal volume flow compares two temporal volumes before and after blurry images to compute their temporal coherence and to search for an image pair with sufficient visual similarities. The selected image pair enhances the accuracy of camera motion recovery using region flow.

1.1.1 Thesis Statement

Accurate co-alignment of real and virtual navigation is achieved through a visually-guided navigation framework, developed by robust egomotion estimation and multi-level visual motion flow from pixel to region to temporal volume.

1.1.2 Fields of Application

VGN has been shown useful for a variety of applications. For example, the Lunar Rover in Fig. 1.2a is an unmanned vehicle for exploring the surface of the moon. Because the rover and its mission are expensive, it is critical to safely guide the rover around obstacles and to move it accurately along planned routes. Military applications also use visually-guided navigation systems. For instance, the Global Hawk, illustrated in Fig. 1.2b, is an unmanned aerial vehicle that can be piloted remotely or fly autonomously. This aircraft can perform attack missions and reconnaissance

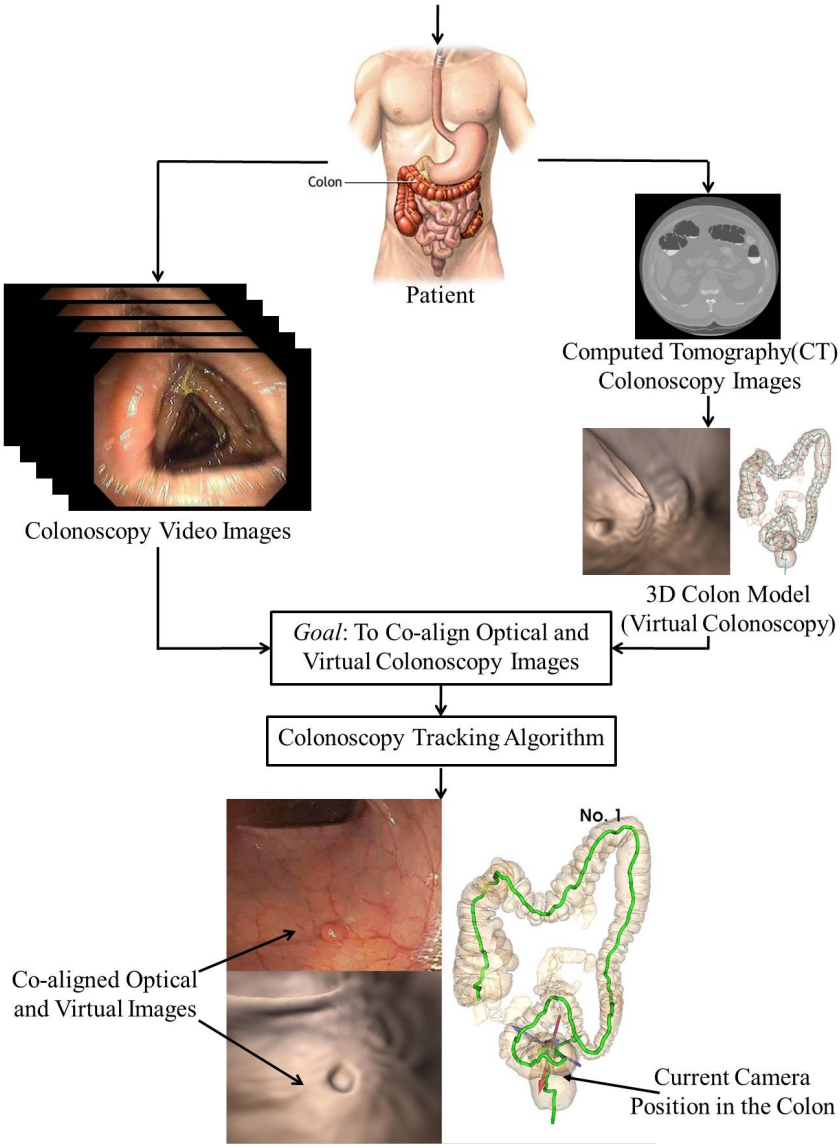
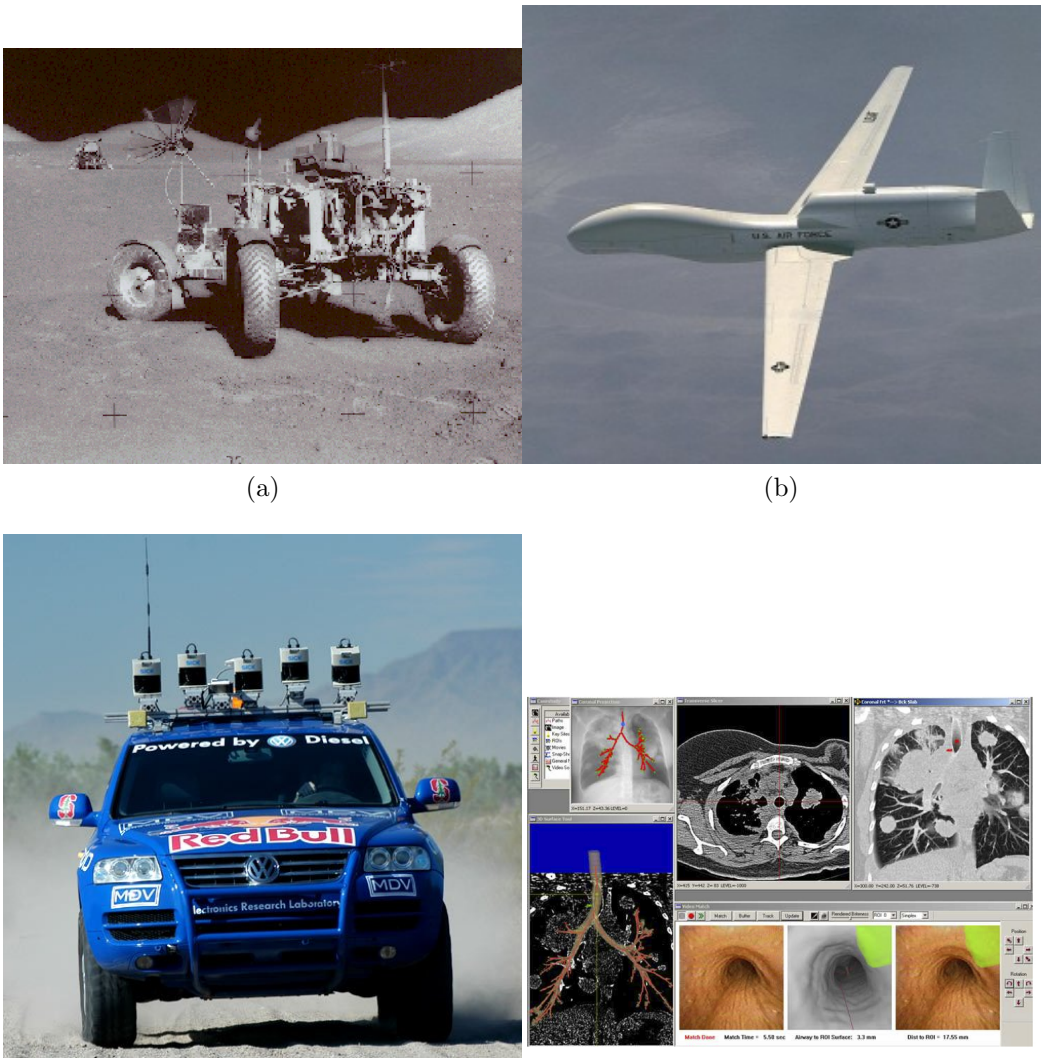


Figure 1.1: *Outline of OC and VC co-alignment.* OC and VC are two technologies to screen the colon. The goal of this work is to automatically co-align corresponding images for presentation to the gastroenterologist. OC (on the left) produces a continuous video stream of images. These images are analyzed to determine camera location and orientation from a colonoscopy tracking system. This information is used in adjusting the virtual camera of the 3D reconstructed CT volume (on the right), resulting in the bottom view of co-aligned OC and VC images, as well as the location of the virtual camera in the external view. The top image of the colon anatomy is reproduced from [223].

without a pilot, thus reducing casualties. Both of the above applications depend on external sensors and human assistance. Cameras on these devices perform only an auxiliary function during navigation, and the video streams assist operators to



(a) (b)

(c) (d)

Figure 1.2: *Possible visually-guided navigation applications.* (a) Exploration of moon – Apollo 17 Lunar Rover[40]; (b)unmanned reconnaissance aircraft – Global Hawk[208]; (c) autonomous car – a car that can drive without any steering[48]; (d) bronchoscopy tracking – a bronchoscopy surgery assisting system(This figure is copied from [91].)

determine the system status, and to make interactive decisions.

Applications based solely on computer vision are still in the conceptual phase, with many unresolved problems. However, as Figs. 1.2c and 1.2d show, such achievements are possible. Fig. 1.2c shows an unmanned car[171] developed by Stanford University. The purpose of this unmanned car is to reduce the incidence of automobile accidents

and to save lives. Despite doubts as to whether the car really achieves this purpose, or whether automobiles will navigate unaided in the future, this car has performed in real situations with encouraging results.

Aligning optical and virtual bronchoscopy images, as shown in Fig. 1.2d, is another VGN application. Optical and virtual bronchoscopy images are continuously matched[25, 88, 180, 57] to update the camera's location in the 3D virtual bronchi model. Moreover, a magnetic sensor can be integrated to enhance the tracking accuracy[159]. The system augments traditional bronchoscopy to accurately biopsy peripheral lung nodules, masses located far from the main branches of the lungs and difficult to reach using standard bronchoscopy. The fusion of optical and virtual bronchoscopy provides visualization of anatomical structures outside the bronchial walls. It also guides the bronchoscopist to follow the planned path and to determine location of the bronchoscope inside the bronchi. Optical and virtual bronchoscopy alignment is clinically useful for cancer detection and biopsy.

VGN is widely used in other areas, including manufacturing [170, 89, 103], entertainment [59, 172], and undersea exploration [169, 237, 238]. These applications show the importance and potential for computer-vision-based approaches, which can perform in many environments and reduce costs as compared to sensor-based methods. In the past three decades, much research has been carried out with respect to computer vision based VGN.

1.2 Overview

In this dissertation, I introduce a new visual motion processing framework to enhance VGN. This framework is applied to tracking a colonoscope, to demonstrate an efficient endoscopy guidance system to reduce the polyp-miss rate. This VGN system involves two main steps: visual motion computation and egomotion estimation. Visual motion is defined as the relative movement between similar visual patterns of an

image pair. Egomotion is the relative movement between a camera and the external world. Unfortunately, a navigation system cannot accurately track a colonoscope if it uses only a single method to compute visual motion and a single method to estimate egomotion. Colonoscopy navigation failure is due to various types of video sequences. In terms of visual contents, a colonoscopy video can be classified as clear or obscure. A clear colonoscopy video is called a fast-camera-motion video if camera translation velocity exceeds $45mm/sec$ or camera rotation velocity is more than $30°/sec$. Otherwise, it is called a slow-camera-motion video. Consequently, a colonoscopy video stream can be classified as one of three types: a slow-camera-motion video sequence; a fast-camera-motion video sequence; or an obscure video sequence.

A slow-camera-motion video stream is generally represented as a sequence of consecutive and clear colonoscopy images. Sparse and dense optical flows are two efficient means in measuring visual motion between successive images. Because the Focus of Expansion(FOE) accurately describes the camera motion in the optical flow field, an egomotion estimation method based on the FOE is developed to separately compute camera rotation and translation velocities. Slow-camera-motion images are accurately tracked.

Two colonoscopy images, interrupted by an obscure video sequence, comprise a fast-camera-motion video sequence. Sparse and dense optical flows fail to represent the visual motion of two such images, because the temporal image derivatives are invalid. Dense matching of region pairs is used instead to accurately measure large visual motion yielding region flow. Unfortunately, FOE-based egomotion estimation fails to directly compute large camera motion. Significant visual motion is subdivided into a sequence of optical flow fields through a strategy based on partial differential equations(PDE), and significant camera motion parameters are incrementally estimated by sequentially performing the FOE-based method on all optical flow fields.

The accuracy of incremental egomotion estimation is determined by the visual sim-

ilarity between two selected images. In order to search for two images with sufficient similarity, temporal volume flow(TVF) is developed to compute temporal coherence between two video sequences before and after blurry images. Finally, a micro-GPS system can be constructed inside the colon by using the estimated camera motion parameters to co-align OC and VC images, as shown in Fig. 1.1.

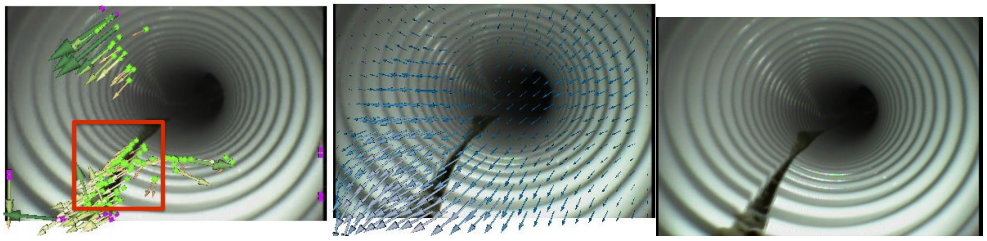
To specify my contributions in detail, I will describe next the challenges of visual motion computation and egomotion estimation, as it relates to the visually-guided navigation. Below, I discuss characteristic difficulties of OC images to provide insight into the proposed framework.

1.2.1 Tracking Consecutive Colonoscopy Images

The primary issue of tracking consecutive colonoscopy images includes sparse and dense optical flow computation as well as FOE-based egomotion estimation. Both are discussed in chapter 4. This section describes technical challenges related to these two problems, in conjunction with inherent difficulties of colonoscopy images.

Sparse optical flow calculation identifies two sets of interest points[79, 147, 12] from an image pair and finds their correspondences to measure visual motion. An interest point[79, 154] is a good feature candidate because it is the intersection of at least two dominant edges. The Harris matrix[79, 194], commonly used for detecting interest points, is a combination of spatial derivatives defined in the image domain. Fig. 1.3a shows interest points detected by the Harris matrix and their sparse optical flow. Interest points are represented as cubes, and optical flow vectors are indicated by arrows. Note that interest points located inside a red rectangle are intersections between the artificial seam and tube rings.

Unfortunately, spatial derivative calculation in the Harris matrix is an ill-posed problem[207] because derivative calculation is mathematically unsound in the discrete image domain. For robust derivative calculations, the image is first smoothed by the



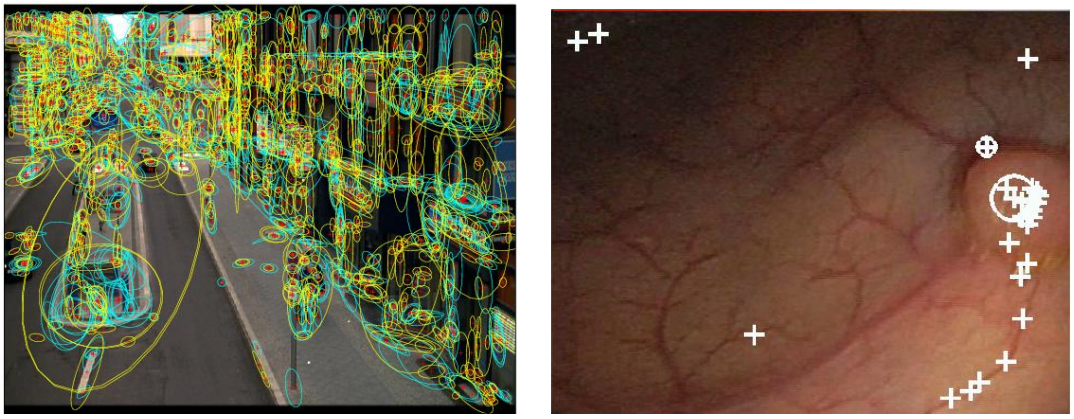
(a) (b) (c)

Figure 1.3: *Two types of visual motion between successive colonoscopy phantom images.* (a) Sparse optical flow in the first frame; (b) dense optical flow in the first frame; (c) the second frame. Here, optical flow vectors are visualized as arrows. In Fig. 1.3a, an interest point is represented as a green cube if its sparse optical flow vector is correctly computed. Otherwise, it is represented as a magenta cube.

Gaussian function. But image smoothing yields another issue: what is the optimal scale parameter of the Gaussian function? If the scale parameter is too large, the image is over-smoothed. If it is too small, image noise or fine image structures will remain. Moreover, matching two sets of interest points involves temporal derivative calculations because it assumes that intensities of corresponding points remain invariant over time. Thus, sparse optical flow computation also requires temporal scale selection. Because the sampling rate is more dense in the spatial domain than in the temporal direction, anisotropic Gaussian scale space is employed to smooth the video stream. One critical issue in sparse optical flow computation is the search for optimal spatial-temporal scales.

Sparse optical flow by itself is insufficient to accurately estimate camera motion between two colonoscopy images. There are three main reasons. First, in contrast to natural world images, colonoscopy images contain insufficient visual cues and generate a small number of interest points. Fig. 1.4 compares the interest points detected by the Harris-affine detector[155] on a street-view image versus the scale invariant feature transform (SIFT) algorithm[147] on a colonoscopy image. There are high-density interest points in Fig. 1.4a. Most of them are actual physical corners, located at building roofs, window corners, moving cars, etc. All these points are good interest

feature candidates because their descriptors are distinctive for sparse optical flow computation. By contrast, there are few geometrical discontinuities in the OC image. Only a few interest points are detected near a polyp and blood vessels in Fig. 1.4b, although the SIFT algorithm usually generates a large number of feature points. Second, the feature descriptors are not distinct because colonoscopy images, generated by a tiny fiber camera, have less intensity variance. False feature matches often occur and produce inaccurate sparse optical flow vectors. Finally, sparse optical flow represents the projected three-dimensional camera motion on the two-dimensional image plane. It is ambiguous to estimate the actual camera motion from only a set of sparse optical flow vectors.



(a) (b)

Figure 1.4: *Comparison of interest points detected in street and colonoscopy images.* (a) The Harris affine features[155] represented by ellipses; (b) the SIFT[147] features indicated by white crosses. The SIFT feature detector usually generates higher density interest points than the affine feature detector. However, there are only a few interest points in Fig (b) because of insufficient visual cues, while massive feature points in Fig (a)(This figure is reproduced from [198])

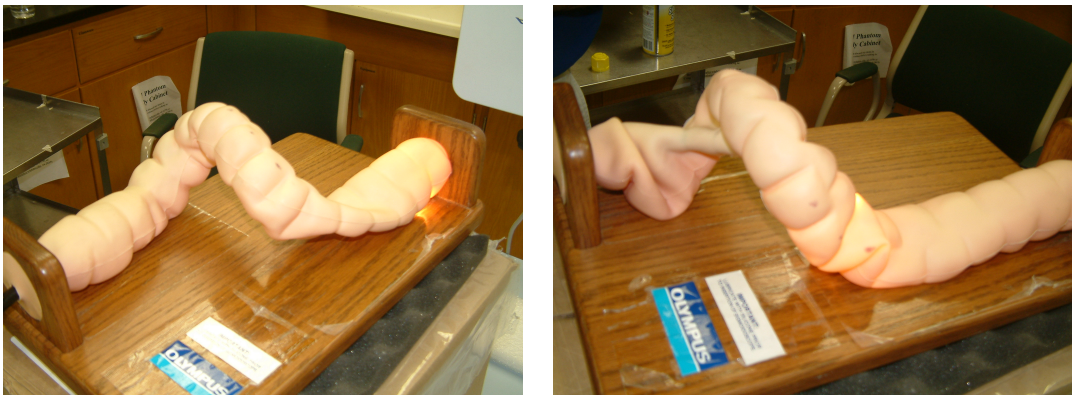
An alternative visual motion computation is to densely match point correspondences and generate a dense optical flow field, as shown in Fig. 1.3b. Dense optical flow can moderate false feature matches because its computation is the minimization of a global energy function over the entire image domain. However, dense optical flow estimation also involves scale selection in the spatial-temporal domain.

The second issue, related to consecutive colonoscopy images, is the estimation of camera velocities from sparse and dense optical flow. Camera velocity estimation refers to egomotion estimation. Egomotion is relative movement between a camera and the external world. However, egomotion is difficult to estimate accurately, even when accurate sparse and dense optical flows are available, because the estimation system is sensitive. Moreover, the ambiguous relationship between visual motion and camera velocity complicates the estimation. For example, it is difficult to distinguish (in the visual motion field) between a camera rotating clockwise around the X-axis and a camera rotating counter-clockwise around the Y-axis.

Colonoscopy images also have inherent challenges that complicate egomotion estimation.

Deformation occurs frequently in the colon. Fig. 1.5 illustrates deformation by stretching or telescoping, camera operation during colonoscopy procedures. In comparison with the colon phantom in Fig. 1.5a, the right portion of this colon phantom expands drastically while its left part contracts in Fig. 1.5b. Most egomotion estimation algorithms assume that objects within the visual field are rigid. Although recent research[195, 206] has made progress in non-rigid motion analysis, most techniques use piecewise rigid-motion assumptions. This piecewise approximation cannot completely estimate non-rigid motion within the colon.

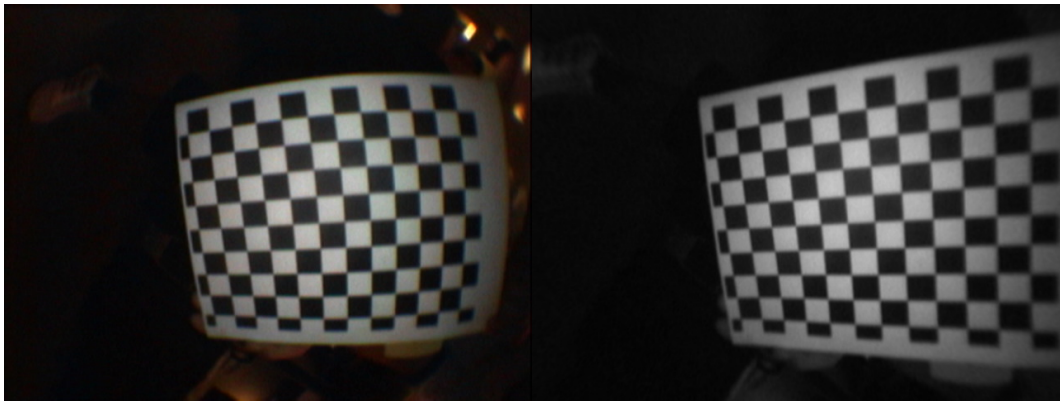
Visual distortion is generated because a tiny camera is installed on the tip of a colonoscope. To maximize the visual field, this camera has a wide-angle lens, causing fish-eye effect. Fish-eye effect severely distorts colonoscopy images, making it nearly impossible to acquire high-quality images. Fig. 1.6 shows a distorted colonoscopy image of a checkerboard. The checkerboard appears bent near the center, although it is physically planar. The distortion seriously affects accuracy of egomotion estimation because it violates the assumption that the



(a) (b)

Figure 1.5: *Illustration of colon deformation.* (Thanks to Dr. Pete Santago from Wake Forest University and Dr. Christopher Wyatt from Virginia Tech for providing these colon phantom images.) (a) The initial state when the colonoscope is inserted into the phantom; (b) another instance when the colonoscope is withdrawn near the center of the phantom. In Fig.(b), one can see that the right colon expands while the left colon contracts.

camera provides a perspective projection.

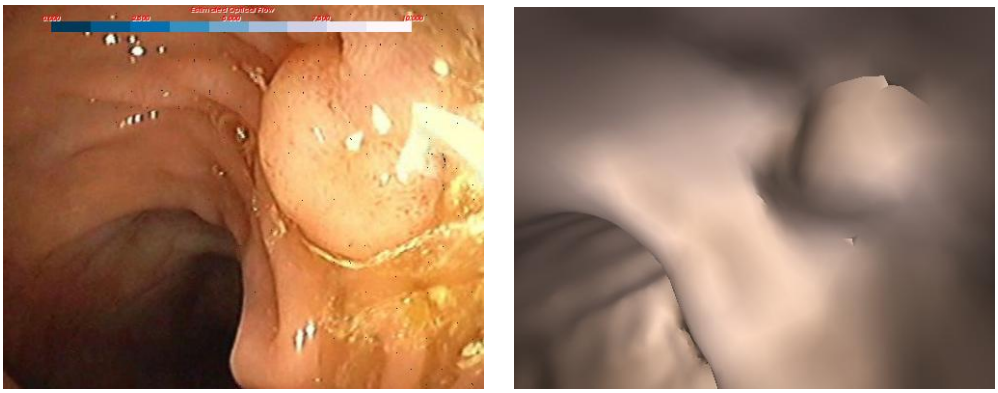


(a) (b)

Figure 1.6: *A colonoscopy image of a planar checkerboard(left image) and its corresponding undistorted image(right image).* Fish-eye effect seriously distorts the image. The checkerboard appears bent near the image center.

Visual variance between OC and VC images is caused by artifacts inside the colon. Artifacts such as fluid, stool, and blood vessel are difficult to simulate in VC images, even if the colon is perfectly segmented from CT data. Fig. 1.7

compares co-aligned OC and VC images. The polyp and folds are synthesized in both images, but yellow fluid and muscle textures are lost in the VC image. This dissimilarity prevents application of the same image registration algorithms[25, 88, 159, 57] that are used in bronchoscopy tracking.



(a) (b)

Figure 1.7: *Comparison between co-aligned OC and VC images.* The yellow fluid and muscle textures are impossible to duplicate in the VC image.

Contribution: In order to tackle these challenges, an egomotion estimation approach must handle errors of sparse and dense optical flow as well as colon artifacts. FOE, the intersection between the camera translational directions and the image plane, is an important feature point that can stabilize the estimation process. Camera motion information at this point is more stable and accurate than anywhere else in the optical flow field[218]. After the FOE is computed, camera translation and rotation parameters can be separately computed. Therefore, the essential problem of egomotion estimation is the identification of the FOE and exploiting it to estimate camera motion parameters.

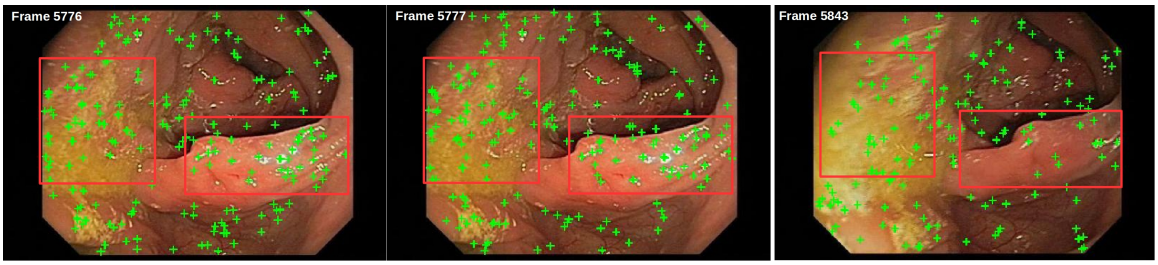
1.2.2 Estimating Large Motion

Blurry images frequently occur in the colonoscopy video stream and cause colonoscopy tracking interruptions, due to the absence of stable visual motion information. The

fundamental issue of continuously co-aligning OC and VC images is the exclusion of blurry images and the estimation of large camera motion parameters from images before and after the blurry sequence, as described in chapter 5. The process includes two main components: visual motion computation based on region flow and incremental egomotion estimation.

Visual motion computation is challenging because discarding blurry images is equivalent to producing large visual motion. Sparse and dense optical flows fail to identify substantial visual motion. There are two main reasons. First, temporal sampling rates are coarse when the camera motion is significant. Image smoothing along the temporal direction results in aliasing. For this reason, temporal derivatives cannot be accurately computed, and the result is inaccurate sparse and dense optical flows. Second, the number of false feature matches in sparse optical flow computation is exponentially increasing because feature descriptors based on intensities are indistinct. Also, interest points at artifacts such as fluid and water are unstable for determining significant visual motion. Fig. 1.8 gives an example. Fig. 1.8a and Fig. 1.8b show two successive colonoscopy images with folds, fluids, and bright blobs. Only the interest points near folds are true features that deliver camera motion information. Other interest points are relatively stable between successive frames, as indicated by red rectangles in Fig. 1.8a and Fig. 1.8b. But those interest points do not continue for very long. It is important to note that blobs in the right rectangle disappear in Fig. 1.8c while fluid in the left rectangle tends to accumulate. These dynamic changes cause improper visual motion estimation, because the results do not stem from actual camera motion.

Contribution: Two important strategies are employed to enhance the accuracy of significant visual motion computation. On the one hand, the SIFT feature descriptor is used to describe an interest point, so as to be insensitive to large visual motion. On the other hand, region flow densely matches all region pairs to compute visual



(a) (b) (c)

Figure 1.8: *Dynamic effects of interest points in colonoscopy images.* Many interest points are detected near the areas with bright blobs or fluids, enclosed in the red rectangles. They are invariant in two successive frames ((a) and (b)), but become unstable in (c) after a blurry image sequence because the blobs and fluids are dynamic.

motion between the selected image pair. As detailed in chapter 5, the accuracy of visual motion is continuously enhanced by performing SIFT feature matching and employing region flow vectors as the predefined matching ranges.

Another critical component is the estimation of large camera motion from significant visual motion. Besides inherent challenges of colonoscopy images, the most difficult issue is the simplified relationship between camera velocity and optical flow, as described in Appendix G. This simplified relationship is a critical equation for deriving most egomotion estimation algorithms. Its use is justified in estimating egomotion between consecutive images, because camera velocities are small. However, this equation is invalid when significant camera motion is artificially caused by the exclusion of blurry images. At this point, the egomotion estimation system becomes unstable and produces inaccurate camera motion parameters. There are two potential solutions to estimate large camera motion. One solution is the development of an egomotion estimation algorithm based on a general mathematical prototype that relates large camera motion parameters and visual motion.

Contribution: Subdivide significant visual motion into a sequence of optical flow fields. This is followed by application of existing egomotion estimation algorithms to incrementally estimate large camera motion from all optical flow fields, as described

in Chapter 5.

1.2.3 Computing Temporal Coherence

Large motion estimation involves selecting two colonoscopy images before and after a blurry image sequence and assumes that they have similar visual patterns. Visual motion is computed by measuring relative image displacements between visual patterns. However, visual similarity is not always guaranteed in the selected images. Temporal coherence can improve the image pair search because it is represented as the concurrence of similar visual patterns in the video stream. Thus, temporal coherence computation is an important topic in this dissertation.

The routine strategy[122, 121, 110, 190, 231] in computing temporal coherence is to detect a set of feature points in the video stream and to define feature descriptors for detected feature points. The temporal coherence computation is thus converted into measuring the distance values between feature descriptors. However, a small set of feature descriptors cannot reliably represent a video stream and false feature matches will cause improper temporal coherence results. Moreover, the image pair is also difficult to select based on a set of sparse feature matches.

Contribution: A PDE-based scheme is used to densely match two video streams before and after blurry images, resulting in temporal volume flow. It represents relative displacements between the corresponding visual patterns, and is used to search for an image pair with maximum visual similarity.

1.3 Organization and Contributions

In this dissertation, I have developed a visually-guided navigation system based on computer vision and show its application to co-align OC and VC images. The system includes multi-scale optical flow(Chapter 4), region flow(Chapter 5), and temporal volume flow(Chapter 6). In an effort to understand the full range of contributions

in these three chapters, I present the reader with a sigmoid colonoscopy sequence, shown in Fig. 1.9, to explain my proposed algorithms.

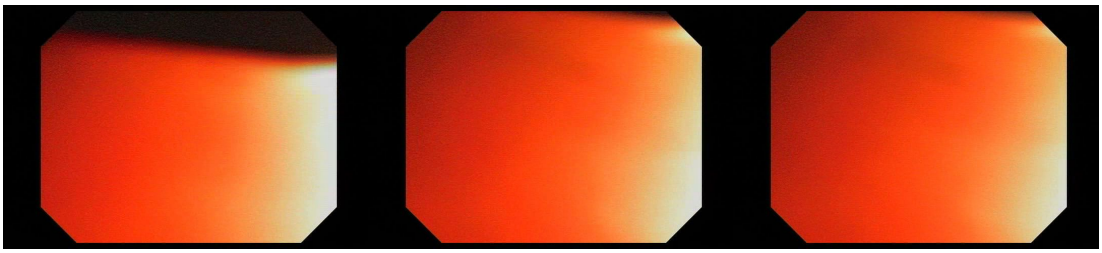
Fig. 1.9a and Fig. 1.9b are two consecutive colonoscopy images, which occur before a blurry sequence. In order to accurately measure their visual motion, multi-scale optical flow is proposed to compute accurate sparse and dense optical flow in optimized spatial-temporal scales. Egomotion estimation based on the FOE uses sparse and dense optical flow to sequentially estimate camera rotation and translation parameters. They are used to accurately track consecutive colonoscopy images, as described in Chapter 4.

However, blurry images typically occur in the colonoscopy video stream. For instance, Fig. 1.9d - Fig. 1.9f are three blurry images in the sigmoid colonoscopy sequence because the colonoscope touches colon wall. These images contain unstable visual motion, and the tracking system fails to accurately estimate egomotion. The tracking system must be able to recover camera motion parameters from such interruptions. The colonoscopy images before and after the blurry sequence have significant visual motion. In chapter 5, region flow is proposed to measure large visual motion by densely matching region pairs. A set of accurate sparse visual motion vectors is determined by using region flow to limit search ranges. Large visual motion is then artificially subdivided into a sequence of optical flow fields. Large camera motion is incrementally estimated by using the FOE-based egomotion estimation on all optical flow fields. Finally, colonoscopy tracking failure is recovered, and the tracking system can continue to track colonoscopy images.

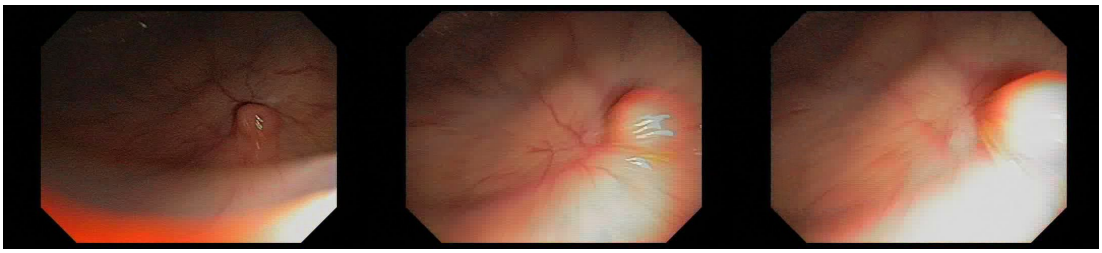
The image pair used for region flow computation is arbitrarily chosen, assuming the two colonoscopy images are similar because of temporal coherence, such as Fig. 1.9c and Fig. 1.9g. However, Fig. 1.9b and Fig. 1.9g are better image pair candidates because they are more visually similar. In order to search for such image pairs, the TVF computation is developed to match two temporal volumes interrupted by

blurry images, as described in Chapter 6.

(a) No.25922 frame (b) No.25923 frame (c) No.25975 frame



(d) No.26051 frame (e) No.26054 frame (f) No.26055 frame



(g) No.26084 frame (h) No.26086 frame (i) No.26087 frame

Figure 1.9: *A sigmoid colonoscopy video segment is used to illustrate the contributions to track OC images in this dissertation.* (a,b): two consecutive colonoscopy images before blurry images; (c) the last clear colonoscopy image before a blurry sequence; (d-f): three blurry images; (g) the first clear colonoscopy image after the blurry sequence; (h,i) two successive colonoscopy images after the blurry sequence. My contributions is centered at developing robust visual motion and egomotion estimation algorithms to robustly track consecutive colonoscopy images as well as to recover tracking system when blurry interruptions happen.

With respect to a better understanding, medical and technical backgrounds are described in Chapter 2 and 3, respectively. The main contributions – three level visual motion flow – are described in chapters 4, 5 and 6. Chapter 7 summarizes my main contributions and discusses future work.

CHAPTER 2: BACKGROUND – COLONOSCOPY

"Of the 22,000 operations I have personally performed, I have never found a single normal colon and of the 100,000 that were performed under my jurisdiction, not over 6 percent were normal."

– John Harvey Kellog, M.D.

This thesis concentrates on the development of visually-guided navigation techniques and its application to colonoscopy tracking, the co-alignment of optical colonoscopy (OC) and virtual colonoscopy (VC) images. To better comprehend this problem, some medical background is needed regarding the colon.

This chapter does not attempt an in-depth discussion of the colon's medical aspects. I describe only those aspects related to colon anatomy, optical colonoscopy, virtual colonoscopy, colon cancer and polyps. Moreover, a similar application, bronchoscopy tracking is discussed to understand similarities and differences. These elements contribute to understanding of the proposed tracking framework.

2.1 Colon Anatomy

The colon is a hollow tube about 5 feet long, as illustrated in Fig. 2.1. It is comprised of six parts,

1. Cecum, the beginning of the colon;
2. Ascending colon, the right vertical portion of the colon;
3. Transverse colon, the portion traversing from right to left;

4. Descending colon, the left vertical descent of the colon;

5. Sigmoid colon, the s-shaped segment of colon;

6. Rectum and anus, the last part of the colon for excreting the solid waste.

The main function of the colon is the storage of unabsorbed food waste, absorbed water and other body fluids before the waste is eliminated.

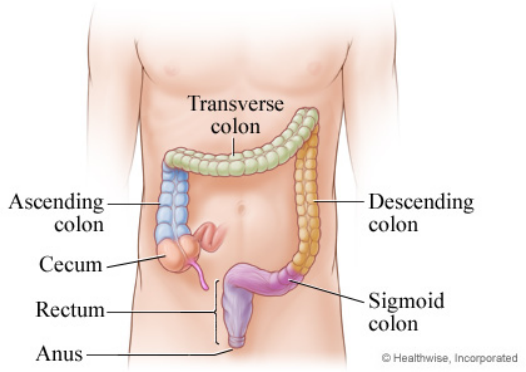
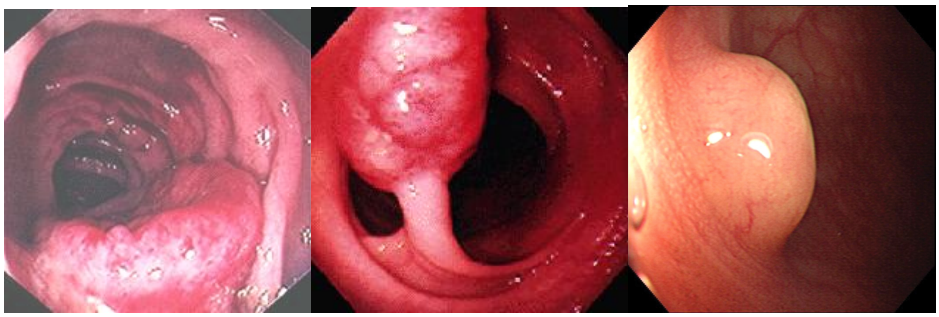


Figure 2.1: *Colon anatomy.* This figure is reproduced from[83]

2.2 Colon Cancer and Polyps

Colon cancer is often discussed with rectal cancer, and together they are referred to as *colorectal cancer*. It was the fifth most common cancer affecting both men and women in 2010[105]. There were $102,900$ colon and $39,670$ rectal cases diagnosed in the United States in 2010, that caused $51,370$ mortalities. It accounted for 10% of cancers in men and 11% in women [132, 174].

Colorectal cancer shown in Fig. 2.2a is usually developed from a colon polyp. A polyp is a flesh growth, and it is extremely common in the colon. Its incidence increases as individuals get older. Colon polyps are displayed in two basic varieties: pedunculated(Fig. 2.2b) and sessile(Fig. 2.2c). Pedunculated polyps are mushroom-like tissue growths that are attached to the surface of the mucous membrane by a long and thin stalk. Sessile polyps sit right on the surface of the mucous membrane.



(a) Colon cancer (b) (c)

Figure 2.2: Examples of (a) the colon cancer([73]); (b) a pedunculated polyp([74]); and (c) a sessile polyp([114]).

Clinical statistics show that 50% of people over the age of 60 have at least one polyp. When polyps grow large enough, they can become cancerous. Screening for colon polyps and removing them early in their development are particularly important for reducing the incidence of colon cancer.

The four most common types of colon polyps are inflammatory, adenomatous (adenoma), hyperplastic, and villous (tubulovillous) adenoma[60]. In addition to these, two less common polyp types are lymphoid and juvenile. Lymphoid polyps are usually rare and benign. Juvenile refers to a type of polyp, not the age at which the polyp first develops.

Inflammatory colon polyps are found most often in patients with an inflammatory bowel disease. They are not actually polyps; but rather, they are a reaction to chronic inflammation in the colon. Inflammatory polyps are benign and the risk of them becoming cancerous is generally low.

Adenomatous polyps are the most common type of polyp and make up about 70% of polyps in the colon. Adenomas can develop into colon cancer, but fortunately, this process typically takes many years.

Hyperplastic refers to the activity of the cells forming the polyp. The cells in this type of polyp are increasing in number. Hyperplasia denotes an abnormal

increase in the number of cells in a tissue, with enlargement of the area. Despite the fact that the cells in hyperplastic polyps are growing and reproducing, these have minor chances of developing into cancers.

Villous or Tubulovillous Adenoma accounts for approximately 15% of polyps that are found and removed by colonoscopy. These polyps are more dangerous because they have the highest likelihood of developing into colon cancer. Villous adenomas may be sessile, or flat, making them extremely hard to remove.

2.3 Optical Colonoscopy

OC is the most common procedure to detect and remove polyps by using a colonoscope. The colonoscope is a flexible and multi-channelled tube attached to a CCD camera or a fiber optic camera on its tip shown in Fig. 2.3.

Figure 2.3: Optical colonoscope.

A colonoscope has several channels including irrigation, instrument port, light, and lens, as illustrated in the right image of Fig. 2.4. Irrigation is a variant of enema treatment, which involves flushing the colon with water in different quantities, temperatures and pressures. As with an enema, the purpose of injecting water is to clean the colon. The top left image in Fig. 2.4 is an example of water injection,

releasing from irrigation channel. The light channel serves to illuminate the colon, which helps the lens to record clear images. Surgical tools are inserted through the instrument port for purposes of colon biopsy or polyp removal. For instance, the center and bottom left images illustrate polyp removal by a snare and biopsy forceps, respectively.

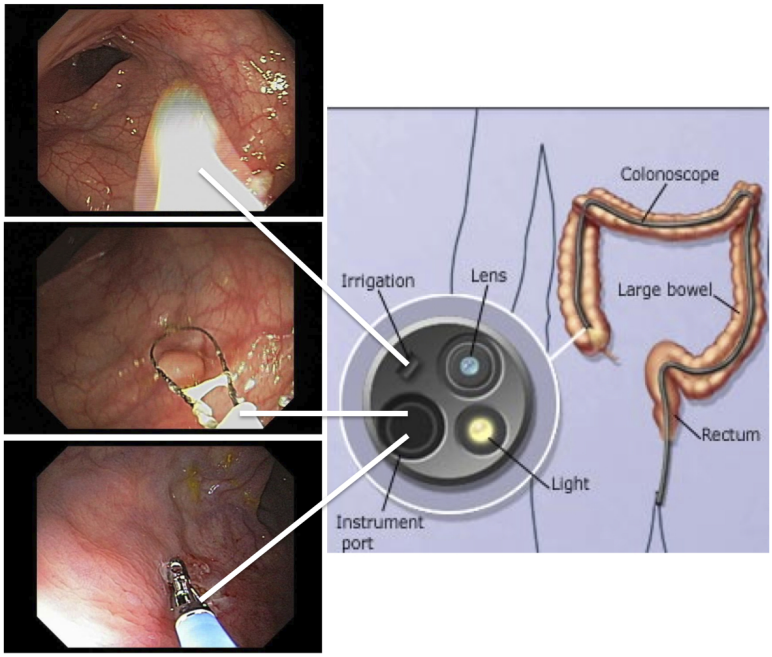


Figure 2.4: *Colonoscope channels.* The right image shows the tip of a colonoscope. It has multiple channels, including irrigation for pushing air or injecting water, an instrument port for inserting surgery tools, light for illumination, and a lens to record video. The left images present three examples. The top left image shows the injection of water, while the center and bottom left images show the insertion of snare and biopsy forceps to remove polyps. Here, the right figure is reproduced from [158].

Before colonoscopy, the colon must be free of solid matter. For one to three days, the patient is required to consume a fluid-only diet. The colonoscope is first inserted though the anus, up the rectum, across the colon (sigmoid, descending, transverse and ascending colon), and ultimately arriving at the cecum. Fig. 2.5a displays the surgical environment. The gastroenterologist is able to observe the inside of the colon and track unusual lesions, providing a lasting record of the exam that may be reviewed. During the procedure, the colon is occasionally insufflated with air to maximize vis-

ibility. Biopsies are frequently taken for histology. For screening purposes, a closer visual inspection is often performed upon withdrawal of the endoscope, and withdrawal takes 20 to 25 minutes. Figs. 2.5b through 2.5d illustrate different phases of colonoscopy examination in the cecum colon, transverse colon, and sigmoid colon.

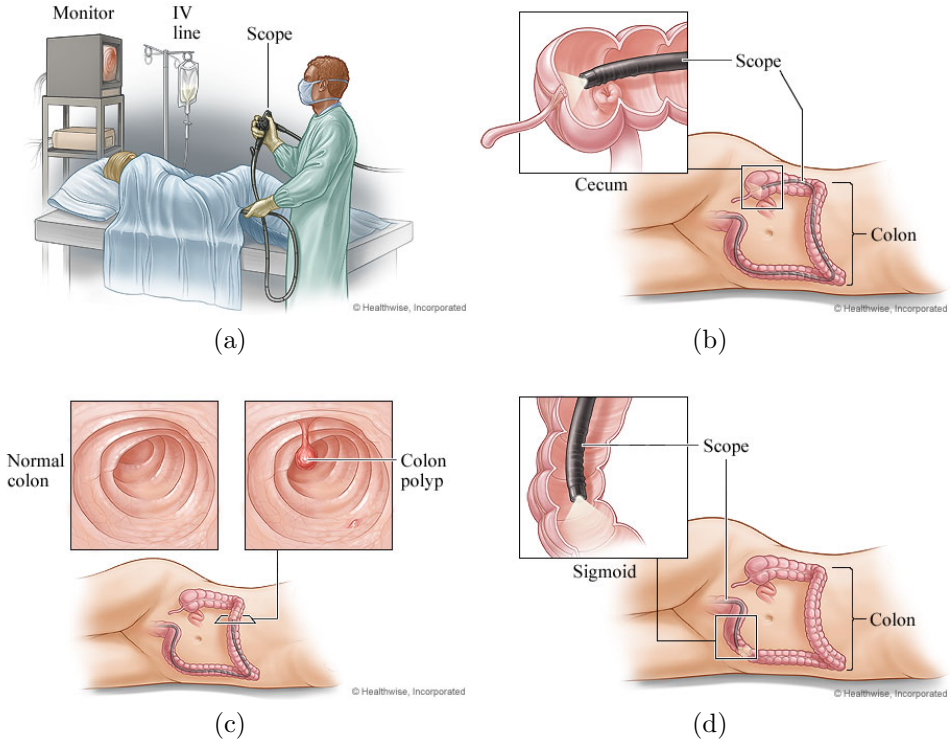


Figure 2.5: *Illustration of a colonoscopy procedure.* (a) The colonoscopy environment. The gastroenterologist inserts the colonoscope into the patient's colon and observes the inside on a video monitor; (b-d) show different phases of colonoscopy surgery in the cecum colon, transverse colon, and sigmoid colon. All figures are reproduced from *Health.com* website([83]).

Because OC provides such an elegant procedure for detecting and removing polyps, many gastroenterologists and patients believe that colonoscopies are nearly infallible. Thus, a person who has a colonoscopy ought to remove all colon polyps and cancer. However, a recent article[115] in the *New York Times* claimed that :

> "The test may miss a type of polyp, a flat lesion or an indented one that nestles against the colon wall. And now, a Canadian study, published in the journal Annals of Internal Medicine, found the test, while still widely

recommended, was much less accurate than anyone expected. The test missed just about every cancer in the right side of the colon, where cancers are harder to detect but about 40 percent arise. And it also missed roughly a third of cancers in the left side of the colon."

2.4 Virtual Colonoscopy

VC[96, 167, 109] is an alternative medical tool designed for examining the colon, building on medical image processing and visualization[95, 10]. It is also called a CT (Computed Tomography) colonoscopy. Gastroenterologists can preview the colon and roughly know the polyp's distributions in advance. A careful surgical plan is thus designed to reduce the polyp missing rate.

The concept of virtual endoscopy was first proposed by Mori[161] and Vining[222, 221], respectively, and developed into early practical systems including virtual bronchoscopy and virtual colonoscopy. Clinical study[113, 112] demonstrated that VC is as effective as OC in detecting a polyp larger than 5mm. Commercial softwares have been applied at research hospitals. For instance, VC developed by the Stony Brook university has been commercialized and evolved into the Viatronix system[219]. In this section, I review some critical technical components to build a VC system, which is summarized in Fig. 2.6. These technical components include procedure preparation, CT data collection, colon segmentation, centerline extraction, and virtual navigation.

Procedure preparation. Similar to preparation for an OC procedure, the patient will be required to take oral agents the day before, to clear stool from the colon. Remaining fecal matter is cleansed from the rectum by a suppository. This process is called *fecal tagging*. It helps the radiologist to better view virtual images because all feces are eliminated, which may otherwise lead to false positive results.

CT data collection. After fecal tagging, CT scan is then performed on the patient. Computed Tomography Imaging, known as a CT scan (or CAT scan), is shown in the

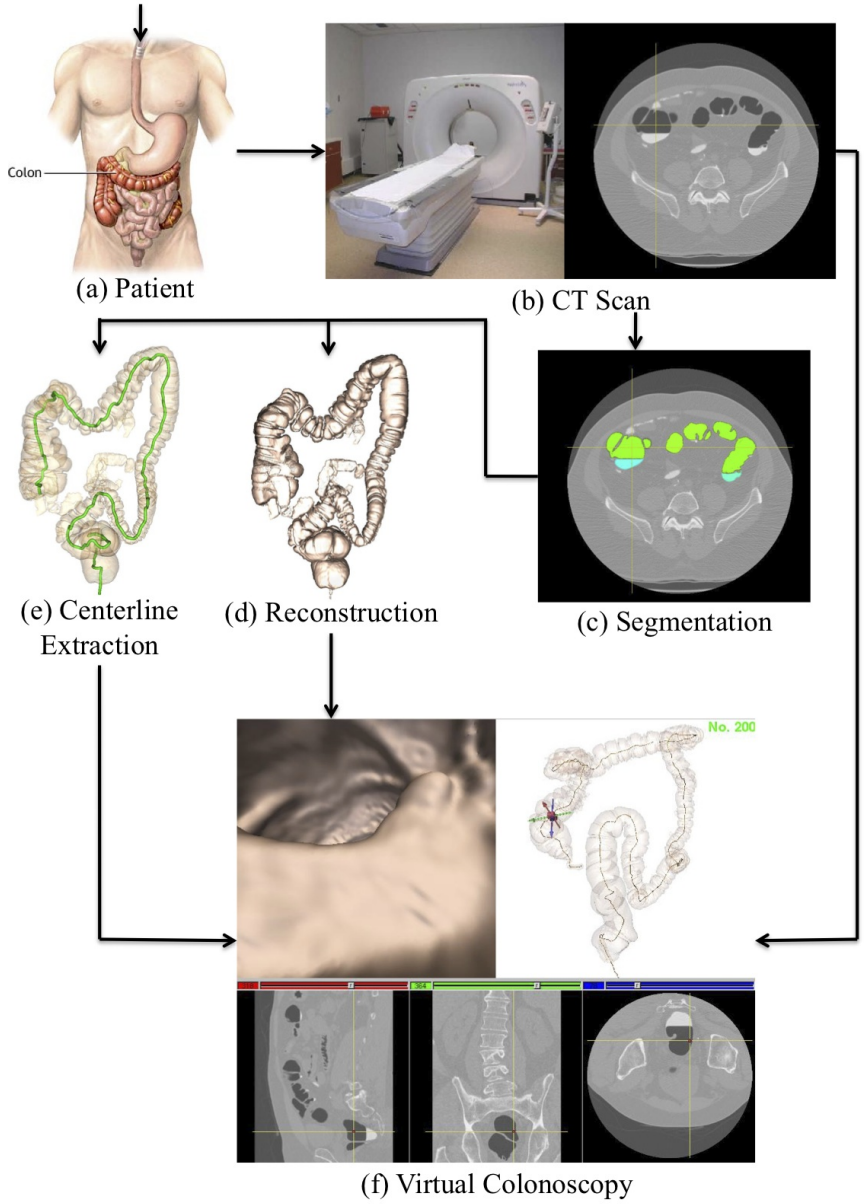


Figure 2.6: *Process of virtual colonoscopy development.* CT scans(b) are first performed on the patient(a). A virtual model(d) is then reconstructed for surface rendering by segmenting the colon from the original CT data(c). The centerline is also extracted from the segmented colon. Finally, all these resources are combined to build VC in figure(f). It includes virtual navigation (top left), external map (top right), and the sagittal, coronal, and transverse planes of the CT data (bottom images) from the current camera position. Figures (a) and (b) are cited from [124] and [223], respectively.

left image of Fig. 2.6(b). A CT scanner includes a portable table on which the patient lies. This table moves the patient through a ring-shaped scanner. A movable ring

is installed on the edge of the scanner, containing the x-ray tube and its associated detectors. During CT scanning, the ring-shaped scanner rotates around the patient with a fine fan of x-ray beams passed through the body from all angles into their associated detectors. Information from all detectors are compiled by a computer into an image which is representative of the particular slice of the body. Once the movable ring rotates 360 degree, a slice has been acquired.

The right image of Fig. 2.6(b) illustrates a slice of an abdomen CT scan, which shows several types of tissue with great clarity. Tissue types include bone, soft tissue and colon lumen(noted by yellow cross lines). When the radiologist performs CT scan on a patient, the patient's CT scan is required to be carried out both face down (prone) and face up (supine) in order to decrease the number of collapsed colonic segments and to improve sensitivity for polyp detection. The scan has high spatial resolutions in every slice, usually about $0.625mm$, and the spacing between two slices is commonly $1.0mm$. Consequently, CT scans can detect and determine the exact size and location of anatomical abnormalities such as polyps, tumors, and lesions. CT data are also generally of a cross-sectional nature, with the ability to produce 3D images of internal structures. Magnetic Resonance Imaging (MRI) is another imaging technology for building a VC system[14]. The feasibility of MRI colonoscopy has been found to be consistent with CT colonoscopy. Due to costs, MRI colonoscopy is seldom used, and it will be ignored in this dissertation.

Colon segmentation. A critical step in the construction of VC is the segmentation of colon regions from CT data. However, colon segmentation poses many challenges. Unlike other organs, the colon contains several materials involving air, fluid, and stool. This causes two extreme intensity ranges in the CT data. The lumen, full of air, is imaged in dark, while the region filled with fluid or stool is relatively bright. CT data may also contain disconnected regions of the colon, due to collapsed segments. Most colon segmentation algorithms[46, 129, 187] start with a region-growing algorithm

to extract colon lumen. Fig. 2.6(c) presents the segmentation results of the CT slice in Fig. 2.6(b), where the green regions are air-filled areas and the cyan regions are fluid-filled areas. After all colon components are extracted, digital cleansing techniques [120, 192] are used to connect fluid-filled and air-filled areas as well as refine boundaries between colon and non-colon regions.

Machine learning techniques [46, 129] are another good choice to supervise the segmentation process. They classify various objects in the CT data, erase non-colon materials, and thus improve accuracy. However, this process is very time-consuming. The level-set method[72, 43, 45] has been widely investigated to refine the boundary between air and fluid objects for avoiding segmentation leakage and reducing computation cost.

Centerline extraction. To easily control the movements of a virtual camera, the centerline is introduced to guide VC navigation. Most centerline extraction algorithms are based on the distance transform, which labels each data point with the distance to the nearest boundary. Therefore, it is often referred to distance from boundary (DFB). Some algorithms also use an additional distance transform, distance from source (DFS), which represents the distance from a source point. Various distance metrics have been used for distance propagation during distance transform, such as 1-2-3 metric[241], 3-4-5 chamfer metric[22], or 10-14-17 metric[44]. Exact voxel distances $(1 - \sqrt{2} - \sqrt{3})$, have been used as the distance metric[225]. A number of researchers have exploited the combination of DFB and DFS distance fields, and Dijsktra's algorithms (shortest path or minimum spanning tree) to extract the object's centerline. The primary idea in these schemes is to transform the object voxels (identified in a pre-processing step) into a weighted graph, with the weights defined by the inverse of the computed distance. Then Dijkstra's algorithm is applied to find the shortest path between specified end points. Chen[44] used this approach but modified the shortest path voxels to the maximal DFB voxels orthogonal to the

path. Zhou[241] chose centerline points among voxel clusters with the same DFS distance. Bitter[18, 17] used a heuristic that combines DFS and DFB distances, with the latter distance being considered a penalty aimed at discouraging the *hugging-corner* problem, which is typical of shortest path based approaches. Wan[224, 225] proposed a method that also used both DFS and DFB distances, but he emphasized the latter distance to keep the centerline close to the center of the tubular structure. More recently, Uitert[216] employed the level-set method on the colon's outer wall instead of the inner wall, in order to improve the centerline accuracy at the sub-pixel level. To reduce dependence on segmentation results, I developed a Gaussian-type probability model[142] to simulate the boundary between colon and non-colon regions. This model builds a more robust distance field and reduces the reliance on segmentation results. Fig. 2.6(e) illustrates my centerline result. All centerline points stay near the actual center of the virtual colon, and the camera has maximum visibility by using the centerline as the camera's trajectory.

Virtual navigation. After centerline and segmented colon are obtained, surface or volume rendering techniques are chosen to achieve interior navigation. Surface rendering is used in my tracking system. The colon mesh shown in Fig. 2.6(d) is reconstructed through the marching cube algorithm[146], followed by mesh smoothing to erase what is known as 'stairway' artifacts. Hong[96, 47] decomposed the colon mesh into cell-and-portal structures and only rendered the cells visible from the current viewpoints, in terms of visibility determination algorithms. Instead of determining real-time visibility in Hong's method, I calculate it prior to virtual navigation[143]. Visibility is computed using spheres in place of irregular portals and is managed through a tree-structure. However, the resulting image quality is usually poor, and some polyps might fade.

Direct volume rendering techniques[61, 125] instead mapped the scalar values of the colon data into RGBA(red, green, blue, and alpha) color space through trans-

fer functions. They composited every pixel value through various rendering strategies, such as ray casting[127, 126], splatting[230], shear warp[118], and texture mapping[90]. Volume rendering can now achieve real-time performance by using a state-of-the-art graphics card[131]. GPU-acceleration has also been applied to colonoscopy rendering[123]. The current research interest has shifted to enhancing navigation images to facilitate the detection of polyps. Fig. 2.6(f) gives an example of my virtual navigation system, where the top left view shows the virtual navigation, and the top right displays an external map. The three bottom images show the sagittal, coronal, and transverse planes of the CT data viewed by the current camera position.

VC is very useful in polyp detection. One important application is supine and prone registration[167, 100, 168]. If a polyp is observed in both supine and prone virtual navigation, it increases the radiologist's confidence that the polyp is real, and not a false result. Because polyps might be hidden by folds, they might be missed in virtual navigation even though the view point can be freely adjusted. Virtual colon flattening[76, 220, 97] is proposed to handle this issue by virtually clipping the colon and projecting it into a plane. Thus, the radiologist can find polyps that might otherwise have gone undetected. Automatic polyp detection[233, 128, 240] is another active research area of VC. The idea is to predefine feature vectors using curvature, intensity, gradient and other properties near polyp-prone areas as well as to select training datasets to designate polyp classifiers. When the optimized parameters are found, the presence of a polyp can be detected based on the trained classifiers.

2.5 Optical Colonoscopy Versus Virtual Colonoscopy

VC has several clinical advantages. First, VC is less invasive to the patient than its optical peer because it does not require body contact. As a result, the patient can return to his/her usual activities immediately after the procedure. More important, VC is free of risk and can be repeated as necessary. The physician can inspect the

colon with unlimited repetitions, in search of hidden polyps. Moreover, about 1 in 10 patients are unable to undergo a full cecum evaluation by OC. Therefore, we need VC to inspect the cecum.

The main disadvantage of VC is that a physician cannot biopsy or remove polyps during VC examination. OC must be performed subsequently if abnormalities are found. VC images also lose details that are presented in OC images, such as muscles or flat polyps. Thus, polyps smaller than $2mm$ might go undetected.

OC is still regarded as the gold standard for colorectal cancer treatment. It is recommended by many professionals because it permits actual viewing of the colon. OC gives an opportunity to identify polyps and cancer, and then to do biopsies or removal of the lesions, sometimes immediately. However, as indicated in Section 2.3, OC can also easily overlook polyps.

Simultaneously co-alignment of OC and VC images can reduce the polyp-miss rate during surgery, because pre-detected polyps in VC images can be merged into the OC procedure. When the colonoscope approaches this area, the physician is warned to observe it carefully. Summers[202] indicated that properly matching polyps on OC and VC also impacts the VC research. Typically, the locations of polyps identified by OC and VC may vary considerably. Such errors can impair computer aided diagnosis research. Benefits such as these ignite interest in the topic of this dissertation, *OC and VC co-alignment, or colonoscopy tracking*.

2.6 Bronchoscopy Tracking

A similar research problem has been extensively studied in the bronchi, which is called bronchoscopy tracking. However, because a bronchi undergoes mostly rigid motion and the bifurcations in the bronchi contain a lot of structural information, most bronchoscopy tracking algorithms explore matching optical and virtual bronchoscopy images. Bricault [25] first proposed a multi-stage tracking algorithm, making full

use of anatomical marks and repeated 2D and 3D registrations to align optical and virtual bronchoscopy images. This method depends on the anatomical features to reconstruct a 3D shape and it is difficult to get accurate results far from the bifurcations. Helferty[88] simplified the matching strategy and suggested an entropy measurement to assess the similarity of 2D images. Rai[180] assumed virtual and optical bronchoscopy images had the same depth values if two views were aligned, and these values were imported into the normalized cross-correlation metric to match virtual and optical bronchoscopy images. But this method is sensitive to depth discontinuities, since the depth accuracy is sensitive to the sampling rate of the Z-buffer. Mori[160] proposed a two-stage bronchoscopy tracking algorithm. Endoscope motion parameters were first estimated through optical flow based epipolar geometry between consecutive optical bronchoscopy images, and they were then refined through matching optical and virtual bronchoscopy images. Alternatively, Deguchi [55, 54] selected some image regions near the bifurcations and utilized sum-of-square difference as a cost metric on selected regions to measure the similarity between virtual and optical bronchoscopy images for the estimation of the camera motion parameters. Nagao [163] exploited the Kalman filter to linearly predict the camera motion by combining registration results from previous frames and reduced the search space of camera motion parameters. Similar to Bricault's method, Deligianni[56] also chose shape-from-shading algorithms to recover a 3D shape of the bronchi through pq-space analysis and performed 3D-2D matching to track bronchoscopy. She further introduced *Consedation* algorithm to compensate for the bronchi's deformation, and in turn, improved registration results.

Recent trends focus on integrating magnetic sensors to assist tracking. Deligianni[57] developed a probabilistic framework to synthesize camera locations and orientations. She combined electromagnetic trackers and an image-based registration algorithm to provide a statistically optimal pose. The deformation of bronchi was compensated

for by active shape models. The accuracy of camera motion was enhanced by removing deformation. Mori[159] employed sensor position and orientation as the starting point for an intensity-based image registration. Camera position and orientation were then estimated by matching optical and virtual bronchoscopy images. Both methods demonstrated that combination of image registration as well as sensor results made the bronchoscopy tracking problem very tractable, and obtained very encouraging results on the entire video sequence of the phantom data. Nevertheless, all these methods assume that virtual and optical bronchoscopy images are roughly the same. They also suppose that the presence of relevant anatomical features inside navigation images can be easily identified during matching.

However, bronchoscopy tracking algorithms cannot be directly applied to colonoscopy tracking. Technically, image registration algorithms fail to function well on colonoscopy images. First, colonoscopy images lack visual cues and do not contain important structural information, such as bifurcations. Secondly, artifacts like fluids and Specularities are difficult to simulate in the VC images, which causes visual difference between OC and VC images. Finally, OC and VC images are also dissimilar due to colon deformation. All these challenges make it difficult to apply image registration algorithms to colonoscopy tracking. From the application viewpoint, bronchoscopy tracking concentrates on improving the accuracy of biopsy procedures so as to accurately remove tumors growing outside the bronchi, which requires virtual and optical bronchoscopy images to look identical. On the contrary, because polyps grow inside the colon, colonoscopy tracking aims towards the simultaneous appearance of important anatomical features like folds or polyps in both optical and virtual colonoscopy images. At this moment, colonoscopy tracking does not need a high level of accuracy. This is the main reason I regard colonoscopy tracking as a visually-guided navigation problem.

CHAPTER 3: BACKGROUND – VISUALLY-GUIDED NAVIGATION

"Mathematics is the queen of the sciences."

– Carl Friedrich Gauss

In this chapter, I explore visually-guided navigation based on computer vision to tackle colonoscopy tracking because this method does not rely on visual similarity between optical colonoscopy(OC) and virtual colonoscopy(VC) images. Instead, it directly estimates camera motion from OC video streams and uses the estimated camera motion to co-align OC and VC images. There are two essential problems in a visually-guided navigation system based on computer vision, including visual motion computation and egomotion estimation. Visual motion computation measures relative image displacements between two images, and egomotion estimation uses visual motion to recover actual camera motion. In this chapter, three underlying computational theories are presented, including calculus of variations, scale-space theory, and Markov random field. Then I discuss some representative algorithms related to visual motion computation and egomotion estimation.

3.1 Computational Theories

Table 3.1 summarizes underlying computational theories used in my proposed algorithms. The proposed algorithms include sparse and dense optical flow, region flow, temporal volume flow, and incremental egomotion estimation. Sparse and dense optical flows are two types of small visual motion representations measuring image displacements between consecutive colonoscopy images. Region flow is a large visual motion expression for estimating significant image displacements between a colonoscopy

Table 3.1: Relationship between underlying computational theories and my proposed methods in this dissertation.

Proposed algorithms	Underlying Computational Theories		
	Scale-space theory	Calculus of variation	Markov random field
Sparse optical flow	×		
Dense optical flow		×	
Region flow			×
Temporal volume flow	×	×	
Incremental egomotion estimation	×	×	

image pair interrupted by blurry images. Temporal volume flow is a field of dense point shifts between two video segments, which assists the search for a colonoscopy image pair used in region flow computation. Incremental egomotion estimation is a partial differential equation based scheme to iteratively estimate substantial egomotion from the selected image pair. Scale-space theory is explored in sparse optical flow computation to determine optimal spatial-temporal scales for derivative calculations. This theory is also used in incremental egomotion estimation and temporal volume flow computation to remove fine image details that steers a minimization process towards local minima. Calculus of variations is used to compute dense optical flow, incremental egomotion estimation, and temporal volume flow using a variational formulation. Markov random field is an alternative mathematical tool to estimate region flow when a variational formulation contains discrete terms, and the Euler-Lagrange equation fails to minimize a discrete function.

Scale-space theory[232, 135] is a formal theory for representing an image as a one-parameter family of smoothed images, which is parameterized by the size of the smoothing kernel used for suppressing fine-scale image structures. The motivation of scale-space theory originates from an observation that an image usually contains objects of different scales. For example, Fig. 3.1a shows an image with many sunflowers. Scale invariant feature transform(SIFT) algorithm[147] is used to detect some feature points and their corresponding scales, where blue crosses indicate SIFT features, and

the radii of blue circles are equal to their detected scales. Obviously, feature points from anthers and leaves have different scales. In addition, the same object is projected with different scales into an image plane when the distance between the object and a camera varies. (Note the same anthers are indicated by red arrows in Fig. 3.1a and Fig. 3.1b). The associated scales are variant because the distance between the objects and a camera of Fig. 3.1a is closer than Fig. 3.1b. The main purpose of scale-space theory is the search for the characteristic object scale, which is defined as the scale parameter corresponding to the extreme scale response[135].

(a) (b)

Figure 3.1: *A set of SIFT features detected in sunflower images adapted from [151, 37].* The location of every feature point is indicated by a blue cross, and the radius of the corresponding circle is the characteristic scale. Anthers and leafs have different scales in Fig. 3.1a. Moreover, the same anther indicated by red arrows also has different scales because the distance between the objects and the camera in Fig. 3.1a is closer than in Fig. 3.1b. Scale-space theory is thereby developed to automatically detect this scale for tackling the vision-front problem[135].

Calculus of variations[63] and Markov random field[130] have been extensively used to handle computer vision problems. Because depth information is lost, almost all vision problems[149, 217, 80, 65] lack unique solutions and are defined as *ill-posed problems*[210, 207]. Smoothness constraints are frequently used to convert ill-posed problems into well-posed problems. After combining smoothness and data constraints, many computer vision problems are explicitly interpreted as the minimization of a

global energy[30].

$$E(\overrightarrow{u}(\boldsymbol{p})) = \int_{\Omega} (\underbrace{M(D^k F, \overrightarrow{u})}_{\text{Data constraint}} + \alpha \underbrace{S(\nabla F, \nabla \overrightarrow{u})}_{\text{Smoothness constraint}}) d\boldsymbol{p} \tag{3.1}$$

where $\boldsymbol{p} = (x_1, x_2, \cdots, x_n)$ denotes a n-coordinate point and $\overrightarrow{u} = (u_1, u_2, \cdots, u_m)$ is a m-tuple to be estimated. Assume $D^k F$ is the set of all partial derivatives of F of order k. $M(D^k F, \overrightarrow{u})$ is a data assumption and $S(\nabla F, \nabla \overrightarrow{u})$ is a data smoothness constraint. α is a constant to balance data and smoothness terms. Dense optical flow, region flow, temporal volume flow, and incremental egomotion estimation are all computed based on the energy function as defined in Eq. 3.1.

Calculus of variations and Markov random field are two common methods for minimizing Eq. 3.1. If $E(\overrightarrow{u}(\boldsymbol{p}))$ is considered as a functional of the function $\overrightarrow{u}(\boldsymbol{p})$, calculus of variations is a mathematical tool that finds extremal function $\overrightarrow{u}(\boldsymbol{p})$ that makes $E(\overrightarrow{u}(\boldsymbol{p}))$ attain a minimum value. Alternatively, if $E(\overrightarrow{u}(\boldsymbol{p}))$ is regarded as an energy function over a graph, Markov random field can be used to minimize Eq. 3.1 to find $\overrightarrow{u}(\boldsymbol{p})$, which is essentially a data labeling process.

3.1.1 Scale Space

Scale-space theory helps us understand the inherent object scales, so as to design proper metrics in search of the characteristic scales to perform computer vision tasks. Many researchers[135, 232] demonstrated that smoothing image to build multi-scale image representation is essentially a diffusion process to evolve the images. Supposing $L(\boldsymbol{p}; \tau) : \mathbb{R}^n \times \mathbb{R}^+ \rightarrow \mathbb{R}$ is a scale space representation of a n-dimensional image $I(\boldsymbol{p}) : \mathbb{R}^n \rightarrow \mathbb{R}$, where τ is the scale parameter, multi-scale image representation is defined as

$$\partial_\tau L = \text{div}(\mathbf{D} \cdot \nabla L) \tag{3.2}$$

Here, the matrix $\mathbf{D}$ is called the diffusion tensor and scale-parameter τ is also called the diffusion time. Variant $\mathbf{D}$ leads to different types of scale spaces, including linear isotropic, nonlinear isotropic, and nonlinear anisotropic. I will briefly describe each of these scale spaces, including their benefits and limitations, and then explain why nonlinear anisotropic is the most appropriate choice for my research.

Linear Isotropic Scale Space

The scale space is linear isotropic if $\mathbf{D} = \mathbf{I}$, where $\mathbf{I}$ is an identity matrix. $\mathbf{D} = \mathbf{I}$ means image smoothing is homogeneous in the entire image domain. Eq. 3.2 is converted into

$$\partial_\tau L = \nabla^2 L \tag{3.3}$$

It is a classical result that the solution of Eq. 3.3 is

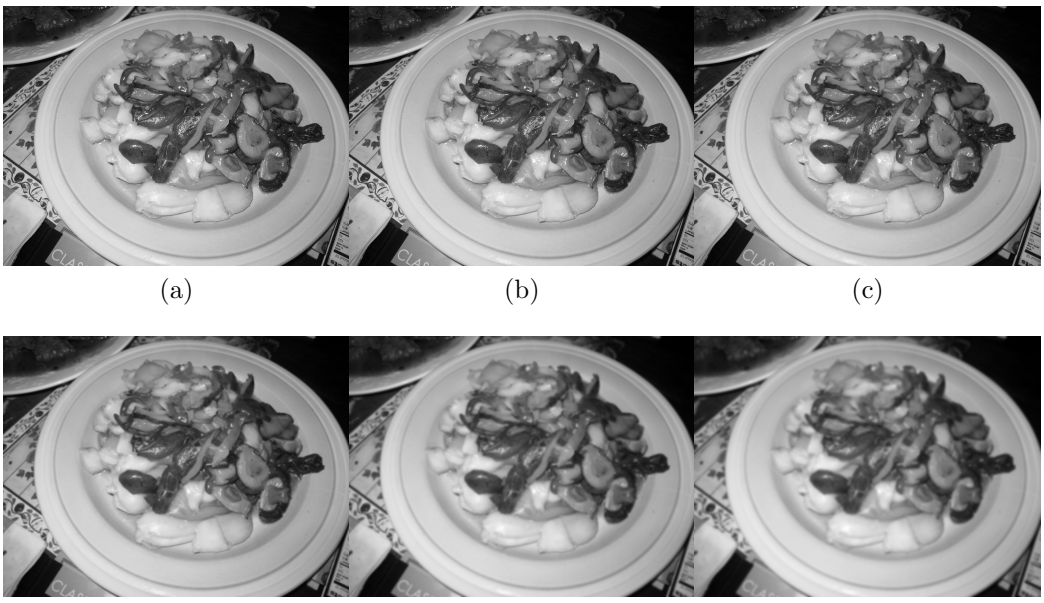
$$L(\boldsymbol{p};\tau) = \begin{cases} I(\boldsymbol{p}) & \tau = 0 \\ I(\boldsymbol{p}) * G(\boldsymbol{p};\tau) & \tau > 0 \end{cases} \tag{3.4}$$

where $G(\boldsymbol{p};\tau)$ is a Gaussian function. Suppose $\tau = \sigma^2$, the Gaussian function is defined as

$$G(\boldsymbol{p};\tau) = \frac{1}{2\pi\tau}\exp^{-(\boldsymbol{p}\cdot\boldsymbol{p})/2\tau} = \frac{1}{2\pi\sigma^2}\exp^{-(\boldsymbol{p}\cdot\boldsymbol{p})/2\sigma^2} = G(\boldsymbol{p};\sigma^2) \tag{3.5}$$

Therefore, linear isotropic image scale space is explicitly represented as the convolution between the Gaussian function $G(\boldsymbol{p};\sigma^2)$ and original image $I(\boldsymbol{p})$. Fig. 3.2 illustrates an example of linear scale space. A Chinese food image is chosen because it contains illumination variance(the surface of mushroom) as well as color change (brown mushroom and white Chinese cabbage). Fig. 3.2a shows the scale-space representation $L(\boldsymbol{p};\sigma^2)$ at scale parameter of $\sigma^2 = 0$. It corresponds to the original image $L(\boldsymbol{p};0) = I(\boldsymbol{p})$. Fig. 3.2b through Fig. 3.2f illustrate the evolution of the scale-space

representation with the increase of scale parameter $\tau = \sigma^2 = 1, 4, 25, 64, 100$, and the image becomes blurrier. Some extreme points, such as bright dots on the mushroom in Fig. 3.2a, disappear in Fig. 3.2f because they are averaged by neighboring points.



(a) (b) (c)

(d) (e) (f)

Figure 3.2: *An example of the Gaussian scale space, where the original image contains mushroom and Chinese cabbage.* (a) Gaussian scale-space representation $L(\boldsymbol{p}; \sigma^2)$ at scale $\sigma^2 = 0$, corresponding to the original image $I(\boldsymbol{p})$; (b) scale-space representation $L(\boldsymbol{p}; \sigma^2)$ at scale $\sigma^2 = 1$; (c) $\sigma^2 = 4$; (d) $\sigma^2 = 25$; (e) $\sigma^2 = 64$; (f) $\sigma^2 = 100$.

Linear isotropic scale space has a simple representation defined in Eq. 3.4 and it can be efficiently implemented in the image domain. Linear isotropic scale space is also demonstrated to have several scale invariant properties described in Appendix B, and it is extremely valuable in tackling vision-front problems, such as corner detection[137, 147, 155], edge extraction[136, 62, 188], image segmentation[204], and optical flow computation[27, 35], etc. However, important image features are often over-smoothed. For instance, image edges between mushroom and Chinese cabbage are hardly preserved in Fig. 3.2f.

Nonlinear Isotropic Scale Space

In order to avoid over-smoothing important features, the diffusion tensor $\mathbf{D}$ is modified to $F(|\nabla L|^2)\mathbf{I}$[176], which adapts to local image gradient magnitudes.

$$F(s^2) = \frac{1}{1 + s^2/\alpha^2} \qquad (\alpha > 0) \tag{3.6}$$

Eq. 3.2 is converted into

$$\partial_\tau L = \mathrm{div}(F(|\nabla L|^2)\nabla L) \tag{3.7}$$

The scale space based on Eq. 3.7 is nonlinear isotropic.

Variant image regions have sharp image boundaries in every scale level, and intra-region smoothing is superior to inter-region smoothing. Important image features can be well kept because they stay at image edges. For instance, image edges between bright dots and mushroom are clearly preserved in Fig. 3.3b because substantial diffusivity forbids smoothing image regions with significant image gradient.

(a) (b)

Figure 3.3: *Comparison between linear and nonlinear isotropic scale-space representations at* $\sigma^2 = 64$. (a) Linear isotropic scale-space representation; (b) nonlinear isotropic scale-space representation. Almost all bright dots of mushroom are preserved in Fig. 3.3b, while they are barely visible in Fig. 3.3a.

However, as indicated by Weickert[226], nonlinear isotropic scale space is unstable. Some new extreme points might be created during the construction of multi-scale

image representation, which fails *causality* property discussed in Appendix B. In addition, the modified diffusion tensor is dependent on local image gradient magnitudes, but orientation is invariant. This uniform diffusion still over-smoothes image features in colonoscopy video streams, preventing the usage of nonlinear isotropic scale space in colonoscopy tracking.

Nonlinear Anisotropic Scale Space

It is desirable to redesign the diffusion tensor $\mathbf{D}$ with respect to local image's anisotropy, which results in anisotropic scale space. The anisotropic scale space is defined as

$$L(\boldsymbol{p};\boldsymbol{\Sigma}) = I(\boldsymbol{p}) * G(\boldsymbol{p};\boldsymbol{\Sigma}) \tag{3.8}$$

It is similar to Eq. 3.4, except that Gaussian kernel is defined as

$$G(\cdot;\boldsymbol{\Sigma}) = \frac{1}{2\pi\sqrt{\det\boldsymbol{\Sigma}}}\exp^{-\frac{\boldsymbol{p}\boldsymbol{\Sigma}^{-1}\boldsymbol{p}^T}{2}} \tag{3.9}$$

where $\boldsymbol{\Sigma}$ is a symmetric positive semi-definite (covariant) matrix.

In order to build multi-scale image representation with respect to local image structures, $\boldsymbol{\Sigma}$ should be related to the anisotropy of local image structures. In the two dimensional image domain, the anisotropy is measured in terms of the *structure tensor* $\mathbf{J}(x, y; \Sigma_s, \Sigma_w)$[138, 226]. It is an anisotropic-scale representation of *Harris matrix* defined in Eq. 3.16. In the three dimensional video stream, exploiting structure tensor to estimate local affinities is very complicated. To simplify anisotropic scale space and to emphasize the anisotropy between spatial and temporal domains, the

scale parameter $\mathbf{\Sigma}$ in anisotropic Gaussian scale space is simplified to

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_s^2 & 0 & 0 \\ 0 & \sigma_s^2 & 0 \\ 0 & 0 & \sigma_t^2 \end{bmatrix} \tag{3.10}$$

where σ_s and σ_t are scale metrics with respect to spatial and temporal domains. Multi-scale optical flow computation will use Eq. 3.8 and Eq. 3.10, as detailed in chapter 4.

3.1.2 Calculus of Variations

Calculus of variations is a field of mathematics that addresses extremizing functionals, and it is generally considered as an approach to minimize functions of function. It has several useful properties. First, all data components can be explicitly integrated into a single mathematical equation as presented in Eq. 3.1. There are no underlying assumptions necessary in this formulation. The whole problem statement is thereby easily comprehended from a single energy formula. The formulation of the energy formula also forces the designer to clearly think about what assumptions can be made[26]. Second, the influence of different data components in Eq. 3.1, can be easily controlled by artificially adjusting their balance parameters. One can either dynamically modify these parameters, or they may remain constant. Therefore, the minimization process is artificially manipulated to proceed towards the designer's purpose.

The last and most important property is that calculus of variations provides a mathematically well-founded technique to minimize the energy. It leads to the so-called Euler-Lagrange equation. For instance, the Euler-Lagrange equation of Eq. 3.1

is

$$\begin{cases} \underbrace{\partial_{u_1} M}_{\text{Data}} - \alpha \underbrace{\text{div}(\partial_{\nabla_n u_1} S)}_{\text{Smoothness}} = 0 \\ \vdots \\ \underbrace{\partial_{u_m} M}_{\text{Data}} - \alpha \underbrace{\text{div}(\partial_{\nabla_n u_m} S)}_{\text{Smoothness}} = 0 \end{cases} \tag{3.11}$$

where div is a divergence operator and $\nabla_n u_i = (\frac{\partial u_i}{\partial x_1}, \cdots, \frac{\partial u_i}{\partial x_n}), i \in [1, \cdots, m]$. Appendix A provides the deduction of the Euler-Lagrange equation. The solution of a complicated Euler-Lagrange equation can be converted into a linear system in terms of some advanced numerical methods[183], such as sequential linearization approach described in chapters 5 and 6. Consequently, calculus of variations not only provides a sound mathematical basis in the modeling process but also gives well-founded numerical solutions.

3.1.3 Markov Random Field

Calculus of variations fails to minimize Eq. 3.1 if data or smoothness terms are discrete. Markov random field is an alternative method to minimize the discrete energy function. Eq. 3.1 is regarded as an energy expression of the Gibbs distribution[130]. The Hammersley-Clifford theorem[77] demonstrated that Gibbs distribution is equivalent to Markov random field, and Li[130] proved that maximizing a posterior solution of Markov random field is equivalent to minimizing Eq. 3.1.

In order to employ Markov random field to minimize Eq. 3.1, it is first converted into

$$E(\overrightarrow{u}) = \sum_{\boldsymbol{p} \in \Omega} M(\overrightarrow{u}(\boldsymbol{p})) + \sum_{(\boldsymbol{p}, \boldsymbol{q}) \in \mathcal{N}} S(\overrightarrow{u}(\boldsymbol{p}), \overrightarrow{u}(\boldsymbol{q})) \tag{3.12}$$

where $\mathcal{N}(\boldsymbol{p})$ is the neighborhood of $\boldsymbol{p}$ and Ω is the entire image domain. Markov random field is defined as a graphical model with a set of random variables fulfilling the following two conditions:

1. $PDF(\overrightarrow{u}(\boldsymbol{p})) > 0$ (positivity)

2. $PDF(\overrightarrow{u}(\boldsymbol{p})|\overrightarrow{u}(\Omega - \boldsymbol{p})) = PDF(\overrightarrow{u}(\boldsymbol{p})|\overrightarrow{u}(\mathcal{N}(\boldsymbol{p})))$ (Markovianity)

Here, $PDF(x)$ is a probability density function. Minimizing Eq. 3.12 through Markov random field[130] is essentially a data labeling process over the graphical model. Belief propagation[67] and graph cut[24] are two main strategies to efficiently minimize Eq. 3.12.

3.2 Optical Flow

In the previous section, I describe the computational theories, calculus of variations and scale-space theory, that underlie the optical flow techniques used in this dissertation. Optical flow refers to the relative movements of the same visual patterns between an image pair. It is a fundamental component in a visually-guided navigation system because optical flow is also considered as the projection of the egomotion onto the image plane. In this section, I describe sparse and dense optical flow computation in more detail. A comprehension of their computation is required in order to understand the specific techniques in this dissertation to calculate multi-scale optical flow, region flow and temporal volume flow.

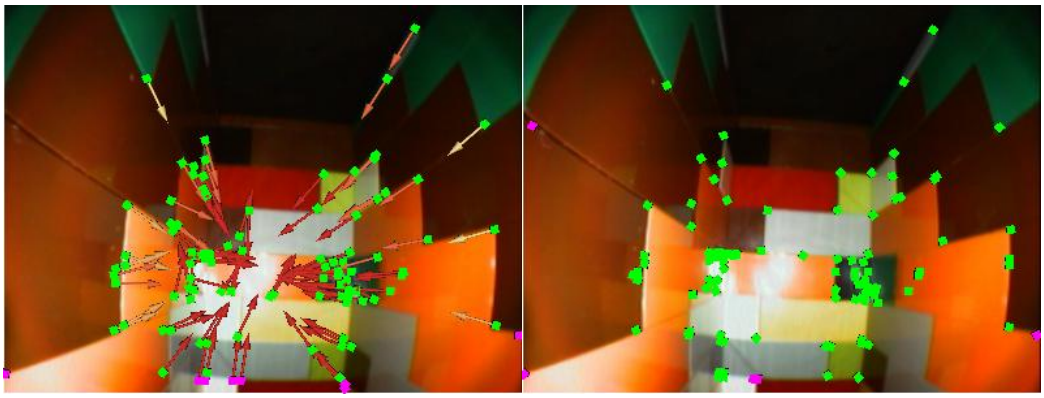
3.2.1 Sparse Optical Flow

Sparse optical flow is a set of visual motion vectors measuring relative movements between two sets of corresponding feature points in an image pair. It is widely used in egomotion estimation[203, 98, 52], video tracking[162, 5], and image mosaic[193, 41]. Sparse optical flow can be accurately computed if there exists two sets of matched feature points. Interest points are usually chosen as feature candidates to calculate sparse optical flow because these points stay at intersections of at least two dominant edges. Sparse optical flow computation assumes that the interest point is sufficiently

distinctive for feature matching. Egomotion can thus be precisely computed from accurate sparse optical flow. The Harris matrix[79, 194] is often used to detect interest points.

$$\mathbf{J} = G(\cdot;\sigma_w) * \begin{pmatrix} (\partial_x I)^2 & \partial_x I \partial_y I \\ \partial_x I \partial_y I & (\partial_y I)^2 \end{pmatrix} \tag{3.13}$$

and an interest point is determined if two eigenvalues of $\mathbf{J}$ are all large. Fig. 3.4 gives an example of sparse optical flow between an image pair of Fig. 3.4a and Fig. 3.4b based on the Harris matrix, where interest points are represented as cubes and sparse optical flow vectors are indicated as arrows.



(a) First frame (b) Second frame

Figure 3.4: *Example of sparse optical flow between two phantom images.* Interest points are represented as cubes and sparse optical flow vectors are visualized as arrows.

Multi-scale space analysis is of particular importance because it enables interest point detection at optimal scales and produces accurate and stable feature points. In this section, I describe two types of multi-scale space techniques. The first type relies on the linear isotropic scale space theory while the second type depends on the nonlinear anisotropic scale space.

Linear isotropic scale space

This section investigates linear isotropic scale space to compute sparse optical flow. Lindeberg[137] improved upon Eq. 3.13 in a linear isotropic scale representation,

$$\mathbf{J}(\cdot;\sigma_s^2,\sigma_w^2) = G(\cdot;\sigma_w^2) * \begin{pmatrix} (\partial_x L(\cdot;\sigma_s^2))^2 & \partial_x L(\cdot;\sigma_s^2)\partial_y L(\cdot;\sigma_s^2) \\ \partial_x L(\cdot;\sigma_s^2)\partial_y L(\cdot;\sigma_s^2) & (\partial_y L(\cdot;\sigma_s^2))^2 \end{pmatrix} \tag{3.14}$$

Here, $L(\cdot,\sigma_s^2)$ is defined in Eq. 3.4, and σ_s^2 and σ_w^2 are derivative and integration scales. In order to reduce the search space of σ_s and σ_w, they are coupled by $\sigma_w = \gamma\sigma_s, \gamma \in [\sqrt{2}, 2]$. The scale-normalized Laplacian operator[154] is used to find optimal scales.

$$\nabla^2_{norm} L(\cdot;\sigma_s^2) = \sigma_s^2 \nabla^2 L(\cdot,\sigma_s^2) \tag{3.15}$$

An iterative multi-scale corner detection algorithm is developed to identify interest points in the optimized scales when corner response (Eq. 3.14) and normalized Laplacian (Eq. 3.15) both achieve extrema. Fig. 3.5 shows corner features detected by multi-scale Harris matrix, where the location of a feature point is the circle's center and its corresponding scale is equal to the circle's radius. Note that all feature points locate at the intersections between two dominant image edges, such as boundaries between the door and the wall. Following the same strategy, lots of interest point detection algorithms [135, 51, 147, 110, 190] have been developed based on the linear scale space.

However, as was indicated in section 3.1.1, linear scale space tends to over-smooth image features, especially when two images have significant visual motion. Fig. 3.6 gives an example. Fig. 3.6(a) and Fig. 3.6(b) are a pair of images undergoing significant viewpoint change. If the linear scale space is built on circular image regions centered at the corner of the 'N' character in two images, then two regions contain

Figure 3.5: *Interest points detected by multi-scale Harris matrix.* Every point locates at the center of a circle, and its corresponding scale is indicated by the circle's radius.

different visual contents, as illustrated in Fig. 3.6(d) and Fig. 3.6(e). Comparing region descriptors built over different image regions yield inaccurate sparse optical flow.

Nonlinear anisotropic scale space

In order to address inherent limitations of the methods based on linear isotropic scale space, nonlinear anisotropic scale space is proposed to remove affine distortion when an image pair undergoes significant visual motion, such as an ellipse image region shown in Fig. 3.6(f). The ellipse image region used by nonlinear anisotropic scale space contains the same visual contents as a circular region shown in Fig. 3.6(a).

The essential idea of the nonlinear anisotropic scale space is that scale parameter Σ should be designed with respect to local affinity. The local affinity is measured based on the affine Harris matrix[155].

$$\mathbf{J}(\cdot;\Sigma_s,\Sigma_w) = G(\cdot;\Sigma_w) * \begin{pmatrix} (\partial_x L(\cdot;\Sigma_s))^2 & \partial_x L(\cdot;\Sigma_s)\partial_y L(\cdot;\Sigma_s) \\ \partial_x L(\cdot;\Sigma_s)\partial_y L(\cdot;\Sigma_s) & (\partial_y L(\cdot;\Sigma_s))^2 \end{pmatrix} \tag{3.16}$$

$\mathbf{\Sigma}_w$ represents the integration scale parameter of the Gaussian window function de-

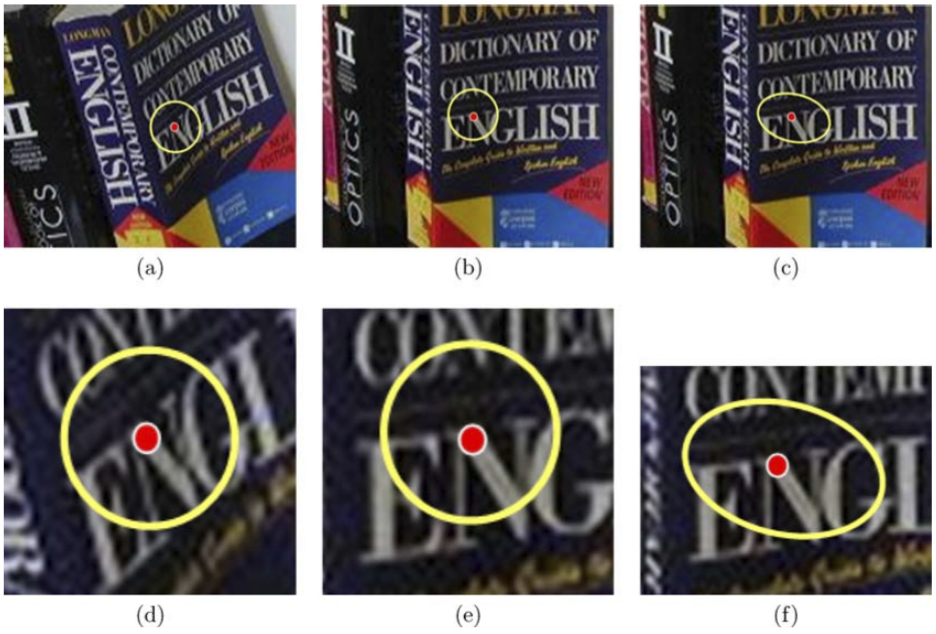


Figure 3.6: *Comparison between linear isotropic and nonlinear anisotropic scale spaces on an image pair with significant image deformation.* (a) First viewpoint; (b, c) second viewpoint. Two patches in (a) and (b) using linear Gaussian scale-space(represented as the fixed circular patches) can not deal with significant viewpoint change because they do not include the same visual contents in (d) and (e). Anisotropic Gaussian scale space can handle this problem. Visual contents in ellipses in (c) and (f) are equal to that in the circles of (a) and (d). All these figures are adapted from Mikolajczyk's work[157].

fined over an image region. $L(\cdot;\Sigma_s)$ is an anisotropic scale space representation defined in Eq. 3.8, and $\mathbf{\Sigma}_s$ is the anisotropic scale parameter. There are many feature detectors[107, 150, 213, 214, 108, 157, 156, 215, 154] based on the nonlinear anisotropic scale space, which utilize Eq. 3.16 to detect affine invariant image regions. Eq. 3.16 has also been extended to determine affine invariant volumes in video streams[121, 231, 111].

Fig. 3.7 illustrates feature matching results of the affine Harris detector[155] in the image domain. A pair of feature points near the cartoon's leg are detected and matched in the left two color images. There is significant affine distortion between this image pair due to substantial camera movements. The center two images illustrate two image regions centered at the selected features(corresponding to two green

rectangles in the left image pair), and the ellipses indicate their local affinities. Instead of directly building anisotropic Gaussian scale space on the ellipse image regions, anisotropic scale space is alternatively constructed by performing linear scale space on the normalized image regions shown in the image pair in the right column of Fig. 3.7, which is called transformation property. Ellipse image regions can be normalized into circular image regions, and only rotation variance exists between corresponding normalized image regions.

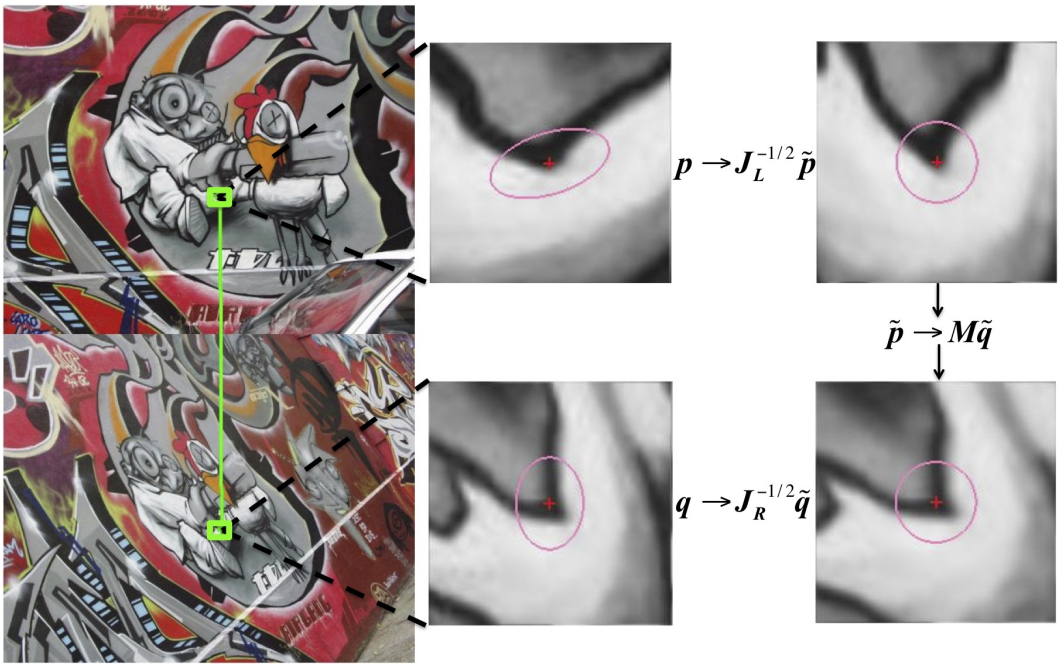


Figure 3.7: *Example of affine Gaussian scale-space over images.* Left column shows a pair of images and two matched feature points linked by a green line, detected by the affine Harris detector. A pair of image regions centered at feature points near the cartoon's leg are illustrated in the middle column. They are copied from the original images and two ellipses indicate their local affinities. Two normalized images are displayed in the right column. Two circles cover the same regions and the only difference between visual contents in two regions is a small amount of rotation. Local affine distortion is thus removed through normalization.

3.2.2 Dense Optical Flow

Dense optical flow is defined as image displacements of the same visual patterns between two images. It is extensively used in object detection and tracking[162], robot

navigation[191], and visual odometry[171]. Dense optical flow is useful in estimating the Focus of Expansion(FOE). The FOE is a key property for stabilizing egomotion estimation because it encompasses the most accurate egomotion information in the optical flow field[218].

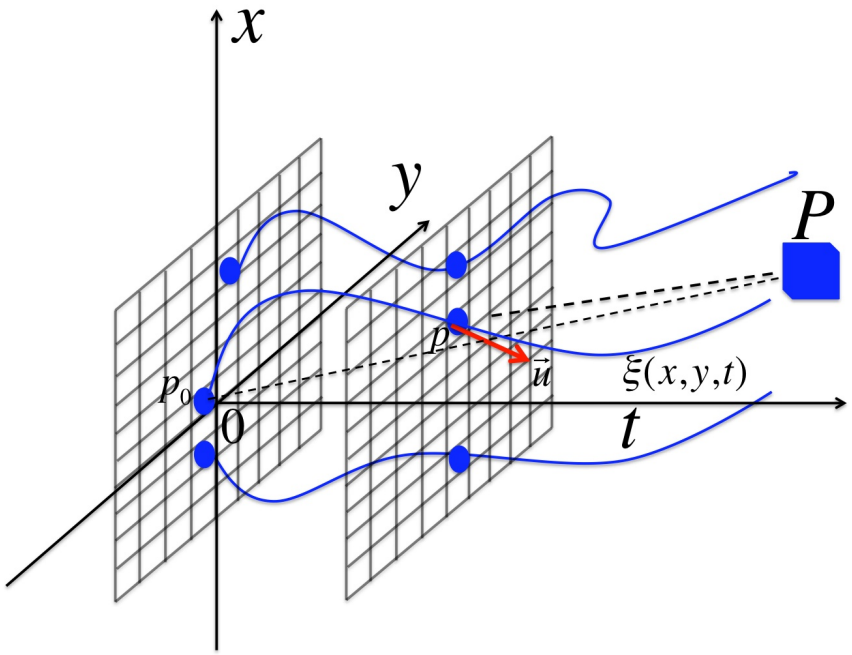


Figure 3.8: *Optical flow as a function of time.* Optical flow estimation is based on the assumption that intensities of image points, such as $\boldsymbol{p}_0$ and $\boldsymbol{p}$ (filled blue circles), projected from the same object point $\boldsymbol{P}$ remain invariant. These projected points form a profile curve, $\xi(x, y, t)$; several such curves are illustrated. The optical flow vector $\vec{u}$ at time t is the tangent vector of $\xi(x, y, t)$, as indicated by a red arrow.

From the definition of dense optical flow, we can realize that dense optical flow cannot be accurately computed unless there are enough duplicated visual patterns between two matched images. Therefore, its computation starts with intensity constancy model, which assumes that the intensity of image points projected from the same objects remain invariant. In Fig. 3.8, all projection points of an object point $\boldsymbol{P}$ at varying times(the filled blue circles along each curve) form a profile curve, $\xi(x, y, t)$ at $[0, t]$. Assuming $\boldsymbol{p}$ is a projection point in $\xi(x, y, t)$ at t, its optical flow vector, $\overrightarrow{u} = (u_x, u_y, u_t)$, is the tangent vector of $\xi(x, y, t)$(the red arrow). Let $\boldsymbol{p}_0 = (x_0, y_0)$

be the projection point at $t = 0$. Thus,

$$\xi = (x_0, y_0, 0) + \int_0^t \overrightarrow{u} d\tau = \begin{bmatrix} x_0 + \int_0^t u_x d\eta \\ y_0 + \int_0^t u_y d\eta \\ \int_0^t u_t d\eta \end{bmatrix} \tag{3.17}$$

and intensity constancy model is represented as two different formulations:

$$\frac{dI(\xi)}{dt} = 0 \Rightarrow$$
$$\begin{cases} \partial_x I u_x + \partial_y I u_y + \partial_t I u_t = 0 & \text{linear model} \\ I(x_0, y_0, t_0) = I(x_0 + \int_0^t u_x d\eta, x_0 + \int_0^t u_y d\eta, x_0 + \int_0^t u_t d\eta) & \text{nonlinear model.} \end{cases} \tag{3.18}$$

Eq. 3.18 is an essential equation in estimating dense optical flow.

However, Eq. 3.18 is an under-constrained problem to determine $\overrightarrow{u}$ because there are two unknown variables u_x and u_y in a single equation. During optical flow computation, u_t is frequently assumed to be 1. This issue is also called an aperture problem[99].

Additional constraints are needed to handle the aperture problem. One frequently used assumption is that optical flow vectors vary smoothly in the image domain except for the areas at depth discontinuities. Two types of smoothness constraints result in two different dense optical flow estimation approaches, including local and global methods. Local method assumes that optical flow vectors remain constant within a local image patch, such as a 5×5 image region. Global approach explicitly integrates intensity constancy model and smoothness constraint into an energy function over the entire image domain.

In this section, I elaborate on some representative algorithms in the image domain

with respect to these two approaches. Phase type methods[81, 71, 82, 75, 70] are out of scope and referred to in some survey papers[9, 13, 19].

Local Methods for Computing Dense Optical Flow

Local methods subdivide an image into several regular image regions, and the combination of the estimated optical flow in each image region forms the final dense optical flow.

Lucas-Kanade[148] proposed the following equation to estimate dense optical flow within an image region.

$$E(\overrightarrow{u}) = G(x, y; \sigma_w^2) * (\partial_x I u_x + \partial_y I u_y + \partial_t I)^2 \tag{3.19}$$

Here, $G(x, y; \sigma_w^2)$ is a Gaussian window function defined over the image region. In terms of intensity constancy assumption in Eq. 3.18, optical flow vectors $\overrightarrow{u}$ should make $E(\overrightarrow{u})$ achieve minimum. Differentiating Eq. 3.19 with respect to u_x and u_y leads to

$$\begin{pmatrix} G(;\sigma_w^2) * (\partial_x I)^2 & G(;\sigma_w^2) * (\partial_x I \partial_y I) \\ G(;\sigma_w^2) * (\partial_x I \partial_y I) & G(;\sigma_w^2) * (\partial_y I)^2 \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} = - \begin{pmatrix} G(;\sigma_w^2) * (\partial_x I \partial_t I) \\ G(;\sigma_w^2) * (\partial_y I \partial_t I) \end{pmatrix} \tag{3.20}$$

If the inverse of left matrix exists, the flow vector $\overrightarrow{u}$ could be computed. Alternatively, the value of Eq. 3.19 is minimized by eigen-analyzing a spatial-temporal linear structure tensor[15, 140].

Local methods based on Eq. 3.19 include the following constraints:

- Optical flow remains invariant within small image regions.
- Optical flow is piece-wise within an image region[19, 20, 21]
- Non-quadratic estimators, such as Lorentzian function[101, 78], are used to

reduce outlier influence in estimating optical flow.

- Optical flow is constant at variant sizes of image regions[164, 29, 140, 234].
- Optical flow undergoes affine visual motion[197, 196, 15, 20, 64].

However, all these constraints assume optical flow vectors are constant within an image region. This condition does not hold in colonoscopy images, especially near colon folds. In addition, nearly all local methods cannot guarantee optical flow is globally smooth and fully dense, eg., image regions having insufficient texture information. Consequently, local methods are not employed to estimate dense optical flow.

Global Methods for Computing Dense Optical Flow

Global methods have elegant properties, which address two issues of local methods: partial optical flow results and motion boundary over-smoothing. Global methods have been extensively studied to estimate both small and large visual motion.

Small Visual Motion Small visual motion approaches are used when we have successive image frames, and two frames only have small pixel shifts. Small visual motion is computed by explicitly integrating intensity constancy and smoothness constraints into a single global energy function.

All these methods can be summarized as a general energy equation like Eq. 3.1

$$E(\overrightarrow{u}) = \iint_{(x,y)\in\mathbb{R}^2} (\underbrace{F(\overrightarrow{u}\mathbf{J}_D\overrightarrow{u}^T)}_{\text{Data term}} + \alpha \underbrace{S(\nabla u_x \mathbf{J}_S \nabla u_x^T + \nabla u_y \mathbf{J}_S \nabla u_y^T)}_{\text{Smoothness term}}) dxdy \qquad (3.21)$$

where $\mathbf{J}_D$ is called *motion tensor*[30]. $\mathbf{J}_S$ is a *diffusion tensor* to control the smoothing

process. The *Euler-Lagrange equation* of Eq. 3.21 is

$$\begin{cases} \underbrace{\partial_x F(\overrightarrow{u}\mathbf{J}_D\overrightarrow{u}^T)}_{\text{Data}} - \alpha\underbrace{\text{div}(\partial_{\nabla u_x} S(\nabla u_x \mathbf{J}_S \nabla u_x^T + \nabla u_y \mathbf{J}_S \nabla u_y^T))}_{\text{Smoothness}} = 0 \\ \underbrace{\partial_y F(\overrightarrow{u}\mathbf{J}_D\overrightarrow{u}^T)}_{\text{Data}} - \alpha\underbrace{\text{div}(\partial_{\nabla u_y} S(\nabla u_x \mathbf{J}_S \nabla u_x^T + \nabla u_y \mathbf{J}_S \nabla u_y^T))}_{\text{Smoothness}} = 0 \end{cases} \tag{3.22}$$

Eq. 3.22 can also be expressed in a diffusion-reaction equation,

$$\begin{cases} \partial_\tau u_x = \underbrace{\text{div}(\partial_{\nabla u_x} S(\nabla u_x \mathbf{J}_S \nabla u_x^T + \nabla u_y \mathbf{J}_S \nabla u_y^T))}_{\text{Diffusion}} - \frac{1}{\alpha}\underbrace{\partial_x F(\overrightarrow{u}\mathbf{J}_D\overrightarrow{u}^T)}_{\text{Reaction}} \\ \partial_\tau u_y = \underbrace{\text{div}(\partial_{\nabla u_y} S(\nabla u_x \mathbf{J}_S \nabla u_x^T + \nabla u_y \mathbf{J}_S \nabla u_y^T))}_{\text{Diffusion}} - \frac{1}{\alpha}\underbrace{\partial_y F(\overrightarrow{u}\mathbf{J}_D\overrightarrow{u}^T)}_{\text{Reaction}} \end{cases} \tag{3.23}$$

Eq. 3.23 shows that global optical flow computation is essentially a diffusion process, where optical flow vectors within textural regions gradually propagate into areas devoid of texture information. The diffusion tensor $\mathbf{J}_S$ steers the direction and intensity of the local diffusion, and the motion tensor $\mathbf{J}_D$ keeps estimated optical flow vectors fulfilling data constraints. A constant α balances data and smoothness constraints to achieve an optimal compromise.

There are two types of approaches to formulate the energy function associated with the key equation in Eq. 3.21, two image driven and two flow driven computational methods[30].

Image-driven methods The assumption behind image-driven approaches is that visual motion boundaries are located at image regions with large gradient magnitudes. Image-driven methods design the diffusion tensor $\mathbf{J}_S$ with respect to local image structures, rather than local optical flow patterns. After $\mathbf{J}_S$ is properly designed, it is imported into either Eq. 3.22 or Eq. 3.23 to compute optical flow field.

Horn[99] assumed that optical flow computation is a homogeneous diffusion process, mathematically represented as $\mathbf{J}_S = \mathbf{I}$. However, this method is prone to

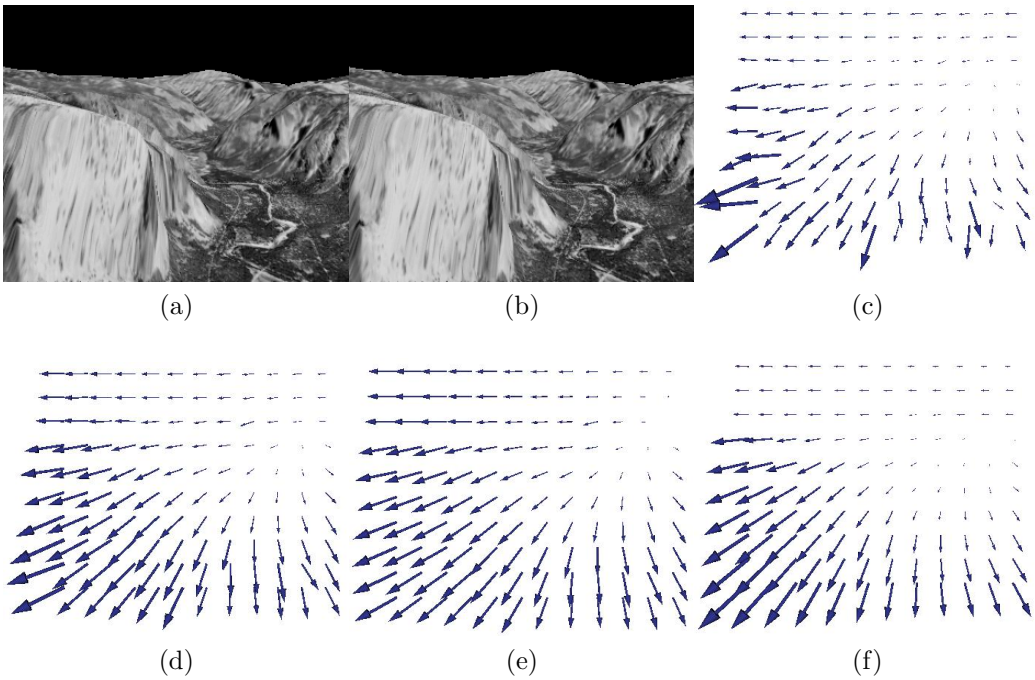


Figure 3.9: *Comparison of dense optical flow results using different motion and diffusion tensors as well as estimators.* (a) and (b) are two successive Yosemite frames; (c) diffusion tensor is an identity matrix (Horn's method[99]); (d) non-quadratic estimator is used for data and smoothness terms to compute flow-driven isotropic optical flow; (e) gradient constancy model is embedded into non-quadratic estimator; (f) nonlinear intensity and gradient constancy models are input into a non-quadratic energy function (Brox's method[27]). Optical flow results become smoother and motion boundaries are also well kept by exploiting sophisticated estimators as well as diffusion tensors.

causes over-smoothing at visual motion boundaries. Two successive Yosemite images (Fig. 3.9a and Fig. 3.9b[7]) are chosen to illustrate dense optical flow results. Here, the camera is moving toward the mountain. Fig. 3.9c gives results from Horn's method, where optical flow is not smooth because visual motion boundaries cannot be well kept. In order to avoid over-smoothing along image edges, Alvarez[3] developed a nonlinear isotropic diffusion tensor to penalize diffusion in image regions with large gradient. Similarly, Lai[119] prohibited optical flow diffusion at image contours[92]. However, nonlinear isotropic, image-driven methods can still over-smooth optical flow field along image edges. Nagel[165, 166, 164] developed an anisotropic image-driven

diffusion tensor, which is orientation variant in terms of local image gradients and their directions. The method prevents optical flow over-smoothing along image edges.

Although the problem of optical flow over-smoothing along image edges is preventable [165, 166, 164], image driven methods are not applicable to colonoscopy tracking because visual motion boundaries are not always located at image edges. For example, the homogeneous areas corresponding to the colon wall in the colonoscopy images have no edges, but visual motion is not constant because of varying depth values. Optical flow vectors at visual motion boundaries will be over-smoothed in these homogeneous areas because diffusion tensor is defined by image gradients.

Flow-driven Methods To prevent over-smoothing visual motion boundaries, a more reasonable strategy is the development of $\mathbf{J}_S$ directly related to optical flow vectors, which is also called the flow-driven method. After $\mathbf{J}_S$ is designed, optical flow can be accurately computed by using Eq. 3.22 or Eq. 3.23. Because large optical flow gradients correspond to visual motion boundaries, over-smoothing can be effectively moderated in flow-driven methods. The accuracy of dense optical flow is thus enhanced, improving the tracking results. Similar to image-driven methods, flow-driven approaches are also comprised of isotropic and anisotropic methods.

An optical flow computation method is isotropic flow-driven if diffusion tensor $\mathbf{J}_S$ is designed in terms of optical flow gradient magnitudes. Cohen[49, 117] suggested to use non-quadratic L_1 term to reduce the smoothness deviation and defined the smoothness term as

$$S(\nabla u_x \mathbf{J}_S \nabla u_x^T + \nabla u_y \mathbf{J}_S \nabla u_y^T) = \sqrt{\nabla u_x \nabla u_x^T} + \sqrt{\nabla u_y \nabla u_y^T} \tag{3.24}$$

Eq. 3.24[153, 58, 6] not only reduces the influence of the image noise, but also forces the diffusivity adaptable to local optical flow variance. This smoothness term results in an isotropic flow-driven method and avoids over-smoothing along visual motion

boundaries. Fig. 3.9d illustrates the flow results of Cohen's method. In comparison to 3.9c, optical flow is significantly improved, particularly in the bottom left corner. Alternatively, Schnorr replaced L_1 norm with a non-quadratic measurement $S(x^2) = \Psi(x^2) = \sqrt{x^2 + \epsilon^2}$, where ϵ is a small constant, for instance, 0.001. Based on this derivation, diffusion tensor, $\Psi(\nabla u_x \nabla u_x^T + \nabla u_y \nabla u_y^T)$, becomes rotation-invariant. Fig. 3.9f depicts optical flow results in terms of this rotation invariant tensor. Optical flow vectors vary smoothly, and they accurately represent the actual visual motion. Thanks to this useful rotation invariant property, many optical flow computation methods are developed from the isotropic flow-driven diffusion tensor, such as [228, 34, 35, 32, 33, 27, 175, 31, 30].

However, optical flow vectors are orientation variant along visual motion boundaries, whereas flow-driven isotropic methods might over-smooth optical flow field along certain directions. Anisotropic flow-driven method is needed to handle this issue. Optical flow computation method is anisotropic flow-driven if smoothness terms are dependent on optical flow gradients as well as their orientations. For instance, Weickert[227] proposed a direction-oriented smoothness term, which belongs to flow-driven anisotropic methods. It is stated as

$$S(\nabla u_x \mathbf{J}_S \nabla u_x^T + \nabla u_y \mathbf{J}_S \nabla u_y^T) = \text{trace}\Psi(\nabla u_x^T \nabla u_x + \nabla u_y^T \nabla u_y) \tag{3.25}$$

The smoothness term defined in Eq. 3.25 is mathematically demonstrated to achieve real flow-driven anisotropic optical flow computation.

Theoretically, flow-driven anisotropic methods generate the most accurate optical flow results. However, it is extremely time-consuming. Flow-driven isotropic methods can produce sufficiently accurate dense optical flow but with lower computation cost. It is the major reason flow-driven isotropic optical flow algorithms are more appropriate for colonoscopy tracking and are used in this dissertation.

Large Visual Motion All global methods I described thus far focus on estimating small visual motion. However, many computer vision tasks require computing significant visual motion, such as estimating large visual motion between a colonoscopy image pair after a sequence of blurry images are disregarded. There are three large displacement optical flow estimation techniques that I utilize in my approach: coarse-to-fine large displacement optical flow, Brox's large displacement optical flow and SIFT flow. In this section, I briefly describe and indicate both the similarities and differences between my approaches and these techniques.

Coarse-to-fine large displacement optical flow[4, 30] Coarse-to-fine is a common strategy to build an image pyramid for estimating optical flow. Calculus of variations computes optical flow from the coarsest pyramid and iteratively updates the computation to the finest level. Region flow computation employs coarse-to-fine strategy to estimate large visual motion between an image pair. Similarly, temporal volume flow computation estimates large visual motion by comparing two multi-resolution temporal volume pyramids.

Instead of using calculus of variations, Markov random field is chosen to minimize the energy function like Eq. 3.21 in estimating region flow. Although temporal volume flow computation also employs calculus of variations, it estimates visual motion between two temporal volumes rather than two images.

Brox's large displacement optical flow[28] Brox segmented the image pair into a set of image regions with uniform motion, and each region was represented as the SIFT[147] feature descriptor. A set of matched regions was determined by measuring the distance between two SIFT descriptors[147, 155, 214]. These region correspondences were then combined into Eq. 3.21 and the Euler-Lagrange equation was used to estimate large displacement optical flow.

Incremental egomotion estimation also applies this strategy to integrate sparse feature correspondences into Eq. 3.21 and uses the Euler-Lagrange equation to estimate

visual motion. However, instead of identifying feature matches through image segmentation, incremental egomotion estimation exploits a set of accurate SIFT feature correspondences from region flow. This strategy can effectively handle the influence of false sparse feature matches on the optical flow computation. In addition, the purpose of integrating sparse feature matches into the incremental egomotion estimation is the subdivision of large visual motion vectors for egomotion estimation, but not the estimation of large displacement optical flow.

SIFT flow[139] This method employs SIFT feature correspondences as the data term in Eq. 3.21 to densely estimate SIFT flow. Belief Propagation[68] is used to minimize the energy function because the data term consisting of SIFT feature correspondences is discrete.

Region flow shares the same idea and also employs Belief Propagation[68] to minimize the discrete energy function. Instead of using SIFT feature correspondences to formulate the data term, efficient normalized cross-correlation is used to measure the similarity between two image regions in region flow computation. Moreover, the purpose of region flow computation is not for image retrieval, but for accurate SIFT feature matching.

3.3 Egomotion Estimation

The previous two sections describe the computational theories needed to estimate optical flow. In this section, I review techniques to calculate egomotion from optical flow. Egomotion is the movement of the camera relative to the external world. We need to know the egomotion in order to accurately determine positions and orientations of the colonoscope camera in the colon.

I first describe the basic mathematical formulation between egomotion and optical flow, and then derive the governing equations for egomotion determination. Let an object point $\boldsymbol{P} = (X, Y, Z)$ be in a camera-centered coordinate as depicted in

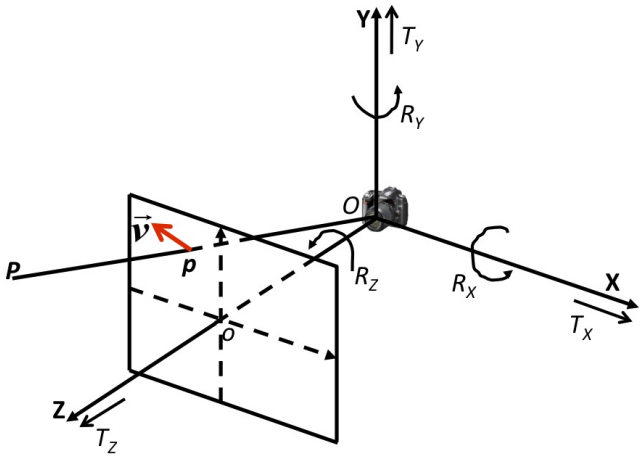


Figure 3.10: An instantaneous camera coordinate relating camera velocities and visual motion.

Fig. 3.10. $\boldsymbol{P}$ is projected into a point $\boldsymbol{p} = (x, y)$ in the image plane. $\overrightarrow{T} = (T_X, T_Y, T_Z)$ and $\overrightarrow{R} = (R_X, R_Y, R_Z)$ are, respectively, the translational and rotational velocities of the camera. The egomotion between the camera and the object point is projected into the image plane. It forms visual motion field $\overrightarrow{v} = (v_x, v_y, v_t)$ in the image plane, which is defined as

$$\overrightarrow{v}(x,y,t) = \begin{bmatrix} v_x(x,y,t) \\ v_y(x,y,t) \\ v_t(x,y,t) \end{bmatrix} = \begin{bmatrix} \frac{T_Z(t)}{Z}(x - \frac{fT_X(t)}{T_Z(t)}) + R_X(t)\frac{xy}{f} - R_Y(t)(f + \frac{x^2}{f}) + R_Z(t)y \\ \frac{T_Z(t)}{Z}(y - \frac{fT_Y(t)}{T_Z(t)}) + R_X(t)(f + \frac{y^2}{f}) - R_Y(t)\frac{xy}{f} - R_Z(t)x \\ \alpha_1 \end{bmatrix} \tag{3.26}$$

where α_1 is the temporal component and is usually assumed to be 1, and Z, the depth value. Eq. 3.26 is a core equation to bridge 3D camera motion and 2D visual motion in the image plane. However, the exact visual motion field is usually unknown, and optical flow is considered an approximation to the visual motion flow. In the other words, visual motion is regarded as the geometric prototype of optical flow. The essential problem of egomotion estimation is thus expressed as the search for $\overrightarrow{T}$ and $\overrightarrow{R}$ that minimizes

$$\iint_{(x,y)\in\mathbb{R}^2} \|\overrightarrow{v} - \overrightarrow{u}\|^2 dxdy \tag{3.27}$$

Minimizing Eq. 3.27 leads to two basic types of egomotion estimation algorithms, including simultaneous translation and rotation estimation as well as sequential translation and rotation estimation. This section surveys some typical algorithms based on this category. Part of their comparison can be found in Tian's work[209].

3.3.1 Simultaneous translation and rotation estimation

There are three basic approaches of simultaneous computation methods. The first method directly minimizes Eq. 3.27 with respect to camera translation and rotation parameters, such as Bruss's method[36]. Depth values are obtained either from depth sources, such as range cameras or the depth buffer of a graphics card, or from mathematical relation to motion parameters, derived from Eq. 3.26. Differentiating with respect to $\overrightarrow{T}$ and $\overrightarrow{R}$ yields a 6×6 linear system, and camera motion parameters can be simultaneously estimated. Later on, Adiv[1, 2] suggested a subdivision method to choose image regions with reliable optical flow fields to improve the accuracy of camera motion parameters and to reject outliers.

The second approach directly recovers motion parameters in avoidance of optical flow computation[98]. The essential idea here is to submit Eq. 3.26 for the linear model of Eq. 3.18 based on the assumption that $\overrightarrow{u} = \overrightarrow{v}$. It yields a sequence of linear equations to compute $\overrightarrow{R}$ and $\overrightarrow{T}$. Helferty[87] applied this method to bronchoscopy tracking and obtained promising tracking results.

Epipolar geometry[80, 65] is the third approach to directly estimate translation and rotation parameters. Given that $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are two image points projected from the same object point $\boldsymbol{P}$, the constraint can be written in the camera coordinate system

$$\boldsymbol{p}_1 \cdot (\overrightarrow{T} \times \mathbf{M}\boldsymbol{p}_2) = 0 \tag{3.28}$$

where $\mathbf{M}$ is a rotation matrix related to rotation velocity $\overrightarrow{R}$. $\overrightarrow{T}$ and $\overrightarrow{R}$ can be determined based on Eq. 3.28 if a set of feature correspondences is known.

These three approaches all assume that optical flow is accurate enough to be considered as the actual visual motion. Camera translation and rotation velocities can be safely estimated from a sequence of linear equations based on this assumption. However, all these approaches are sensitive to optical flow errors because the linear system combining translation and rotation parameters is highly unstable. I empirically and mathematically demonstrate this in Appendix G.

3.3.2 Sequential translation and rotation estimation

Sequential egomotion estimation methods separately compute translation and rotation parameters. The robustness of an egomotion estimation can be significantly enhanced over the simultaneous method because the search for camera translation and rotation parameters is reduced from a 6×6 linear system to two 3×3 linear systems. Decreasing search space reduces the ambiguity of egomotion estimation.

This separation is either based on the FOE, or other important, non-FOE optical flow features. Next, I describe these approaches along with their advantages and disadvantages.

FOE-Based Approaches

FOE-based approaches are egomotion estimation methods using the FOE to separate camera translation and rotation estimation. The FOE is defined as the intersection between the camera translation direction and an image plane.

FOE-based methods can accurately estimate translation and rotation parameters because the FOE encompasses the most stable camera motion information in the optical flow field. These methods employ different visual motion properties to determine the FOE, such as motion parallax, collinearity, etc. Several examples are described here:

- Heeger[86, 84, 85, 106] assumed that a set of optical flow vectors could be

detected in the image plane and the combination of these optical flow vectors are orthogonal to $\overrightarrow{T}$. Based on this property, the FOE can be determined. However, this method is very time-consuming if a dense optical flow field is used to detect the expected set of optical flow vectors and to determine the FOE. By contrast, if sparse optical flow is utilized, this method might cause estimation bias.

- Vitoria[52, 53] realized that FOE is located at the zero sum of the collinear optical flow vectors and used this property to determine the FOE. However, the search for collinear points involves lots of computation, and the sum of optical flow vectors amplifies estimation errors.

- Sundareswaran[203, 102] observed that the curl of optical flow field was only related to camera rotation parameters, and translation components of an optical flow field could be excluded. The FOE was thus determined according to the translation components of optical flow.

- The FOE can also be determined based on the assumption that flow differences between two points near depth discontinuities point to the FOE, due to motion parallax[181]. Chapter 4 improves this method to track consecutive colonoscopy images. The enhancement includes: 1) development of a covariance matrix to measure region confidence; and 2) use of multi-scale optical flow to estimate camera motion parameters.

Because the FOE is mathematically demonstrated to contain the most accurate and stable camera motion information in the optical field[218], camera translation and rotation parameters can be accurately computed. In terms of the FOE, tracking a long colonoscopy image sequence becomes possible.

Other Separation-Estimation Methods

Other approaches include egomotion estimation methods that use features in the optical flow field besides FOE to separate camera translation and rotation estimation. Various algorithms have used the following visual motion features:

- Tomasi[211, 212, 239] noticed that visual angle changes were only dependent on camera rotation. Therefore, camera rotation and translation can be separately computed by measuring visual angle changes.

- Prazdny[178, 179] derived a mathematical constraint independent on camera rotation parameter from a triple of image points, and rotation parameter is estimated from his mathematical equation. Once camera rotation parameter is known, it is substituted back into Eq. 3.26 to estimate camera translation parameter.

- Lim[133, 134] realized that the optical flow sum of two antipodal points is devoid of camera translation parameters. This property is very important for large field-of-view cameras i.e., range camera or endoscope camera.

These separation algorithms have similar benefits to FOE-based approaches in robustly estimating camera motion parameters. However, they are somehow still sensitive to optical flow errors. For instance, Tomasi's method requires selected feature points well distributed in the image domain, so as to build robust visual angles. Therefore, these types of separation methods are not used in my colonoscopy tracking system.

3.4 Summary

This chapter reviewed techniques related to visual motion computation and egomotion estimation in a visually-guided navigation system. Underlying computational

theories were first presented, including scale-space theory, calculus of variations, and Markov random field. After underlying computational theories were described, I discussed several representative algorithms related to visual motion computation and egomotion estimation. Visual motion computation included sparse and dense optical flow calculations. Egomotion estimation approaches were comprised of simultaneous translation and rotation estimation, as well as FOE-based and other sequential egomotion estimation methods.

Chapter 4 through 6 explore the described techniques to develop a robust colonoscopy tracking system. Chapter 4 investigates scale-space theory to determine optimized scales and to detect distinctive interest points. It also computes sparse and dense optical flows between consecutive colonoscopy images. Motion parallax is used to detect the FOE in the optical flow field and to separately estimate camera translation and rotation velocities.

When a colonoscopy video stream is interrupted by blurry image sequences, Markov random field is an important mathematical tool to compute region flow to measure significant visual motion between two images. I discuss this in chapter 5. Incremental egomotion estimation employs calculus of variations and scale-space theory to subdivide significant visual motion from region flow into a sequence of optical flow fields. Large egomotion between image pairs is estimated incrementally using the FOE-based egomotion estimation method on the optical flow sequence. Temporal volume flow, described in chapter 6, also uses calculus of variations and scale-space theory to estimate temporal volume flow for an image pair interrupted by blurry images. The accuracy of colonoscopy tracking failure recovery can be enhanced by using the selected image pair.

CHAPTER 4: CONTRIBUTION ONE – MULTI-SCALE OPTICAL FLOW

"Even a journey of a thousand miles begins with a small step."

– Chinese saying

In the previous chapter, I described the theoretical and technical background pertaining to a visually-guided navigation system. In this chapter, these techniques are utilized to tackle the first problem of a colonoscopy tracking system, Vis. co-alignment of consecutive optical colonoscopy(OC) and virtual colonoscopy(VC) images. Because OC images are lacking in visual cues, optical flow is chosen as a means of focusing on smaller features such as folds and polyps for visual motion computation. The tracking algorithm employs scale-space theory to search for optimal spatial-temporal scales. A set of sparse optical flow vectors is computed at the optimal scales to accurately represent image displacements between consecutive OC images. Dense optical flow is also computed at the optimal scales.

The proposed tracking system must be stable in estimating egomotion, in order to be useful for co-aligning OC and VC images. Focus of Expansion(FOE) encompasses the most accurate camera motion information in the optical flow field; therefore, FOE-based egomotion estimation is explored in this chapter. Motion parallax is investigated to determine the FOE, which is used to separately compute camera translation and rotation velocities. Finally, the position and orientation of the colonoscope are determined by integrating the estimated velocities. Straight and curved phantoms are designed to quantitatively validate the tracking accuracy of the proposed method. Five clinical colonoscopy image sequences are used to verify robustness and accuracy.

4.1 Problem Statement

The goal of this chapter is to track consecutive OC images and guide OC by co-aligning OC and VC images. Colonoscope's positions and orientations are required in order to achieve OC and VC co-alignment. The positions and orientations are determined by integrating all egomotion parameters between two consecutive colonoscopy images. Egomotion is the relative motion between a camera and the external world. To calculate egomotion, we need to calculate optical flow from the OC video stream. Accurate optical flow calculation requires a lot of stable visual cues in OC images, such as interest points.

However, colonoscopy images manifest a number of challenges in visual motion computation and egomotion estimation, mainly due to lack of stable visual cues and significant colon deformation. Visual cues consist of geometrical and texture discontinuities. Because the colon is a tubular structure, geometrical discontinuities rarely exist in OC images. Moreover, they contain a few texture discontinuities due to indistinct intensity variance of OC images. For these reasons, only a small amount of interest points can be detected near blood vessels in colonoscopy images, as illustrated in the left image of Fig. 1.4b.

Colon deformation also seriously affects colonoscopy tracking. In Fig. 4.1, when the colonoscope is inserted, the folds are originally presented as triangle shapes (Fig. 4.1a), and they gradually become inflated and convert into ellipse forms (Fig. 4.1b). These non-rigid deformations affect egomotion estimation algorithms because egomotion assumes that objects undergo rigid motion. In order to track colonoscopy images, egomotion estimation algorithms must be resistant to colon deformation.

Given the lack of prominent features in the colon, optical flow is chosen as the means of focusing on smaller features (folds, polyps) for visual motion computation. In this chapter, I propose a multi-scale optical flow based colonoscopy tracking al-

(a) (b)

Figure 4.1: *Two transverse colonoscopy images illustrating colon deformation.* A triangle fold in 4.1a expands to a ellipse fold in 4.1b because the colon is pushed by the colonoscope.

gorithm that combines sparse and dense optical flow techniques, resulting in a more robust and accurate egomotion estimation.

4.2 Optical Flow Based Colonoscopy Tracking Algorithm

Fig. 4.2 shows the framework for tracking consecutive colonoscopy images. Scale-space theory is used to determine *characteristic spatial-temporal scales* across multiple anisotropic scale representations for each OC image. A set of accurate sparse optical flow vectors is then computed at the characteristic scales. The chosen scales are applied to compute dense optical flow, which determines the FOE. FOE and sparse optical flow are jointly used to estimate camera rotation velocities. Camera translation velocities are then computed by eliminating camera rotation components from sparse optical flow and using depth values from a colon-like cylinder model. OC and VC images are manually co-aligned at $t = 0$ to determine the initial camera position and orientation. The position and orientation corresponding to the image being tracked at that time are computed by integrating the estimated camera translation and rotation velocities. The bottom right image of Fig. 4.2 illustrates the final colonoscopy tracking system by using camera position and orientation parameters,

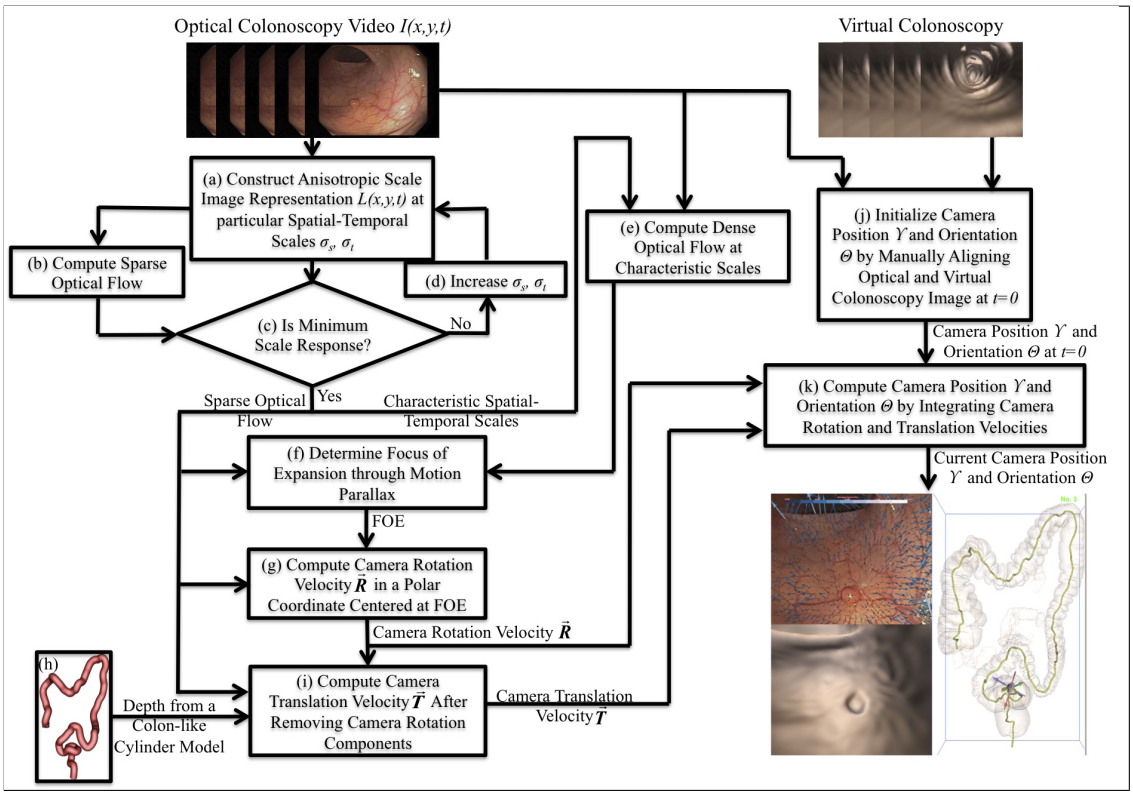


Figure 4.2: *Colonoscopy tracking algorithm.* The input consists of OC video stream and VC images. Scale-space analysis is performed to compute the characteristic spatial-temporal scales for each OC image, prior to computation of the sparse optical flow, using the Harris metric. These characteristic scales are used to determine the dense optical flow, which is then used to compute the FOE. FOE and sparse optical flow are used to determine the camera rotation velocity. After removal of camera rotation velocity from the optical flow field, camera translation velocity is determined, using depth values from a colon model. Camera positions and orientations are computed by integrating estimated camera velocities, and are transformed into CT volume coordinates to adjust the VC camera, illustrated in the bottom right.

including the optical image(top), the tracked VC image (bottom) and the camera location in the colon (right).

I will describe colonoscopy tracking framework under two general headings: multi-scale optical flow and FOE-based egomotion estimation. Multi-scale optical flow includes calculating sparse and dense optical flow. FOE-based egomotion estimation consists of FOE determination from dense optical flow as well as camera rotation velocity computation followed by camera translation velocity calculation based on the FOE.

4.2.1 Multi-scale Optical Flow

This section first describes a multi-scale selection metric to compute sparse and dense optical flows and then validates their accuracy on some VC images because their ground-truth optical flows are known.

Algorithm Description

Multi-scale optical flow computation encompasses calculating sparse and dense optical flow.

Anisotropic Gaussian scale space: Nonlinear, anisotropic Gaussian scale space is constructed over an OC video stream. Anisotropic Gaussian scale space representation $L : \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R}^2_+ \rightarrow \mathbb{R}$[122] is built by convolving the OC video stream $I(x, y, t)$ with an anisotropic Gaussian function.

$$L(x, y, t; \sigma_s^2, \sigma_t^2) = I(x, y, t) * G(x, y, t; \sigma_s^2, \sigma_t^2) \tag{4.1}$$

where

$$G(x, y, t; \sigma_s^2, \sigma_t^2) = \frac{e^{\left(\frac{-(x^2+y^2)}{2\sigma_s^2} - \frac{t^2}{2\sigma_t^2}\right)}}{\sqrt{(2\pi)^3 \sigma_s^4 \sigma_t^2}} \tag{4.2}$$

and the semicolon in $G(x, y, t; \sigma_s^2, \sigma_t^2)$ implies that the convolution is performed only over x, y, t, while σ_s^2 and σ_t^2 are spatial and temporal scale parameters. In the implementation, anisotropic Gaussian scale space is built on a group of images.

Temporal scale is critical in optical flow computation because 1) intensity constancy model defined in Eq. 3.18 involves a temporal derivative calculation; 2) a colonoscope moves unevenly during a colonoscopy procedure, which results in various sampling rates in the colonoscopy video stream. In order to preserve local details in the video stream as well as keep computational costs reasonable, a group of 11 colonoscopy images centered at the current colonoscopy image is selected to build

the anisotropic scale space. Anisotropic scale space is used for sparse optical flow calculation. The goal here is to search for the characteristic spatial-temporal scale parameters (σ_s, σ_t). They are initialized with $\sigma_s^2 = 0.5, \sigma_t^2 = 0.3$.

Sparse optical flow computation: Anisotropic scale space is now used in calculating sparse optical flow. Sparse optical flow computation involves two main components: interest point detection and matching. The Harris matrix is frequently used in interest point detection and is defined in the spatial-temporal scale space as

$$\mathbf{J}(x, y, t; \sigma_s^2, \sigma_t^2) = G(x, y; \sigma_w^2) * \begin{pmatrix} (\partial_x L)^2 & \partial_x L \partial_y L \\ \partial_x L \partial_y L & (\partial_y L)^2 \end{pmatrix} \tag{4.3}$$

where $G(x, y; \sigma_w^2)$ is a Gaussian window function and L is defined in Eq. 4.1. Lucas-Kanade method[148] is used to match interest points and compute sparse optical flow $\overrightarrow{u} = (u_x, u_y)$

$$\begin{aligned} E(\overrightarrow{u}) &= G(x, y; \sigma_w^2) * \left[L(x, y, t; \sigma_s^2, \sigma_t^2) - L(x + u_x, y + u_y, t + 1; \sigma_s^2, \sigma_t^2)\right]^2 \\ &\approx G(x, y; \sigma_w^2) * \left[(\partial_x L)u_x + (\partial_y L)u_y + (\partial_t L)\right]^2 \end{aligned} \tag{4.4}$$

Sparse optical flow is computed by first detecting a set of interest points according to Eq. 4.3 and calculating the first derivative of Eq. 4.4 with respect to (u_x, u_y) of the detected interest points. By setting the first derivative to zero, a 2×2 linear system is obtained to compute sparse optical flow. The linear system is expressed as follows.

$$\begin{aligned} G(\cdot; \sigma_w^2) * \begin{bmatrix} (\partial_x L)u_x + (\partial_y L)u_y + (\partial_t L)(\partial_x L) \\ (\partial_x L)u_x + (\partial_y L)u_y + (\partial_t L)(\partial_y L) \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \Rightarrow G(\cdot; \sigma_w^2) * \begin{bmatrix} (\partial_x L)^2 & (\partial_x L \partial_y L) \\ (\partial_x L \partial_y L) & (\partial_y L)^2 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} &= \begin{bmatrix} G(\cdot; \sigma_w^2) * (\partial_t L \partial_x L) \\ G(\cdot; \sigma_w^2) * (\partial_t L \partial_y L) \end{bmatrix} \end{aligned} \tag{4.5}$$

Spatial-temporal scale selection: Accurate sparse optical flow exhibits maximum variance in the spatial domain while maintaining a minimum difference along the temporal direction. Based on this, I combine Eqs. 4.3 and 4.4, and propose the following scale selection metric to search for characteristic spatial-temporal scales in computing sparse optical flow,

$$N(x,y,t;\sigma_s^2,\sigma_t^2) = \frac{G(x,y;\sigma_w^2) * \left[L(x,y,t;\sigma_s^2,\sigma_t^2) - L(x+u_x,y+u_y,t+1;\sigma_s^2,\sigma_t^2)\right]^2}{\sqrt{|C(x,y,t;\sigma_s^2,\sigma_t^2)|} + 1.0} \quad (4.6)$$

where

$$C(x,y,t;\sigma_s^2,\sigma_t^2) = \det(\mathbf{J}(x,y,t;\sigma_s^2,\sigma_t^2)) - \alpha \mathrm{Trace}^2(\mathbf{J}(x,y,t;\sigma_s^2,\sigma_t^2)) \quad (4.7)$$

and $\alpha = 0.04$[79].

Converting the numerator through Taylor expansion, Eq. 4.6 is transformed into

$$N(x,y,t;\sigma_s^2,\sigma_t^2) \approx \frac{G(x,y;\sigma_w^2) * \left[(\partial_x L)u_x + (\partial_y L)u_y + (\partial_t L)\right]^2}{\sqrt{|C(x,y,t;\sigma_s^2,\sigma_t^2)|} + 1.0} \quad (4.8)$$

The numerator in Eq. 4.8 represents the similarity between corresponding interest points, while the denominator measures how distinct the interest points are in their local neighborhoods. Good corresponding feature points should make the numerator(temporal difference) as small as possible and the denominator(spatial distinctiveness) as large as possible. Thus, the smaller the response of $N(x,y,t;\sigma_s^2,\sigma_t^2)$, the better the match. Another critical property of Eq. 4.8 is that it is invariant to scale changes and characteristic spatial-temporal scales can be determined if Eq. 4.8 also attains the minimum. In Eq. 4.8, its numerator consists of multiple derivatives including $\partial_x^2, \partial_y^2, \partial_t^2, \partial_x\partial_y, \partial_x\partial_t$ and $\partial_y\partial_t$, while the denominator is the root of derivative combination of $\partial_x^4, \partial_y^4$ and $\partial_x^2\partial_y^2$. The derivative order is two in both numerator and denominator. Therefore, the scale response of Eq. 4.8 remains invariant to scale changes and is the basis for spatial and temporal scale selection.

Scale response $N(x, y, t; \sigma_s^2, \sigma_t^2)$ is computed by substituting $\overrightarrow{u}$ for Eq. 4.8. Assume the current scale index is k and the finer scale index is $k-1$. A set of sparse optical flow vectors and scale responses are known in the finer spatial-temporal scales, $(\sigma_s^{k-1}, \sigma_t^{k-1})$. The current spatial and temporal scales (σ_s^k, σ_t^k) are set as $(\sqrt{2}\sigma_s^{k-1}, \sqrt{2}\sigma_t^{k-1})$ to compute the second set of sparse optical flow vectors and scale response. The factor $\sqrt{2}$ has been experimentally demonstrated for constructing a smooth multi-scale space without losing important image structures[188]. The third set of sparse optical flow vectors as well as scale response are determined again in the coarser level $(\sigma_s^{k+1}, \sigma_t^{k+1}) = (\sqrt{2}\sigma_s^k, \sqrt{2}\sigma_t^k)$. Therefore, we obtain three scale response values. If the scale response value of the current scale level is a local minimum in comparison with the finer and coarser scale response values, then the current spatial-temporal scales are optimal with respect to sparse optical flow computation. If the scale response value is not the local minimum, set the coarser scales as the current scales, and repeat response computation and comparison until the current scale response value is the local minimum.

Algorithm 1 illustrates the pseudo code and parameter details on scale selection and sparse optical flow computation.

Algorithm 1: Multi-scale sparse optical flow computation

Input: Video stream $I(x, y, t)$.
Output: Sparse optical flow vectors (u_x, u_y) and characteristic spatial-temporal scales σ_s^2 and σ_t^2.

```
begin
    Initialize σ_s^2 = 0.5 and σ_t^2 = 0.3;
    while (u_x, u_y, σ_s^2, σ_t^2) ∉ arg min_{u_x,u_y;σ_s^2,σ_t^2} N(x, y, t; σ_s^2, σ_t^2) do
        begin
            Construct anisotropic scale space L(x, y, t; σ_s^2, σ_t^2) (as per Eq. 4.1);
            Select interest points with the largest values C(x, y, t; σ_s^2, σ_t^2) (as per
            Eq. 4.7);
            Estimate optical flow vectors u⃗ = (u_x, u_y) (as per Eq. 4.5);
            σ_s^2 ← √2.0 σ_s^2, σ_t^2 ← √2.0 σ_t^2;
        end
end
```

Dense optical flow computation: The colonoscopy video stream is smoothed with the chosen spatial and temporal scales, and Horn's method[99] is used to compute the dense optical flow over the smoothed video stream at the optimal spatial-temporal scales. Although Horn's method is unable to avoid over-smoothing visual motion boundaries, the accuracy of dense optical flow is reasonable for egomotion estimation, and has low computational costs.

Example Demonstration

A VC image sequence was used to examine the effectiveness of the scale selection metric and optical flow computation. Scale selection results are illustrated in Fig. 4.3. Fig. 4.3(d) shows a response curve plotted as a function of the two spatial and temporal scale parameters. It can be seen that the response curve first decreases to a local minimum and then gradually increases. There are also three navigation images overlaid with ground truth sparse optical flow vectors (red) and the estimated sparse optical flow vectors(blue). Small green cubes indicate the positions of the chosen interest points. Fig. 4.3(a), corresponding to point A in Fig. 4.3(d), shows the results with fine spatial and temporal scales, where large vectors deviate from the ground truth because the scales are not sufficient to eliminate the noise or large intensity variation; because the chosen scales are too coarse, small areas with varying motion are merged. Therefore, small vectors diverge in Fig. 4.3(c), which corresponds to point C in Fig. 4.3(d). Spatial-temporal scales at the local minima are a means to balance between these two extremes, and as seen in Fig. 4.3(b) (point B in Fig. 4.3(d)), generate sparse optical flow vectors close to the ground-truth.

Fig. 4.4 illustrates dense optical flow results on the characteristic spatial-temporal scales. Fig. 4.4a gives the ground-truth dense optical flow in the current VC image. Dense optical flow that is computed in fine spatial-temporal scales is illustrated in Fig. 4.4b. There are many large optical flow vectors because small image structures

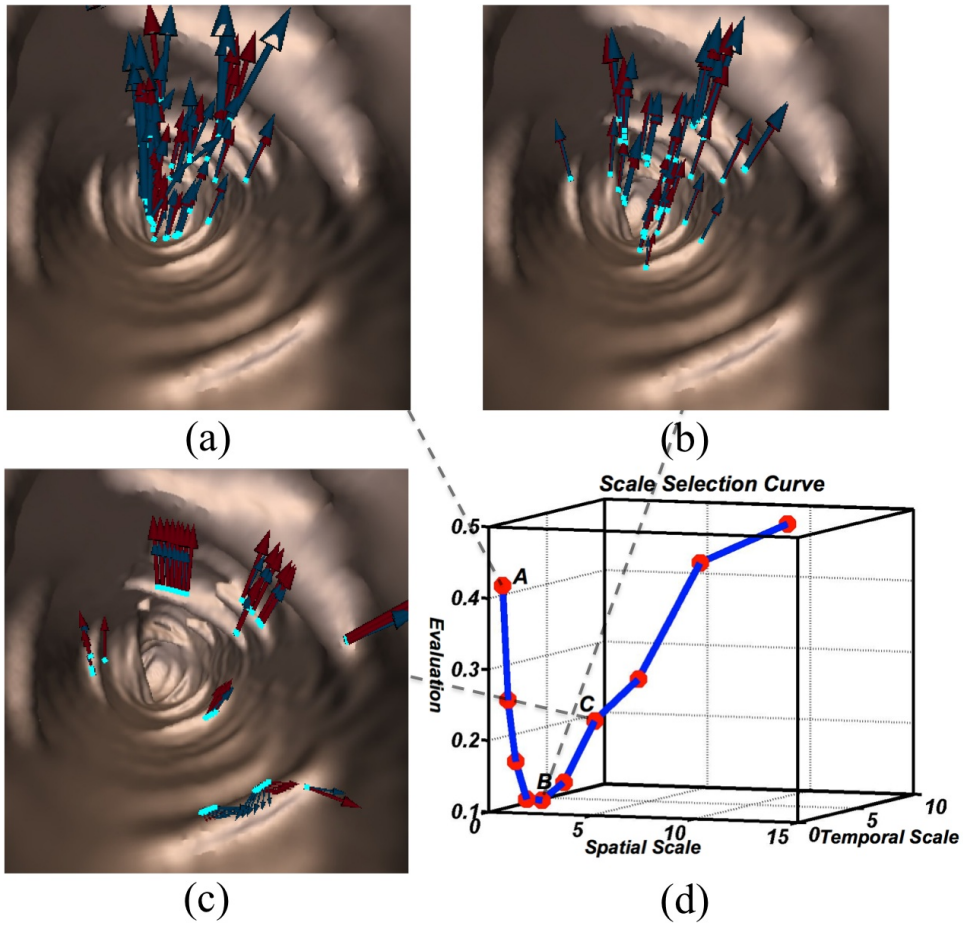


Figure 4.3: *Relationship between spatial-temporal scales and the scale selection metric.* Ground-truth optical flow vectors are in red and estimated flow vectors are in blue. Green cubes represent the selected feature points, (a) Results with fine spatial and temporal scales, (b) Results with characteristic scales, (c) Results with coarse scales, (d) The response curve between spatial-temporal scales and the scale metric; the scale values at points A, B and C correspond to images (a), (b), and (c) respectively.

are smoothed insufficiently. Fig. 4.4c shows dense optical flow computed in coarse spatial-temporal scales. Dense optical flow becomes smooth except in area marked A. In this circle, inverted optical flow vectors are generated because of improper smoothing along the temporal direction. Some optical flow vectors also disappear in area B because this area is homogeneous and fails to contain sufficient visual cues. Fig. 4.4d shows a reasonable dense optical flow computed in the characteristic spatial-temporal scales, and inverted optical flow vectors disappear in area marked A, with

exception of homogeneous area B.

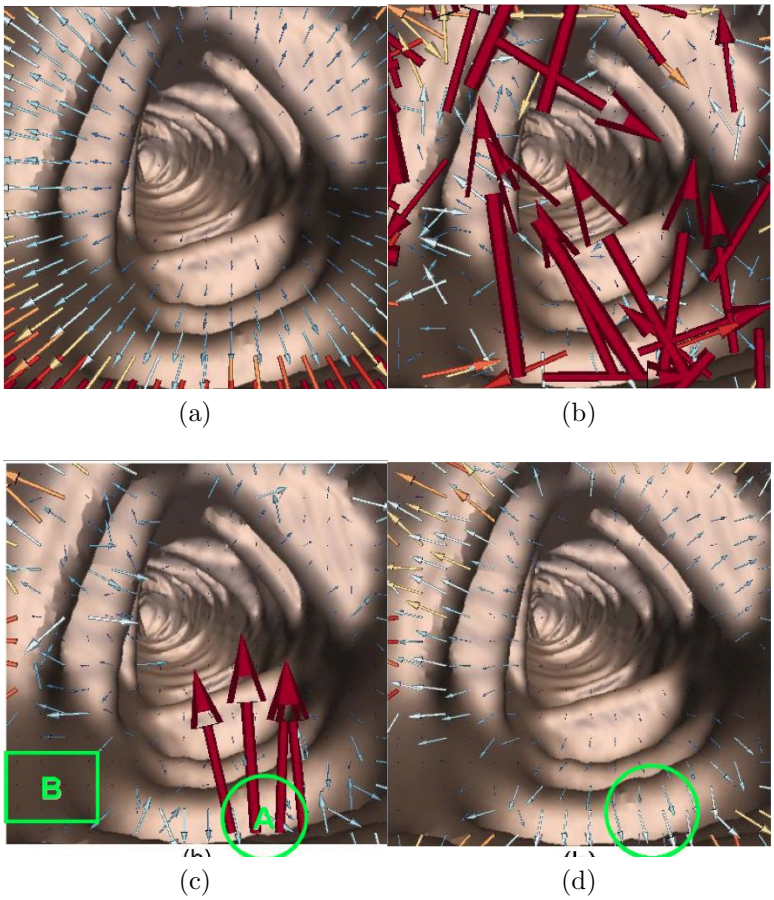


Figure 4.4: *Dense optical flow computed in different spatial-temporal scales.* (a) Ground-truth dense optical flow; (b) estimated dense optical flow in the fine spatial-temporal scales; (c) the coarse scales; (d) the characteristic scales;

4.2.2 FOE Based Egomotion Estimation

This section describes a FOE-based egomotion estimation algorithm by using sparse and dense optical flow, followed by a validation of a VC image sequence.

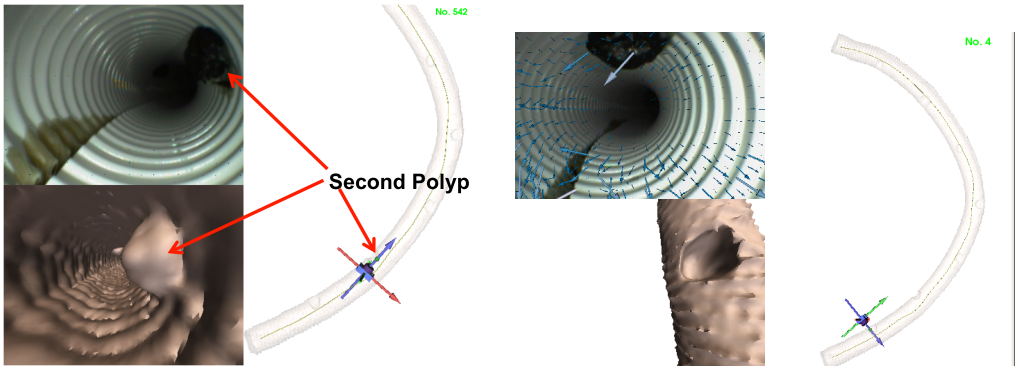
Algorithm Description

FOE-based egomotion estimation consists of determining the FOE, and computing camera rotation and translation velocities. Optical flow vectors at the FOE encom-

pass the most accurate egomotion information, and FOE is determined by camera translation velocities alone, independent of camera rotation velocities. Thus, the FOE can separate camera translation and rotation computation. This property is important in estimating egomotion because it significantly reduces the search space of camera translation and rotation parameters. FOE-based egomotion estimation can improve both accuracy and robustness over non-FOE based methods, such as Bruss and Horn's method[36].

Fig. 4.5 shows the comparison between FOE and non-FOE based methods on an 1000-image sequence of a colon phantom. This phantom is a bent tube with artificial polyps glued to its interior surface, and it is imaged using both CT and an endoscope. Fig. 4.5(a) shows the camera being tracked at frame 540 by using the FOE-based egomotion estimation method, while in Fig. 4.5(b), Bruss and Horn's method is shown in the 4th frame. It can be seen that the camera has moved out of the colon phantom (external view on the right shows the camera at the boundary of the colon phantom). In this sequence, FOE-based egomotion estimation was able to track the phantom images between the first and the second polyps. Note that the second polyp is displayed in the optical and the virtual images, as marked by red arrows in Fig. 4.5(a). Bruss and Horn's method fails to accurately estimate phantom images because 1) this method uses an estimator based on the least sum of squares, and the estimator is sensitive to optical flow errors; and 2) a 6×6 linear system from this estimator is also sensitive. The sensitivity is mathematically demonstrated in Appendix D.

FOE-based egomotion estimation is investigated to compute camera velocities in my colonoscopy tracking system. The intersection between the camera translation velocity $\overrightarrow{T}$ and the image plane is defined as Focus of Expansion when the camera moves towards objects; and it is called Focus of Contraction when the camera moves away from the objects; and the intersection is at infinity if $\overrightarrow{T}$ is parallel to the image



(a) (b)

Figure 4.5: *Comparison between my approach and Bruss and Horn's method on a phantom image sequence.* (a) Tracking results at frame 540, showing the camera reaching the second artificial polyp, (b) Bruss and Horn's algorithm results at frame 4, where the camera is out of the phantom. In Figs. 4.5a and 4.5b, the phantom images are displayed in the top left images, the tracked VC images are illustrated in the bottom left images, and the external views are depicted in the right images.

plane. The FOE[145] makes it possible to separate translation and rotation components from visual motion flow, the geometrical prototype of optical flow, because it is determined by the translation velocity $\overrightarrow{T}$ alone. This property is a key to estimating camera motion parameters.

To derive the FOE location in the optical flow field, visual motion vector $\overrightarrow{v}$ can be split into two vector components, $\overrightarrow{v}^T$ and $\overrightarrow{v}^R$, caused solely by camera translation and rotation velocities,

$$\overrightarrow{v} = \overrightarrow{v}^T + \overrightarrow{v}^R \tag{4.9}$$

where

$$\overrightarrow{v}^T = \begin{bmatrix} v_x^T \\ v_y^T \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{T_Z}{Z}\left(x - \frac{fT_X}{T_Z}\right) \\ \frac{T_Z}{Z}\left(y - \frac{fT_Y}{T_Z}\right) \\ 0 \end{bmatrix} \tag{4.10}$$

and

$$\overrightarrow{v}^R = \begin{bmatrix} v_x^R \\ v_y^R \\ 1 \end{bmatrix} = \begin{bmatrix} R_X\frac{xy}{f} - R_Y(f + \frac{x^2}{f}) + R_Z y \\ R_X(f + \frac{y^2}{f}) - R_Y\frac{xy}{f} - R_Z x \\ 1 \end{bmatrix} \tag{4.11}$$

It can be seen from Eq. 4.10 that the spatial components of $\overrightarrow{v}^T$ intersect at $(f\frac{T_X}{T_Z}, f\frac{T_Y}{T_Z})$, which is the FOE.

The motion parallax theory proposed by Longuet[145] can be used to estimate the FOE. Suppose there are two object points $\boldsymbol{P}$ and $\boldsymbol{Q}$ at distinct depths Z_1 and Z_2 but with the same projection point $\boldsymbol{p} = (x, y)$, where their visual motion vectors are $\overrightarrow{v}_1$ and $\overrightarrow{v}_2$; the difference between spatial components of $\overrightarrow{v}_1$ and $\overrightarrow{v}_2$ is given by

$$\frac{v_{1x} - v_{2x}}{v_{1y} - v_{2y}} = \frac{v_{1x}^T - v_{2x}^T}{v_{1y}^T - v_{2y}^T} = \frac{x - f\frac{T_X}{T_Z}}{y - f\frac{T_Y}{T_Z}} \tag{4.12}$$

because $v_{1x}^R = v_{2x}^R$ and $v_{1y}^R = v_{2y}^R$. Eq. 4.12 indicates that the direction of spatial visual motion vector difference $[v_{1x} - v_{2x}, v_{1y} - v_{2y}]$ of two adjacent points at depth discontinuity would point to FOE.

FOE determination: This step applies the motion parallax theory to determine the FOE in dense optical flow. I propose a subdivision method similar to[181] to search for the FOE, as illustrated in Fig. 4.6. The image plane is subdivided into rectangular regions. The estimated spatial optical flow vector difference $\Delta\overrightarrow{u} = (\Delta u_x, \Delta u_y) = (u_x(x_c, y_c) - u_x(x, y), u_y(x_c, y_c) - u_y(x, y))$ between the center point $\boldsymbol{p}_c = (x_c, y_c)$ and its neighbors $\boldsymbol{p} = (x, y)$ are tabulated. The covariance matrix, $\mathbf{C}(\Delta\overrightarrow{u})$ in each sub-region is formed,

$$\mathbf{C}(\Delta\overrightarrow{u}) = \begin{bmatrix} \sum(\Delta u_x)^2 & \sum \Delta u_x \Delta u_y \\ \sum \Delta u_x \Delta u_y & \sum(\Delta u_y)^2 \end{bmatrix} \tag{4.13}$$

The eigenvector(represented by the major axis of ellipses in Fig. 4.6b), corresponding to the largest eigenvalue, is the direction joining the center point of an image region to the FOE, based on principal component analysis. Eigen-ratio of this matrix $\delta = \|\lambda_{small}/\lambda_{large}\|$ represents the confidence of the computed direction, and is thresholded to select the sub-regions with high confidence. A line fitting procedure is

(a) (b) (c)

Figure 4.6: *Determining the FOE.* (a) Dense optical flow; (b) anisotropy of the covariance matrix defined in Eq. 4.13 in each grid, indicated by ellipses; the principal orientation within each region is indicated by the long axis of each ellipse and the confidence of this orientation is the inverse of the ratio between the minor and the major axes; (c) *FOE*, the intersection of the green lines, is determined by least-squares fitting the high confidence regions. Most of these regions are near depth discontinuities.

performed on the selected regions and the intersection of these lines (shown in green) is the estimated FOE in Fig. 4.6c. Note that most of the selected subdivision regions are near colon folds, which are areas of depth discontinuity.

Camera velocity computation: After the FOE is determined, camera rotation velocity can be first estimated through a polar coordinate centered at the FOE. Let $\overrightarrow{d} = [d_x, d_y]$ be a 2D vector joining the current feature point $\boldsymbol{p}$ to the FOE, and $\overrightarrow{d}_\perp = [d_{\perp x}, d_{\perp y}]$ is perpendicular to $\overrightarrow{d}$. Including the temporal component, two 3D vectors, $\overrightarrow{e} = [d_x, d_y, 0]$ and $\overrightarrow{e}_\perp = [d_{\perp x}, d_{\perp y}, 0]$, are defined in the spatial-temporal domain. Because $\overrightarrow{e}$ is parallel to $\overrightarrow{v}^T$ from Eq. 4.10, $\overrightarrow{v}^T \cdot \overrightarrow{e}_\perp = 0$. Camera translation velocity is eliminated as follows:

$$\overrightarrow{v}^R \cdot \overrightarrow{e}_\perp = \overrightarrow{v}^R \cdot \overrightarrow{e}_\perp + \overrightarrow{v}^T \cdot \overrightarrow{e}_\perp = [\overrightarrow{v}^T + \overrightarrow{v}^R] \cdot \overrightarrow{e}_\perp = \overrightarrow{v} \cdot \overrightarrow{e}_\perp = \overrightarrow{u} \cdot \overrightarrow{e}_\perp \quad (4.14)$$

Substituting Eq. 4.11 into Eq. 4.14, we obtain

$$\begin{bmatrix} R_X\frac{xy}{f} - R_Y(f+\frac{x^2}{f}) + R_Z y \\ R_X(f+\frac{y^2}{f}) - R_Y\frac{xy}{f} - R_Z x \\ 1 \end{bmatrix} \cdot \overrightarrow{e}_{\perp} = \overrightarrow{u} \cdot \overrightarrow{e}_{\perp} \tag{4.15}$$

The sparse optical flow is used to determine camera rotation velocity, due to its accuracy. Substituting all sparse optical flow vectors for Eq. 4.15 leads to a sequence of linear equations,

$$\begin{cases} \begin{bmatrix} R_X\frac{x_1y_1}{f} - R_Y(f+\frac{x_1^2}{f}) + R_Z y_1 \\ R_X(f+\frac{y_1^2}{f}) - R_Y\frac{x_1y_1}{f} - R_Z x_1 \\ 1 \end{bmatrix} \cdot \overrightarrow{e}_{\perp 1} = \overrightarrow{u}(x_1,y_1) \cdot \overrightarrow{e}_{\perp 1} \\ \vdots \\ \begin{bmatrix} R_X\frac{x_ny_n}{f} - R_Y(f+\frac{x_n^2}{f}) + R_Z y_n \\ R_X(f+\frac{y_n^2}{f}) - R_Y\frac{x_ny_n}{f} - R_Z x_n \\ 1 \end{bmatrix} \cdot \overrightarrow{e}_{\perp n} = \overrightarrow{u}(x_n,y_n) \cdot \overrightarrow{e}_{\perp n} \end{cases} \tag{4.16}$$

where n is the number of sparse optical flow vectors. Singular value decomposition is then applied to compute camera rotation velocity $\overrightarrow{R}$.

If OC and VC images are well co-aligned, I can safely assume that depth values in VC images are equal to those in optical colonoscopy images[180, 141]. This property is critical for determining camera translation computation because Eq. 4.19 requires the depth value of each interest point is known. In order to generalize the colonoscopy tracking algorithm, a cylinder-like colon model is used to approximate the colon and to generate depth values. The core of the colon model is the centerline from the virtual colon, and the radius is the average distance of all centerline points to the colon boundary. Rather than using patient specific parameters, an alternative is to use a radius that is averaged over patients, further generalizing the model, or a

model with varying radii that typically represent the different segments of the colon anatomy.

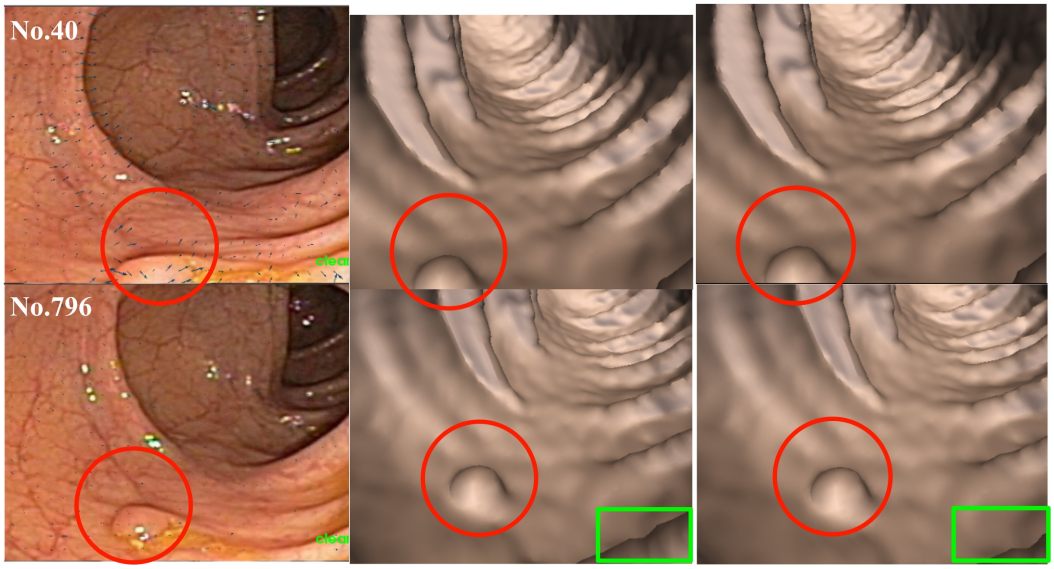
The sensitivity of the cylinder-like colon model is assessed through comparing the tracking results using depth values from the actual CT colon model with the cylinder-like colon model. Fig. 4.7a shows two snapshots from a sequence of 796 OC images containing a polyp (marked inside red circles) in the descending colon. I compare tracking results between the use of depth values from the virtual colon(Fig. 4.7b) and cylinder model(Fig. 4.7c). In both cases, tracking results are quite reasonable in frame 40 (top row). At frame 796, the only noticeable difference between the OC and VC images is the appearance of the fold, located in the bottom right corner(green squares). It is smaller using the cylinder model (bottom right image) versus the colon model(bottom center image). While a detailed study on this issue is beyond the scope of this work, experiments on additional clinical datasets illustrates no major differences or errors introduced through the use of depth values from a cylinder model.

I first substitute estimated camera rotation velocity $\overrightarrow{R}$ for Eq. 4.11 to compute $\overrightarrow{v}^R$. Next, $\overrightarrow{v}^R$, depth value Z from the colon model, and sparse optical flow $\overrightarrow{u}$ are substituted for Eq. 3.27 to compute the square of the difference between visual motion and optical flow components caused by translation only, ε,

$$\begin{aligned}\varepsilon &= \iint \left\| \overrightarrow{v}^T - (\overrightarrow{u} - \overrightarrow{v}^R) \right\|^2 dxdy \\ &= \iint \left\| \begin{bmatrix} \frac{T_Z}{Z}\left(x - f\frac{T_X}{T_Z}\right) \\ \frac{T_Z}{Z}\left(y - f\frac{T_Y}{T_Z}\right) \\ 0 \end{bmatrix} - (\overrightarrow{u} - \overrightarrow{v}^R) \right\|^2 dxdy \end{aligned} \tag{4.17}$$

Set $\frac{\partial}{\partial \overrightarrow{T}}\varepsilon = 0$ to minimize Eq. 4.17, and a 3×3 linear system is obtained,

$$\mathbf{B}\overrightarrow{T} = \overrightarrow{g} \tag{4.18}$$



(a) (b) (c)

Figure 4.7: *Comparison of tracking results by using depth values from a cylinder-like colon model.* To generalize the tracking algorithm, I use a cylinder like model derived from the 3D virtual colon. Results shown at two different frames. Left column illustrate the optical images, middle column shows results using depth from the virtual colon, and right column shows results using the colon model. A round polyp (red circle) is used as a reference to evaluate the tracking accuracy. Tracking results are comparable by using different depth sources, as the polyp is tracked well.

where

$$\mathbf{B} = \begin{bmatrix} -\iint \frac{f}{Z} dxdy & 0 & \iint \frac{x}{Z} dxdy \\ 0 & -\iint \frac{f}{Z} dxdy & \iint \frac{y}{Z} dxdy \\ -\iint \frac{xf}{Z} dxdy & -\iint \frac{yf}{Z} dxdy & \iint \frac{(x^2+y^2)}{Z} dxdy \end{bmatrix} \tag{4.19}$$

and

$$\overrightarrow{g} = \begin{bmatrix} \iint [u_x - v_x^R] dxdy \\ \iint [u_y - v_y^R] dxdy \\ \iint [x(u_x - v_x^R) + y(u_y - v_y^R)] dxdy \end{bmatrix} \tag{4.20}$$

Substituting sparse optical flow vectors to discretize Eq. 4.19 and Eq. 4.20, I obtain

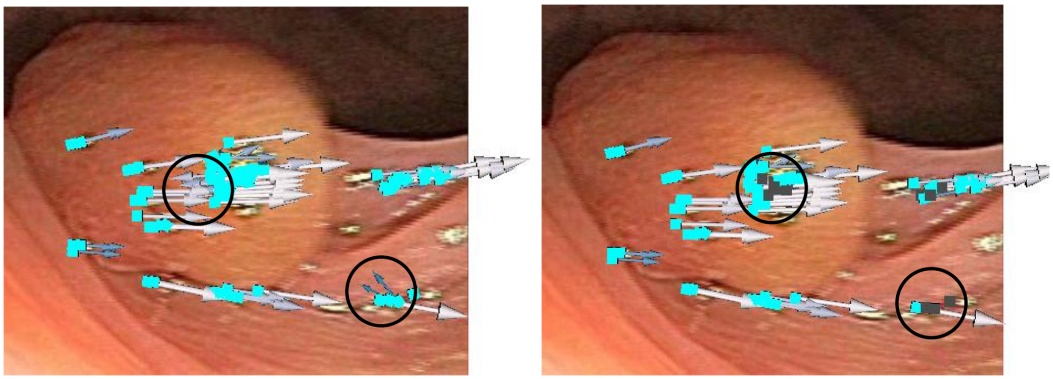
$$\mathbf{B} = \sum_{i=1}^{n} \begin{bmatrix} -f/Z_i & 0 & x_i/Z_i \\ 0 & -f/Z_i & y_i/Z_i \\ -x_i f/Z_i & -y_i f/Z_i & (x_i^2 + y_i^2)/Z_i \end{bmatrix} \tag{4.21}$$

$$\overrightarrow{g} = \sum_{i=1}^{n} \begin{bmatrix} u_x(x_i, y_i) - v_x^R(x_i, y_i) \\ u_y(x_i, y_i) - v_y^R(x_i, y_i) \\ x_i(u_x(x_i, y_i) - v_x^R(x_i, y_i)) + y_i(u_y(x_i, y_i) - v_y^R(x_i, y_i)) \end{bmatrix} \tag{4.22}$$

where Z_i is the depth value of the i-th feature point. Camera translation velocity $\overrightarrow{T}$ can be obtained through solving $\mathbf{B}\overrightarrow{T} = \overrightarrow{g}$.

A least sum of squares(LS) estimator is used to compute Eqns. 4.15 and 4.18. However, this estimator is unable to identify outliers inside the sparse optical flow vectors, and it employs all flow vectors to estimate camera velocities. Fig. 4.8a shows an example of features(in cyan) used by the LS method. A few optical flow vectors are pointing to the wrong (left) direction shown by two black circles, which are outlier vectors and have to be excluded from estimating camera velocities.

In order to remove outlier optical flow vectors, a least median of squares(LMS) estimator[185] is used to enhance camera rotation and translation computation. The LMS estimator iteratively analyzes and converges toward the main distribution of optical flow vectors while disregarding outliers. The application of the LMS method to camera motion computation is described in Appendix E. Fig. 4.8 shows the selected feature points of the LS and LMS methods within an OC image. The right image illustrates the features used by the LMS method, where the marked features in black indicate outliers that are discarded. The black circles illustrate features that are wrongly directed and are detected as outliers. They are excluded from camera motion estimation by using the LMS estimator, and the accuracy of camera velocities are



(a) Features selected by the LS method. (b) Features selected by the LMS method.

Figure 4.8: *Selected features (in cyan) used by the LS (a) and the LMS (b) methods on a tracked OC image in the transverse colon.* Each cube represents the selected feature point. Black features are outliers and black circles indicate areas with outliers, where some feature vectors are pointing to the wrong direction (towards left), which is incorrect.

improved.

Camera location and orientation computation: Colonoscopy tracking system assumes that OC and VC images are well co-aligned in the first OC image. Automatically initializing OC and VC cameras is infeasible at this time, and I have to manually adjust virtual camera to co-align virtual and OC images. Therefore, the initial camera position $\Upsilon(0) = [\Upsilon_X(0), \Upsilon_Y(0), \Upsilon_Z(0)]$ and camera orientation $\Theta(0) = [\Theta_X(0), \Theta_Y(0), \Theta_Z(0)]$ of the OC are known after manual co-alignment.

The current camera position $\Upsilon(t) = [\Upsilon_X(t), \Upsilon_Y(t), \Upsilon_Z(t)] : \mathbb{R}^+ \rightarrow \mathbb{R}^3$, and camera orientation $\Theta(t) = [\Theta_X(t), \Theta_Y(t), \Theta_Z(t)] : \mathbb{R}^+ \rightarrow \mathbb{R}^3$ are determined by

$$\begin{aligned} \Upsilon(t) &= \Upsilon(0) + \int_0^t \overrightarrow{T}^W(\tau)d\tau, \\ \Theta(t) &= \Theta(0) + \int_0^t \overrightarrow{R}^W(\tau)d\tau \end{aligned} \tag{4.23}$$

where $\overrightarrow{T}^W$ and $\overrightarrow{R}^W$ are corresponding velocities for $\overrightarrow{T}$ and $\overrightarrow{R}$ in the world coordinate,

and computed as

$$\begin{aligned} \overrightarrow{T}^W &= \mathbf{M}^T \overrightarrow{T} \\ \overrightarrow{R}^W &= \mathbf{M}^R \overrightarrow{R} \end{aligned} \tag{4.24}$$

$\mathbf{M}^T$ and $\mathbf{M}^R$ represent the affine transformation between the camera and world coordinates, and defined in Appendix. F.

$\Upsilon(t)$ and $\Theta(t)$ are used to drive the VC camera, and OC and VC images are co-aligned in the bottom image of Fig. 4.2.

Example Demonstration

A sequence of 807 OC images containing a polyp in the sigmoid colon was used for the colonoscopy tracking demonstration. I chose this dataset because it contains only a single polyp, as confirmed by the gastroenterologist in both OC and VC reports. In addition, the polyp is relatively large and easily recognized during evaluation. Fig. 4.9 shows four frames from this sequence. Column one illustrates the OC images and column two shows the tracked VC images. As the ground truth was not known, the virtual camera was interactively adjusted in order to match the corresponding OC image, and column three shows the resulting VC image. By measuring the differences in camera velocities that produced the images in columns two and three, I can get an estimate of the error from my tracking algorithm.

From frame 1 to 200, the colonoscope moves toward and rotates around the polyp, and the OC image is tracked accurately. From frame 200 to 400 (second row), the colonoscope is mostly stationary, focused on the polyp; from 200 to 246, the OC and corresponding VC images are nearly stationary due to the small motion of the colonoscope. From frame 242 to 322, video recording is suspended, thus the OC image is frozen. At frame 322, there is a significant motion change as recording is resumed. As the tracking system is currently unable to handle such large motion changes, the

tracking is stopped at frame 322 because motion changes exceed predefined thresholds (currently, 10mm for translation and 6° for rotation). Exceeding these thresholds is considered a tracking failure in the current system and will be further investigated in chapter 5. From frame 322 to 400, the colonoscope slightly rotates around the polyp and this motion is tracked precisely. After frame 400, the colonoscope moves closer to the polyp and rotates around it again. Tracked VC images (third row) show that the VC camera also moves toward the polyp. From frame 600 to 800, the colonoscope moves away from the polyp and has another significant rotation at frame 765. Note the translational error in the polyp location between the OC and VC images; this is due to colon stretching or flattening. The tracking is stopped again. Finally, these two big motion changes and colon deformation cause a 7° rotation error. At frame 807, the fold appears in the top OC image while it is only partly visible in the VC image. Other than that, the colonoscope is always tracked and the most important feature, polyp, is always observable in the OC and VC images. Table 4.1 illustrates the measured error between tracked and adjusted VC cameras at five particular frames along the sequence. The drift error gradually increases, but it is within *2mm* and the polyp is always tracked.

Table 4.1: *Accuracy evaluation using a clinical colonoscopy dataset.* Translation and rotation errors illustrated between the tracked and interactively adjusted VC cameras at five frames along the image sequence.

Frame	Translation(mm)			Rotation(degree)		
	X	Y	Z	X	Y	Z
100	0	0	0	0	0	0
300	-1.0	0	0.0	0	0	0
500	0.5	0	1.0	0	0	0
700	0.0	0.0	0.5	-2	0	0
800	0	0	-0.5	-7	0	0

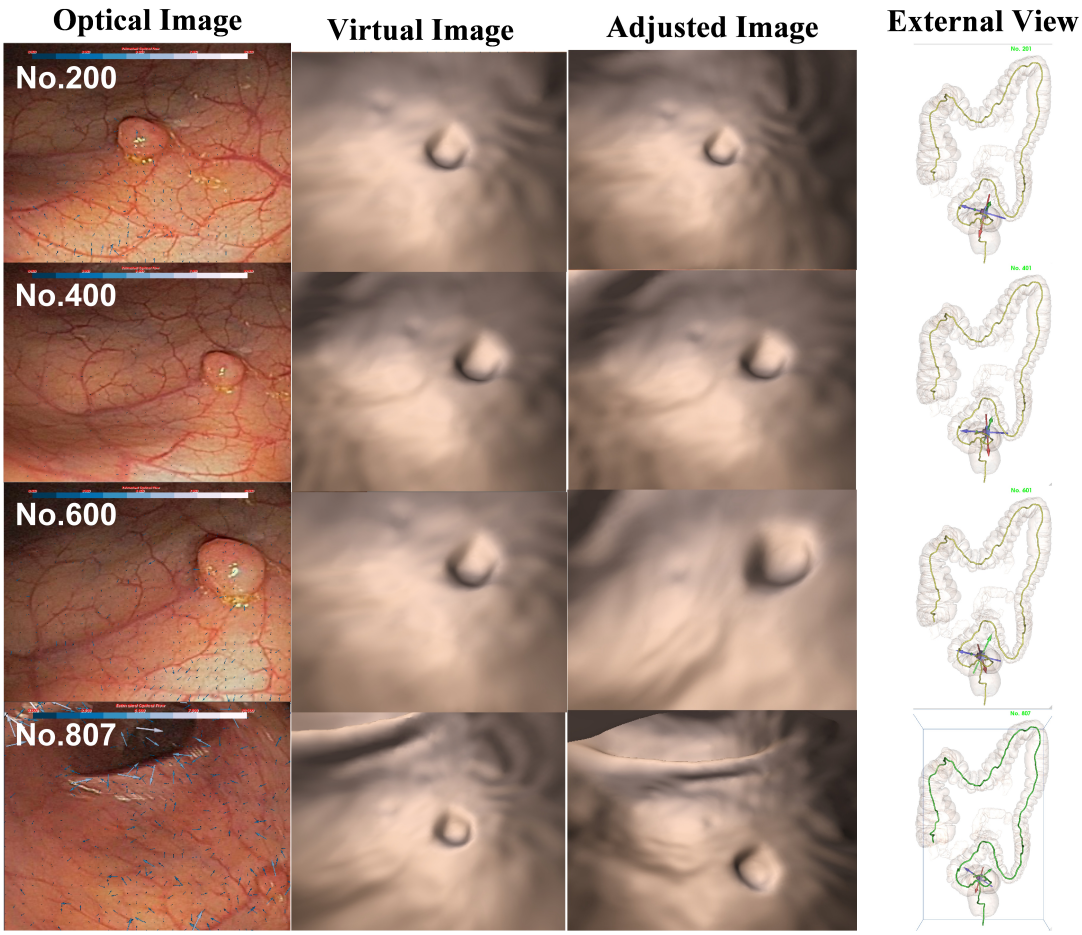


Figure 4.9: *Accuracy evaluation using a clinical colonoscopy dataset.* An 807-image sequence in the the sigmoid colon segment, containing a polyp. *Column 1:* OC images, *Column 2:* tracked VC images; *Column 3:* VC image(from column 2) after interactively adjusting virtual camera to match corresponding OC image (column 1), in order to measure errors between OC and tracked VC images, *Column 4:* Camera position within the virtual colon. The drift error is no more than *2mm* and the polyp is always tracked.

4.3 Phantom Validation

The colon phantom used in Fig. 4.5 can only qualitatively evaluate colonoscopy tracking accuracy by visually inspecting OC and VC polyps. These qualitative results are insufficient in understanding tracking accuracy because statistical analysis cannot be performed with respect to tracking accuracy. Designing phantom models with

known camera motion parameters is therefore extremely important to quantitatively analyze the tracking algorithm. In this section, straight and curved phantoms are developed for this purpose.

4.3.1 Phantom Design

Before I present the physical setup of the phantom experiments, let me first describe the experimental requirements.

1. Instantaneous camera motion velocities can be measured for the evaluation of tracking algorithms.

2. A colonoscope should navigate inside the phantom models in order to simulate an actual colonoscopy procedure.

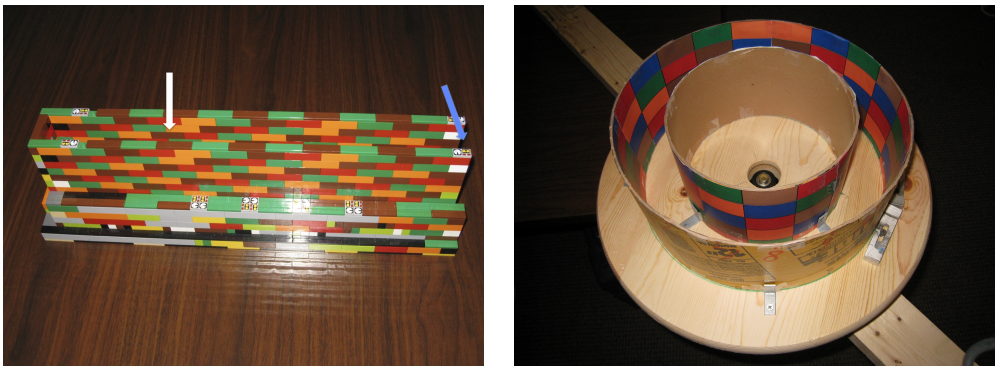
3. The phantom image must contain large amounts of image edges and corners for optical flow computation.

4. Phantom models must be reproducible whenever needed. As a result, the phantom design can contribute to endoscopy research.

5. Phantom experiments must be repeatable at the same conditions. The confidence of the tracking results can thus be statistically evaluated.

6. The colonoscope's movements inside the phantom models must be capable of being simulated in two general motions, along a straight line and in a curved path. Moreover, the velocities of the colonoscope in the phantom models should approximate the velocities that occur during an actual colonoscopy procedure.

Based on the requirements listed above, I designed two types of colon phantoms, including straight and curved phantoms, as shown in Fig. 4.10.



(a) Straight phantom (b) Curved phantom

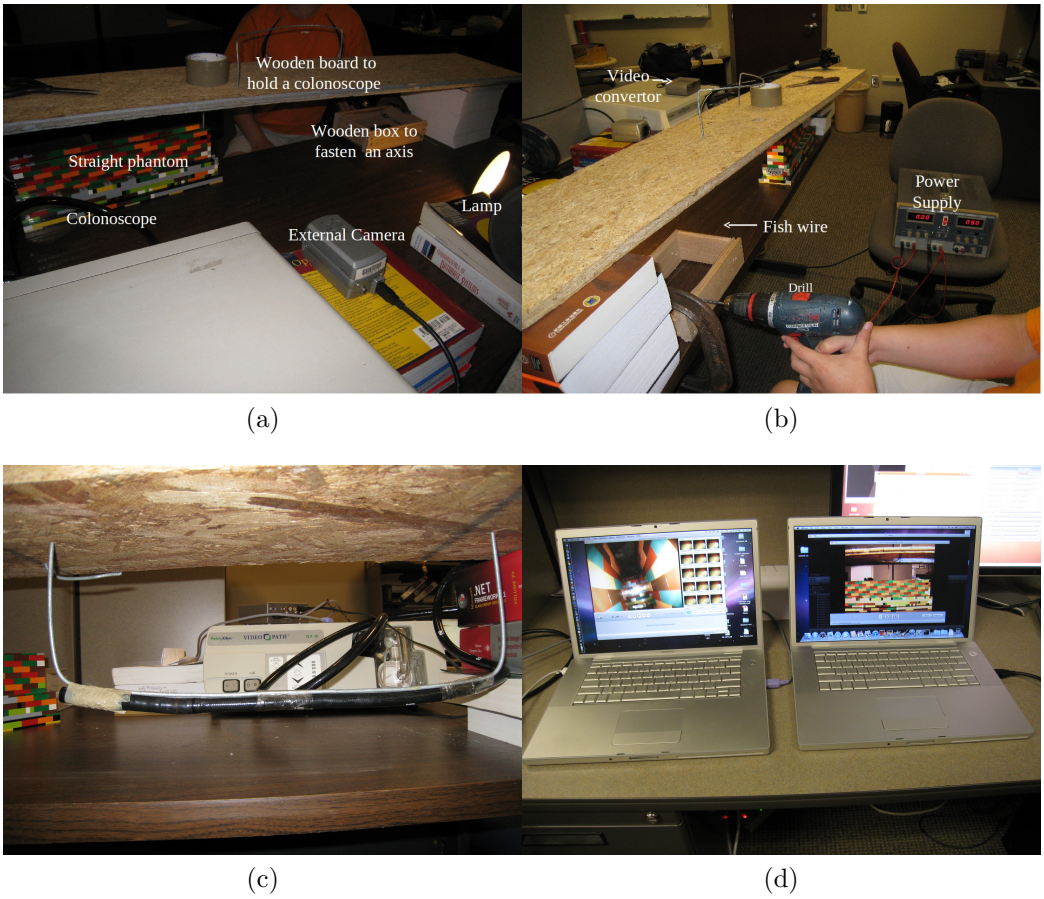
Figure 4.10: *Two types of colon phantoms.* In the straight phantom, one end is open and is used to insert the colonoscope(indicated by a blue arrow). The top(a white arrow) is also uncovered for the purpose of observing the colonoscope's movements.

Lego bricks[1] are selected to build a straight tunnel phantom because Lego products have several useful properties that easily fulfill the experimental requirements. First, they are uniform in size. Second, the straight model is made of different colored bricks, which yield many interest points and edges near the bricks' boundaries. Visual motion can be easily measured by identifying interest points. In addition, Lego bricks facilitate actual camera motion determination because the colonoscope's displacement can be visually measured as it passes a brick. Fig. 4.10a shows a straight-tunnel phantom made of 4×1 Lego bricks, where each Lego brick is 32mm×9mm×8mm. The interior of the straight-tunnel phantom is 105mm×32mm×384mm.

Because it is difficult to use Lego bricks to build a curved tunnel, two tubes made of cardboard with different radii were used, as illustrated in Fig. 4.10b. Their radii are 158.5mm and 102.5mm, respectively. The height of each tube is 125mm. In order to generate distinctive interest points for visual motion computation, papers printed with different colored squares are pasted inside the curved phantom(the inner walls of the large tube and the outer walls of the small tube). Here, the size of each colored square is $54mm \times 28mm$.

[1] `http://www.lego.com/`

After straight and curved phantoms are designed, a colonoscope is placed inside each phantom to collect datasets. For repeatability, the colonoscope in the phantom experiments is controlled by a motor. As it is more complex to manipulate the motor to precisely move the colonoscope at different speeds; straight and curved phantoms are instead driven by the motor, while the colonoscope is kept stationary during data collection.



(a) (b)

(c) (d)

Figure 4.11: (a,b) The straight phantom experiment setup; (c) the colonoscope's placement; (d) Two laptop computers that collect exterior and interior colonoscopy navigation images.

Straight phantom experiment. Fig. 4.11 illustrates the experimental setup for moving the straight phantom. It includes six main steps:

1. A long wooden board is placed on top of two stacks of books of equivalent height,

and a straight iron wire is installed on the board to place the colonoscope. Fig. 4.11c gives a close view of the colonoscope's placement. The colonoscope is firmly taped to the iron wire.

2. The straight phantom is placed under the wooden board, and the colonoscope in conjunction with the iron wire is inserted into the straight phantom shown in Fig. 4.11a.

3. A wooden box is also placed under the wooden board and is fastened to a table by a clamp, as illustrated in Fig. 4.11b. A steel axis is fixed inside this box, and one end of a fish wire is wound around the axis. The fish wire then passes through a small hole in the wooden box and the other end is connected to the straight phantom. Constant rotation of the axis can, therefore, move the straight phantom at a uniform speed.

4. An external video camera is placed on top of a stack of books, and it points to the straight phantom. A lamp is used to enhance the brightness and to improve the recording of the external video camera, as shown in Fig. 4.11a.

5. A drill rotates the axis. Its current and voltage are adjusted by a power supply as shown in Fig. 4.11b, thus controlling the speed of the drill.

6. Phantom images recorded from the colonoscope are analog. A video converter is used to transform the analog images into digital images. The digital images are then imported into the left laptop computer, shown in Fig. 4.11d. The images captured by the external video camera are input into the right laptop computer. A straight phantom trial is completed when the colonoscope exits the phantom.

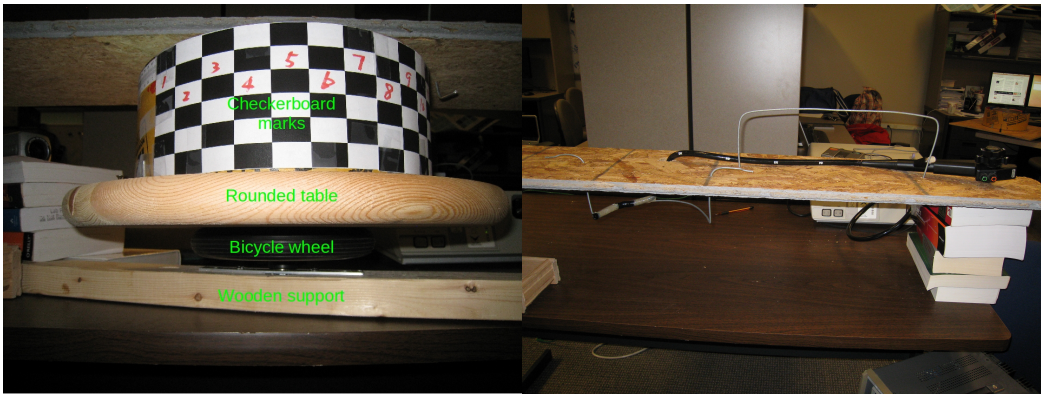
In the straight phantom experiment, the phantom is pulled by the fish wire at speeds of $10mm/sec, 15mm/sec$, and $20mm/sec$: three typical colonoscope speeds

used during a colonoscopy procedure. Let me explain how these three speeds are determined. A colonoscopy consists of an insertion phase (from sigmoid colon to cecum colon) and a withdrawal phase (from cecum colon to sigmoid colon.) The procedure lasts about half an hour. The actual colon examination is performed in the withdrawal phase. In the withdrawal phase, the main colonoscope's motion is the withdrawal. This phase also includes insertion, therapy, biopsy, and observation. The average pure withdrawal motion lasts about 6.5 minutes, based on clinical studies[8, 182]. However, even a 6.5-minute period includes adjusting time when blurry images appear, and the colonoscope is manipulated away from colon walls or fluid. The period of blurry images should be excluded from the withdrawal time calculation. JungHwan[173] applied image retrieval techniques to find that approximately 40% of colonoscopy images are blurry. Therefore, the colonoscope's average velocity is about $1500mm/(6.5 \times 60 \times (1 - 40\%)) = 6.4mm/sec$ because the colon is about $1500mm$ long. Considering that the colonoscope is not withdrawn at a constant speed, and is sometimes stopped, three speeds were chosen: $10mm/sec$, $15mm/sec$ and $20mm/sec$.

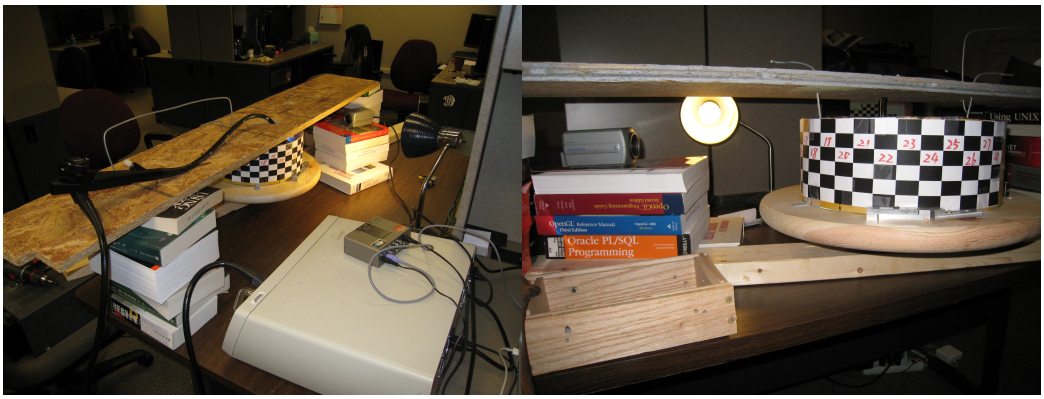
Twenty-five trials are collected for each individual speed level. Five trials are selected from these twenty-five trials based on two criteria: 1) the phantom's displacement divided by the time length of the trial approximates the speed within an error of $2mm/sec$; 2) the time difference between five selected trials is less than 0.3 seconds. Finally, fifteen exterior and interior straight phantom image sequences are collected at speeds of $10mm/sec$, $15mm/sec$, and $20mm/sec$(five at each speed.)

Curved phantom experiment. Fig. 4.12 shows the setup of the curved phantom experiment and involves five steps.

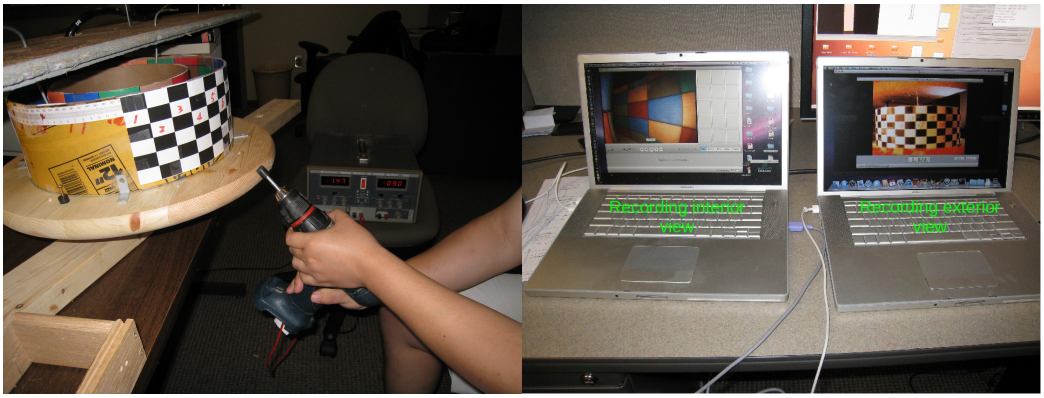
1. The curved phantom is first fixed on top of a turntable, and a bicycle wheel is installed under the turntable. The curved phantom, the turntable, and the bicycle wheel comprise the entire rotating apparatus, shown in Fig. 4.12a, and they share the same center. This bicycle wheel can produce smooth, slow rota-



(a) (b)



(c) (d)



(e) (f)

Figure 4.12: (a) The curved phantom setup; (b) the colonoscope's placement; (c) the colonoscope inside the curved phantom; (d) the external view recording; (e) curved phantom rotation; (f) two laptop computers that collect exterior and interior phantom images.

tion. Finally, the apparatus is fixed on top of a wooden board, which is fastened to the table by a clamp.

2. The colonoscope is fastened to a curved iron wire, to keep it stationary during phantom experiments, as illustrated in Fig. 4.12b.

3. The colonoscope and the curved iron are placed inside the curved phantom. An external video camera is also placed on top of a stack of books and pointed toward the curved phantom, as shown in Fig. 4.12c and Fig. 4.12d. A lamp is utilized to enhance the lighting for recording the external view. Papers printed in checkerboard patterns are wrapped around the outer wall of the large tube, facilitating ground-truth motion determination.

4. A drill controlled by the power supply is again used to motorize the spinning table shown in Fig. 4.12e. First, I determine what rotational speeds of the turntable will cause the colonoscope to translate at the three selected speeds. Supposing the colonoscope is placed at the medial axis of the curved phantom, the radius about the medial axis is $(158.5mm + 102.5mm)/2 = 130.5mm$ where $158.5mm$ is the radius of the big tube and $102.5mm$ is the radius of the small tube shown in Fig. 4.10b. Therefore, the length of the medial axis is $2\pi \times 130.5 = 820mm$. After one complete revolution of the spinning table, the colonoscope will traverse $820mm$ inside the curved phantom. In order for the colonoscope to be $10mm/sec$, the angular velocity of the turntable should be $\frac{360^\circ}{(819.54mm)/(10mm/sec)} = 4.4^\circ/second$. In order to achieve this slow speed, a drive system was designed to reduce the turntable's angular speed(The motor is unstable at slow speeds.) Fig. 4.12e illustrates this simple but effective speed reduction system. A small wheel of radius $0.6mm$ is attached to the end of the drill. This wheel then touches against the side wall of the turntable. Because the radius of the turntable is $240mm$, this drive system can reduce the speed $240mm/0.6mm = 400$ times. Therefore, the colonoscope can move at $10mm/sec$ while the drill can still rotate at $\frac{400 \times 4.4^\circ/second}{360^\circ} \approx$

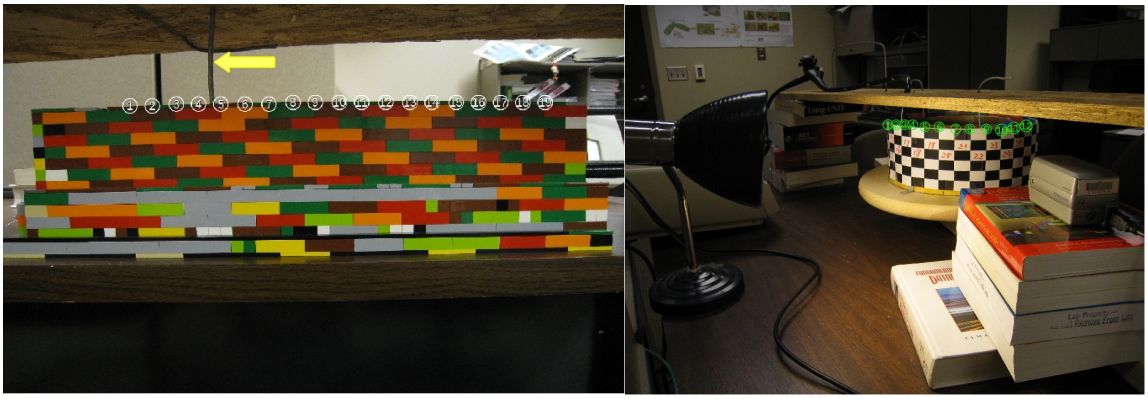
$5 revolutions/second$.

5. Exterior and interior colonoscopy navigation images are imported into two laptop computers, as illustrated in Fig. 4.12f.

Similar to straight phantom data collection, twenty-five trials are performed at each speed. Five trials are chosen based on the two criteria described above. Therefore, five exterior and five interior curved phantom image sequences are obtained at each of the three speeds.

4.3.2 Straight Phantom Results

Exterior phantom image sequences are used to determine the actual colonoscope motion, and interior image sequences are used to estimate the colonoscope's motion by the proposed tracking algorithm. Thus, the accuracy of the tracking algorithm can be analyzed by comparing the actual and estimated colonoscope motions. This section describes the determination of the ground-truth colonoscope motion from the external phantom image sequences in the straight phantom. It also describes accuracy of the estimated camera motion from the interior image sequences.



(a) (b)

Figure 4.13: *Marked points used for determining actual colonoscope motion in the straight and curved phantoms.* In (a), the vertical iron wire is indicated by a yellow arrow.

The initial portion of the external video sequences is excluded from ground-truth motion determination because the drill is in the acceleration phase. After the acceleration phase is eliminated, I select nineteen marked locations that are boundaries between Lego tiles and bricks, as shown in Fig. 4.13a. An image sequence in the straight phantom is employed to validate the tracking algorithm, if the vertical iron wire remains between No.1 and No.19 marked locations of the external phantom image. The distance between consecutive locations is $16mm$(half of one brick length), and the length from locations 1 to 19 is $16mm \times 18 = 288mm$. Because the drill rotates smoothly in the selected external phantom sequences, the colonoscope is assumed to move constantly between successive marked locations. The time required for the colonoscope to move between two marked locations is determined by finding two images in which the vertical iron wire arrives at these two locations. Then I measure the time difference between the two images. The ground-truth colonoscope velocities are determined by dividing the distance traveled, $16mm$, by the time used in traveling this distance. The time is calculated by counting the number of video frames to travel a half-brick length. The ground-truth colonoscope displacements are then calculated by accumulating ground-truth velocities.

After the ground-truth camera motion has been determined from the exterior phantom images, the corresponding interior video streams are used by the FOE-based egomotion estimation algorithm. Because the colonoscope camera has a strong fish-eye effect, colonoscopy images are distorted, affecting the tracking results. This section analyzes the effect of this distortion on the colonoscopy tracking algorithm. Matlab Toolbox[23] is used to calibrate the camera and to remove distortion from OC images. Fish-eye effect can thus be understood by comparing the tracking results on the same colonoscopy image sequences, with and without camera calibration.

A virtual straight model is created through VTK (Visualization Toolkit)[189] to generate depth values needed in camera translation estimation. The view angle of the

virtual camera is set to 65°, so that the VC images look very similar to OC images.

In order to adapt the FOE-based egomotion estimation algorithm to the current phantom setup, the algorithm is enhanced in the following two aspects. First, a weighted least-sum-of-squares estimator is used to highlight the feature points that are close to the camera and to reduce the influence of feature points far away. This weighting of feature points is because optical flow vectors close to the camera are more accurate. The weight of each feature point is defined as

$$f(Z) = \frac{1.}{1. + (Z/\alpha)^6} \tag{4.25}$$

where $\alpha = 100$ and Z is the depth value of the current feature point. Second, I remove some feature points in egomotion estimation when the estimated camera translation velocity from these points exceeds certain thresholds.

Fig. 4.14 shows the camera velocity curves on five straight phantom trials at a speed of about $10mm/sec$, where the blue band represents the ground-truth colonoscope motion, and the red and green bands represent the estimated camera velocity curves on the original and calibrated phantom image sequences (five each). The lower and upper curves of each band indicate minimum and maximum camera velocities of five trials. The solid center curve represents the average velocities. At about $10mm/sec$, average velocity error is less than $2mm/sec$ after 750 phantom images have been tracked. Maximum velocity error is less than $8mm/sec$ on both original and calibrated phantom image sequences. Table 4.2a presents the average, maximum, and minimum estimated camera velocity errors of each of five trials. Fig. 4.15 shows camera displacement curves at about $10mm/sec$. Average displacement error is less than $7mm$ on five original phantom image sequence, and it is less than $9mm$ on the calibrated image sequence. Maximum displacement error is less than $15mm$ on both original and calibrated image sequences. Table 4.2b gives the average, maximum, and

minimum estimated camera displacement errors of each of five trials.

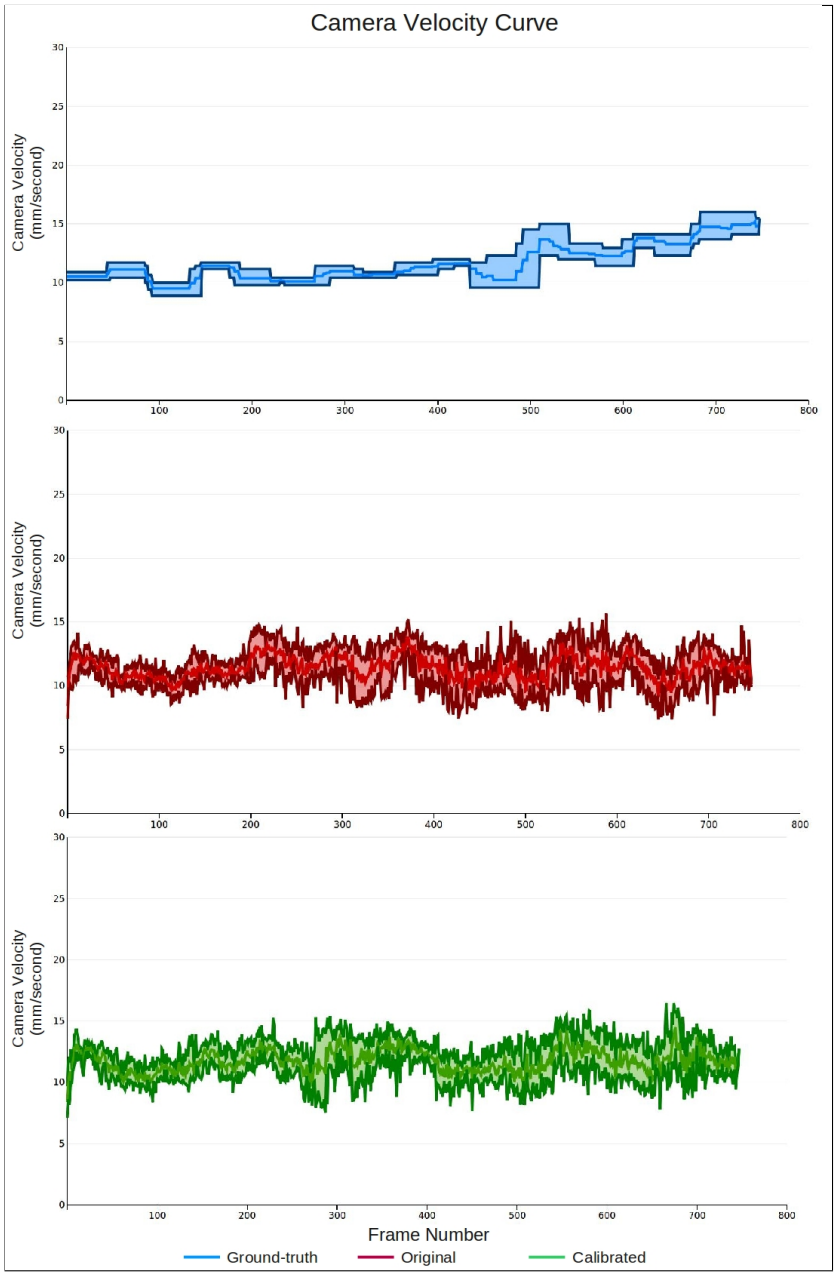


Figure 4.14: *Camera velocity curves at about* $10mm/sec$ *in the straight phantom.* The blue band represents the ground-truth camera velocities of five trials, and the red and green bands show the estimated velocities on the original and calibrated phantom image sequences(five each), respectively. The bottom and upper curves represent the minimum and maximum velocities of five trials in each band. The solid center curve represents the average velocities. Average velocity error is less than $2mm/sec$ on both original and calibrated phantom image sequences after 750 images have been tracked. Maximum velocity error is less than $8mm/sec$ on both sequences.

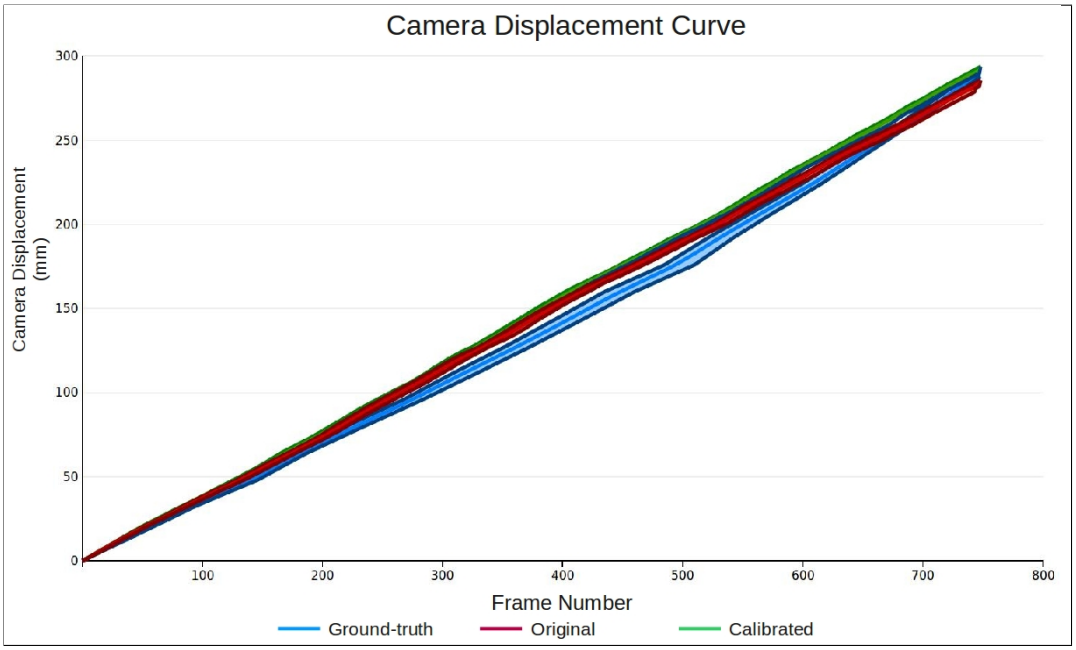


Figure 4.15: *Camera displacement curves at about* $10mm/sec$ *in the straight phantom.* The blue band represents the ground-truth camera displacements of five trials, and the red and green bands indicate the estimated displacements on the original and calibrated phantom image sequences, respectively. The bottom and upper curves represent the minimum and maximum displacements of five trials in each band. The solid center curve represents the average displacements of five trials. Average displacement error of five trials is less than $7mm$ on the original phantom image sequences after 750 images have been tracked, and it is less than $9mm$ on the calibrated image sequences. Maximum error of five trials is less than $15mm$ on both original and calibrated image sequences.

Fig. 4.16 and Fig. 4.17 illustrate the camera velocity and displacement curves at about $15mm/sec$, respectively, after 580 phantom images have been tracked. Average velocity error of five trials is less than $1.5mm/sec$ on both original and calibrated image sequences, and maximum velocity error is less than $7mm/sec$. Average displacement error of five trials is less than $3mm$ on the original phantom image sequence, and it is less than $5mm$ on the calibrated image sequences. Maximum displacement error of five trials is less than $6mm$ on the original image sequences and less than $8mm$ on the calibrated image sequences. Table 4.3a presents the average, maximum, and minimum estimated camera velocity errors of each of five trials. Table 4.3b shows their average, maximum, and minimum estimated camera displacement errors.

Table 4.2: The average, maximum, and minimum estimated camera **velocity** and **displacement** errors of original and calibrated straight phantom image sequences at about $10mm/sec$ after 750 images have been tracked.

(a) Camera velocity

Image sequence	Original Images(mm/sec)			Calibrated Images(mm/sec)		
	average	maximum	minimum	average	maximum	minimum
1	1.68	6.36	0.004	1.66	5.93	0.01
2	1.52	5.23	0.005	1.6	5.05	0.002
3	1.97	7.2	0.01	1.99	7.4	0.0007
4	1.6	6.06	0.003	1.63	5.61	0.0006
5	1.64	6.11	0.003	1.53	5.81	0.003

(b) Camera displacement

Image sequence	Original Images(mm)			Calibrated Images(mm)		
	average	maximum	minimum	average	maximum	minimum
1	5.92	11.89	0.0	8.7	12.33	0.0
2	4.26	8.72	0.0	7.46	9.87	0.0
3	6.55	10.44	0.0	8.58	11.91	0.0
4	6.5	12.19	0.0	8.4	13.3	0.0
5	6.87	14.65	0.0	8.21	14.57	0.0

Fig. 4.18 and Fig. 4.19 show the camera velocity and displacement curves at about $20mm/sec$, respectively. Average velocity error is less than $2mm/sec$ on both original and calibrated image sequences after 400 images have been tracked, and maximum velocity error is less than $7mm/sec$. Average displacement error is less than $2mm$ on the original phantom image sequences, and it is less than $4mm$ on the calibrated sequences. Maximum displacement error is less than $5mm$ on the original image sequences and less than $7mm$ on the calibrated image sequences. Table 4.4a presents the average, maximum, and minimum estimated camera velocity errors of the original and calibrated phantom image sequences. Table 4.4b shows their average, maximum, and minimum estimation errors of the camera displacements.

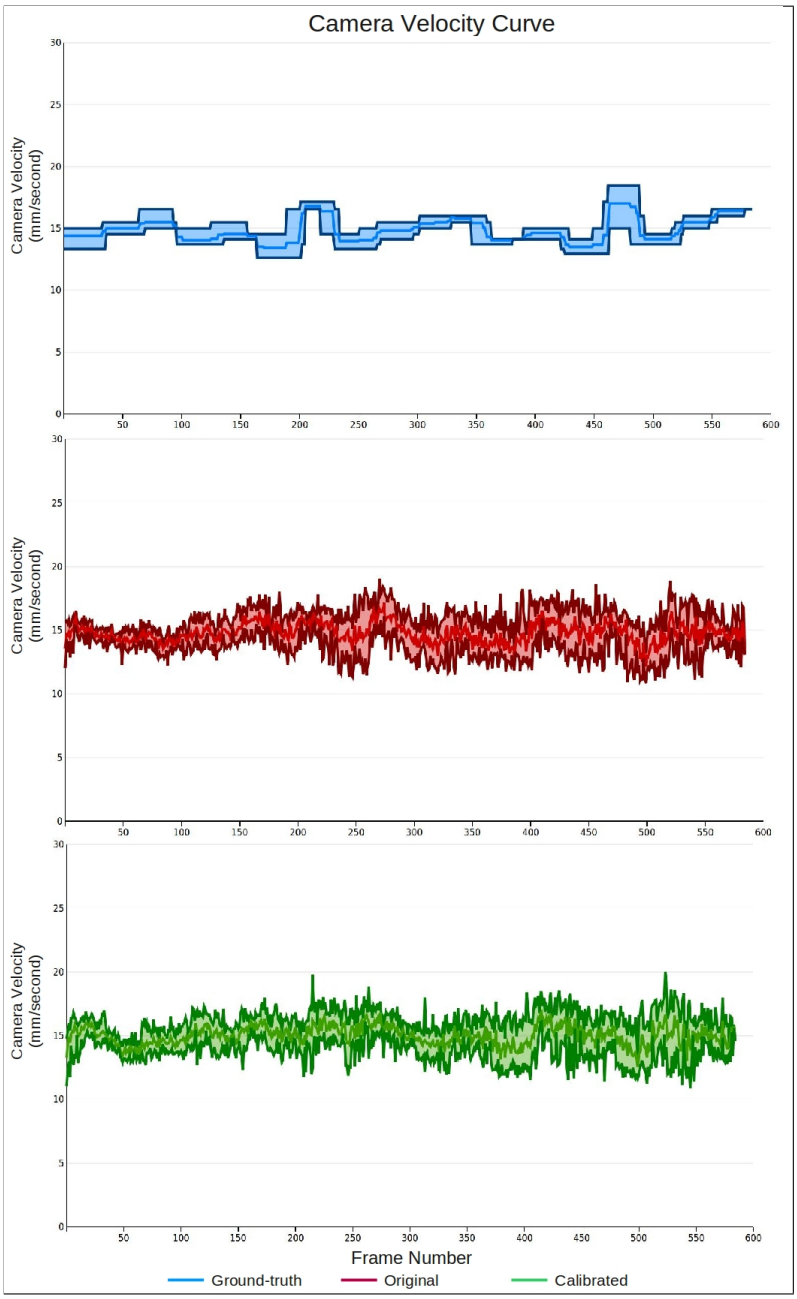


Figure 4.16: *Camera velocity curves at about* $15mm/sec$ *in the straight phantom.* The blue band represents the ground-truth camera velocities of five trials, and the red and green bands indicate the estimated velocities on original and calibrated phantom image sequences, respectively. The bottom and upper curves represent the minimum and maximum velocities of five trials in each band. The solid center curve shows their average velocities. Average velocity error is less than $1.5mm/sec$ on both original and calibrated phantom image sequences after 580 images have been tracked, and maximum velocity error is less than $7mm/sec$ on both sequences.

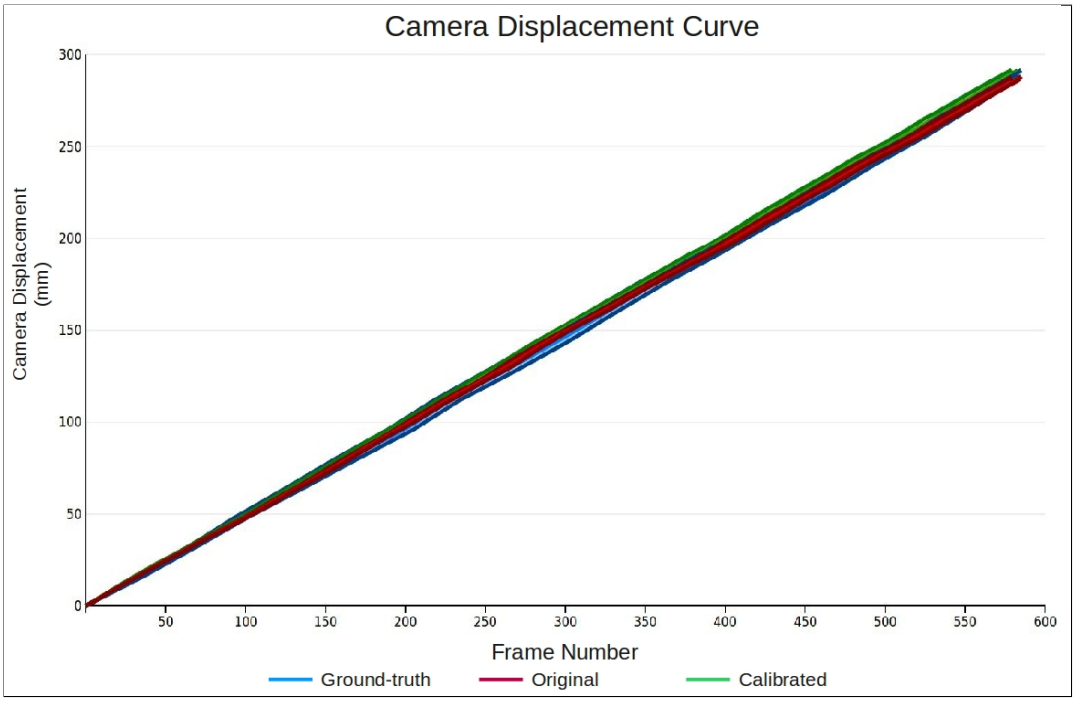


Figure 4.17: *Camera displacement curves at about* $15mm/sec$ *in the straight phantom.* The blue band represents the ground-truth camera displacements of five trials, the red and green bands indicate the estimated displacements on the original and calibrated phantom image sequences, respectively. The bottom and upper curves represent the minimum and maximum displacements in each band. The solid center curve represents the average displacements. Average displacement error is less than $3mm$ on the original phantom image sequences after 580 images have been tracked, and it is less than $5mm$ on the calibrated sequences. Maximum displacement error is less than $6mm$ on the original image sequences and less than $8mm$ on the calibrated sequences.

4.3.3 Curved Phantom Results

Because colored squares are pasted inside the curved phantom and there are no symmetrical squares on the outside, the outer walls of the big tube are wrapped by checkerboard images for the determination of the actual camera motion, as shown in Fig. 4.13b. Here, the size of each black or white square is $29mm \times 19mm$. After excluding the acceleration portion of the external videos, the video streams are chosen for the ground-truth motion computation when the vertical iron wire stays between No.5 and No.17 squares. The displacement that the colonoscope moves across

Table 4.3: The average, maximum, and minimum estimated camera **velocity** and **displacement** errors of original and calibrated straight phantom image sequences at about $15mm/sec$ after 580 images have been tracked.

(a) Camera velocity

Image sequence	Original Images(mm/sec)			Calibrated Images(mm/sec)		
	average	maximum	minimum	average	maximum	minimum
1	1.43	6.71	0.001	1.61	7.06	0.008
2	1.43	5.39	0.004	1.54	6.43	0.004
3	1.29	4.33	0.0041	1.4	4.47	0.009
4	1.3	4.47	0.004	1.43	5.07	0.0005
5	1.09	4.13	0.002	1.17	4.29	0.001

(b) Camera displacement

Image sequence	Original Images(mm)			Calibrated Images(mm)		
	average	maximum	minimum	average	maximum	minimum
1	1.12	3.64	0.0	2.87	7.11	0.0
2	1.76	4.52	0.0	3.23	6.94	0.0
3	2.76	5.47	0.0	4.76	7.97	0.0
4	1.69	3.77	0.0	2.45	5.46	0.0
5	1.63	3.38	0.0	1.52	4.98	0.0

a square is $29mm \times 130.5mm/158.5mm = 23.88mm$, where $130.5mm$ is the radius of the medial axis and $158.5mm$ is the radius of the large tube. The total movement of the colonoscope is $(17-5) \times 23.88mm = 286.56mm$ in the curved phantom image sequences. The colonoscope's instantaneous speed is determined based on an assumption that the colonoscope moves constantly inside the square. The time period that it takes the colonoscope to pass the square is measured by finding two images in which the vertical iron wire just arrived at the square's boundaries. Dividing the time period between two selected images by $23.88mm$ yields the ground-truth velocity when the colonoscope moves inside the square. Finally, ground-truth colonoscope displacements are built by accumulating ground-truth colonoscope velocities.

A virtual curved model is also created through VTK to generate depth values needed in camera translation estimation. Eq. 4.25 is also used to enhance egomotion estimation in the curved phantom image sequences. Fig. 4.20 shows the camera

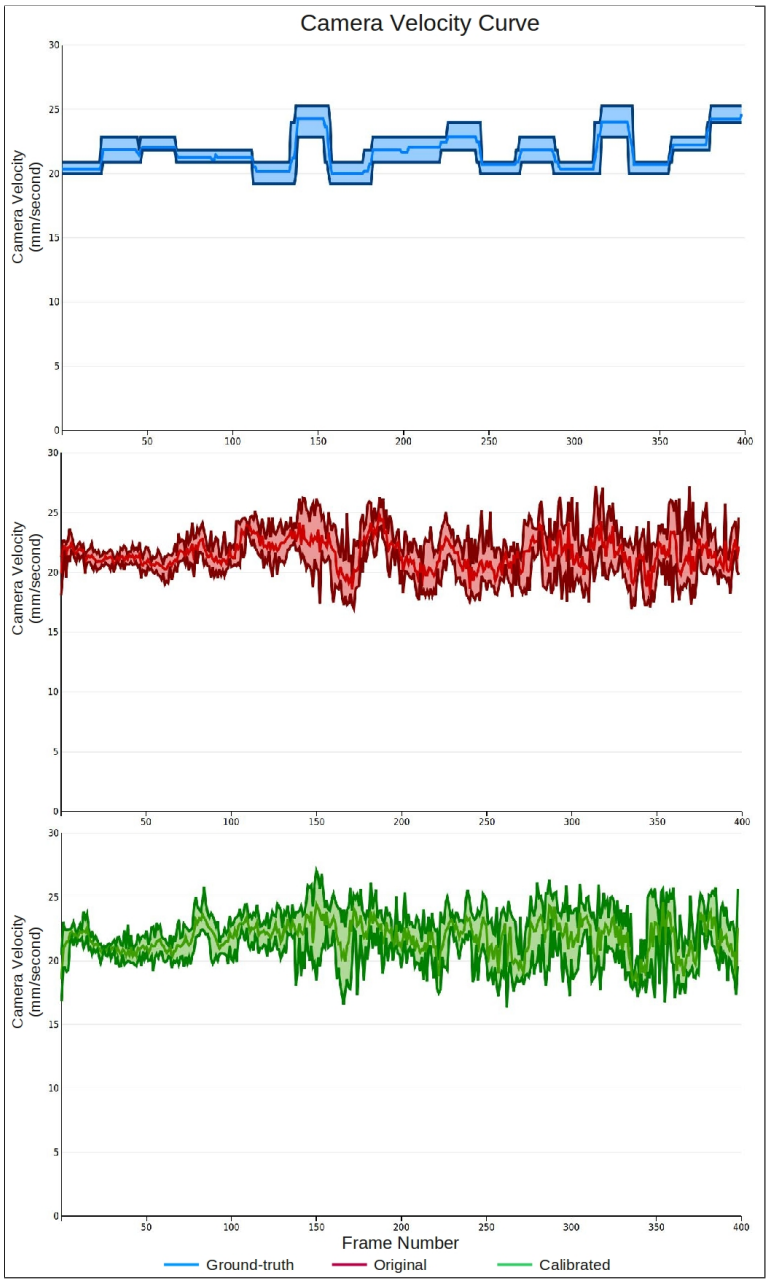


Figure 4.18: *Camera velocity curve at about* $20mm/sec$ *in the straight phantom.* The blue band represents the ground-truth camera motion of five trials, and the red and green bands indicate the estimated camera velocities on the original and calibrated phantom image sequences, respectively. The bottom and upper curves represent the minimum and maximum velocities of five trials in each band. The solid center curve shows the average velocities. Average velocity error is less than $2mm/sec$ on both original and calibrated phantom image sequences after 400 images have been tracked, and maximum error is less than $7mm/sec$.

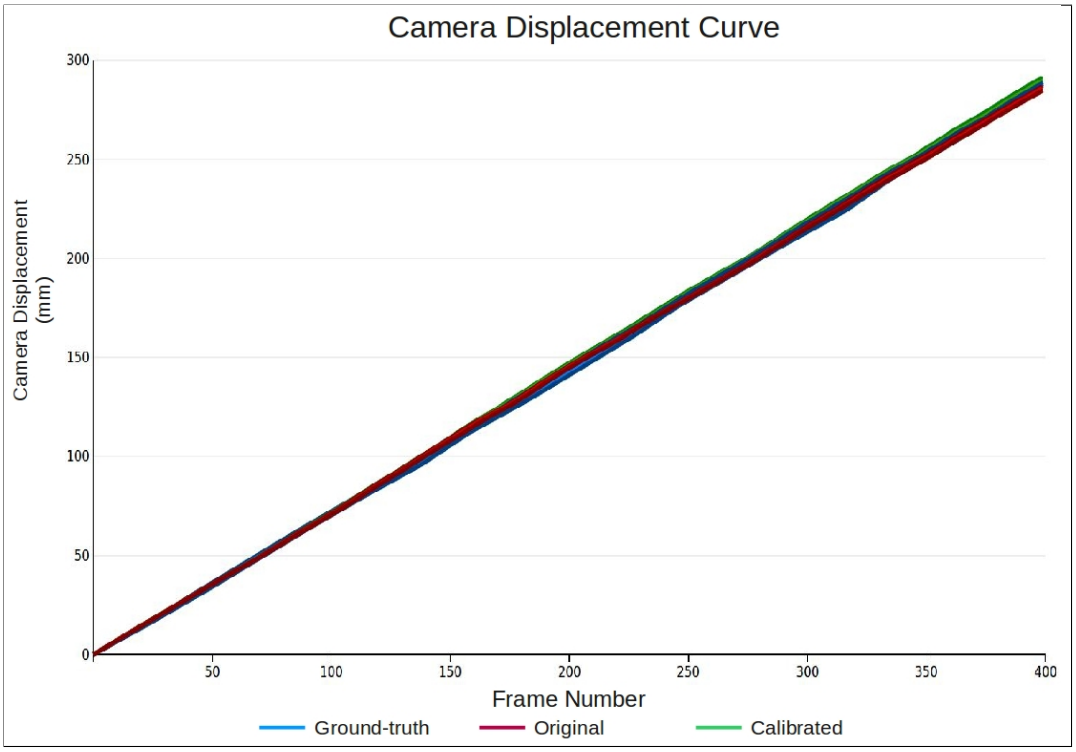


Figure 4.19: *Camera displacement curves at about* $20mm/sec$ *in the straight phantom.* The blue band represents the ground-truth camera displacements of five trials, and the red and green bands indicate the estimated camera displacements on the original and calibrated phantom image sequences, respectively. The bottom and upper curves in each band represent the minimum and maximum displacements of five trials. The solid center curve represents the average displacements. Average displacement error is less than $2mm$ on the original phantom image sequences after 400 images have been tracked, and it is less than $4mm$ on the calibrated image sequences. Maximum displacement error is less than $5mm$ on the original image sequences and less than $7mm$ on the calibrated sequences.

velocity curves on five curved phantom trials at a speed of about $10mm/sec$. Average velocity error is less than $2mm/sec$ on both original and calibrated phantom image sequences after 750 images have been tracked, and the maximum velocity error is less than $8mm/sec$. Fig. 4.15 shows the camera displacement curves at a speed of about $10mm/sec$. Average displacement error is less than $2mm$ on the original phantom image sequences, and it is less than $4mm$ on the calibrated image sequences. Maximum displacement error is less than $5mm$ on the original image sequences and less than $10mm$ on the calibrated image sequences. Table 4.2a presents the average,

Table 4.4: The average, maximum, and minimum estimated camera **velocity** and **displacement** errors of original and calibrated straight phantom image sequences at about $20mm/sec$ after 400 images have been tracked.

(a) Camera velocity

Image sequence	Original Images(mm/sec)			Calibrated Images(mm/sec)		
	average	maximum	minimum	average	maximum	minimum
1	2.	7.04	0.004	2.0	6.85	0.015
2	1.57	6.6	0.019	1.69	5.9	0.005
3	1.7	5.9	0.005	1.82	6.79	0.001
4	1.71	5.47	0.001	1.83	5.94	0.021
5	1.52	5.91	0.001	1.69	6.64	0.01

(b) Camera displacement

Image sequence	Original Images(mm)			Calibrated Images(mm)		
	average	maximum	minimum	average	maximum	minimum
1	1.15	3.59	0.0	3.14	7.0	0.0
2	0.8	2.0	0.0	2.05	4.7	0.0
3	1.37	4.08	0.0	2.68	5.01	0.0
4	1.85	4.37	0.0	2.76	4.98	0.0
5	0.81	2.13	0.0	2.16	5.44	0.0

maximum, and minimum estimation errors of the camera velocity on original and calibrated phantom image sequences. Table 4.2b gives the average, maximum, and minimum estimation errors of the camera displacements.

Fig. 4.22 and Fig. 4.23 illustrate the camera velocity and displacement curves at about $15mm/sec$. Average velocity error is less than $3mm/sec$ on both original and calibrated phantom image sequences after 550 images have been tracked. Maximum velocity error is less than $9mm/sec$ on the original image sequences, and it is less than $8.0mm/sec$ on calibrated image sequences. Average displacement error is less than $7mm$ on the original phantom image sequences, and it is less than $9mm$ on the calibrated sequences. Maximum displacement error is less than $10mm$ on the original image sequences and less than $13mm$ on the calibrated sequences. Table 4.6a presents the average, maximum, and minimum estimation errors of the camera velocity on each of five trials. Table 4.6b shows the average, maximum, and minimum estimation

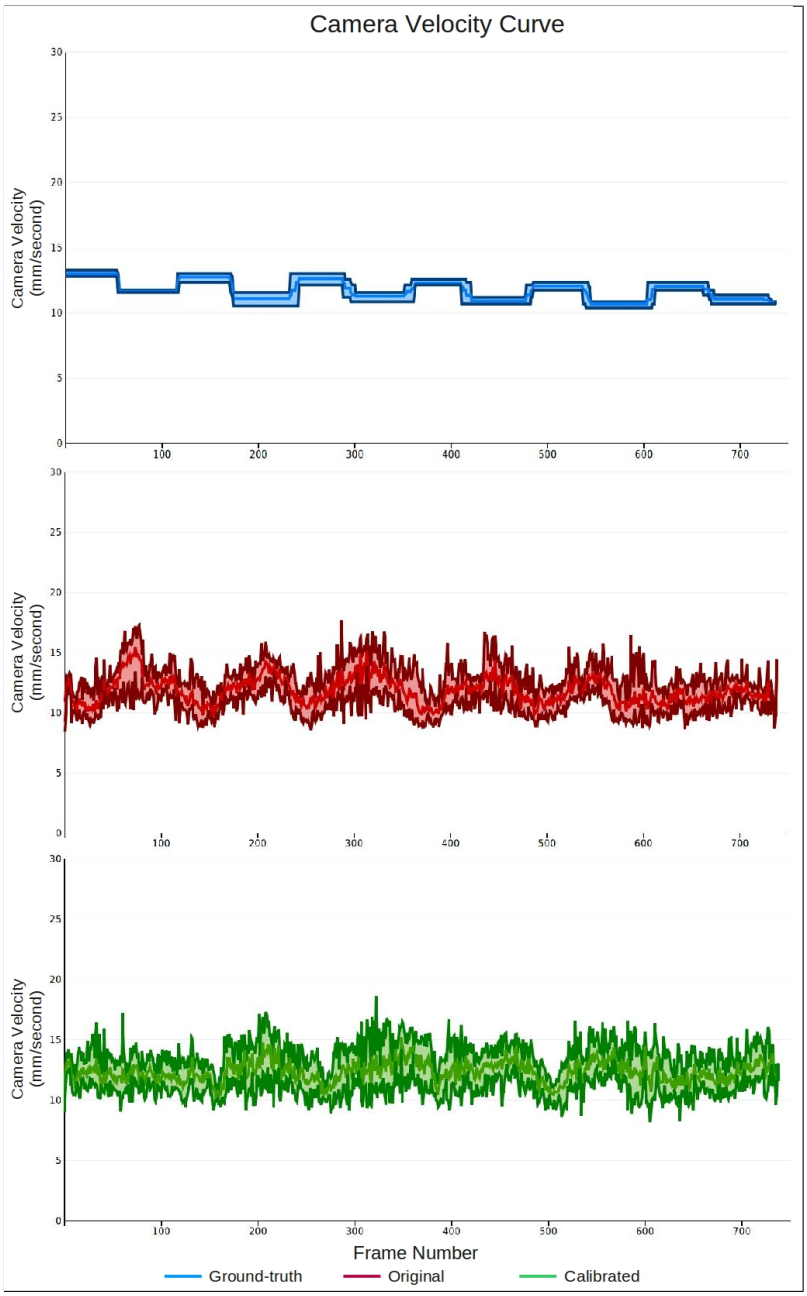


Figure 4.20: *Camera velocity curves at about* $10mm/sec$ *in the curved phantom.* The blue band represents the ground-truth camera velocities of five trials, and the red and green bands indicate the estimated camera velocities on the original and calibrated phantom image sequences, respectively, after 750 images have been tracked. The bottom and upper curves in each band represent the minimum and maximum velocities of five trials, and the solid center curve represents the average velocities. Average velocity error is less than $2mm/sec$ on both original and calibrated image sequences after 750 images have been tracked, and maximum velocity error is less than $8mm/sec$.

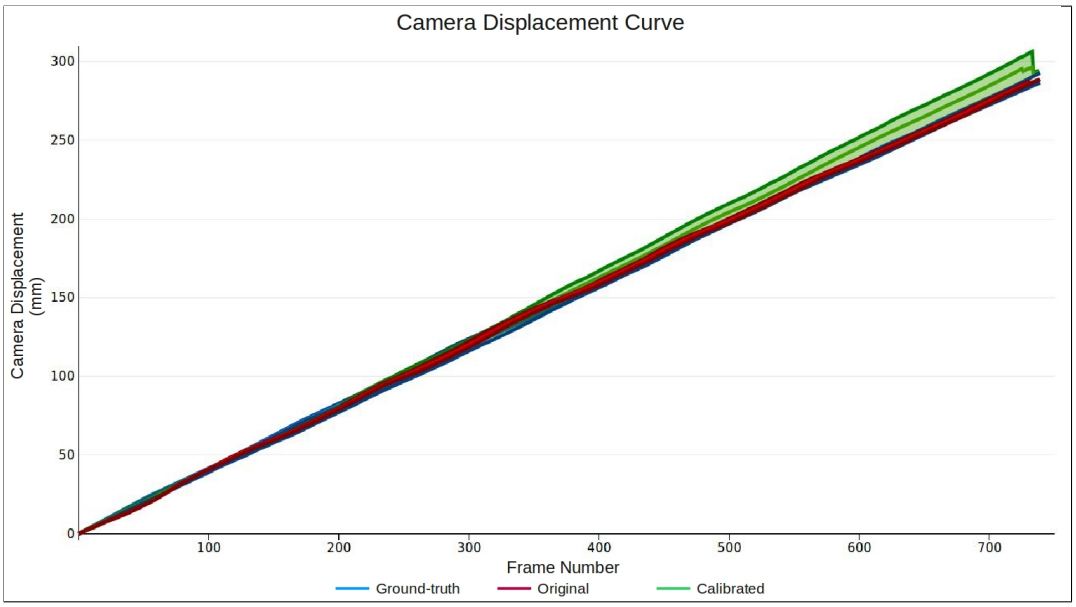


Figure 4.21: *Camera displacement curves at about* $10mm/sec$ *in the curved phantom.* The blue band represents the ground-truth camera displacements of five trials, and the red and green bands indicate the estimated camera displacements on the original and calibrated phantom image sequences, respectively. The bottom and upper curves in each band represent the minimum and maximum displacements of five trials, and the solid center curve represents the average displacements. Average displacement error is less than $2mm$ on the original phantom image sequences after 750 images have been tracked, and it is less than $4mm$ on the calibrated image sequences. Maximum displacement error is less than $5mm$ on the original image sequences and less than $10mm$ on the calibrated image sequences.

errors of the camera displacements.

Fig. 4.24 and Fig. 4.25 show the camera velocity and displacement curves at about $20mm/sec$ in the curved phantom. Average velocity error is less than $3mm/sec$ on both original and calibrated image sequences after 470 images have been tracked. Maximum velocity error is less than $9mm/sec$ on the original image sequences and less than $8.0mm/sec$ on calibrated image sequences. Average displacement error is less than $6mm$ on the original phantom image sequences, and it is less than $7mm$ on the calibrated image sequences. Maximum displacement error is less than $9mm$ on the original image sequences and less than $10mm$ on the calibrated image sequences. Table 4.7a presents the average, maximum, and minimum estimation errors of the

Table 4.5: The average, maximum, and minimum estimated camera **velocity** and **displacement** errors of original and calibrated curved phantom image sequences at about $10mm/sec$ after 750 images have been tracked.

(a) Camera velocity

Image sequence	Original Images(mm/sec)			Calibrated Images(mm/sec)		
	average	maximum	minimum	average	maximum	minimum
1	1.39	6.08	0.002	1.57	7.42	0.00
2	1.52	5.44	0.00	1.6	5.47	0.003
3	1.59	5.71	0.0015	1.31	5.43	0.002
4	1.49	5.69	0.0009	1.41	5.6	0.003
5	1.57	6.03	0.0004	1.56	6.62	0.004

(b) Camera displacement

Image sequence	Original Images(mm)			Calibrated Images(mm)		
	average	maximum	minimum	average	maximum	minimum
1	1.3	3.86	0.0	2.82	7.45	0.0
2	0.99	3.27	0.0	1.23	9.8	0.0
3	1.32	4.22	0.0	2.84	6.18	0.0
4	1.64	3.85	0.0	3.76	10.2	0.0
5	1.29	3.8	0.0	1.94	4.44	0.0

camera velocity on each of five trials. Table 4.7b shows the the average, maximum, and minimum estimation errors of the camera displacements.

I can draw the following conclusions based on the straight and curved phantom results from figures 4.14 through 4.25 and tables 4.5a through 4.7b.

1. In both straight and curved phantom experiments, average estimated velocity error is less than $3mm/sec$ on the original and calibrated phantom image sequences at speeds of $10mm/sec$, $15mm/sec$, and $20mm/sec$. Average displacement error is less than $7mm$ over $288mm$, the actual translation distance of the colonoscope in the straight phantom. In the curved phantom, average displacement error is less than $7mm$ over $286.56mm$. As was described in chapter 2, the colon is generally $1500mm$ long and it has six colon segments. Phantom experiments validated that the proposed algorithm could accurately track the length of a colon segment.

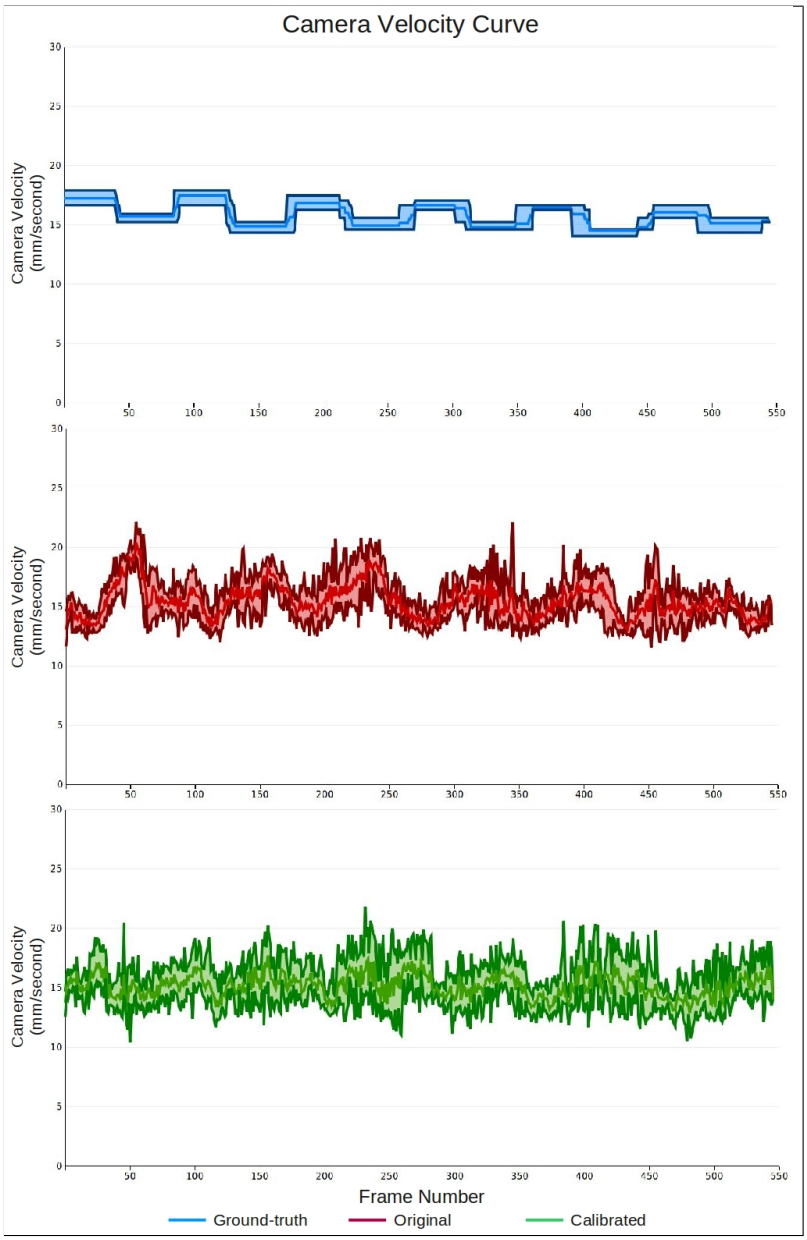


Figure 4.22: *Camera velocity curves at about* $15mm/sec$ *in the curved phantom.* The blue band represents the ground-truth camera velocities of five trials, and the red and green bands indicate the estimated camera velocities on the original and calibrated phantom image sequences, respectively. The bottom and upper curves in each band represent the minimum and maximum velocities of five trials, and the solid center curve represents the average velocities. Average velocity error is less than $2mm/sec$ on both original and calibrated curved phantom image sequences after 550 images have been tracked. Maximum velocity error is less than $8mm/sec$ on the original image sequences, and it is less than $9.0mm/sec$ on calibrated image sequences.

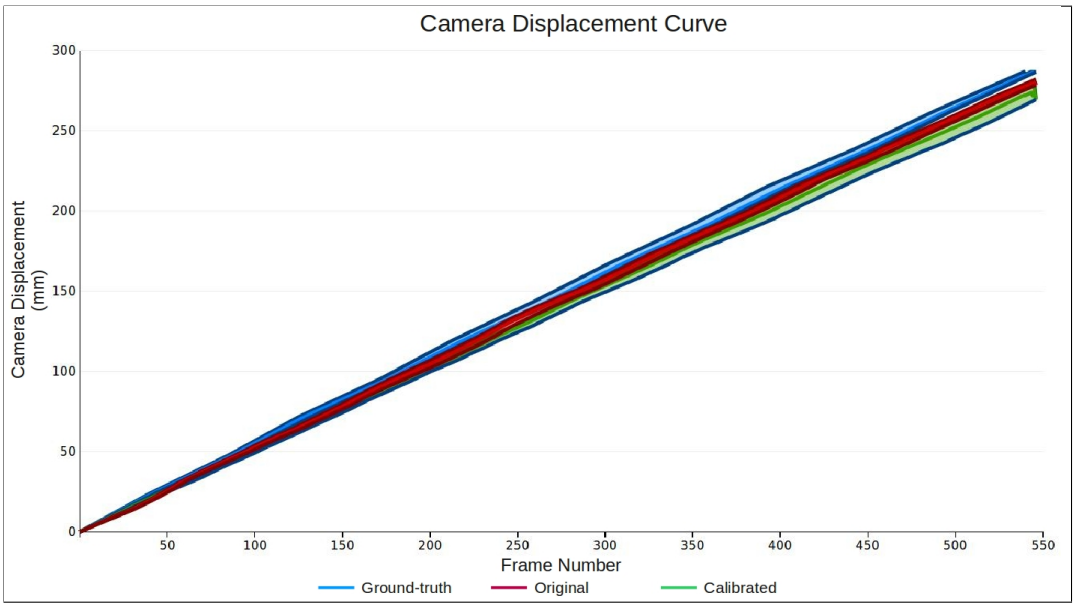


Figure 4.23: *Camera displacement curves at about* $15mm/sec$ *in the curved phantom.* The blue band represents the ground-truth camera velocities of five trials, and the red and green bands indicate the estimated camera displacements on the original and calibrated phantom image sequences, respectively. The bottom and upper curves in each band represent the minimum and maximum displacements of five trials, and the solid center curve represents the average displacements. Average displacement error is less than $7mm$ on the original phantom image sequences after 550 images have been tracked, and it is less than $9mm$ on the calibrated image sequences. Maximum displacement error is less than $10mm$ on the original image sequences and less than $13mm$ on the calibrated image sequences.

2. There is no significant difference between the camera velocity and displacement errors in the original and calibrated phantom image sequences, in both straight and curved phantom experiments. The estimated velocities from the calibrated phantom image sequences are slightly larger than those from the original sequences. The velocity enlargement arises because phantom images are artificially scaled up after the calibrated parameters are used to remove the image distortion, which increases the optical flow magnitudes also. But the tracking results are very comparable because average displacement error is less than $7mm$ in both original and calibrated phantom image sequences. Therefore, the proposed tracking algorithm is insensitive to image distortion caused

Table 4.6: The average, maximum, and minimum estimated camera **velocity** and **displacement** errors of original and calibrated curved phantom image sequences at about $15mm/sec$ after 550 images have been tracked.

(a) Camera velocity

Image sequence	Original Images(mm/sec)			Calibrated Images(mm/sec)		
	average	maximum	minimum	average	maximum	minimum
1	1.75	5.62	0.014	1.72	5.74	0.003
2	1.79	5.6	0.003	1.75	6.2	0.003
3	1.82	6.9	0.001	1.86	8.9	0.0006
4	1.63	7.51	0.004	1.67	5.55	0.012
5	1.88	5.67	0.0016	1.76	5.64	0.0022

(b) Camera displacement

Image sequence	Original Images(mm)			Calibrated Images(mm)		
	average	maximum	minimum	average	maximum	minimum
1	3.35	6.33	0.0	5.94	8.33	0.0
2	6.13	9.26	0.0	8.46	13.0	0.0
3	2.43	4.96	0.0	4.6	7.0	0.0
4	4.51	8.15	0.0	6.45	11.	0.0
5	3.6	7.78	0.0	6.7	8.5	0.0

by fish-eye effect.

3. Despite the variance of the ground-truth camera motion at three speed levels, the estimated camera velocity curves follow the same trend except that the velocity amplitudes are different. These results indicate that the estimated camera motion parameters rely on the actual camera motion as well as texture distributions inside the phantom. Periodic texture distributions can artificially affect the estimated camera velocities. Therefore, a random texture distribution enhances estimation accuracy.

4. The number of tracked colonoscopy images dominates the errors in estimated camera displacements, while the actual speed of the colonoscope has a lesser effect. Camera displacement errors decrease with the increase of actual camera velocities because the slower the velocities, the greater the number of colonoscopy

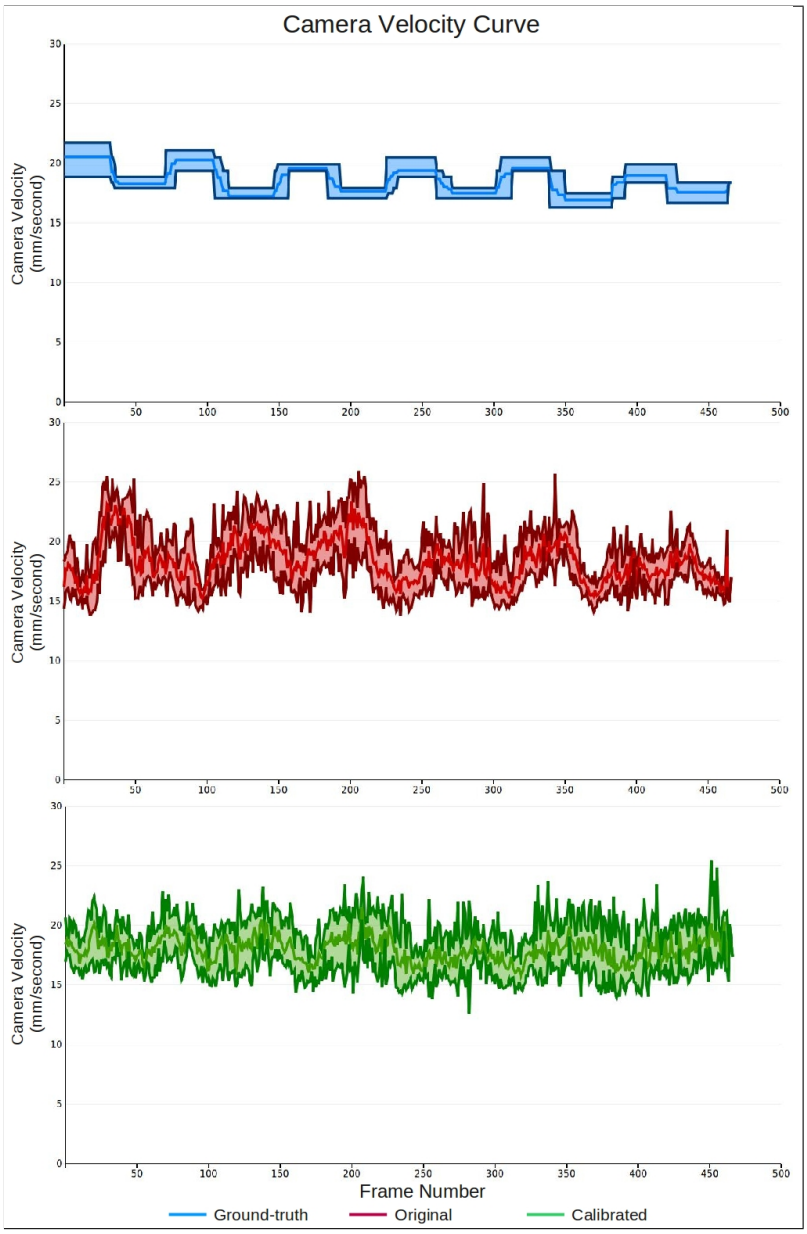


Figure 4.24: *Camera velocity curves at about* $20mm/sec$ *in the curved phantom.* The blue band represents the ground-truth camera velocities of five trials, and the red and green bands indicate the estimated camera velocities on the original and calibrated phantom image sequences, respectively. The bottom and upper curves in each band represent the minimum and maximum velocities of five trials, and the solid center curve represents the average velocities. Average velocity error is less than $3mm/sec$ on both original and calibrated image sequences after 470 images have been tracked. Maximum velocity error is less than $9mm/sec$ on the original image sequences, and it is less than $8.0mm/sec$ on calibrated image sequences.

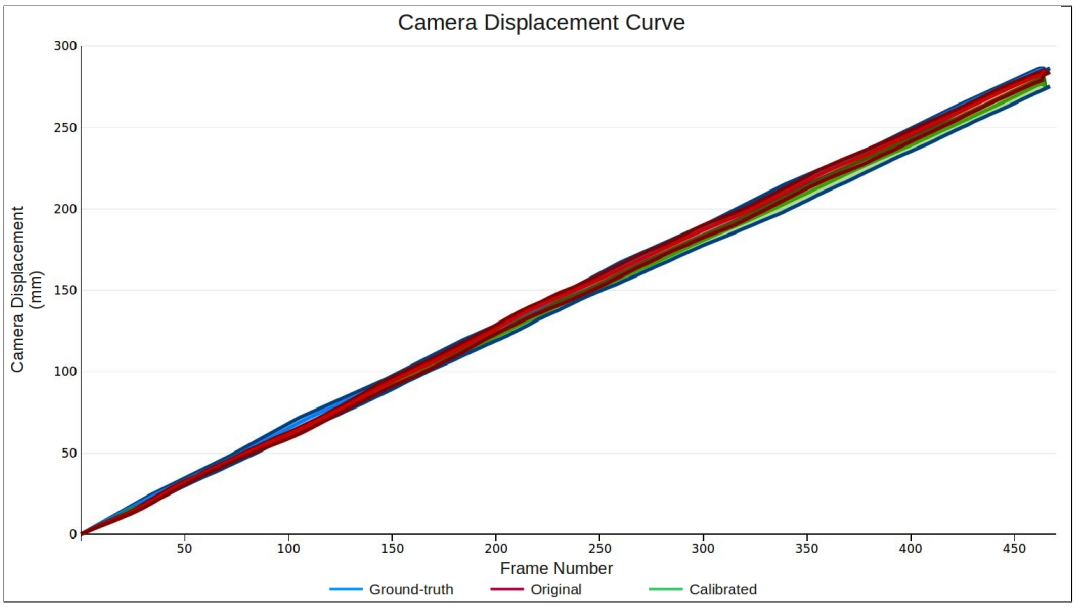


Figure 4.25: *Camera displacement curves at about* $20mm/sec$ *in the curved phantom.* The blue band represents the ground-truth camera displacements of five trials, and the red and green bands indicate the estimated camera displacements on the original phantom image sequences The bottom and upper curves in each band represent the minimum and maximum displacements, and the solid center curve shows the average displacements. Average displacement error is less than $6mm$ on the original phantom image sequences after 470 images have been tracked, and it is less than $7mm$ on the calibrated image sequences. Maximum displacement error is less than $9mm$ on the original image sequences and less than $10mm$ on the calibrated sequences.

images.

5. The modified egomotion estimation algorithm based on Eq. 4.25 is phantom-oriented. It makes little difference on clinical colonoscopy image sequences. The lack of effect is because the field of view inside phantom images is farther than in colonoscopy images. Depth variance in the phantom images is larger than in the clinical colonoscopy images. The modified egomotion estimation algorithm based on depth weights, defined in Eq. 4.25, can enhance the tracking accuracy on the phantom images while it might be inappropriate for the OC images. Fig. 4.26 compares the tracking results by using the original and modified egomotion estimation approaches on an OC image sequence. There are no major differences between the tracked VC images at frame 200 and 400. At

Table 4.7: The average, maximum, and minimum estimated camera **velocity** and **displacement** errors of original and calibrated curved phantom image sequences at about $20mm/sec$ after 470 images have been tracked.

(a) Camera velocity

Image sequence	Original Images(mm/sec)			Calibrated Images(mm/sec)		
	average	maximum	minimum	average	maximum	minimum
1	2.05	6.99	0.002	1.89	5.6	0.002
2	2.35	7.98	0.008	2.18	6.14	0.0006
3	2.33	8.12	0.004	2.05	7.97	0.004
4	2.15	6.87	0.013	1.85	6.17	0.02
5	2.18	7.36	0.006	1.7	5.1	0.0003

(b) Camera displacement

Image sequence	Original Images(mm)			Calibrated Images(mm)		
	average	maximum	minimum	average	maximum	minimum
1	5.33	8.83	0.0	5.45	9.95	0.0
2	1.71	5.51	0.0	5.11	8.1	0.0
3	2.02	6.01	0.0	6.19	9.74	0.0
4	1.25	3.4	0.0	4.52	8.52	0.0
5	1.6	4.9	0.0	4.18	8.26	0.0

frame 600, the tracked VC image using the depth weights is slightly better than the original tracked result by visually measuring the polyp's size. However, the original method produces more accurate results than the updated approach at frame 807, because the top fold is included more in the center image than in the right image. Nevertheless, tracking results based on these two egomotion estimations are very comparable in the OC sequence.

4.4 Clinical Data Evaluation

In this section, the proposed tracking algorithm is evaluated on the OC image sequences[2]. Without doubt, this is important because accurately tracking OC images is the ultimate goal of the dissertation. However, unlike clear phantom images, OC images exhibit a number of characteristics that pose significant challenges to any

[2]All clinical datasets are from National Cancer Institute(`http://www.cancer.gov/`).

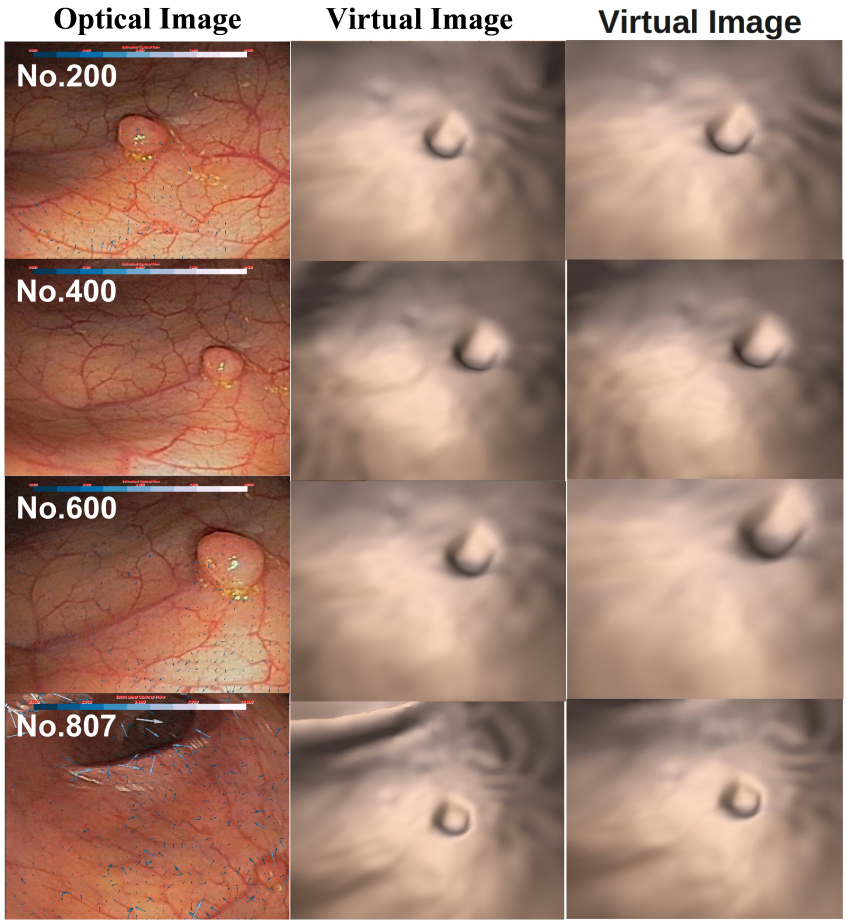


Figure 4.26: *Comparison between the tracking results using the original and updated egomotion estimation algorithms based on Eq. 4.25.* The 807-frame sigmoid colonoscopy sequence used in Fig. 4.9 is employed here. *Column 1:* OC images, *Column 2:* tracked VC images using original egomotion estimation algorithm; *Column 3:* modified egomotion estimation approach.

tracking system. These include deformation and structural changes due to patient movement or simply from the fact that OC and VC are separate acquisitions over time. Image artifacts include specularities (extremely bright regions), blurriness due to the endoscope facing a wall or very fast motion. Pre and post surgery images are another instance I will consider. I illustrate the robustness of my tracking algorithm under many of these conditions in the following sections, using 4 clinical colonoscopy sequences from 4 different patients.

4.4.1 Colon Deformation

A sequence of 174 OC images were acquired in the transverse colon to illustrate and evaluate the impact of deformation. Fig. 4.27 shows the OC(top row) and VC (bottom row) images of four frames from this sequence. VC and OC images are manually co-aligned at the first fold marked with a cyan triangle(column 1). Column 2 indicates the colonoscope arriving at the second fold, marked by the yellow triangle; the fold becomes inflated and is elliptical in the OC image, while it is still triangular in the VC image. In column 3 OC and VC images show the fold reverting to a triangular shape. The images in the fourth column show the colonoscope at the third fold, labeled by the blue ellipse. Here, the orientation of the marked triangle and ellipse do not necessarily represent camera rotation, since the deformation also contributes to the change in shape of the fold. Although this case is difficult to evaluate, the results demonstrate that my algorithm is not very sensitive to the fold or other structural changes in the colon, since the tracking system is able to keep the OC and VC images in sync, *i.e.* they reach the same fold.

4.4.2 Fluid and Illumination Artifacts, Blurriness

A sequence of 272 OC images were captured between two folds in the ascending colon. This sequence contained images with fluid and illumination artifacts, as well as blurry frames. These artifacts are not present in the VC images, as they have been segmented out. Fig. 4.28 shows four frames from this sequence. Yellow fluid (region marked A in column 1), strong illumination band (colon moving close to the wall, region marked B in column 2), and blurriness (colon moving fast, column 3) are some of the difficulties that face the tracking system, while the corresponding VC images are devoid of these artifacts. Despite these artifacts, it can be seen in column 4 the colonoscope is close to the second fold (area marked D), and in sync with the VC

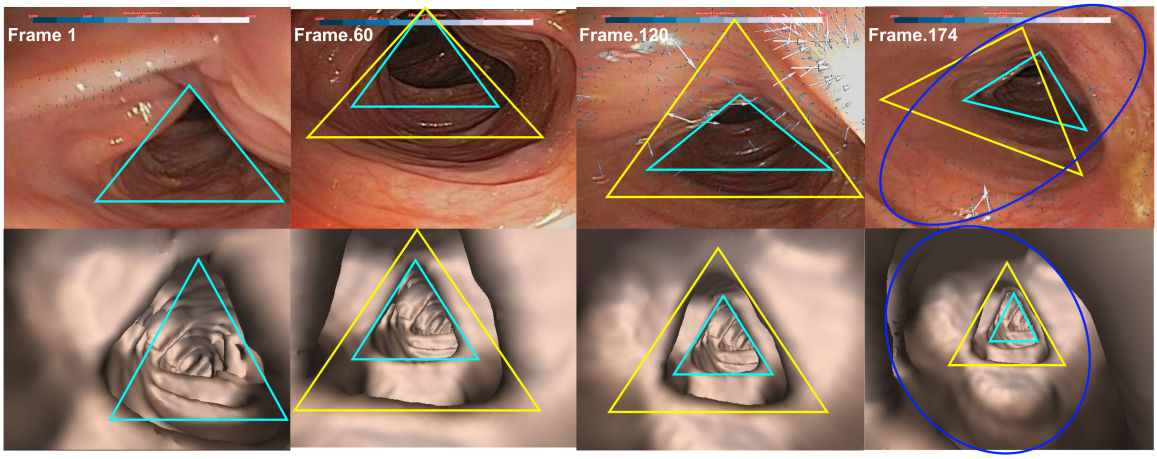


Figure 4.27: *Robustness evaluation: colon deformation.* A sequence of 174 OC images in transverse colon were used to evaluate algorithm sensitivity to deformation in colon fold shape. *Column 1:* First fold marked with a cyan triangle in both OC and VC images, *Column 2:* Colonoscope at second fold(yellow triangle) in both OC and VC. Fold shape in OC is elliptical, but triangular in VC. *Column 3:* Fold in OC becomes triangular with colonoscope near the colon's wall and VC still stays near the second fold, *Column 4:* Both VC and OC arrive at the third fold, marked by blue ellipse.

image (area marked E). Also note the artificial hole in the VC image of column 2 (area marked C), a segmentation error. But it does not influence my tracking results, and this fact is important because perfect segmentations are almost never achievable[72].

4.4.3 Surgery Induced Structural Changes

OC is a screening as well as treatment procedure. Removal of polyps changes the structure of the colon. Fig. 4.29 illustrates this. The OC image in column 1 shows the circled polyp, and the OC image in column 4 shows the area where the polyp was removed. It is important that a tracking algorithm continue to work under these conditions. In this example, I have selected two image sequences acquired before and after the removal of the polyp in the sigmoid colon. The left two columns of images in Fig. 4.29 show the tracking results of the first and last images of the sequence before the polyp removal, while the right two columns illustrate the results after the

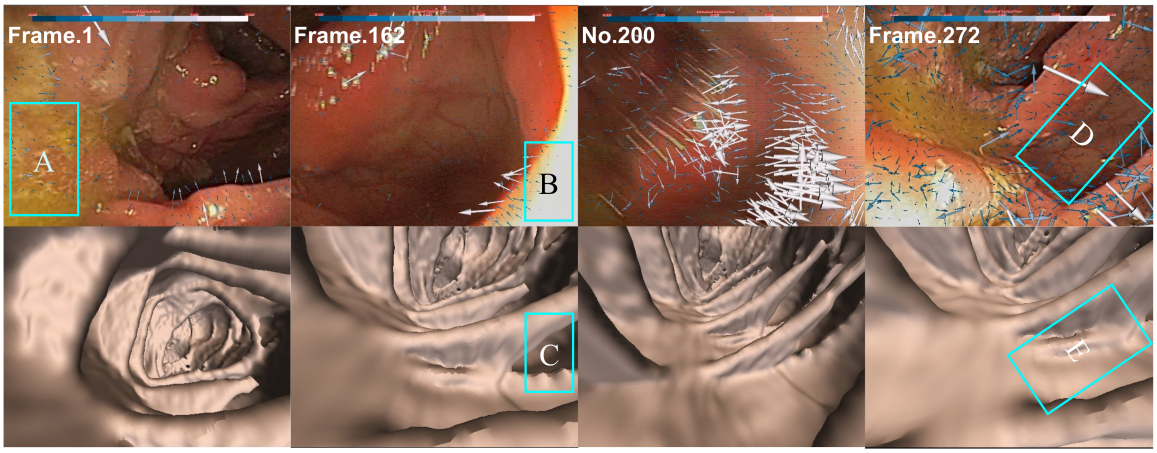


Figure 4.28: *Robustness evaluation: fluid and illumination artifacts.* A 272 frame sequence from ascending colon is used to demonstrate algorithm sensitivity to fluid presence(area marked A), illumination band(area marked B), segmentation error results in artificial hole(area marked C), and blurry image(column three, top row). At the end of the sequence, images are tracked well, as shown by corresponding areas D and E in the OC and VC images of column 4.

polyp removal. The positions of the polyp are marked in both OC and VC images with cyan circles. It can be seen that the tracking algorithm continues to function successfully despite these structural changes induced by surgery.

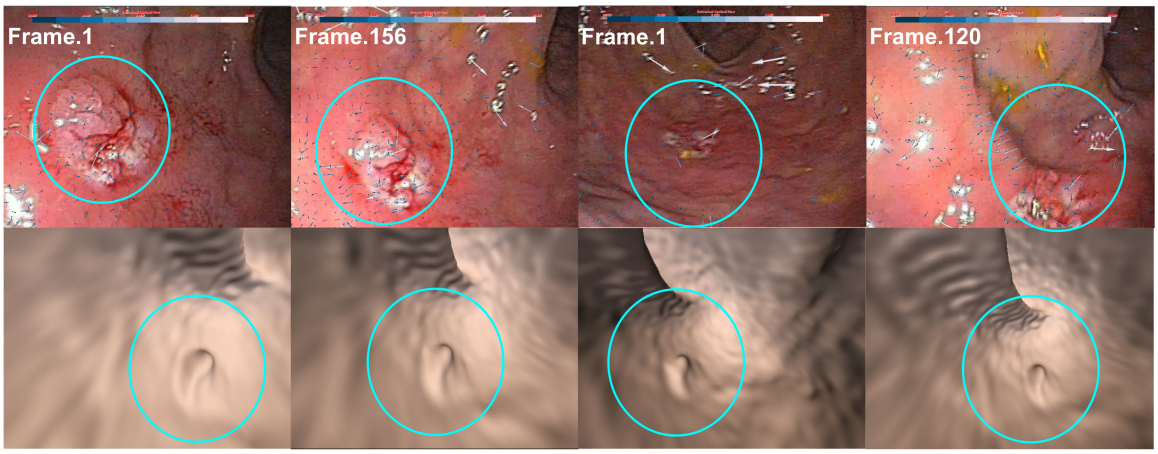


Figure 4.29: *Robustness evaluation: surgery related structural changes.* Illustration of tracking two image sequences corresponding to pre and post polyp removal in the sigmoid colon. Left two columns show the tracking results of the first and last frame before the polyp is removed, while the right two columns show the results after polyp removal. Polyp positions are marked by the cyan circle.

4.4.4 Multi-Object Motion Induced by Surgical Tools

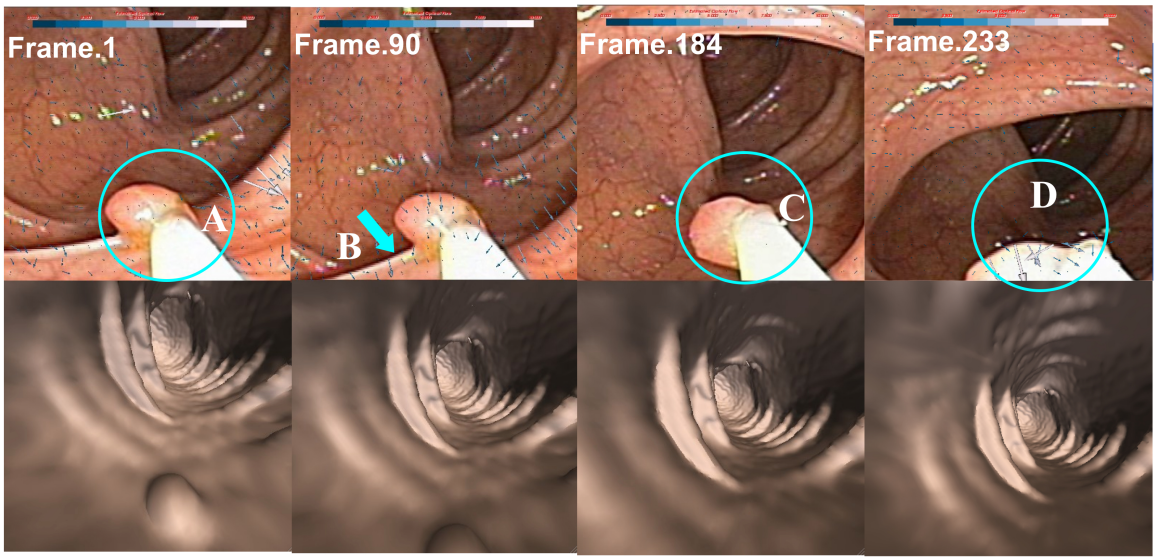


Figure 4.30: *Robustness evaluation: simultaneous motion of colonoscope and surgical tools in optical flow field.* A sequence of optical images in the descending colon to illustrate effectiveness of tracking system under conditions that break egomotion determination due to simultaneous motion of both colonoscope and surgical tools. *Column 1:* Snare inserted into colon to circle polyp, marked A, *Column 2:* Polyp is lifted up for removal, and VC image continues to track motion, *Column 3:* Polyp is in the OC, but disappears in the VC image due to colonoscope's motion, *Column 4:* Withdrawal of snare and polyp, but the VC image continues to track the optical image.

During surgical procedures, tools will appear in the optical images. An example is illustrated in the top row images of Fig. 4.30. Both the colonoscope and the tool are simultaneously influencing the visual motion field captured by the optical flow. Theoretically, this breaks the condition of egomotion determination. However, if the affected region is relatively small and localized, then the optical colonoscope can be successfully tracked. I attempted to test this with a sequence of optical images captured in the descending colon. As illustrated in Fig. 4.30 (left column), a snare is inserted into the colon to enclose the polyp, shown in the area marked A. VC image is initially co-aligned based on the polyp's position. Since the tissue near the polyp

(area pointed to by the arrow B) is stretched, the gastroenterologist has to lift the polyp in order to remove it. The polyp in the tracked VC image is partly hidden because the egomotion in the OC image is translating along the Y direction. In the area marked C(third column), the polyp is removed and attached by the snare, and the polyp disappears in the tracked VC of the third column. However, the VC still follows the actual egomotion of the optical image, while the polyp continues to stay in the OC image. The images in the fourth column show both the tool and the polyp withdrawn from the colon. However, this does not affect the tracking, and VC continues to follow the egomotion of the optical sequence.

In this section, I evaluated colonoscopy tracking algorithms on four colonoscopy image sequences from four different patients. My colonoscopy tracking algorithm is demonstrated to be robust to colon deformation, colon artifacts, structural changes, and multi-object motion. All these properties are very important to show its application to clinical practice.

4.5 Conclusions

In this chapter, I have presented an optical flow based colonoscopy tracking algorithm to co-align OC and VC images. This algorithm uses a combination of sparse and dense optical flows with the FOE, resulting a highly robust and stable tracking algorithm. Optimal spatial-temporal scales are determined for each image during the sparse optical flow computation, which is also used to compute the dense optical flow. The dense optical flow is employed to compute the FOE, utilizing the full visual motion information in the optical flow field. The FOE permits separation of camera rotation and translation velocities, contributing to the mathematical robustness of the algorithm. Camera motion parameters are estimated using the sparse optical flow and the FOE. This algorithm has also been augmented with a regression method, LMS, to accurately estimate camera motion parameters. The regression method permits

detection of outliers among the chosen feature points. This leads to better estimates of rotation and translation parameters. Accumulation errors are reduced, making it possible to track longer colonoscopy image sequences.

I have performed extensive experimental results, on (1) straight and curved phantoms with known camera motion parameters, and (2) 4 clinical colonoscopy image sequences. Phantom validation indicated that the average camera velocity errors were less than $3mm/sec$ at speeds of about $10, 15$, and $20mm/sec$, and the average camera displacement errors were less than $7mm$ over a total displacement of $288mm$ in the straight phantom and $7mm$ over $286.56mm$ in the curved phantom. Over 800 frames of a clinical dataset were successfully tracked with a maximum error of $2mm$. Specific challenges posed by OC data include the presence of fluid, illumination and blurred images, and deformation of colon tissue due to patient position changes between OC and VC images. These artifacts make it difficult to accurately track colonoscopy images. Through four image sequences from four different patients, I have shown the reliability of my tracking algorithm under these conditions.

CHAPTER 5: CONTRIBUTION TWO – REGION FLOW

"As far as the laws of mathematics refer to reality, they are not certain, and as far as they are certain, they do not refer to reality."

– Albert Einstein

In the previous chapter, an egomotion estimation framework based on multi-scale optical flow was proposed to track consecutive colonoscopy images. But the method fails when blurry images appear. In this chapter, I describe the key problem of estimating large motion despite interruption by blurry images. Large motion estimation includes two components: region flow computation and incremental egomotion estimation. Phantom and clinical image sequences are used to verify the robustness and accuracy of the proposed algorithm for estimating large egomotion.

5.1 Problem Statement

The appearance of blurry images is a common occurrence in the colonoscopy video stream. This situation arises because a colonoscopy is an interior navigation inside a narrow environment with many artifacts, including fluids and stools. When the colonoscope touches these artifacts, it results in blurry images. For instance, a colonoscope immerses into yellow fluid(Fig. 5.1a), touching the wall(Fig. 5.1b), lens covered by water(Fig. 5.1c), extreme lighting conditions(Fig. 5.1d and Fig. 5.1e).

Blurry images shown in Fig. 5.1 contain unstable visual motion information because there are no folds or blood vessels for interest point detection. Intensity distributions of blurry images are also uniform, which makes it difficult to match two interest points using the intensity constancy model, because there exist several feature

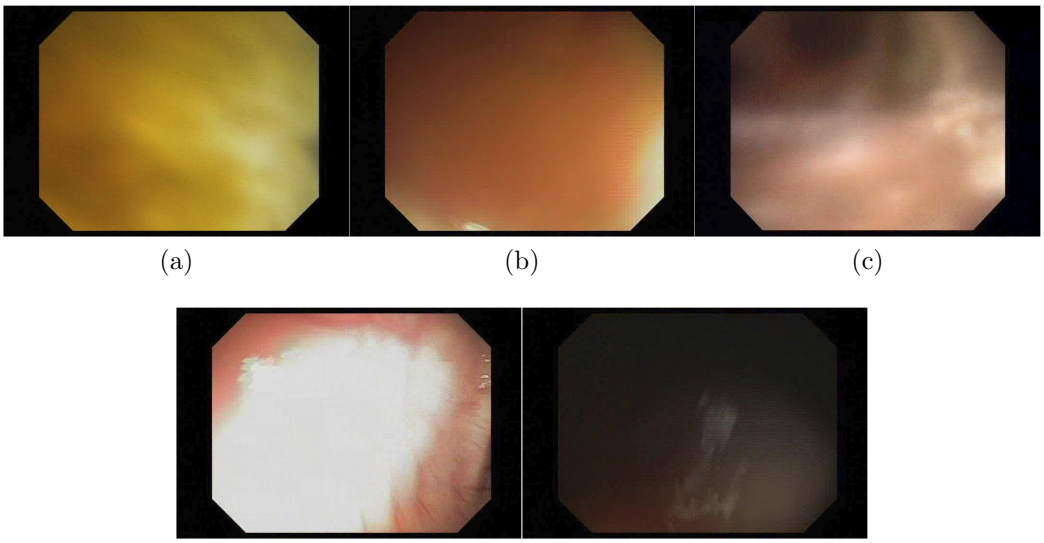
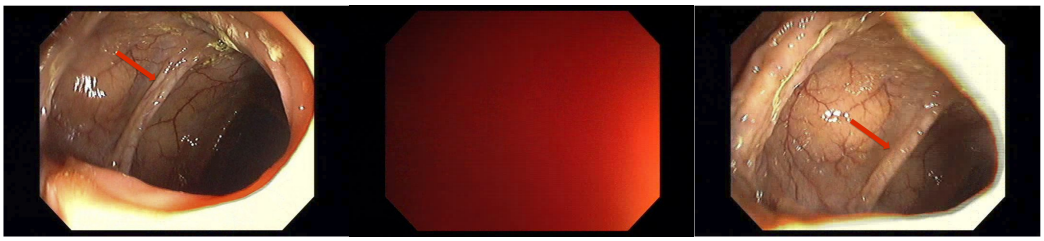


Figure 5.1: *Five types of blurry images* frequently occur in the colonoscopy video stream because of (a) colonoscope immersing into the fluid; (b) colonoscope touching the wall; (c) colonoscope's lens covered by water; (d) strong light, causing bright regions; (e) weak light, causing dark areas.

candidates with the same intensity value. Such properties of blurry images prohibit multi-scale optical flow computation from accurately identifying visual motion. As a result, the colonoscopy tracking system introduced in chapter 4 fails to compute camera motion parameters using blurry images, due to inaccurate optical flow.

Consequently, the goal of this chapter is to continuously co-align optical colonoscopy (OC) and virtual colonoscopy (VC) images after blurry images appear. Blurry images must be excluded from the colonoscopy tracking system. Also, two colonoscopy images interrupted by a blurry image sequence have significant visual motion, such as Fig. 5.2a and Fig. 5.2c. Note the large image displacements between two corresponding folds, as indicated by red arrows.

Multi-scale optical flow is unable to compute large visual motion from these two selected colonoscopy images. The difficulty is that the temporal derivative calculation is an ill-posed problem when two images have significant visual motion. Many affine-invariant feature descriptors[155, 147, 11, 213, 214] assume that regions of matched



(a) Frame 6430 (b) Frame 6470 (c) Frame 6515

Figure 5.2: *An OC image sequence with blurry images.* (a) A clear colonoscopy image before blurry images, (b) a blurry image, (c) a colonoscopy image after blurry images.

features undergo affine transform during large visual motion. The resulting feature descriptors are insensitive to affine distortion. They are used along with wide-baseline image matching techniques and have been extensively studied and used to compute large visual motion. The accuracy of visual motion computation depends on the distinctiveness of feature descriptors and feature matching sizes. However, feature descriptors are indistinct in the colonoscopy images because colonoscopy images contain insufficient visual cues. Matching sizes are also unknown because image displacements are unpredictable between two colonoscopy images interrupted by a blurry image sequence. For these reasons, there are many false feature matches in the colonoscopy images.

Prior knowledge of the feature matching ranges can significantly reduce the number of false feature matches. Accordingly, the determination of feature matching ranges is an important problem addressed in this chapter. Another issue is related to egomotion estimation. The Focus of Expansion(FOE) based egomotion estimation algorithm designed in chapter 4 can accurately estimate small camera motion, but it fails when the camera motion is large. The estimation failure is due to a simplified model, defined in Eq. 3.26, that assumes camera motion is small. This model fails when there is large camera motion between two colonoscopy images. One solution is the subdivision of large visual motion into a sequence of small optical flow fields. The FOE-based egomotion estimation approach can then be applied to each individual

small optical flow field. Therefore, another issue in this chapter is how to subdivide large visual motion into a sequence of small optical flow fields. Then, all the small optical flow fields are used to incrementally estimate large camera motion.

In order to resolve these two challenges, I propose a strategy for estimating large motion due to blurry image interruption, which is based on region flow. Region flow is used to pre-determine the feature matching ranges. Accurate visual motion can thus be computed by using the determined ranges. A partial-differential-equations(PDE) based framework is then developed to subdivide large visual motion into a sequence of optical flow fields, to incrementally recover large egomotion. Fig. 5.3 details various components of the proposed framework for estimating large motion when blurry images appear. It first detects blurry images and selects a colonoscopy image pair before and after the blurry image sequence. Region flow describes large visual motion between the two selected images. Incremental egomotion estimation is developed to compute significant camera motion parameters.

I will thereby describe a strategy for large motion estimation under three general headings: blurry image detection, region flow based visual motion computation, and incremental egomotion estimation.

5.2 Blurry Image Detection

This section describes different image filters to detect blurry images in a colonoscopy video stream. I also present colonoscopy image selection after excluding blurry images in this section. The effectiveness of blurry image detection is demonstrated by a long colonoscopy image sequence.

5.2.1 Algorithm Description

Assuming No.k image is a clear colonoscopy image and k is the image index, No.$k+1$ image is imported into the blurry image detector. The proposed approach modifies

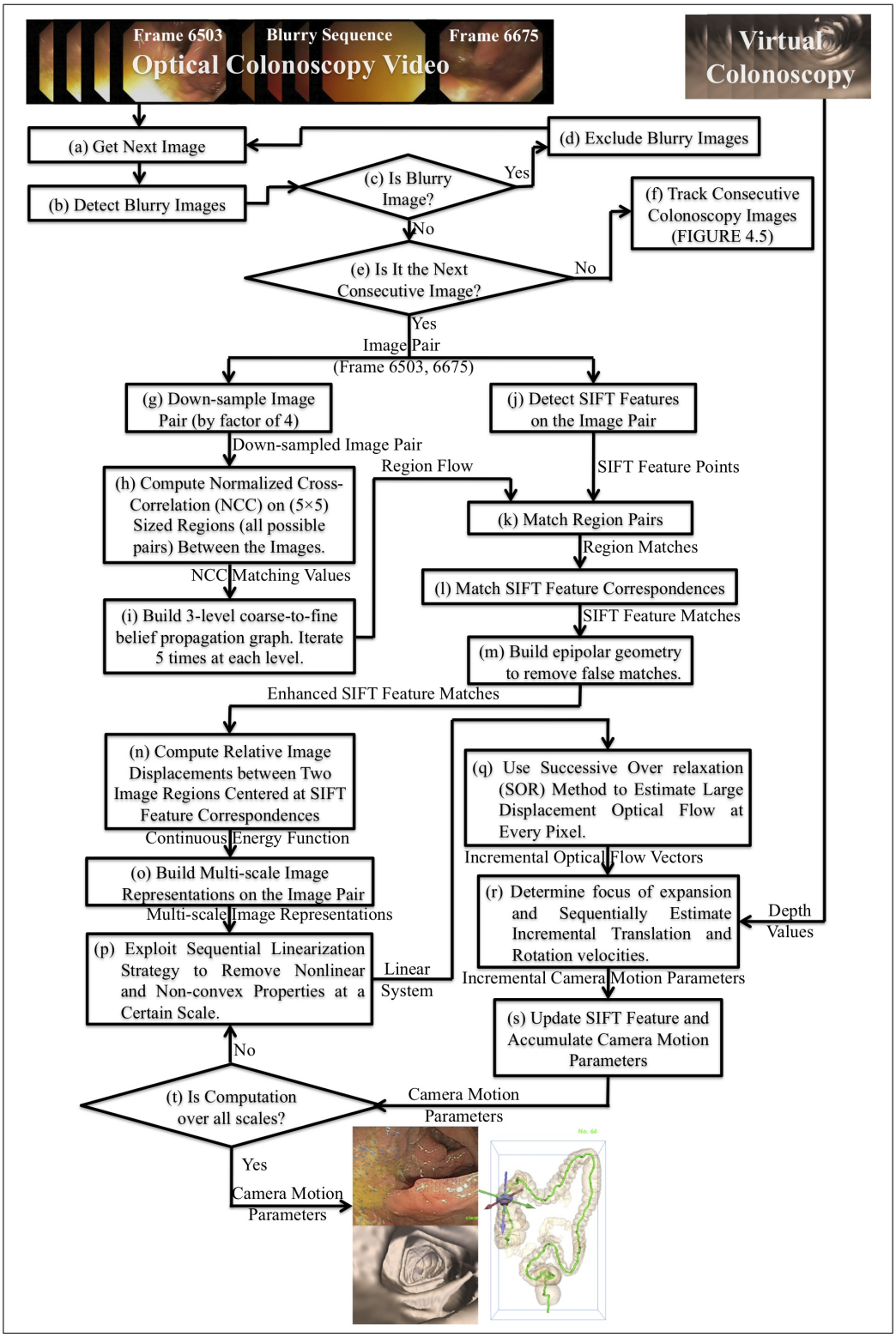


Figure 5.3: Strategy for estimating large motion to continuously co-align optical and virtual colonoscopy images when blurry images appear.

the blurry image detection algorithm described in [144] by ignoring machine learning process. This modification is necessary in order to fulfill real-time requirements of the colonoscopy tracking system. It is comprised of a saturation filter, edge filter, and intensity filter.

- **Saturation filter:** Saturation measures the vividness of color images, and is defined as

$$F^{saturation} = 1 - \frac{3}{R+G+B}[\min(R,G,B)] \tag{5.1}$$

at every pixel. R, G, and B are its intensities of red, green, and blue channels. Saturation measurements can efficiently detect blurry images when the colonoscope immerses into fluid or touches the wall. The detection works because these images have less vivid color than unblurred images. The saturation detector subdivides the input image into many regions. The size of each region is 25 pixels $\times$25 pixels. The average saturation ($\bar{F}^{saturation}$) of each image region is then measured, and blurry regions are detected if $\bar{F}^{saturation} >= 0.6$. Finally, the number of blurry regions are counted as N_s. Assuming N be the total number of image regions, an image is blurry if $N_s/N > 0.5$.

- **Edge filter:** Blurry images have less intensity variance when the colonoscope's lens is covered with water. These types of blurry images can be measured through edge detection. The Canny edge detection algorithm[39] is used to extract edges in a colonoscopy image. It converts the image into a binary image that includes only edge/non-edge pixels. I again use the subdivision strategy to decompose the binary image into several image regions. An image region is blurry if it contains no edge pixels. Let's define N_e to be the number of blurry regions. An image is classified as blurry if $N_e/N > 0.7$.

- **Intensity filter:** Extreme lighting conditions also generate blurry images. These types of blurry images can be measured through intensity distribution.

Again, the colonoscopy image is subdivided into a set of image regions. Let $\bar{I}$ be the average intensity value of an image region. Assuming the intensity range is $[0, 255]$, an image region is blurry if $\bar{I} > 220$ or $\bar{I} < 30$.

Intensity variance is also exploited to classify blurry image regions. Let I_{max} and I_{min} be the maximum and minimal intensities of an image region. Then the intensity variance is defined as

$$F^{intensity} = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} \tag{5.2}$$

An image is blurry if $\sum_{i=1}^{N} \frac{F^{intensity}(i)}{N} < 0.05$, where $i \in [1, N]$.

If the current image is blurry, the colonoscopy tracking system is temporally halted. This image is excluded and the next image is retried to determine if it is blurry too. This process continues until the first non-blurry image is found. If the current image is not blurry, determine whether the current image index is $k+1$. If yes, the colonoscopy tracking algorithm described in Chapter 4 is exploited. Otherwise, k^{th} colonoscopy image and the current clear colonoscopy image form a colonoscopy image pair for estimating large camera motion.

5.2.2 Example Demonstration

Fig. 5.4 illustrates an example of blurry image detection results on a colonoscopy image sequence using saturation and edge filters. All blurry images are accurately classified in this sequence. In addition, all experimental results in this dissertation also demonstrated that the proposed filters are sufficient to identify blurry images.

5.3 Region Flow Based Visual Motion

This section describes a region flow algorithm to accurately identify feature matching ranges for scale-invariant feature transform(SIFT)[147] algorithm. As a result,

Figure 5.4: *Results of blurry image detection* based on edge and saturation filters. Solid red and blue lines represent the response curves of edge and saturation filters, respectively. Thresholds 0.5(saturation) and 0.7(edge) are used for these two filters, corresponding to blue and dash lines.

SIFT features can be accurately matched between a colonoscopy image pair by employing identified matching ranges, and large visual motion can be computed by measuring relative image displacements between matched SIFT features. This section concludes with the comparison of SIFT feature matches with and without region flow.

5.3.1 Algorithm Description

Visual motion determination is comprised of region flow computation and SIFT feature matching. After blurry images are detected and eliminated, two non-consecutive, clear colonoscopy images are chosen for estimating large motion. Visual motion computation starts from these two images. Because colonoscopy images contain insufficient visual cues, SIFT feature descriptors are indistinct for feature matching and many false feature matches are generated in visual motion computation. Central to

large visual motion computation is the use of region flow, a dense feature matching strategy. It provides a framework to predefine feature matching size and to limit the search space for accurately matching corresponding features.

Let me first mathematically describe region flow calculation and assume $I_1(x, y)$ and $I_2(x, y)$ be a pair of selected images. In order to reduce the effects of illumination variance, images are normalized as

$$\hat{I}_1(x, y) = \frac{I_1(x, y) - \bar{I}_1}{\sigma_{I_1}} \qquad \hat{I}_2(x, y) = \frac{I_2(x, y) - \bar{I}_2}{\sigma_{I_2}} \tag{5.3}$$

where $\bar{I}_1$ and $\bar{I}_2$ are mean values, and σ_{I_1} and σ_{I_2} are standard deviations.

The similarity between two regions of $\hat{I}_1(x, y)$ and $\hat{I}_2(x, y)$ can be measured by normalized cross-correlation(NCC)[186],

$$NCC(x, y, r_x, r_y) = \iint \hat{I}_2(x + r_x, y + r_y)\hat{I}_1(x, y)dxdy \tag{5.4}$$

where $\overrightarrow{r} = (r_x, r_y)$ represents a region flow vector at point (x, y). The range of NCC values belongs to $[-1, 1]$, and two regions are matched if the NCC value is maximized. In order to fit NCC measurement into the minimization framework of region flow computation, I transform Eq. 5.4 into $1.0 - NCC(x, y, r_x, r_y)$. Similar to optical flow computation[99], a global energy function is proposed to compute region flow, within a minimization framework,

$$E(r_x, r_y) = \iint \underbrace{\min(|1.0 - NCC(x, y, r_x, r_y)|, \alpha)}_{\text{Data constraint}} + \lambda \underbrace{\min((|\nabla r_x|^2 + |\nabla r_y|^2), \beta)}_{\text{Smoothness constraint}} dxdy \tag{5.5}$$

where α and β are truncation values to prevent over-smoothing, and λ is a parameter to balance data and smoothness constraints.

Region flow is computed on the selected image pair. In order to reduce computational cost, the selected two colonoscopy images are first down-sampled by a factor of 4. The computation begins by calculating NCC as defined in Eq. 5.4 to match image regions in the down-sampled source image to corresponding regions in

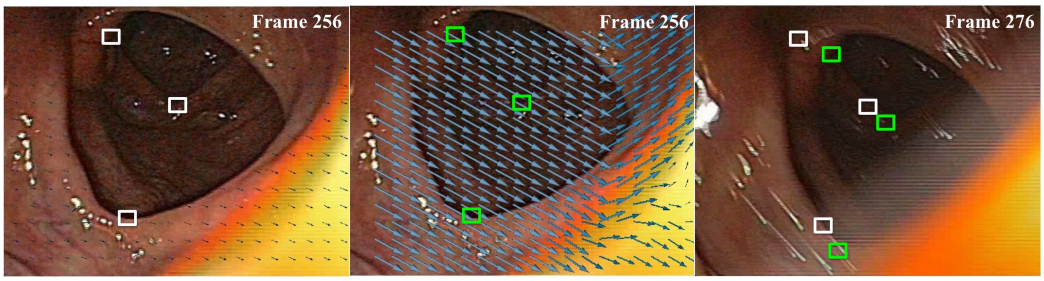
the down-sampled target image at every pixel. Its computational cost is $O(n^4)$ for $N \times N$ sized images.

Because data constraint does not involve derivatives, the Euler-Lagrange equation fails for Eq. 5.5. Note that the data term is represented in $L1$-form and the smoothness term in $L2$-form. The outliers in the data term can be suppressed and Eq. 5.5 can be alternatively minimized by Markov random field(MRF), introduced in chapter 3.

Efficient belief propagation(BP)[68], a Markov random field method, is applied to minimize Eq. 5.5. BP is a message spreading process around the 4-connected image graph. Messages are being iteratively passed through all graph nodes. In order to reduce computational cost, three-level undirected graphs are constructed at the sizes of $N/4 \times N/4$, $N/2 \times N/2$, and $N \times N$. A generalized distance transform[69] is exploited to update BP messages. Each node in the undirected graph is initialized with minimum NCC matching value, $\min(|1.0 - NCC(x, y, r_x(\boldsymbol{q}), r_y(\boldsymbol{q}))|, \alpha)$. A quadratic distance metric[68] is used to update messages. The two-dimensional transform is computed by first performing a one-dimensional transform along each column of the image, and then performing a one-dimensional transform along each row of the result (or vice versa). Therefore, region flow computation can be efficiently computed through a two-pass one-dimensional distance transform with respect to r_x and r_y components.

The algorithm spreads BP messages for five iterations at the current graph level, and uses the estimated results to initialize region flow vectors in the fine graph level. The number of iterations(five), is experimentally determined. The same process is repeated through the finest level, and then region flow is computed. Fig. 5.5b illustrates an example colonoscopy image with overlaid region flow vectors. The region flow vectors represent the visual motion between Fig. 5.5b and Fig. 5.5c. Three corner points are manually selected and indicated by white boxes in Fig. 5.5a and green boxes in Fig. 5.5b. The white squares in Fig. 5.5c represent corresponding pairs generated by

the optical flow field. They do not match up with the green squares, which roughly represent the positions of the true corresponding pairs.

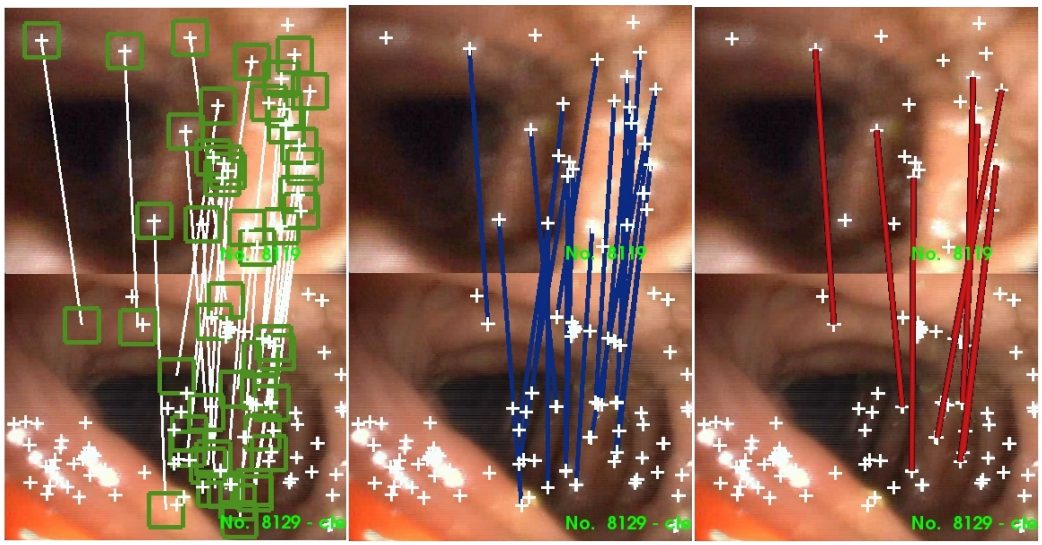


(a) (b) (c)

Figure 5.5: *Region flow vs. optical flow for describing large motion*, (a) source image with overlaid optical flow vectors, (b) source image with overlaid region flow vectors, (c) target image after a 20 frame blurry sequence. White and green squares in the target image represent 3 selected regions in the image, and correspond to the white and green squares in the source images, after application of optical and region flow vectors. Region flow does a better job tracking the image motion. The lengths of the vectors in the source images represent the magnitude of the motion velocity.

Fig. 5.6 illustrates the process of feature matching based on region flow. Two sets of SIFT feature points are detected on the *original sized* colonoscopy image pair, illustrated as white crosses in Fig. 5.6. The choice of SIFT algorithm[147] is because this method usually generates a sufficient number of feature points. This property is useful for colonoscopy tracking, considering that colonoscopy images often lack sufficient visual cues.

Region-to-Region Matching. In this step, corresponding regions are identified using the region flow field and a local matching procedure. The corresponding regions of SIFT feature points in the target image are identified using the region flow vectors and a local neighborhood search. In Fig. 5.6a, the green squares joined by the white lines represent corresponding regions containing at least one SIFT feature point in the source image and 0 or more SIFT feature points in the target image. In the implementation, the mapped region is locally adjusted using NCC as a metric to find the best region match.



(a) (b) (c)

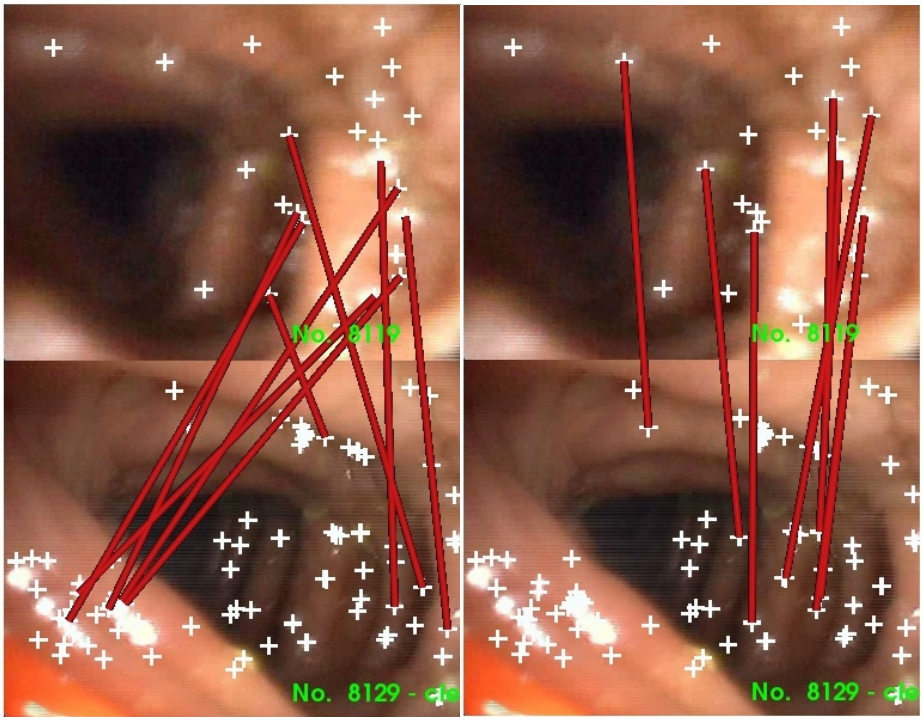
Figure 5.6: *Corresponding pairs computation.* Top and bottom images represent images before and after the blurry image sequence, (a) Region-to-Region matching. Green squares indicate the matched regions using the region flow field. Local search using NCC is performed to find the best region pair. (b) Point-to-Point feature matching. Using SIFT descriptor as a metric, the best SIFT feature point pair is determined between source and target regions. (c) False feature match rejection using epipolar geometry.

Point-to-Point Feature Matching. In this step, each corresponding region pair is refined to a corresponding point pair. If the target region does not contain a SIFT feature point, it is removed. For target regions with multiple SIFT feature point candidates, the candidate with the closest SIFT descriptor(a distance metric) is chosen as the best candidate. Fig. 5.6b illustrates the selected feature point pairs after this step.

False Feature Match Rejection. With the chosen feature point pairs, epipolar geometry is built using the RANSAC algorithm[155]. Outliers that do not satisfy the epipolar geometry constraints are removed, as seen in Fig. 5.6c.

5.3.2 Example Demonstration

Fig. 5.7 illustrates the comparison of the same image pair using original SIFT feature matching and region flow based SIFT feature matching. It can be seen that original SIFT feature matching generates significant mismatches in Fig. 5.7a because the matching size is uncertain and the SIFT feature descriptor is indistinct. In comparison, region flow generates accurate SIFT feature matches in Fig. 5.7b because region flow vectors predefine feature matching ranges and limit false feature matches. As a result, visual motion is accurately computed by measuring relative image displacements between precisely matched SIFT features.



(a) (b)

Figure 5.7: *Comparison between (a) original SIFT feature matching and (b) region flow based SIFT feature matching.* Top and bottom images represent images before and after the blurry image sequence. Obviously, original SIFT matching using only (locally defined) SIFT feature descriptors, contains significant errors.

5.4 Incremental Egomotion

Large visual motion computed by region flow is next imported into an incremental egomotion estimation strategy. This strategy estimates significant camera motion from the selected colonoscopy image pair, so as to continuously track OC images. It is evaluated by comparing it with the FOE-based egomotion estimation algorithm described in chapter 4.

5.4.1 Algorithm Description

Incremental egomotion estimation encompasses visual motion subdivision and egomotion estimation. Most existing egomotion estimation algorithms, including the FOE-based egomotion estimation described in chapter 4, can accurately estimate small camera motion. But they fail when camera motion is large. They are derived from the basic equation(Eq. 3.26) relating camera motion parameters and optical flow and this equation is valid if the camera motion is small. This property is demonstrated in Appendix G. Unfortunately, Eq. 3.26 fails in a colonoscopy image pair interrupted by blurry images. A better strategy, incremental egomotion estimation, is proposed to estimate significant camera motion by artificially decomposing large visual motion into a sequence of optical flow fields through a PDE-based framework. Significant camera motion can thereby be incrementally estimated by using all optical flow fields. The advantage of this PDE-based framework is that it provides a general framework to enable existing egomotion estimation algorithms to compute significant camera motion.

I first mathematically define this PDE-based framework. Assuming a colonoscopy video stream $I(x, y, t)$ has a blurry image sequence at $[t_0, t_n]$, the goal is to estimate $\overrightarrow{T}$ and $\overrightarrow{R}$ between $I(x, y, t_0)$ and $I(x, y, t_n)$. Similar to optical flow computation, two corresponding points $\boldsymbol{p}_0 = (x_0, y_0, t_0)$ and $\boldsymbol{p}_n = (x_n, y_n, t_n)$ satisfy the intensity

constancy model.

$$I(\boldsymbol{p}_n) = I(\boldsymbol{p}_0) \tag{5.6}$$

$\boldsymbol{p}_0$ and $\boldsymbol{p}_n$ are also insensitive to the illumination variance, and fulfill

$$\nabla I(\boldsymbol{p}_n) = \nabla I(\boldsymbol{p}_0) \tag{5.7}$$

An additional data term is SIFT feature matches

$$ST(\boldsymbol{p}_n) = ST(\boldsymbol{p}_0) \tag{5.8}$$

where $ST(\boldsymbol{p})$ is a SIFT feature descriptor if $\boldsymbol{p}_0$ and $\boldsymbol{p}_n$ are two matched SIFT feature points.

Next, I introduce two smoothness terms. Let $\overrightarrow{u} = (u_x, u_y)$ be a large optical flow vector between the selected image pair interrupted by a blurry image sequence. $\overrightarrow{u}$ is the sum of a sequence of optical flow vectors during visual motion subdivision. $\overrightarrow{s} = (s_x, s_y)$ is the visual motion vector between two matched SIFT features. The first smoothness term assumes that large displacement optical flow vectors vary smoothly. It is defined as

$$SM1 = (\nabla u_x)^2 + (\nabla u_y)^2 \tag{5.9}$$

and the second term assumes that large displacement optical flow vectors approximate the visual motion vectors from SIFT feature matches. It is defined as

$$SM2 = (u_x - s_x)^2 + (u_y - s_y)^2 \tag{5.10}$$

All these data and smoothness terms are integrated into a variational function,

which leads to

$$
\begin{aligned}
E(u_x, u_y) = \iint & \underbrace{\Psi((I(x+u_x, y+u_y, t+1) - I(x,y,t))^2)}_{\text{Intensity constraint}} \\
& + \gamma \underbrace{\Psi((\nabla I(x+u_x, y+u_y, t+1) - \nabla I(x,y,t))^2)}_{\text{Gradient constraint}} \\
& + \delta(x,y) \underbrace{|ST(x+s_x, y+s_y, t+1) - ST(x,y,t)|^2}_{\text{SIFT constraint}} \\
& + \underbrace{\alpha\Psi((\nabla u_x)^2 + (\nabla u_y)^2) + \beta\Psi(|u_x - s_x|^2 + |u_y - s_y|^2)}_{\text{Smoothness constraint}} dxdy
\end{aligned}
\tag{5.11}
$$

where γ and α are constants, and

$$
\delta(x,y) = \begin{cases} 1 & \text{if } ST(x+s_x, y+s_y, t+1) \text{ and } ST(x,y,t) \text{ are matched} \\ 0 & \text{otherwise} \end{cases}
\tag{5.12}
$$

$\Psi(s^2) = \sqrt{s^2 + \epsilon^2}, \epsilon = 0.001$ allows the computation to handle occlusions and other non-Gaussian deviations of the matching criterion. In order to ensure the large visual motion vector $\overrightarrow{s}$ is evenly decomposed into a sequence of optical flow vectors, β has to be adjusted inversely to the difference between $\overrightarrow{s}$ and $\overrightarrow{u}$ in Eq. 5.11. The embedding of SIFT feature matches into Eq. 5.11 is a key contributor to avoid local minimums, while intensity and gradient assumptions preserve local details.

However, minimizing Eq. 5.11 presents many challenges. First, Eq. 5.11 contains a discrete term, the invariance of SIFT feature correspondence, which prevents using the Euler-Lagrange equations. Second, this equation is non-linear due to nonlinear intensity and gradient constancy models. Finally, it is also non-convex because there are several local minima in this energy function. In this section, I investigate some advanced numerical techniques to tackle the minimization challenges: region-to-region image matching to remove discrete data term, multi-scale image representation, and sequential linearization.

Discrete term removal: The discrete data term, SIFT feature correspondence, must be removed from Eq. 5.11 while retaining its influence on the final minimization results. Region-to-region image matching on every SIFT feature correspondence serves this purpose. For each SIFT feature correspondence, two regions centered at matched feature points are first built. The size of each region is 41 pixels × 41 pixels. Brox's optical flow computation method[27] is employed to compute relative image displacements between two regions, which are also called patch flow. Fig. 5.8 shows the patch flow between a SIFT feature correspondence. Fig. 5.8(a) illustrates a colonoscopy image pair, where white crosses indicate the detected SIFT feature points and blue lines represent the matched SIFT feature correspondences. Fig. 5.8(b) and (c) give two regions centered at the matched SIFT features near a polyp. Fig. 5.8(d) illustrates the warped patch of Fig. 5.8(c) in terms of the patch flow. The image displacements between two regions are accurately computed because the warped region in the target image is very similar to the region in the source image.

Large displacement optical flow is initialized by the addition of original SIFT feature displacements and the estimated patch flow, which is $\overrightarrow{s} + \overrightarrow{p}$, where $\overrightarrow{s}$ is the visual motion vector from SIFT feature matches, and $\overrightarrow{p}$ is the patch flow vector. Assuming $\overrightarrow{g} = \overrightarrow{s} + \overrightarrow{p}$ to be the transformed visual motion vector from SIFT feature matches after region-to-region image matching and removing discrete SIFT feature matches, Eq. 5.11 is rewritten as

$$\begin{aligned} E(u_x, u_y) = \iint & \underbrace{\Psi((I(x+u_x, y+u_y, t+1) - I(x,y,t))^2)}_{\text{Intensity constraint}} \\ & + \gamma \underbrace{\Psi((\nabla I(x+u_x, y+u_y, t+1) - \nabla I(x,y,t))^2)}_{\text{Gradient constraint}} \\ & + \underbrace{\alpha \Psi((\nabla u_x)^2 + (\nabla u_y)^2) + \beta \Psi(|u_x - g_x|^2 + |u_y - g_y|^2)}_{\text{Smoothness constraint}} dxdy \end{aligned} \tag{5.13}$$

Both data and smoothness terms are continuous in Eq. 5.13 and therefore the

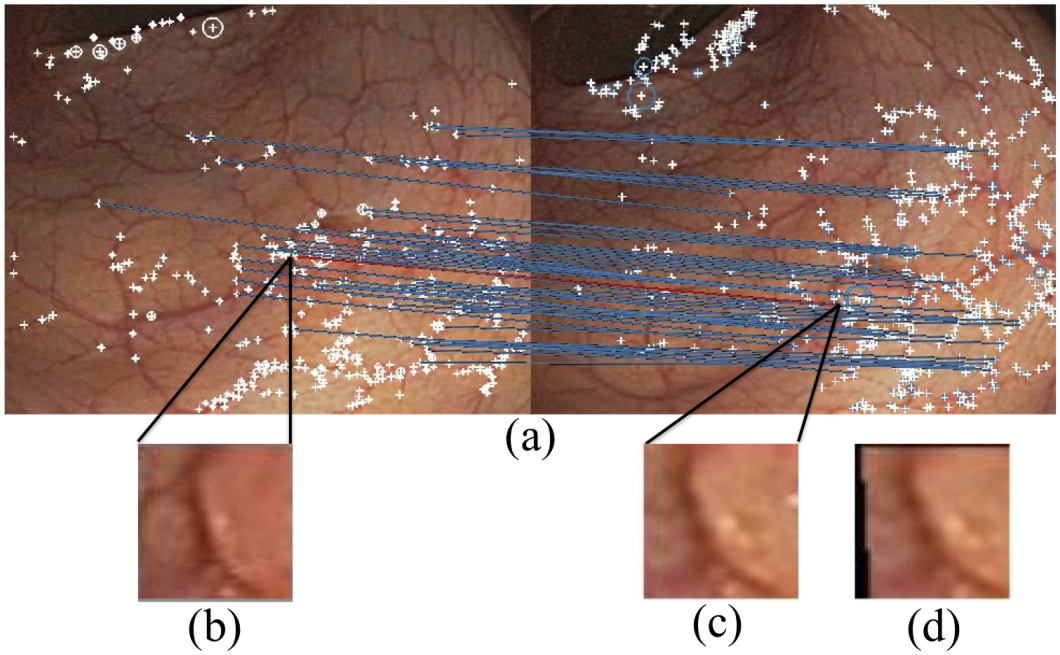


Figure 5.8: *Region-to-region image matching*. (a) SIFT feature matches between two OC images where white crosses indicate the detected SIFT features and blue lines represent matched feature pairs; (b) an image region centered at the selected feature point in the left image; (c) the corresponding image region in the right image; (d) the warped image region in terms of the patch flow. The warped region is very similar to the region in the left image, which means that the patch flow is accurately estimated.

Euler-Lagrange equation can be applied. Meanwhile, importing the transformed visual motion vectors into Eq. 5.13 also initializes large displacement optical flow vectors near the actual visual motion. Region-to-region image matching can remove the discrete SIFT feature match term. It also keeps the influence of this discrete term on the final minimization results. After the discrete term is removed, I obtain a continuous energy function shown in Eq. 5.13.

Multi-scale image representation: Multi-scale space is then constructed on the colonoscopy images for minimizing the continuous energy function from the previous step, in order to reduce the non-convexity influence. Non-convexity means that an energy function has local minima, in addition to the global minimum value. If large displacement optical flow is improperly initialized, the minimization process might fail to steer the energy function towards the global minimum.

Local minima are generally caused by fine image structures. Multi-scale image representation is an efficient tool for removing fine structures. Because an optical flow vector in Eq. 3.26 is non-linearly related to image coordinates, the two camera motion parameters estimated from different images resolutions are non-linearly related and thus cannot be directly added. In order to simplify the addition of two camera motion parameters, multi-scale space is built by iteratively convolving the original-sized image with the Gaussian function.

Let $\overrightarrow{u}^k = (u_x^k, u_y^k)$ be the large displacement optical flow vector at the scale level k and $\overrightarrow{u}^0 = (0, 0)$. The minimization process starts from the coarse scale and gradually assigns optical flow results to the fine scale.

Sequential linearization: After the discrete term has been removed from Eq. 5.11, the Euler-Lagrange equations for Eq. 5.11 with respect to $\overrightarrow{u}$ are given by

$$\begin{aligned}
&\Psi'((\partial_t I)^2)\partial_t I\partial_x I + \gamma\Psi'((\partial_{xt} I)^2 + (\partial_{yt} I)^2)(\partial_{xx} I\partial_{xt} I + \partial_{xy} I\partial_{yt} I) \\
&+ \beta\Psi'(|u_x - g_x|^2 + |u_y - g_y|^2)(u_x - g_x) - \alpha\mathrm{div}(\Psi'(|\nabla u_x|^2 + |\nabla u_y|^2)\nabla u_x) = 0 \\
&\Psi'((\partial_t I)^2)\partial_t I\partial_y I + \gamma\Psi'((\partial_{xt} I)^2 + (\partial_{yt} I)^2)(\partial_{xy} I\partial_{xt} I + \partial_{yy} I\partial_{yt} I) \\
&+ \beta\Psi'(|u_x - g_x|^2 + |u_y - g_y|^2)(u_y - g_y) - \alpha\mathrm{div}(\Psi'(|\nabla u_x|^2 + |\nabla u_y|^2)\nabla u_y) = 0
\end{aligned} \tag{5.14}$$

where

$$\begin{aligned}
\Psi'(x^2) &= \frac{1}{2\sqrt{x^2 + \epsilon^2}} \qquad \partial_x I = \frac{\partial I_2(x + u_x, y + u_y, t + 1)}{\partial x} \\
\partial_{xy} I &= \frac{\partial^2 I_2(x + u_x, y + u_y, t + 1)}{\partial x\partial y} \qquad \partial_y I = \frac{\partial I_2(x + u_x, y + u_y)}{\partial y} \\
\partial_{xx} I &= \frac{\partial^2 I_2(x + u_x, y + u_y, t + 1)}{\partial x^2} \qquad \partial_{yy} I = \frac{\partial^2 I_2(x + u_x, y + u_y, t + 1)}{\partial y^2} \\
\partial_t I &= I_2(x + u_x, y + u_y, t + 1) - I_1(x, y, t) \\
\partial_{xt} I &= \frac{\partial I_2(x + u_x, y + u_y, t + 1)}{\partial x} - \frac{\partial I_1(x, y, t)}{\partial x}
\end{aligned}$$

$$\partial_{yt} I = \frac{\partial I_2(x + u_x, y + u_y, t + 1)}{\partial y} - \frac{\partial I_1(x, y, t)}{\partial y}$$

Let

$$\begin{aligned} \Psi_I &= \Psi((\partial_t I)^2) \\ \Psi_G &= \Psi((\partial_{xt} I)^2 + (\partial_{yt} I)^2) \\ \Psi_M &= \Psi((u_x - g_x)^2 + (u_y - g_y)^2) \\ \Psi_S &= \Psi(|\nabla u_x|^2 + |\nabla u_y|^2) \end{aligned} \tag{5.15}$$

Note that Eq. 5.14 is nonlinear because $\Psi'_I, \Psi'_G, \Psi'_M$, and Ψ'_S are nonlinear terms with respect to $\overrightarrow{u}$. Sequential linearization is used to remove non-linearity from the Euler-Lagrange equation. It is represented as two nested fixed point iterations to gradually remove non-linearity in Eq. 5.14. Let l denote the outer iteration index and k the current image scale level. The purpose of the outer iteration is the decomposition of large displacement optical flow vector $u_x^{k,l+1} = u_x^{k,l} + du_x^{k,l}$ and $u_y^{k,l+1} = u_y^{k,l} + du_y^{k,l}$ through Taylor expansion. $\overrightarrow{u}^{k,l+1}$ is subdivided into a known large displacement optical flow vector $\overrightarrow{u}^{k,l}$ in the previous iteration and an unknown incremental optical flow $d\overrightarrow{u}^{k,l} = (du_x^{k,l}, du_y^{k,l})$. Correspondingly, nonlinear terms, $\Psi'_I, \Psi'_G, \Psi'_M$, and Ψ'_S, are also transformed in terms of large displacement optical flow decomposition. Let the index of inner iteration be m. The incremental optical flow vector $d\overrightarrow{u}^{k,l}$ is rewritten as $d\overrightarrow{u}^{k,l,m} = (du_x^{k,l,m}, du_y^{k,l,m})$. In the inner iteration, $(\Psi'_I)^{k,l,m}$, $(\Psi'_G)^{k,l,m}$, $(\Psi'_M)^{k,l,m}$ and $(\Psi'_S)^{k,l,m}$ are further derived and only related to incremental optical flow $(du_x^{k,l,m}, du_y^{k,l,m})$ at previous iteration m. Eq. 5.14 is finally converted into a linear equation with respect to $(du_x^{k,l,m+1}, du_y^{k,l,m+1})$ after sequential linearization. The details of this technique can be found in Appendix H. Therefore, each image point has two linear equations with respect to $(du_x^{k,l,m+1}, du_y^{k,l,m+1})$, which formulates a massive sparse linear system.

Visual motion subdivision: After non-linearity, non-convexity, and discrete terms have been removed, large displacement optical flow can be computed by adding a sequence of incremental optical flow vectors. In order to efficiently estimate incremental optical flow, the successive over-relaxation method[236] is used to solve the massive linear system, placing this computation in the most inner iteration. Let n indicate the index for this iteration.

$$
\begin{aligned}
(du_x)_{\boldsymbol{p}}^{k,l,m,n+1} &= (1-\omega)(du_x)_{\boldsymbol{p}}^{k,l,m,n} \\
&+ \omega \left(\sum_{\boldsymbol{q}\in\mathcal{N}^-(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} \left((u_x)_{\boldsymbol{q}}^{k,l,m} + (du_x)_{\boldsymbol{q}}^{k,l,m,n} \right) + \sum_{\boldsymbol{q}\in\mathcal{N}^+(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} \left((u_x)_{\boldsymbol{q}}^{k,l,m} + (du_x)_{\boldsymbol{q}}^{k,l,m,n} \right) \right. \\
&- \sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} (u_x)_{\boldsymbol{p}}^{k,l,m} - \frac{1}{\alpha} (\Psi'_I)_{\boldsymbol{p}}^{k,l,m} (\partial_x I^{k,l,m})_{\boldsymbol{p}} \left((\partial_y I^{k,l,m})_{\boldsymbol{p}} (du_y)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_t I^{k,l,m})_{\boldsymbol{p}} \right) \\
&- \frac{\gamma}{\alpha} (\Psi'_G)_{\boldsymbol{p}}^{k,l,m} \left((\partial_{xx} I^{k,l,m})_{\boldsymbol{p}} \left((\partial_{xy} I^{k,l,m})_{\boldsymbol{p}} (du_y)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_{xt} I^{k,l,m})_{\boldsymbol{p}} \right) \right. \\
&+ \left. (\partial_{xy} I^{k,l,m})_{\boldsymbol{p}} \left((\partial_{yy} I^{k,l,m})_{\boldsymbol{p}} (du_y)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_{yt} I^{k,l,m})_{\boldsymbol{p}} \right) \right) - \left. \frac{\beta}{\alpha} (\Psi'_M)_{\boldsymbol{p}}^{k,l,m} ((u_x^{k,l,m})_{\boldsymbol{p}} - (g_x)_{\boldsymbol{p}}) \right) \\
&/ \left(\sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} + \frac{1}{\alpha} (\Psi'_I)_{\boldsymbol{p}}^{k,l,m} (\partial_x I^{k,l,m})_{\boldsymbol{p}}^2 + \frac{\gamma}{\alpha} (\Psi'_G)_{\boldsymbol{p}}^{k,l,m} ((\partial_{xx} I^{k,l,m})_{\boldsymbol{p}}^2 + (\partial_{xy} I^{k,l,m})_{\boldsymbol{p}}^2) + \frac{\beta}{\alpha} (\Phi'_M)_{\boldsymbol{p}}^{k,l,m} \right) \\
(du_y)_{\boldsymbol{p}}^{k,l,m,n+1} &= (1-\omega)(du_y)_{\boldsymbol{p}}^{k,l,m,n} \\
&+ \omega \left(\sum_{\boldsymbol{q}\in\mathcal{N}^-(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} \left((u_y)_{\boldsymbol{q}}^{k,l,m} + (du_y)_{\boldsymbol{q}}^{k,l,m,n} \right) + \sum_{\boldsymbol{q}\in\mathcal{N}^+(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l} \left((u_y)_{\boldsymbol{q}}^{k,l,m} + (du_y)_{\boldsymbol{q}}^{k,l,m,n} \right) \right. \\
&- \sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} (u_y)_{\boldsymbol{p}}^{k,l,m} - \frac{1}{\alpha} (\Psi'_I)_{\boldsymbol{p}}^{k,l,m} (\partial_y I^{k,l,m})_{\boldsymbol{p}} \left((\partial_x I^{k,l,m})_{\boldsymbol{p}} (du_x)_{\boldsymbol{p}}^{k,l,m,n+1} + (\partial_t I^{k,l,m})_{\boldsymbol{p}} \right) \\
&- \frac{\gamma}{\alpha} (\Psi'_G)_{\boldsymbol{p}}^{k,l,m} \left((\partial_{xy} I^{k,l,m})_{\boldsymbol{p}} \left((\partial_{xx} I^{k,l,m})_{\boldsymbol{p}} (du_x)_{\boldsymbol{p}}^{k,l,m,n+1} + (\partial_{xt} I^{k,l,m})_{\boldsymbol{p}} \right) \right. \\
&+ \left. (\partial_{yy} I^{k,l,m})_{\boldsymbol{p}} \left((\partial_{xy} I^{k,l,m})_{\boldsymbol{p}} (du_x)_{\boldsymbol{p}}^{k,l,m,n+1} + (\partial_{yt} I^{k,l,m})_{\boldsymbol{p}} \right) \right) - \left. \frac{\beta}{\alpha} (\Psi'_M)_{\boldsymbol{p}}^{k,l,m} ((u_y^{k,l,m})_{\boldsymbol{p}} - (g_y)_{\boldsymbol{p}}) \right) \\
&/ \left(\sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} + \frac{1}{\alpha} (\Psi'_I)_{\boldsymbol{p}}^{k,l,m} (\partial_y I^{k,l,m})_{\boldsymbol{p}}^2 + \frac{\gamma}{\alpha} (\Psi'_G)_{\boldsymbol{p}}^{k,l,m} \left((\partial_{xy} I^{k,l,m})_{\boldsymbol{p}}^2 + (\partial_{yy} I^{k,l,m})_{\boldsymbol{p}}^2 \right) + \frac{\beta}{\alpha} (\Phi'_M)_{\boldsymbol{p}}^{k,l,m} \right)
\end{aligned}
\tag{5.16}
$$

Here, $\mathcal{N}^+(\boldsymbol{p})$ denotes the neighbors $\boldsymbol{q}$ of $\boldsymbol{p}$ if the indices of $\boldsymbol{q}$ are larger than those of $\boldsymbol{p}$. $\mathcal{N}^-(\boldsymbol{p})$ denotes the neighbors $\boldsymbol{q}$ of $\boldsymbol{p}$ if the indices of $\boldsymbol{q}$ are smaller than those of $\boldsymbol{p}$. $\omega \in (0, 2)$ is the relaxation parameter to guide the convergence of the massive linear system. As suggested by Young[236], values close to 2 give the best performance. In my implementation, it is chosen as 1.99.

Iteratively, large displacement optical flow $\overrightarrow{u}$ is incrementally estimated and represented as

$$\overrightarrow{u}^{k,l,m+1} = \overrightarrow{u}^{0,0,0} + \sum_{k,l,m} du^{k,l,m} \tag{5.17}$$

where $\overrightarrow{u}^{0,0,0} = (0,0)$. Eq. 5.10 implies that visual motion vectors from SIFT feature matches approximate large displacement optical flow vectors. In other words, $\overrightarrow{s} \approx \overrightarrow{u}^{k,l,m+1}$. Because $\overrightarrow{u}^{k,l,m+1}$ is comprised of a sequence of incremental optical flow vectors $du^{k,l,m}$ in Eq. 5.17, visual motion vectors from SIFT feature matches can be decomposed into a set of incremental optical flow vectors.

$$\overrightarrow{s} \approx \overrightarrow{u}^{0,0,0} + \sum_{k,l,m} du^{k,l,m} \tag{5.18}$$

Egomotion Estimation: A modified FOE based egomotion estimation method is used for computing incremental camera rotation parameter $d\overrightarrow{T}^{k,l,m}$ and incremental camera translation parameter $d\overrightarrow{R}^{k,l,m}$, from incremental optical flow vectors computed in the previous step. The FOE is determined by computing incremental optical flow vector difference in the image regions centered at matched SIFT feature points. The choice of these image regions is because incremental optical flow vectors are accurate in these regions. Also, these regions are near depth discontinuities.

A polar coordinate can be constructed by using the FOE as the origin. $d\overrightarrow{R}^{k,l,m}$ is first estimated from the incremental optical flow vectors at matched SIFT feature points, as was described in the previous chapter. After $d\overrightarrow{R}^{k,l,m}$ is estimated, camera rotation components are excluded from the incremental optical flow vectors. $d\overrightarrow{T}^{k,l,m}$ is next determined by using the incremental optical flow vectors resulting from camera translation alone and depth values from a colon-like cylinder model.

Assume $\overrightarrow{T}^{k,l,m}$ and $\overrightarrow{R}^{k,l,m}$ to be the estimated camera motion parameters at the previous iteration. Incremental camera motion parameters are added to the estimated

parameters and represented as

$$\begin{aligned} \overrightarrow{T}^{k,l,m+1} &= \overrightarrow{T}^{k,l,m} + d\overrightarrow{T}^{k,l,m+1} \\ \overrightarrow{R}^{k,l,m+1} &= \overrightarrow{R}^{k,l,m} + d\overrightarrow{R}^{k,l,m+1} \end{aligned} \tag{5.19}$$

After camera motion parameters are accumulated, the locations of matched SIFT feature points are iteratively updated by adding their locations to the incremental optical flow vectors. Meanwhile, their depth values are dynamically changed by using bilinear interpolation.

The purpose of this step is to determine whether incremental egomotion estimation should be repeated. Let the finest image scale level be K. The maximum numbers of outer and inner iterations of sequential linearization are L and M, respectively. If the current scale index $k \neq \mathsf{K}$, or inner iteration index $m \neq \mathsf{M}$, or outer iteration index $l \neq \mathsf{L}$, then repeat visual motion subdivision and incremental egomotion estimation. Otherwise, output the camera motion parameters. Algorithm 2 summarizes the complete incremental egomotion estimation.

5.4.2 Example Demonstration

I have tested incremental egomotion estimation on two clinical colonoscopy image sequences containing significant camera motion. The estimated camera motion parameters are also compared against results from the FOE-based egomotion estimation described in chapter 4. Because actual camera motion is unknown, the estimation error was qualitatively evaluated by visually inspecting the co-aligned VC images.

Fig. 5.9 shows an example of a colonoscopy image pair interrupted by blurry images. There is significant egomotion between these two images. Fig. 5.9a shows the co-aligned OC and VC images prior to the blurry images. Fig. 5.9b illustrates co-aligned OC and VC images *after* the blurry images, by using the FOE-based egomotion estimation method. Fig. 5.9c displays the results using the incremental

Algorithm 2: Incremental Egomotion Estimation

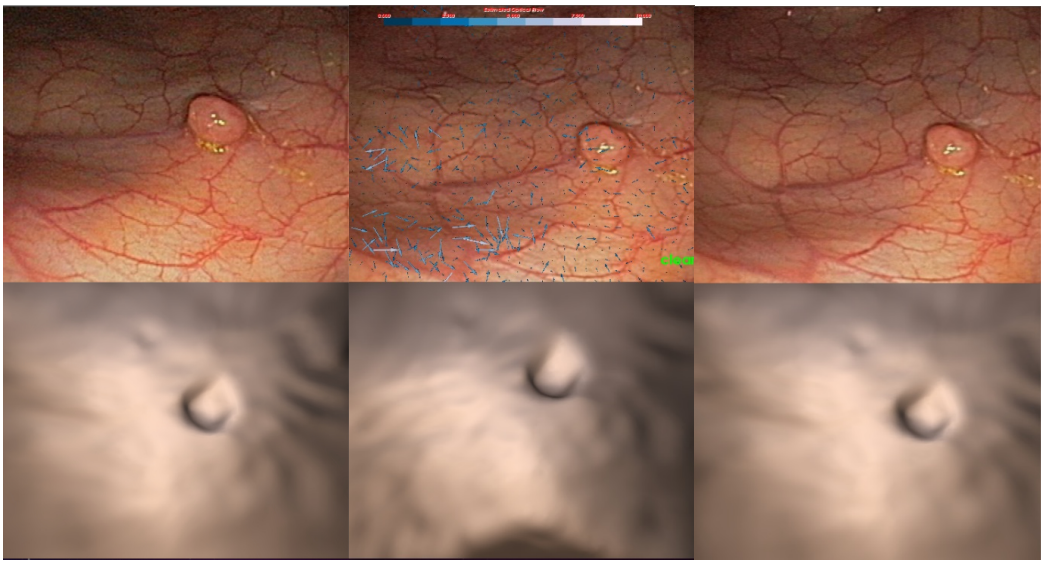
Data: $I(x, y, t)$ at t_0 and t_n

Result: Motion parameters $\overrightarrow{T}$ and $\overrightarrow{R}$.

Perform region-to-region image matching to initialize SIFT feature correspondences and update SIFT feature correspondences to obtain $\overrightarrow{g}$;
Build K-level multi-scale image representation for $I(x, y, t_0)$ and $I(x, y, t_n)$;
for $k \leftarrow 1$ **to** K **do**
    Initialize k-th level large displacement optical flow;
    for $l \leftarrow 1$ **to** L **do**
        Compute derivatives $\partial_t I^{k,l}, \partial_{xt} I^{k,l}$, and $\partial_{yt} I^{k,l}$ through bilinear interpolation;
        for $m \leftarrow 1$ **to** M **do**
            Compute diffusion terms $\Psi_I'^{k,l,m}, \Psi_G'^{k,l,m}, \Psi_M'^{k,l,m}$ and $\Psi_S'^{k,l,m}$;
            Initialize incremental optical flow $d\overrightarrow{u}^{k,l,m} = 0$;
            for $n \leftarrow 1$ **to** N **do**
                Update incremental optical flow $d\overrightarrow{u}^{k,l,m,n}$ in Eq. 5.16 at every pixel through SOR Method;
            Use incremental flow field $d\overrightarrow{u}^{k,l,m}$ to determine the FOE;
            Compute camera motion parameters $\Delta\overrightarrow{T}^{k,l,m}$ and $\Delta\overrightarrow{R}^{k,l,m}$ in terms of incremental optical flow vectors near SIFT feature points;
            Update large displacement optical flow $\overrightarrow{u}^{k,l,m+1} = \overrightarrow{u}^{k,l,m} + d\overrightarrow{u}^{k,l,m}$;
            Update camera motion parameters $\overrightarrow{T}^{k,l,m+1} = \overrightarrow{T}^{k,l,m} + \Delta\overrightarrow{T}^{k,l,m}$ and $\overrightarrow{R}^{k,l,m+1} = \overrightarrow{R}^{k,l,m} + \Delta\overrightarrow{R}^{k,l,m}$;

egomotion method. In this image sequence, there is a round polyp for reference in the sigmoid colon, as seen in the top row of Fig. 5.9. It can be seen that the polyp matches more closely (in position) in Fig. 5.9c, in comparison to Fig. 5.9b; thus, the colonoscope's motion is being tracked more closely using the incremental estimation scheme.

Fig. 5.10 illustrates another example in the descending colon during a biopsy procedure. The dominating visual motion between the colonoscopy image pair is downward displacement (as seen by the marked polyps). Figs. 5.10b and 5.10c illustrate respectively the results using the FOE-based egomotion estimation method and the incremental egomotion estimation method. It is clearly seen that the downward displacement is more accurately computed and displayed in the virtual images (bot-



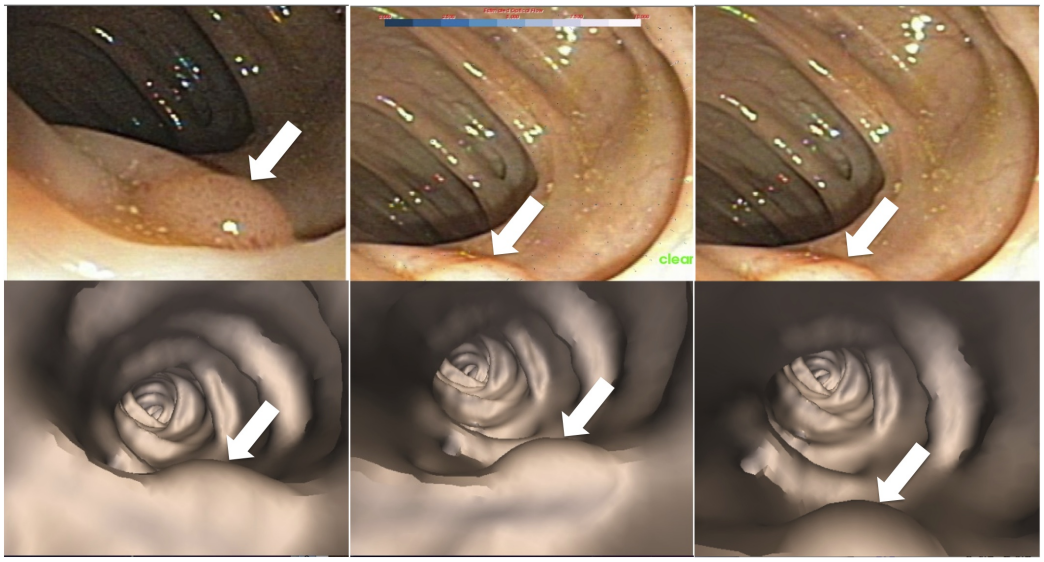
(a) (b) (c)

Figure 5.9: *Egomotion estimation results on a polyp biopsy in the sigmoid colon.* Note the rightward shift of the polyp between the top OC images of (a) and (c), (a) An OC image prior to blurry images, and its co-aligned VC image; (b) egomotion estimation results (after blurry images), using the FOE-based egomotion estimation method. The polyp has shifted up and a little rightward; (c) egomotion estimation results using the incremental method. The polyp in the VC image is nearly in the same location as in the OC image.

tom frames of Fig. 5.10b and Fig. 5.10c) by using the incremental egomotion method. Note that there is a scale variation in the polyp size, likely due to deformation. Deformation is a significant issue in colonoscopy tracking, reflecting changes between the pre-acquired CT and the OC images.

5.5 Phantom Validation

Straight and curved phantom image sequences described in chapter 4 are also used to validate the region flow based strategy for large motion estimation. This section attempts to validate the accuracy of my algorithm in estimating significant camera motion. Only image sequences at the speed of about $20mm/sec$ are utilized. In each straight phantom image sequence, I choose 19 images to formulate a new straight phantom image sequence. Each image represents the instant when the colonoscope



(a) (b) (c)

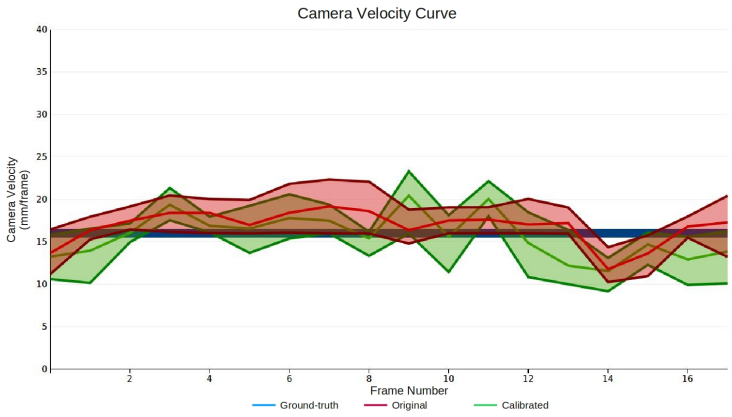
Figure 5.10: *Egomotion estimation results on a polyp biopsy in the descending colon.* Note the downward displacement of the marked polyp between (a) and (c), representing frames before and after blurry images. (a) Co-aligned OC and VC images before a blurry sequence. (b) Egomotion estimation results (after blurry images) using the FOE-based egomotion estimation method. The polyp in the VC image has hardly moved. (c) Egomotion estimation results using the incremental estimation method. The polyp has shifted downward, in comparison to the VC image in (a).

has just arrived at one of the 19 marked locations illustrated in Fig. 4.13a. Similarly, 13 phantom images corresponding to 13 marked locations shown in Fig. 4.13b are selected from each curved phantom image sequence, to comprise a new curved phantom image sequence. Phantom images between two consecutive selected images are eliminated, simulating the exclusion of blurry images. The distance between two sequentially marked locations is $16mm$ in the straight phantom and $23.88mm$ in the curved phantom. Therefore, the colonoscope moves $16mm$ between any two consecutive images in the new straight phantom image sequences. The colonoscope moves $23.88mm$ in the new curved phantom image sequences. The total displacements are $12 \times 18 = 288mm$ in the new straight phantom and $23.88 \times 12 = 286.56mm$ in the new curved phantom.

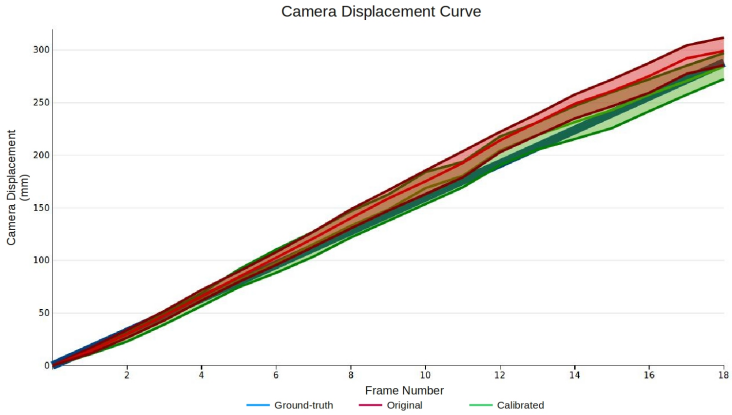
Fig. 5.11a and Fig. 5.11b show the tracking results of the incremental egomotion

estimation algorithm on the new straight phantom image sequences. Here, the camera velocity is measured in ***mm/frame***, not $mm/second$. The measurement is chosen because region flow and incremental egomotion estimation are designed to estimate large camera motion of *an image pair*, instead of tracking an entire OC video stream. Fig. 5.11a indicates that the maximum estimated velocity error is about $5mm/frame$ on both original and calibrated phantom image sequences(five each) in the straight phantom, after 19 phantom images have been tracked. Average velocity error is less than $3mm/frame$ on the original image sequences and less than $4mm/frame$ on the calibrated image sequences. Average displacement error is less than $7mm$ on the original image sequences and less than $8mm$ in the calibrated image sequences. Maximum displacement error is less than $13mm$ on both original and calibrated image sequences. Table 5.1a presents the average, maximum, and minimum estimated camera velocity errors of each of five trials. Table 5.1b shows their average, maximum and minimum errors of the estimated camera displacements.

In comparison with straight phantom image sequences, large camera motion parameters are more challenging to estimate in the new curved phantom image sequences, because the colonoscope moves $23.88mm$ between two successive phantom images. Fig. 5.12a shows that there is a significant estimation error at frame 1 in the original phantom image sequences and at frame 2 in the calibrated sequences(indicated by black arrows). As a result, average velocity error is about $6mm/frame$ on both original and calibrated image sequences, and maximum velocity error approximates $15mm/frame$ on both sequences. Average displacement error is about $14mm$ on the original image sequences and $16mm$ on the calibrated image sequences. Maximum displacement error achieves to $26mm$ on both original and calibrated image sequences. Tables. 5.2a and 5.2b reveal that significant errors of the camera velocities and displacements are caused by the second phantom image sequence.



(a)



(b)

Figure 5.11: *Comparison between the ground-truth and estimated camera motion parameters on the original and calibrated straight phantom image sequences.* (a) Camera velocity curves; (b) camera displacement curves. After eliminating a number of phantom images to simulate the exclusion of blurry images, the colonoscope moves $16mm$ between two consecutive, selected phantom images. Here, the camera motion parameter is either camera velocity in (a) or camera displacement in (b). The blue line represents the ground-truth camera motion parameters, and the red and green bands indicate the estimated camera motion parameters on the original and calibrated phantom image sequences, respectively. The bottom and upper curves in each band indicate the minimum and maximum camera motion parameters of five trials, and the solid center curve represents the average camera motion parameters. Average velocity error is less than $3mm/frame$ on the original image sequences after 19 phantom images have been tracked, and it is less than $4mm/frame$ in the calibrated image sequences. Maximum velocity error is less than $5mm/frame$ on both original and calibrated image sequences. Average displacement error is less than $7mm$ on the original image sequences and less than $8mm$ on the calibrated image sequences. Maximum displacement error is less than $13mm$ on both original and calibrated image sequences.

Table 5.1: The average, maximum, and minimum estimated camera **velocity** and **displacement** errors on the original and calibrated straight phantom image sequences at a speed of $16mm/frame$ after 19 images have been tracked.

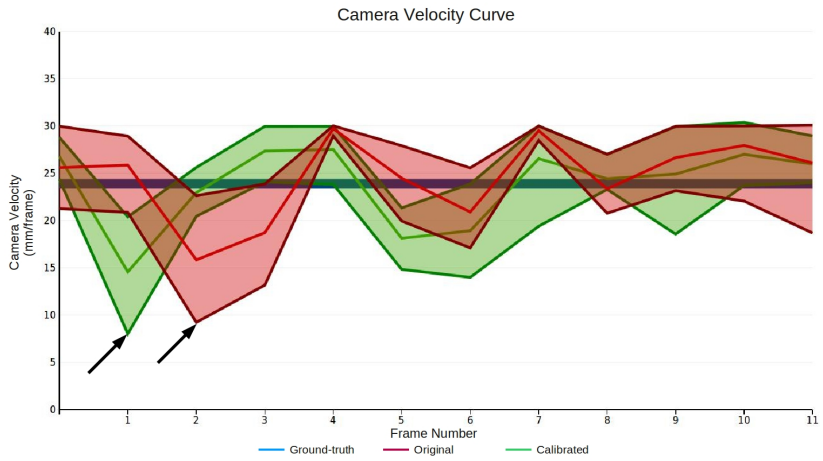
(a) Camera velocity

Image sequence	Original Images($mm/frame$)			Calibrated Images($mm/frame$)		
	average	maximum	minimum	average	maximum	minimum
1	2.06	4.07	0.0008	3.55	4.29	0.26
2	2.2	4.33	0.024	2.08	5.0	0.05
3	2.3	4.33	0.0007	2.81	4.67	0.19
4	2.25	4.82	0.014	2.08	4.58	0.15
5	1.53	4.43	0.00076	2.12	4.16	0.0073

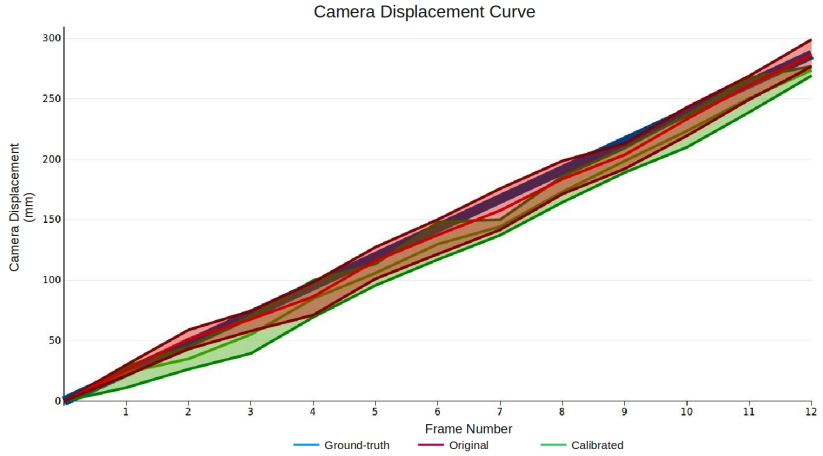
(b) Camera displacement

Image sequence	Original Images(mm)			Calibrated Images(mm)		
	average	maximum	minimum	average	maximum	minimum
1	5.45	10.99	0.0	4.85	11.08	0.0
2	6.49	12.62	0.0	7.86	15.6	0.0
3	5.92	10.28	0.0	5.49	10.83	0.0
4	5.81	10.745	0.0	5.17	10.9	0.0
5	2.82	7.54	0.0	4.1	7.63	0.0

Frame 1 of the second sequence is chosen to investigate the estimation errors, as illustrated in Fig. 5.13. There is a vertical curve highlighted by a red ellipse in the left phantom image, which is located in the inner wall of the curved phantom. Several SIFT feature points are detected at the vertical curve. They disappear in the right image because the vertical curve no longer exists. The vertical curve is occluded in the right image because the colonoscope is moving quickly toward the image center. There are some false SIFT feature correspondences that chooses the feature points detected at the vertical curve although their matched points are occluded. For instance, a false SIFT feature correspondence is connected by a red line, where the SIFT feature point in the left image is detected at the inner wall while its matched feature point is located at the outer wall in the right image. The visual motion vector calculated from this false SIFT feature match would point to the image center, corresponding to the visual motion when the colonoscope moves backward. However, the colonoscope is



(a)



(b)

Figure 5.12: *Comparison between the ground-truth and estimated camera motion parameters on the original and calibrated curved phantom images.* (a) Camera velocity curves; (b) camera displacement curves. The colonoscope moves 23.88mm between two consecutive, selected phantom images. The blue line represents the ground-truth camera motion parameters of five trials. The red and green bands indicate the estimated camera motion parameters on the original and calibrated phantom image sequences, respectively. The bottom and upper curves in each band indicate the minimum and maximum camera motion parameters of five trials, and the solid center curve represents the average camera motion parameters. Average velocity error is less than $4mm/frame$ on both original and calibrated image sequences after removing two drop points indicated by black arrows. The two significant estimation errors are caused by false SIFT feature matches due to the occlusion of feature points. Maximum velocity error is less than $8mm/frame$ on both image sequences. Average displacement error is less than $6mm$ on the original image sequences and less than $7mm$ on the calibrated sequences. Maximum displacement error is less than $13mm$ on both image sequences.

Table 5.2: The average, maximum, and minimum estimated camera **velocities** and **displacement** errors on the original and calibrated curved phantom image sequences at a speed of $23.88mm/frame$ after 13 images have been tracked.

(a) Camera velocity

Image sequence	Original Images($mm/frame$)			Calibrated Images($mm/frame$)		
	average	maximum	minimum	average	maximum	minimum
1	2.79	6.54	0.024	2.05	6.5	0.48
2	5.62	14.65	0.73	5.12	14.26	0.045
3	2.71	7.2	0.17	2.09	5.9	0.013
4	2.8	6.55	0.01	2.35	6.13	0.27
5	3.03	7.28	0.1	2.5	6.76	0.46

(b) Camera displacement

Image sequence	Original Images(mm)			Calibrated Images(mm)		
	average	maximum	minimum	average	maximum	minimum
1	4.54	12.49	0.0	3.96	10.34	0.0
2	13.37	24.4	0.0	15.46	25.9	0.0
3	5.98	12.5	0.0	5.56	13.44	0.0
4	4.0	9.57	0.0	5.15	10.92	0.0
5	7.7	12.73	0.0	6.61	12.24	0.0

moving forward. All these false feature matches reduce the estimated camera motion parameters during egomotion estimation. For this reason, there is a significant drop of the estimated camera velocity in the original phantom image sequences, indicated by the black arrow in Fig. 5.12a.

With the exception of the second trial, camera velocities are reasonably estimated in all other curved phantom image sequences. Average velocity error is less than $3mm/frame$ in both original and curved phantom image sequences. Maximum velocity error is less than $8mm/frame$ and $7mm/frame$ on the original and calibrated phantom image sequences, respectively. Average camera displacement error is less than $8mm$ on the original phantom image sequences and less than $7mm$ on the calibrated sequences. Maximum camera displacement error is less than $14mm$ on both original and calibrated curved phantom image sequences. Similar to the straight phantom results, there is no significant variance between the estimation results on

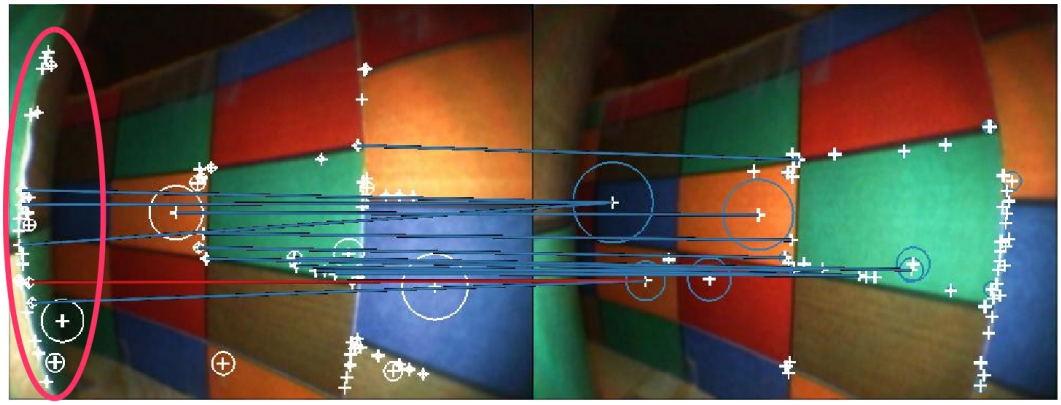


Figure 5.13: *SIFT feature matches between two successive phantom images (left and right images).* Some SIFT feature points enclosed by a red ellipse are detected at a vertical curve in the left image, while all these SIFT feature points disappear in the right image because the vertical curve is occluded. Therefore, some SIFT feature correspondence from these SIFT features are false feature matches, which cause the estimation errors of camera motion parameters. A SIFT feature correspondence connected by a red line gives an example.

the original and calibrated curved phantom images.

From the phantom experimental results, I can draw the following three conclusions.

1. Both straight and curved phantom results demonstrate that region flow based strategy is able to accurately recover significant camera motion. Average velocity error is $3mm$ of $16mm$ traveled between two consecutive images in the straight phantom and also $3mm$ of $23.88mm$ traveled in the curved phantom. If the colonoscope is moving at the the speed of $10mm/second$, the straight phantom experiments simulate the exclusion of $\frac{16mm \times 30 frames/second}{10mm/second} = 48$ frames and the curved phantom experiments simulate the exclusion of $\frac{23.88mm \times 30 frames/second}{10mm/second} \approx 72$ frames. Therefore, the proposed strategy can accurately estimate large camera motion within $\frac{3mm}{23.88mm} = 12.6\%$ relative error after excluding 72 blurry images at a speed of $10mm/sec$. In the real clinical situation, this strategy is able to estimate large motion within more blurry images because a colonoscope usually moves slightly or remains stationary during the appearance of blurry

images. The proposed strategy is sufficient to recover most blurry interruption in a colonoscopy video stream, because a gastroenterologist can quickly adjust the colonoscope.

2. The accuracy of large motion estimation is dependent on the amount of the colonoscope's movement. Large camera motion will cause a large portion of SIFT features to be occluded. Feature occlusion increases the possibility of false feature matches. The occurrence of false feature matches leads to less accuracy in estimating large motion.

3. There is no significant variance in results between the original and calibrated colonoscopy image sequences.

5.6 Clinical Data Evaluation

Region flow based large motion estimation is also demonstrated through several colonoscopy image sequences. Fig. 5.14 illustrates four examples of colonoscopy sequences with blurry images. The top rows illustrate optical images before and after the blurry sequences. The corresponding VC images are in the bottom rows. Regions marked by green circles indicate corresponding features, to verify accuracy.

Experiment 1: Polyp Surgery in the Sigmoid Colon. This sequence contains 520 images, with a blurry image sequence from frame 304 to 361, due to the colonoscope touching the colon wall. In Fig. 5.14a the polyp can be clearly seen in the OC and VC images, including scale changes in the polyp. The fold in the virtual image is likely due to deformation.

Experiment 2: Polyp Removal in the Sigmoid Colon. This sequence represents the removal of the polyp, and contains 160 images, with a blurry image sequence between 90 and 111. Injection of water (bright area) in the vicinity of the removed polyp causes the blurry image. Though somewhat harder to see, the green circles

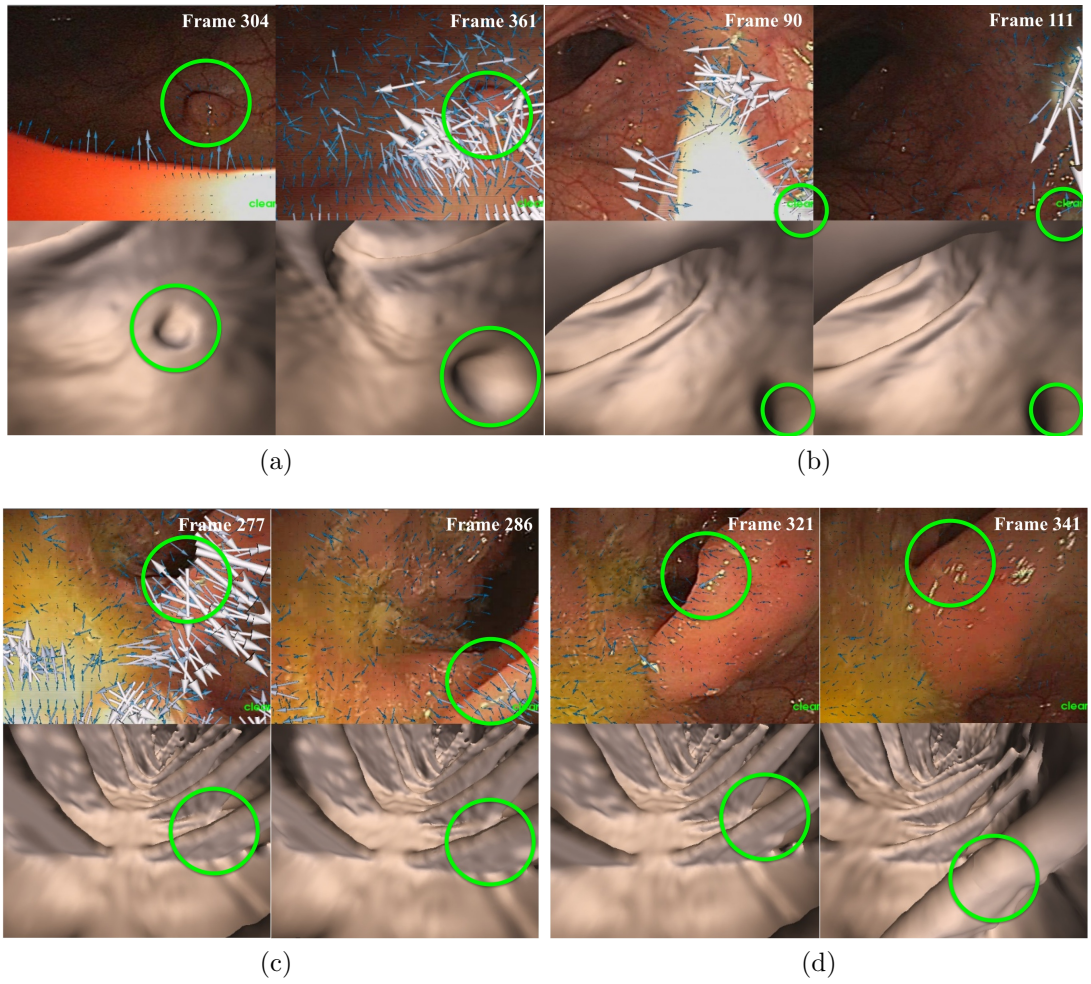


Figure 5.14: *Results in four colonoscopy sequences.* OC-VC image pairs before and after blurry sequences, (a) 520 image sequence of polyp surgery in the sigmoid colon with a sequence of 57 blurry images, (b) 160 image sequence of polyp removal in the sigmoid colon with a sequence of 21 blurry images, (c,d) 450 image sequence with 2 blurry image sequences of 9 and 19 images. The tracking system tracked successfully through both blurry image sequences.

estimate the locations of the polyp quite well in the OC and VC images.

Experiments 3,4: Ascending Colon. This sequence in the ascending colon contains 450 images and two blurry sequences, 277-286 and 321-340. In both cases, the colonoscope is very close to a fold. My algorithm is able to track continuously through the two blurry sequences, as seen by the well co-aligned OC and VC images in Figs. 5.14c and 5.14d.

The initial results are very promising. Although these sequences contain large

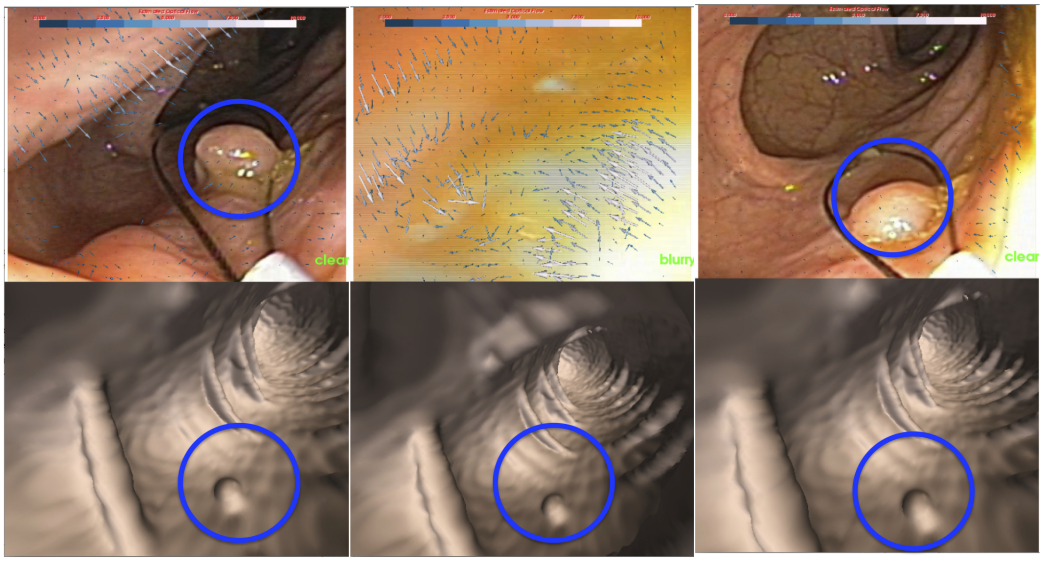
changes and artifacts (especially deformation), region flow accurately represents the global motion characteristics, facilitating the SIFT feature matching. In all these experiments, it is possible to identify features (folds, polyps, etc.) which provide confidence in the tracking system and qualitative accuracy.

Previous clinical results demonstrate that large motion estimation enables the colonoscopy tracking system to function in short colonoscopy image sequences. A 1980 descending colonoscopy image sequence is chosen to illustrate that this recovery strategy also works well in a long colonoscopy image sequence. In this sequence, the gastroenterologist detects a round polyp in the descending colon and used a snare to remove it. From frame 1414 to 1425, there is a short blurry sequence because the colonoscope's lens is covered by fluid, which is shown in Fig. 5.15b. Fig. 5.15a and Fig. 5.15c show a pair of colonoscopy images separated by blurry images to recover the colonoscope's actual motion as the colonoscope moves toward the polyp. Substantial variance in the shape of the fold indicates significant deformation between the two images. Fig. 5.15a and Fig. 5.15c demonstrate that the polyp's relative movement in the VC images exactly follows the colonoscope's motion in the OC images.

5.7 Conclusions

In this chapter, I have presented a region flow based strategy for estimating large motion when blurry images appear. Blurry images occur when the colonoscope touches a wall or fold, or when the colonoscope is immersed in fluid. Region flow provides the computational basis for accurate and robust corresponding features computation, which in turn permits estimating camera motion parameters.

An incremental egomotion estimation algorithm is designed to recover large camera motion from two images interrupted by blurry images. The core idea of incremental egomotion estimation is the subdivision of large visual motion into a sequence of optical flow fields. FOE-based egomotion estimation, introduced in chapter 4, is



(a) (b) (c)

Figure 5.15: *Large motion estimation results on a descending colonoscopy sequence.* (a) A colonoscopy image before a blurry sequence; (b) a blurry image; (c) the colonoscopy image after the blurry image sequence. The rounded polyp is highlighted by blue circles.

used to estimate camera motion parameters from each individual optical flow field. Combining the estimated camera motion parameters yields the final camera motion parameters.

Phantom experiments demonstrate that the proposed strategy can estimate significant camera motion with less than 12.6% relative error, when 72 frames are excluded at the speed of $10mm/sec$. Four short clinical colonoscopy sequences demonstrate the effectiveness of the proposed recovery strategy. It keeps the tracking system continuously co-aligning OC and VC images as it encounters blurry image sequences. In my experiments, blurry image sequences range from 9 to 57 consecutive images. A long colonoscopy image sequence demonstrates that the proposed strategy significantly improves upon the consecutive colonoscopy tracking algorithm, which is described in chapter 4. Qualitative results based on visual inspection of the tracked VC images are promising and outperform the FOE-based egomotion estimation method.

CHAPTER 6: CONTRIBUTION THREE – TEMPORAL VOLUME FLOW

"It is not knowledge, but the act of learning, not possession but the act of getting there, which grants the greatest enjoyment."

– Carl Friedrich Gauss

In the previous chapter, I developed a sophisticated strategy for large motion estimation based on region flow. It estimates large camera motion during a sequence of blurry images, and uses the region flow method to continuously track colonoscopy images. This strategy assumes that the two selected colonoscopy images contain enough similarity, and that significant visual motion and camera motion can be accurately estimated from the matched visual patterns.

But the selected colonoscopy images before and after a blurry image sequence do not always fulfill the requirement of having sufficient similarity. In this chapter, I present a temporal volume flow(TVF) algorithm for choosing a colonoscopy image pair with the maximum amount of similarity. The proposed algorithm is evaluated by studying different parameters controlling the computation of TVF. The comparison of colonoscopy tracking results, with and without TVF, is also presented.

6.1 Problem Statement

Fig. 6.1 illustrates two colonoscopy image sequences with blurry images shown in column (c). The region flow strategy described in chapter 5 artificially chooses two colonoscopy images, which are illustrated in columns (b) and (d) just before and after the blurry images. The region flow based strategy fails to estimate camera motion

parameters because there are nearly no feature correspondences between the images in columns (b) and (d).

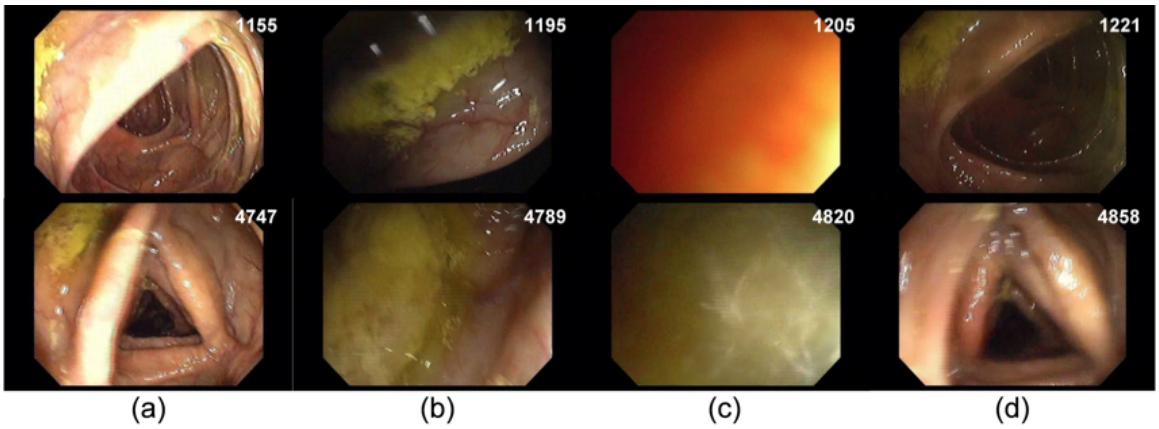


Figure 6.1: *Temporal coherence of two colonoscopy image sequences.* Column (a) and column (d) are two selected images before and after blurry images; column (b): the last clear colonoscopy images before the blurry sequences; column (c): blurry images. Temporal coherence guarantees that a pair of similar colonoscopy images can always be found.

In order to robustly estimate large motion during blurry image interruption, the selected image pair should contain sufficient similarity, such as colonoscopy images shown in columns (a) and (d) in Fig. 6.1. Temporal coherence in a colonoscopy video stream is the key property that ensures visual patterns co-occur before and after blurry images. The essential problem is thus to design an intelligent strategy to compute temporal coherence. An image pair with a large degree of similarity can be automatically identified, thus enhancing both stability and accuracy of large motion estimation.

In this chapter, "temporal volume flow", or dense voxel shifts between two temporal volumes, is proposed to efficiently exploit temporal coherence and determine the image pair, with sufficient similarity.

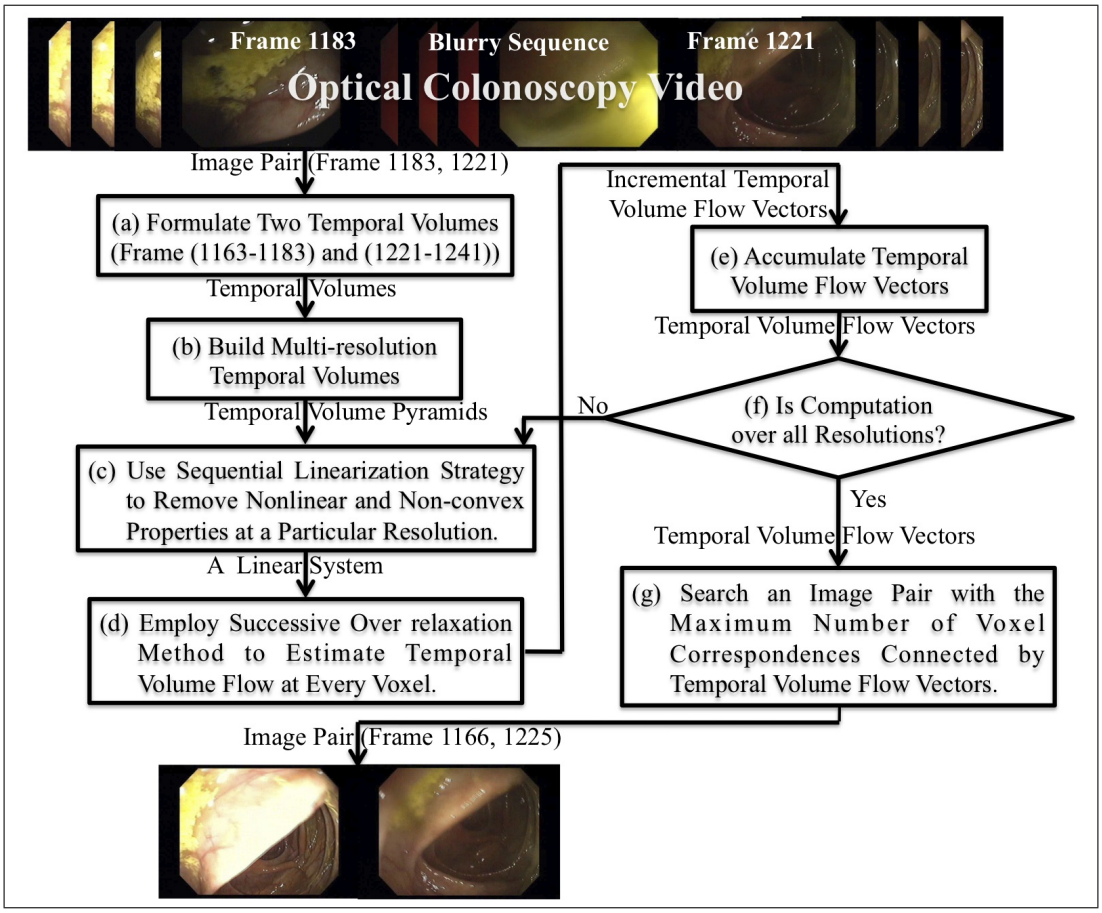


Figure 6.2: Flowchart of image pair search based on temporal volume flow.

6.2 Temporal Volume Flow Based Image Pair Search

Fig. 6.2 shows the flowchart of determining a colonoscopy image pair with the maximum amount of similarity. TVF computation is the core technique of image pair search. TVF computation includes temporal volume formulation; multi-resolution temporal volume pyramid construction; sequential linearization; and successive over relaxation. After TVF is calculated, it is then used to search an image pair with sufficient similarity. I will describe these components under two general headings: TVF computation and image pair search, followed by a section on model parameter tuning.

6.2.1 Temporal Volume Flow Computation

TVF computation is a partial differential equation(PDE) based framework to densely match two temporal volumes before and after blurry images; the number of voxel correspondences are determined in terms of TVF. The image pair with the maximum number of voxel correspondences are chosen as the image pair for large motion estimation. The main contribution of this chapter, TVF computation, is described and validated by a colonoscopy image pair in this section.

Algorithm Description

Assume there is a colonoscopy video stream $I(x, y, t)$ interrupted by a blurry image sequence at (t_1, t_2). Two temporal volumes can be built by collecting colonoscopy images at $(t_1 - \Delta t, t_1)$ and $(t_2, t_2 + \Delta t)$, as illustrated in Fig. 6.3. Without loss of generality, define ρ to represent the artificial time of a temporal volume stream. All temporal volumes formulate a continuous *four-dimensional temporal volume stream*, $V(x, y, t, \rho)$. The purpose of TVF computation is to densely match $V(x, y, t, \rho)$ at time ρ_1 and ρ_2.

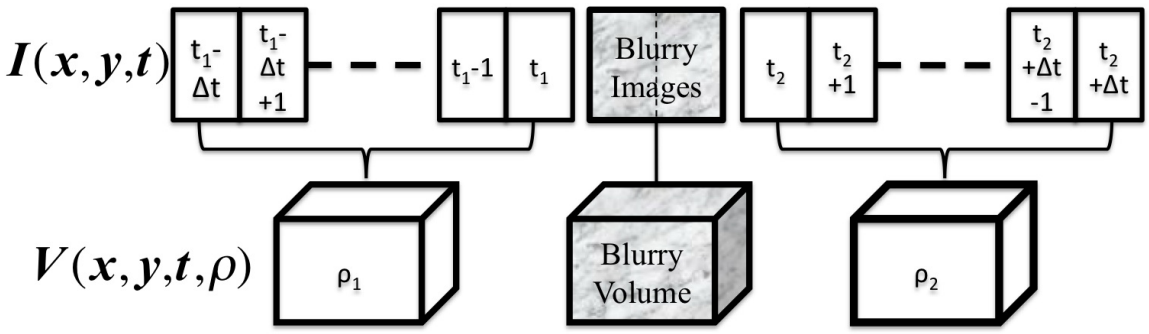


Figure 6.3: *Process of temporal volume construction.* Top row illustrates a video stream $I(x, y, t)$ with a blurry image sequence at (t_1, t_2). Bottom row shows the construction of temporal volume stream by grouping images, for example $(t_1 - \Delta t, t_1)$ and $(t_2, t_2 + \Delta t)$.

Similar to optical flow computation, TVF calculation assumes that the intensity at

a voxel in a temporal volume at ρ_1 remains invariant after this voxel shifts to another position in the temporal volume at ρ_2. The gradient of this voxel is also constant during the voxel shift. Therefore, TVF is a visual motion field that measures relative displacements between matched voxels in two temporal volumes at ρ_1 and ρ_2 with the same intensity and gradient.

Based on TVF definition, I can mathematically describe TVF computation in a PDE-based metric. Let me first express mathematical prototypes that comprise this PDE metric. Assume $\overrightarrow{w} = (w_x, w_y, w_t, w_\rho)$ to be the TVF vector at a point $\boldsymbol{p} = (x, y, t, \rho)$. Here, w_ρ is a constant and equal to 1. TVF begins with the intensity constancy assumption,

$$V(x, y, t, \rho) = V(x + w_x, y + w_y, t + w_t, \rho + 1) \tag{6.1}$$

The linearised formulation of Eq. 6.1 is

$$\partial_x V w_x + \partial_y V w_y + \partial_t V w_t + \partial_\rho V = 0 \tag{6.2}$$

The gradient constancy model is defined as

$$\nabla_3 V(x, y, t, \rho) = \nabla_3 V(x + w_x, y + w_y, t + w_t, \rho + 1) \tag{6.3}$$

where $\nabla_3 = (\partial_x, \partial_y, \partial_t)$.

TVF vectors vary smoothly except near visual motion boundaries, and a smoothness constraint is defined as

$$SM = |\nabla_3 w_x|^2 + |\nabla_3 w_y|^2 + |\nabla_3 w_t|^2 \tag{6.4}$$

Combining all these mathematical prototypes into a PDE-based metric, it leads to

$$
\begin{aligned}
E(\overrightarrow{w}) = \iint_{(x,y,t)\in\mathbb{R}^2\times\mathbb{R}_+} & (\Psi((V(x+w_x, y+w_y, t+w_t, \rho+1) - V(x,y,t,\rho))^2 \\
& + \gamma(\nabla_3 V(x+w_x, y+w_y, t+w_t, \rho+1) - \nabla_3 V(x,y,t,\rho))^2) \\
& + \alpha\Psi(|\nabla_3 w_x|^2 + |\nabla_3 w_y|^2 + |\nabla_3 w_t|^2))dxdydt
\end{aligned} \tag{6.5}
$$

where $\Psi(x^2) = \sqrt{x^2+\epsilon^2}, \epsilon = 0.001$ is a modified $L1$ norm and allows the computation to handle occlusions and other non-Gaussian deviations of the matching criterion. α and γ are two constants to balance different components in Eq. 6.5. Minimizing Eq. 6.5 with respect to $\overrightarrow{w}$ generates TVF.

Eq. 6.5 is similar to Eq. 5.13 except that the smoothness constraint contains SIFT feature matches in Eq. 5.13. Eq. 6.5 can be considered as the extension of a PDE-based model used by incremental egomotion estimation to the spatial-temporal domain. It determines relative displacements between a temporal volume pair. Eq. 6.5 is a non-convex and non-linear equation with respect to $\overrightarrow{w}$ due to the nonlinear intensity constancy model(Eq. 6.1) and the nonlinear gradient constancy model(Eq. 6.3). In addition, $\Psi(x^2)$ is also a nonlinear equation with regards to x^2. Eq. 6.5 is therefore difficult to minimize because its non-convexity and non-linearity produce multiple local minima. The minimization process easily becomes trapped in a local minimum and generates inaccurate TVF results.

In order to remove non-linearity and non-convexity from Eq. 6.5, the two advanced numerical computation strategies described in chapter 5 are employed: multi-scale image representation and sequential linearization.

Temporal volume formulation: TVF computation starts with temporal volume construction. Fig. 6.3 conceptualizes the construction process. For instance, an OC video stream shown in the top image of Fig. 6.2 has blurry images from frame 1184 to frame 1220, and two temporal volumes are constructed by collecting two sequences

of 20 colonoscopy images, 1163-1183 and 1221-1241.

Temporal volume pyramid construction: The underlying temporal volumes identified in the previous step must be smoothed to remove small details. The small details are responsible for local minima in Eq. 6.5, which result from non-convexity. One strategy to handle the non-convexity is the construction of the Gaussian scale space on the original sized temporal volume as was used in incremental egomotion estimation in chapter 5. However, this strategy is computationally consuming and uses memory inefficiently. Smoothing a temporal volume can be alternatively achieved by continuously down-sampling temporal volumes to build multi-resolution temporal volume pyramids. Down-sampling temporal volume is equivalent to using a Gaussian function to smooth the original temporal volumes. But it significantly reduces the computational cost. The sampling rate is akin to the Gaussian scale parameter. The smaller the sampling rate, the coarser the temporal volumes will be. In my implementation, the sampling rate is chosen as 0.75. Finally, two temporal volume pyramids are constructed, based on this continuous down-sampling strategy.

Sequential linearization: Minimizing Eq. 6.5 can be mathematically solved through the Euler-Lagrange equation. It reads, with respect to x, y and t components,

$$
\begin{aligned}
&\Psi'((\partial_\rho V)^2 + \gamma((\partial_{x\rho} V)^2 + (\partial_{y\rho} V)^2 + (\partial_{t\rho} V)^2))(\partial_x V \partial_\rho V + \gamma(\partial_{xx} V \partial_{x\rho} V + \partial_{xy} V \partial_{y\rho} V \\
&+ \partial_{xt} V \partial_{t\rho} V)) - \alpha \mathrm{div}_3 \left(\Psi'(|\nabla_3 w_x|^2 + |\nabla_3 w_y|^2 + |\nabla_3 w_t|^2)\nabla_3 w_x\right) = 0 \\
&\Psi'((\partial_\rho V)^2 + \gamma((\partial_{x\rho} V)^2 + (\partial_{y\rho} V)^2 + (\partial_{t\rho} V)^2))(\partial_y V \partial_\rho V + \gamma(\partial_{xy} V \partial_{x\rho} V + \partial_{yy} V \partial_{y\rho} V \\
&+ \partial_{yt} V \partial_{t\rho} V)) - \alpha \mathrm{div}_3 \left(\Psi'(|\nabla_3 w_x|^2 + |\nabla_3 w_y|^2 + |\nabla_3 w_t|^2)\nabla_3 w_y\right) = 0 \\
&\Psi'((\partial_\rho V)^2 + \gamma((\partial_{x\rho} V)^2 + (\partial_{y\rho} V)^2 + (\partial_{t\rho} V)^2))(\partial_t V \partial_\rho V + \gamma(\partial_{xt} V \partial_{x\rho} V + \partial_{yt} V \partial_{y\rho} V \\
&+ \partial_{tt} V \partial_{t\rho} V)) - \alpha \mathrm{div}_3 \left(\Psi'(|\nabla_3 w_x|^2 + |\nabla_3 w_y|^2 + |\nabla_3 w_t|^2)\nabla_3 w_t\right) = 0 \qquad (6.6)
\end{aligned}
$$

where the derivatives related to $\partial_{*\rho}V$ are defined as temporal difference.

$$\begin{aligned}
\partial_\rho V &= V(x+w_x, y+w_y, t+w_t, \rho+1) - V(x,y,t,\rho) \\
\partial_{x\rho} V &= \partial_x V(x+w_x, y+w_y, t+w_t, \rho+1) - \partial_x V(x,y,t,\rho) \\
\partial_{y\rho} V &= \partial_y V(x+w_x, y+w_y, t+w_t, \rho+1) - \partial_y V(x,y,t,\rho) \\
\partial_{t\rho} V &= \partial_t V(x+w_x, y+w_y, t+w_t, \rho+1) - \partial_t V(x,y,t,\rho)
\end{aligned} \tag{6.7}$$

Let me abbreviate data and smoothness terms in Eq. 6.5 to simplify the description,

$$\begin{aligned}
\Psi_D &= \Psi((V(x+w_x, y+w_y, t+w_t, \rho+1) - V(x,y,t,\rho))^2 \\
&\quad + \gamma(\nabla_3 V(x+w_x, y+w_y, t+w_t, \rho+1) - \nabla_3 V(x,y,t,\rho))^2) \\
\Psi_S &= \Psi(|\nabla_3 w_x|^2 + |\nabla_3 w_y|^2 + |\nabla_3 w_t|^2)
\end{aligned} \tag{6.8}$$

Non-linearity happens in Ψ'_D and Ψ'_S. Sequential linearization is an efficient strategy to remove non-linearity in Eq. 6.6, so as to easily minimize Eq. 6.5. Sequential linearization involves two nested fixed point iterations[27]. Let l denote the outer iteration index at temporal volume pyramid level k, and define $\overrightarrow{w}^{k,l} = (w_x^{k,l}, w_y^{k,l}, w_t^{k,l}, 1)$. This iteration subdivides the TVF vector $\overrightarrow{w}^{k,l+1} = \overrightarrow{w}^{k,l} + d\overrightarrow{w}^{k,l}$, where $d\overrightarrow{w}^{k,l} = (dw_x^{k,l}, dw_y^{k,l}, dw_t^{k,l}, 0)$. It also removes non-linearity from intensity and gradient constancy constraints in Eq. 6.5. The estimation of $\overrightarrow{w}^{k,l+1}$ is converted into the computation of the incremental TVF vector $d\overrightarrow{w}^{k,l}$.

However, Eq. 6.6 still remains nonlinear with respect to $d\overrightarrow{w}^{k,l}$, which is caused by $\Psi_D'^{k,l}$ and $\Psi_S'^{k,l}$. Another inner iteration is introduced to remove their non-linearity. Assume m to be the iteration index, and let $\Psi_D'^{k,l,m}$ and $\Psi_S'^{k,l,m}$ denote the updated abbreviation parameters and $d\overrightarrow{w}^{k,l,m} = (dw_x^{k,l,m}, dw_y^{k,l,m}, dw_t^{k,l,m}, 0)$ be the updated incremental temporal volume flow vector. The inner iteration removes $dw_x^{k,l,m+1}$, $dw_y^{k,l,m+1}$ and $dw_t^{k,l,m+1}$ from $\Psi_D'^{k,l,m}$ and $\Psi_S'^{k,l,m}$. Eq. 6.6 is finally converted into a

linear equation with respect to $dw_x^{k,l,m+1}$, $dw_y^{k,l,m+1}$ and $dw_t^{k,l,m+1}$. The details of sequential linearization are described in Appendix H. Finally, each voxel has three linear equations, which leads to a massive sparse linear system to compute TVF.

Numerical Calculation: This massive sparse linear system system is again efficiently solved by applying the successive over-relaxation method[236] in the most inner iteration. Let n be the index for this iteration. We can obtain the following equation to explicitly compute the incremental TVF vector $d\overrightarrow{w}^{k,l,m,n+1}$

$$
\begin{aligned}
(dw_x)_{\boldsymbol{p}}^{k,l,m,n+1} &= (1-\omega)(dw_x)_{\boldsymbol{p}}^{k,l,m,n} \\
&+ \omega \left(\sum_{\boldsymbol{q}\in\mathcal{N}^+(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} \left((w_x)_{\boldsymbol{q}}^{k,l} + (dw_x)_{\boldsymbol{q}}^{k,l,m,n} \right) \right. \\
&+ \sum_{\boldsymbol{q}\in\mathcal{N}^-(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} \left((w_x)_{\boldsymbol{q}}^{k,l} + (dw_x)_{\boldsymbol{q}}^{k,l,m,n+1} \right) - \sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} (w_x)_{\boldsymbol{p}}^{k,l} \\
&- \frac{1}{\alpha} (\Psi'_D)_{\boldsymbol{p}}^{k,l,m} \left((\partial_x V^{k,l})_{\boldsymbol{p}} \left((\partial_y V^{k,l})_{\boldsymbol{p}} (dw_y)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_t V^{k,l})_{\boldsymbol{p}} (dw_t)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_\rho V^{k,l})_{\boldsymbol{p}} \right) \right. \\
&+ \gamma (\partial_{xx} V^{k,l})_{\boldsymbol{p}} \left((\partial_{xy} V^{k,l})_{\boldsymbol{p}} (dw_y)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_{xt} V^{k,l})_{\boldsymbol{p}} (dw_t)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_{x\rho} V^{k,l})_{\boldsymbol{p}} \right) \\
&+ \gamma (\partial_{xy} V^{k,l})_{\boldsymbol{p}} \left((\partial_{yy} V^{k,l})_{\boldsymbol{p}} (dw_y)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_{yt} V^{k,l})_{\boldsymbol{p}} (dw_t)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_{y\rho} V^{k,l})_{\boldsymbol{p}} \right) \\
&\left. \left. + \gamma (\partial_{xt} V^{k,l})_{\boldsymbol{p}} \left((\partial_{yt} V^{k,l})_{\boldsymbol{p}} (dw_y)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_{tt} V^{k,l})_{\boldsymbol{p}} (dw_t)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_{t\rho} V^{k,l})_{\boldsymbol{p}} \right) \right) \right) \\
&/ \left(\sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} + \frac{1}{\alpha} (\Psi'_D)_{\boldsymbol{p}}^{k,l,m} \left((\partial_x V^{k,l})_{\boldsymbol{p}}^2 + \gamma \left((\partial_{xx} V^{k,l})_{\boldsymbol{p}}^2 + (\partial_{xy} V^{k,l})_{\boldsymbol{p}}^2 + (\partial_{xt} V^{k,l})_{\boldsymbol{p}}^2 \right) \right) \right) \\
(dw_y)_{\boldsymbol{p}}^{k,l,m,n+1} &= (1-\omega)(dw_y)_{\boldsymbol{p}}^{k,l,m,n} \\
&+ \omega \left(\sum_{\boldsymbol{q}\in\mathcal{N}^+(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} \left((w_y)_{\boldsymbol{q}}^{k,l} + (dw_y)_{\boldsymbol{q}}^{k,l,m,n} \right) \right. \\
&+ \sum_{\boldsymbol{q}\in\mathcal{N}^-(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} \left((w_y)_{\boldsymbol{q}}^{k,l} + (dw_y)_{\boldsymbol{q}}^{k,l,m,n+1} \right) - \sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} (w_y)_{\boldsymbol{p}}^{k,l} \\
&- \frac{1}{\alpha} (\Psi'_D)_{\boldsymbol{p}}^{k,l,m} \left((\partial_y V^{k,l})_{\boldsymbol{p}} \left((\partial_x V^{k,l})_{\boldsymbol{p}} (dw_x)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_t V^{k,l})_{\boldsymbol{p}} (dw_t)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_\rho V^{k,l})_{\boldsymbol{p}} \right) \right. \\
&+ \gamma (\partial_{xy} V^{k,l})_{\boldsymbol{p}} \left((\partial_{xx} V^{k,l})_{\boldsymbol{p}} (dw_x)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_{xt} V^{k,l})_{\boldsymbol{p}} (dw_t)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_{x\rho} V^{k,l})_{\boldsymbol{p}} \right) \\
&+ \gamma (\partial_{yy} V^{k,l})_{\boldsymbol{p}} \left((\partial_{xy} V^{k,l})_{\boldsymbol{p}} (dw_x)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_{yt} V^{k,l})_{\boldsymbol{p}} (dw_t)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_{y\rho} V^{k,l})_{\boldsymbol{p}} \right) \\
&\left. \left. + \gamma (\partial_{yt} V^{k,l})_{\boldsymbol{p}} \left((\partial_{xt} V^{k,l})_{\boldsymbol{p}} (dw_x)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_{tt} V^{k,l})_{\boldsymbol{p}} (dw_t)_{\boldsymbol{p}}^{k,l,m,n} + (\partial_{t\rho} V^{k,l})_{\boldsymbol{p}} \right) \right) \right) \\
&/ \left(\sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} + \frac{1}{\alpha} (\Psi'_D)_{\boldsymbol{p}}^{k,l,m} \left((\partial_y V^{k,l})_{\boldsymbol{p}}^2 + \gamma \left((\partial_{xy} V^{k,l})_{\boldsymbol{p}}^2 + (\partial_{yy} V^{k,l})_{\boldsymbol{p}}^2 + (\partial_{yt} V^{k,l})_{\boldsymbol{p}}^2 \right) \right) \right) \\
(dw_t)_{\boldsymbol{p}}^{k,l,m,n+1} &= (1-\omega)(dw_t)_{\boldsymbol{p}}^{k,l,m,n} \\
&+ \left(\sum_{\boldsymbol{q}\in\mathcal{N}^+(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} \left((w_t)_{\boldsymbol{q}}^{k,l} + (dw_t)_{\boldsymbol{q}}^{k,l,m,n} \right) \right.
\end{aligned}
$$

$$
\begin{aligned}
&+\sum_{\boldsymbol{q}\in\mathcal{N}^-(\boldsymbol{p})}(\Psi'_S)^{k,l,m}_{\boldsymbol{p}\sim\boldsymbol{q}}\left((w_t)^{k,l}_{\boldsymbol{q}}+(dw_t)^{k,l,m,n+1}_{\boldsymbol{q}}\right)-\sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})}(\Psi'_S)^{k,l,m}_{\boldsymbol{p}\sim\boldsymbol{q}}(w_t)^{k,l}_{\boldsymbol{p}}\\
&-\frac{1}{\alpha}(\Psi'_D)^{k,l,m}_{\boldsymbol{p}}\left((\partial_t V^{k,l})_{\boldsymbol{p}}\left((\partial_x V^{k,l})_{\boldsymbol{p}}(dw_x)^{k,l,m,n}_{\boldsymbol{p}}+(\partial_y V^{k,l})_{\boldsymbol{p}}(dw_y)^{k,l,m,n}_{\boldsymbol{p}}+(\partial_\rho V^{k,l})_{\boldsymbol{p}}\right)\right.\\
&+\gamma(\partial_{xt}V^{k,l})_{\boldsymbol{p}}\left((\partial_{xx}V^{k,l})_{\boldsymbol{p}}(dw_x)^{k,l,m,n}_{\boldsymbol{p}}+(\partial_{xy}V^{k,l})_{\boldsymbol{p}}(dw_y)^{k,l,m,n}_{\boldsymbol{p}}+(\partial_{x\rho}V^{k,l})_{\boldsymbol{p}}\right)\\
&+\gamma(\partial_{yt}V^{k,l})_{\boldsymbol{p}}\left((\partial_{xy}V^{k,l})_{\boldsymbol{p}}(dw_x)^{k,l,m,n}_{\boldsymbol{p}}+(\partial_{yy}V^{k,l})_{\boldsymbol{p}}(dw_y)^{k,l,m,n}_{\boldsymbol{p}}+(\partial_{y\rho}V^{k,l})_{\boldsymbol{p}}\right)\\
&\left.+\gamma(\partial_{tt}V^{k,l})_{\boldsymbol{p}}\left((\partial_{xt}V^{k,l})_{\boldsymbol{p}}(dw_x)^{k,l,m,n}_{\boldsymbol{p}}+(\partial_{yt}V^{k,l})_{\boldsymbol{p}}(dw_y)^{k,l,m,n}_{\boldsymbol{p}}+(\partial_{t\rho}V^{k,l})_{\boldsymbol{p}}\right)\right)\\
&/\left(\sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})}(\Psi'_S)^{k,l,m}_{\boldsymbol{p}\sim\boldsymbol{q}}+\frac{1}{\alpha}(\Psi'_D)^{k,l,m}_{\boldsymbol{p}}\left((\partial_t V^{k,l})^2_{\boldsymbol{p}}+\gamma\left((\partial_{xt}V^{k,l})^2_{\boldsymbol{p}}+(\partial_{yt}V^{k,l})^2_{\boldsymbol{p}}+(\partial_{tt}V^{k,l})^2_{\boldsymbol{p}}\right)\right)\right)
\end{aligned}
\tag{6.9}
$$

Here, $\mathcal{N}^+(\boldsymbol{p})$ denotes the neighbors $\boldsymbol{q}$ of $\boldsymbol{p}$ if the indices of $\boldsymbol{q}$ are larger than those of $\boldsymbol{p}$. $\mathcal{N}^-(\boldsymbol{p})$ denotes the neighbors $\boldsymbol{q}$ of $\boldsymbol{p}$ if the indices of $\boldsymbol{q}$ are smaller than those of $\boldsymbol{p}$. $\omega \in (0,2)$ is the relation parameter to guide the convergence of the massive linear system. As suggested by Yong[236], values close to 2 give the best performance. In my implementation, it is chosen as 1.99.

Assume the current pyramid level be k, and the outer and inner iteration indices be l and $m+1$, respectively. The TVF vectors are sequentially accumulated by adding incremental TVF vectors

$$
\overrightarrow{w}^{k,l,m+1} = \overrightarrow{w}^{k,l,m} + d\overrightarrow{w}^{k,l,m} \tag{6.10}
$$

TVF computation starts from the coarsest level and gradually assigns TVF vectors through the finer level. The up-sampling of the TVF vectors is performed by using bilinear interpolation.

The purpose of this step is to determine when the accumulation of TVF vectors may stop. Let the finest temporal volume pyramid level be K. The maximum numbers of the outer and inner sequential linearization iteration are L and M, respectively. If the temporal volume pyramid index $k \neq \mathsf{K}$, or the inner iteration index $m \neq \mathsf{M}$, or the outer iteration index $l \neq \mathsf{L}$, then repeat the sequential linearization and numerical calculation. Otherwise, output the final TVF.

TVF computation is summarized in algorithm 3.

Algorithm 3: Temporal Volume Flow Computation

Data: $V(x, y, t, \rho)$ at ρ_1 and ρ_2
Result: Temporal volume flow $\overrightarrow{w}(x, y, t, \rho)$.

```
Build K level temporal volume pyramids for V(x,y,t,ρ1) and V(x,y,t,ρ2);
for k ← 1 to K do
    Initialize k-th level TVF vectors;
    for l ← 1 to L do
        Compute derivatives ∂ρV^{k,l}, ∂xρV^{k,l}, ∂yρV^{k,l} and ∂tρV^{k,l} through bilinear
        interpolation;
        for m ← 1 to M do
            Compute diffusion terms Ψ'^{k,l,m}_D and Ψ'^{k,l,m}_S ;
            Initialize incremental TVF vector dw^{k,l,m} = 0;
            for n ← 1 to N do
                Compute incremental TVF vector dw^{k,l,m,n} based on Eq. 6.9 at
                every voxel through successive over-relaxation method;
            Update TVF vector w^{k,l,m+1} = w^{k,l,m} + dw^{k,l,m};
```

Example Demonstration

Volume rendering techniques are used to visualize two temporal volumes and to composite the two images illustrated in Fig. 6.4. Arrows in the left image show the TVF results. The gastroenterologist attempted to rotate the colonoscope towards the colon wall in the left temporal volume. A blurry sequence was produced because the colonoscope touched the colon wall. This blurry image sequence disappeared when the gastroenterologist rotated the colonoscope back in the right temporal volume. Note the folds' movements between two temporal volumes. Flow vectors accurately capture relative movements between two folds(point in the up direction).

6.2.2 Image Pair Search

The following two subsections describe the image pair selection algorithm and present an example.

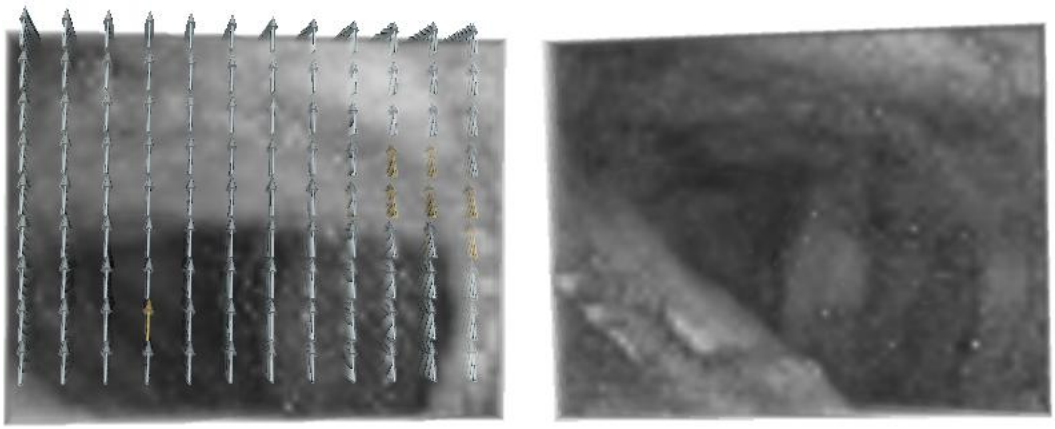


Figure 6.4: *TVF results of two temporal volumes before and after a blurry image sequence*. Here, volume rendering techniques are used to visualize temporal volumes and to composite left and right images. TVF pointing in the up direction accurately reflects relative displacements between colon folds in the left and right images.

Algorithm Description

After TVF is computed, I track all possible voxel displacements between two temporal volumes. Then I count the number of all possible voxel correspondences connected by TVF vectors, between any image pairs in two temporal volumes. Thus, if there are N images in both temporal volumes, there are $N \times N$ pairs of images that will be considered. I select the image pair that has the largest number of voxel correspondences. These frames are then input to the region flow algorithm described in chapter 5.

Example Demonstration

Fig. 6.5 shows an image pair in the descending colon. The image pair, which is separated by 10 blurry images, was selected by the TVF vectors illustrated in Fig. 6.4. These two images have 1) similar intensity distribution; 2) similar scale variance; and 3) the appearance of important features like folds in the bottom left image regions. They are very useful for computing large visual motion, which is described in Chapter. 5.

(a) (b)

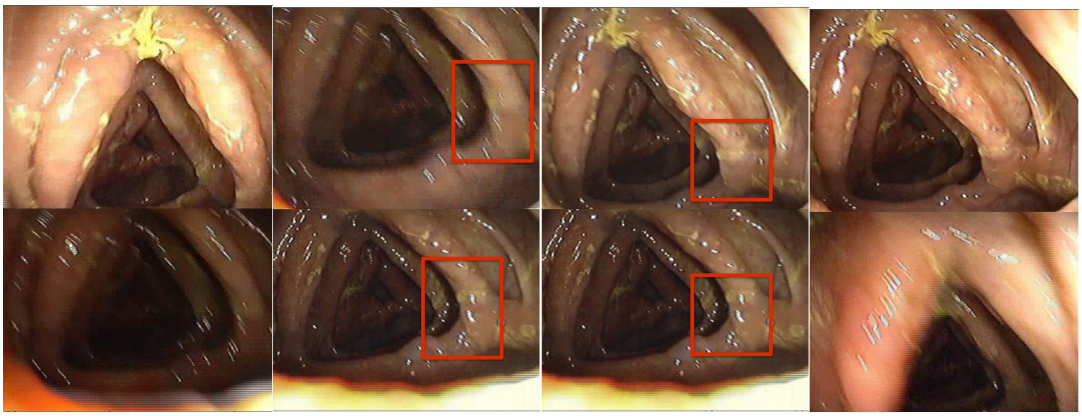
Figure 6.5: *Image pair selected by TVF.* These two images have similar intensity distribution, and main structures such as folds appear in both images. They are critical features for visual motion computation.

6.2.3 Model Parameter Tuning

TVF computation is a time-consuming process, due to solving the huge linear system defined in Eq. 6.9. However, clinical applications require colonoscopy tracking recovery to be as quick as possible. It is desirable that the size of this linear system be reduced while the accuracy of TVF results is maintained. There are two important parameters which serve these goals: 1) the number of colonoscopy images to form a temporal volume; and 2) the down-sampling rate to build a temporal volume pyramid. The trade-off between these two parameters is empirically designed to obtain sufficient accuracy while minimizing computational complexity.

Let me first investigate the influence of the size of temporal volume. I build temporal volumes with $10, 20, 40$ and 60 colonoscopy images. The resolution of a colonoscopy image is 500×390. Fig. 6.6 shows the selected image pairs from temporal volumes of different size. Images in the top row are selected before a blurry image sequence. Images in the bottom row are selected after the blurry image sequence. The two images are quite different in Fig. 6.6a when a temporal volume is comprised of 10 frames. This improper selection result is because there are insufficient frames

to search for an accurate image pair. The selected images are also quite different in Fig. 6.6d, where the size of the temporal volume is 60. At this point, TVF is inaccurate, because the TVF computation becomes trapped in local minima when the size of the temporal volume is too large. Fig. 6.6b and Fig. 6.6c show the selection results when the size of the temporal volume is equal to 20 and 40, respectively. Now the image pair selection results are more reasonable for large motion estimation, because visual features such as folds are maintained in both images. For example, an important geometric feature, a "T"-junction fold, indicated by red rectangles, is presented in the image pairs. Visual motion between these two images is relatively small. Two selected images in Fig. 6.6c are more similar than in Fig. 6.6b, but at the cost of more computational efforts. The computational timing is listed in table 6.1. Based on these experimental results, 20 is an optimal experimental parameter for TVF computation. This parameter selects two similar images, while conserving computational cost. Therefore, my TVF computation selects 20 colonoscopy images to formulate a temporal volume.



(a) 10 images (b) 20 images (c) 40 images (d) 60 images

Figure 6.6: *Image pairs selected by TVF on the same colonoscopy image sequence by varying frame numbers to formulate temporal volumes.* The top row shows the selected images before blurry images. The bottom row shows the images after the blurry image sequence.

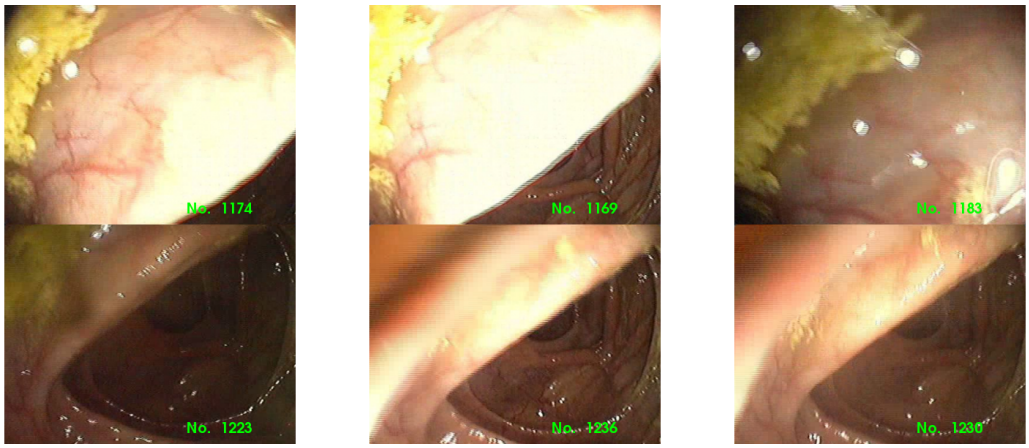
Another important parameter in TVF computation is the down-sampling rate to

Table 6.1: The time required for TVF computation, for various numbers of frames in the temporal volume.

Temporal Volume Size	10	20	40	60
Timing	64.32s	121.13s	260.65s	375.24s

construct a temporal volume pyramid. Down-sampling the temporal volume reduces computational cost. It also smoothes temporal volumes, similar to constructing a Gaussian scale space. Therefore, the sampling rate can be considered as a scale parameter. The smaller the sampling rate, the more coarsely the temporal volume is smoothed.

An ascending colonoscopy image sequence is chosen to show how image pair selection results are affected by sampling rates, as shown in Fig. 6.7. Fig. 6.7a shows the selected image pair when the sampling rate is 0.9, which is equivalent to using the fine scale to smooth temporal volumes. A triangular fold appears in the bottom image, while it only partly shows up in the top image. Small image structures are not smoothed, causing the TVF computation to become trapped in local minima. Fig. 6.7c gives another improper selection result, where the sampling rate is 0.5. This rate is akin to using the coarse scale to smooth the temporal volume. At this point, some important image structures are over-smoothed in temporal volumes, and the TVF computation is improperly initialized. Compared to the previous selection results, 0.75 selects two similar images, illustrated in Fig. 6.7b. This value is experimentally determined to be optimal and is used in constructing temporal volume pyramids. Table 6.2 reports the computational time required with different sampling rates. It takes about 2 minutes to compute TVF when the down-sampling rate is 0.75.



(a) 0.9 (b) 0.75 (c) 0.5

Figure 6.7: *Image pairs selected by TVF on the same colonoscopy image sequence by varying sampling rates to construct temporal volume pyramids.* The top row shows the selected images before blurry images. The bottom row shows the images after the blurry image sequence.

Table 6.2: The time required for TVF computation, for various sampling rates in constructing temporal volume pyramids.

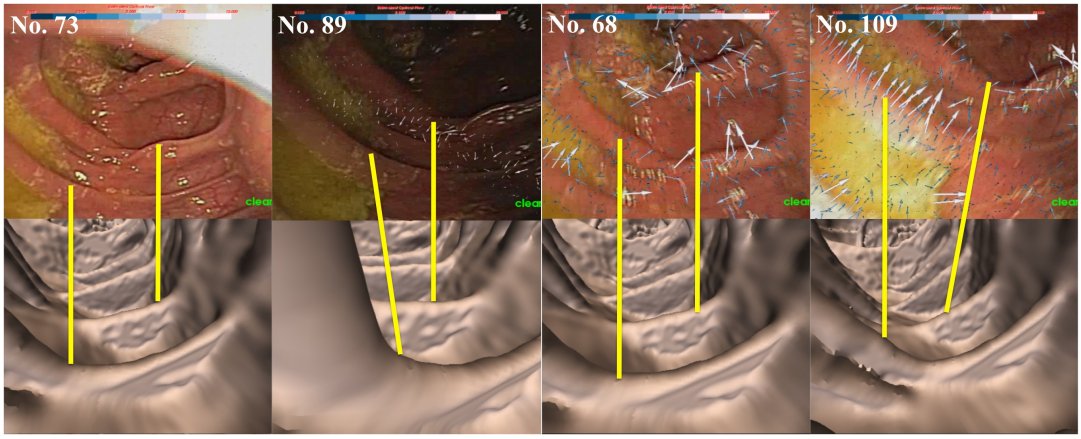
Down-sampling rate	0.5	0.75	0.9
Timing	67.72s	121.13s	220.28s

6.3 Clinical Data Evaluation

I have tested the TVF algorithm on three OC image sequences from three patients, and have shown tracking accuracy improvements over region flow method.

Sequence 1: Ascending Colon. This sequence contained 235 images in the ascending colon and had 12 blurry images between 77 and 88. Frames 73 and 89 are two colonoscopy images just before and after the blurry image sequence. They are chosen by the region flow method in chapter 5 for computing camera motion parameters. TVF selected a better image pair – Frame 68 and Frame 109. The tracking system can continuously co-align OC and VC images by using both image pairs. Corresponding folds appear in OC and VC images after the blurry image sequence, as seen in

the yellow lines in Fig. 6.8. However, the image pair not selected by TVF resulted in a large rotation error illustrated in Fig. 6.8a. In contrast, the image pair selected by using TVF exhibited no such error, as seen in Fig. 6.8b.

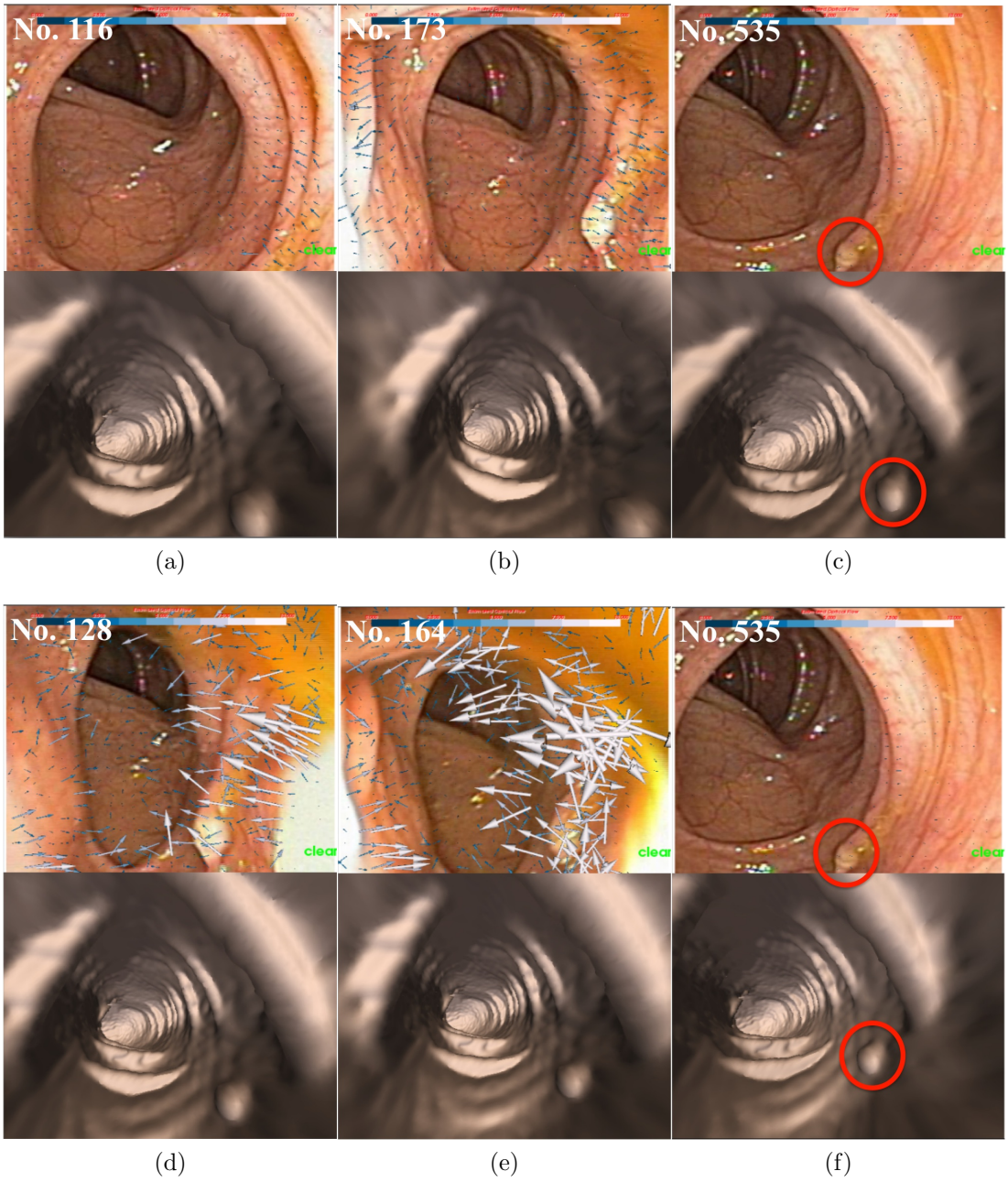


(a) (b)

Figure 6.8: *Comparison of tracking results using the image pair not selected by TVF vs. selected by TVF in the ascending colon.* (a) Original image pair (73 and 89), (b) TVF image pair (68 and 109). Corresponding folds are connected by yellow lines in OC and VC images. There is significant rotation error in the left image, in contrast to the right image.

Sequence 2: Descending Colon. Similar results are observed in a 535-image sequence with a rounded polyp in the descending colon, shown in Fig. 6.9. This sequence contains 24 blurry images, caused by fluid, from frame 130 to 153. The colon also undergoes contraction as well as expansion in this sequence. The tracking system can continuously co-align OC and VC images after blurry images. At frame 535, although the polyp is visible in both OC and VC images, TVF had less error. The polyp in the VC image is more similar to that in the OC image in Fig. 6.9c.

Sequence 3: Descending Colon. Fig. 6.10 illustrates a 580-image sequence in the descending colon after polyp removal. There is a long blurry image sequence from 82 to 353. The locations of the polyp are highlighted by red rectangles in Fig. 6.10a. Region flow fails to recover the motion parameters if two colonoscopy images are not selected by TVF, because there are insufficient feature correspondences from



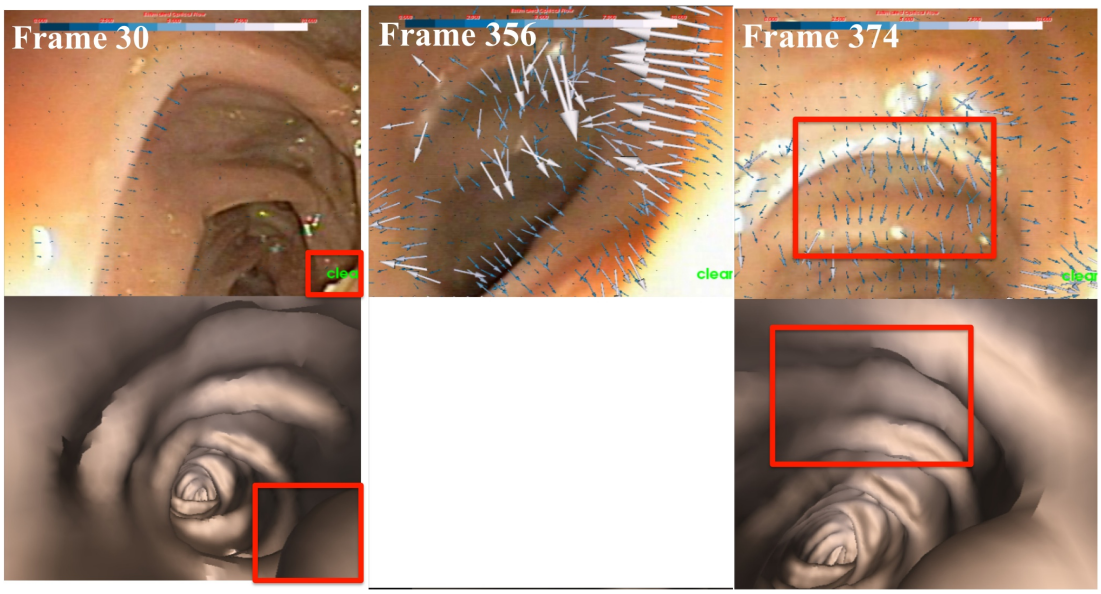
(a) (b) (c)

(d) (e) (f)

Figure 6.9: *Comparison of the tracking results with TVF(top row, a-c) and without TVF(bottom row, d-f) in the descending colon with a rounded polyp.* (a,b) Frames 116 and 173 selected by TVF, (c) tracking results at frame 535. Tracking accuracy is improved since polyp is close to the bottom of image, (d,e) tracking results of the selected image pair at frames 128 and 164, (f) tracking results at frame 535, polyp highlighted by red circles in OC and VC images.

the selected frames, 30 and 356. By contrast, TVF successfully continues to track, selecting frames 30 and 374, as seen in in Fig. 6.10c. Note the same folds inside the

rectangles appear in both OC and VC images.



(a) (b) (c)

Figure 6.10: *Comparison of the tracking results with and without TVF in the descending colon after polyp removal.* (a) OC and VC images at frame 30 before a blurry sequence; red rectangles indicate polyp locations, (b) region flow fails to track after the blurry sequence, selecting frame 356 to match frame 30, (c) TVF chooses frame 374 to successfully continue tracking, because the same folds (red rectangles) appear in both OC and VC images.

6.4 Summary

In this chapter, I have presented a TVF approach to continue tracking in colonoscopy video sequences which encounter blurry images. TVF computation employs nonlinear intensity and gradient constancy models, which are combined into an energy function. The energy function is minimized through multi-resolution temporal volume pyramid and sequential linearization schemes. The image pair with the maximum amount of voxel correspondences, before and after a blurry image sequence, is identified and used to compute camera motion parameters.

I have evaluated several important parameters related to TVF computation. I first studied the influence of the number of frames to formulate temporal volume.

Experimental results indicated that 20 frames optimized the performance with respect to search accuracy and computational time. Another parameter was sampling rate to construct temporal volume pyramids. Experiments showed that 0.75 is the optimal value to maintain important image structures during image smoothing and to reduce computational cost.

Three clinical sequences were chosen to evaluate the TVF algorithm. The first two clinical sequences demonstrate that accuracy can be improved through the TVF. The third sequence showed that improper choice of image pair can cause the region flow algorithm to fail. The improper image pair could be the two colonoscopy images just before and after a blurry image sequence. By contrast, the image pair chosen by TVF contained sufficient corresponding features, and the system continued to track through the end of the sequence.

CHAPTER 7: SUMMARY AND FUTURE WORK

"Whenever a new finding is reported to the world people say: It is probably not true. Later on, when the reliability of a new finding has been fully confirmed, people say: OK, it may be true but it has no real significance. At last, when even the significance of the finding is obvious to everybody, people say: Well, it might have some significance, but the idea is not new."

– Michel de Montaigne

This dissertation focuses on development of a robust visually-guided navigation system, showing its application to the co-alignment of virtual and optical colonoscopy images. It presents visual motion processing framework from pixel to region to temporal volume, to develop a robust visually-guided navigation system to accurately co-align real and virtual navigation. Phantom experiments validated that the proposed visually-guided navigation system can successfully track an optical colonoscope moving $288mm$ inside a straight phantom and $286.56mm$ curved paths, with less than $10mm$ error. Clinical data evaluation demonstrated that the proposed colonoscopy tracking system could track the most portion of the sigmoid(top row) and descending colons(bottom row), as illustrated in Fig. 7.1. There are several blurry image sequences in these two examples, and my tracking system can recover from these blurry interruptions and achieve accurate co-alignment of optical colonoscopy(OC) and virtual colonoscopy(VC) images. In the descending colonoscopy image sequence, both OC and VC reach the same fold; in the sigmoid colonoscopy image sequence, the polyp enclosed by green circles appears in both OC and VC images. These results show that the ultimate goal of tracking an entire colonoscopy sequence, is solvable in the

future, through combining the proposed tracking system with manual reinitialization, plus strategies to recover from tracking failures.

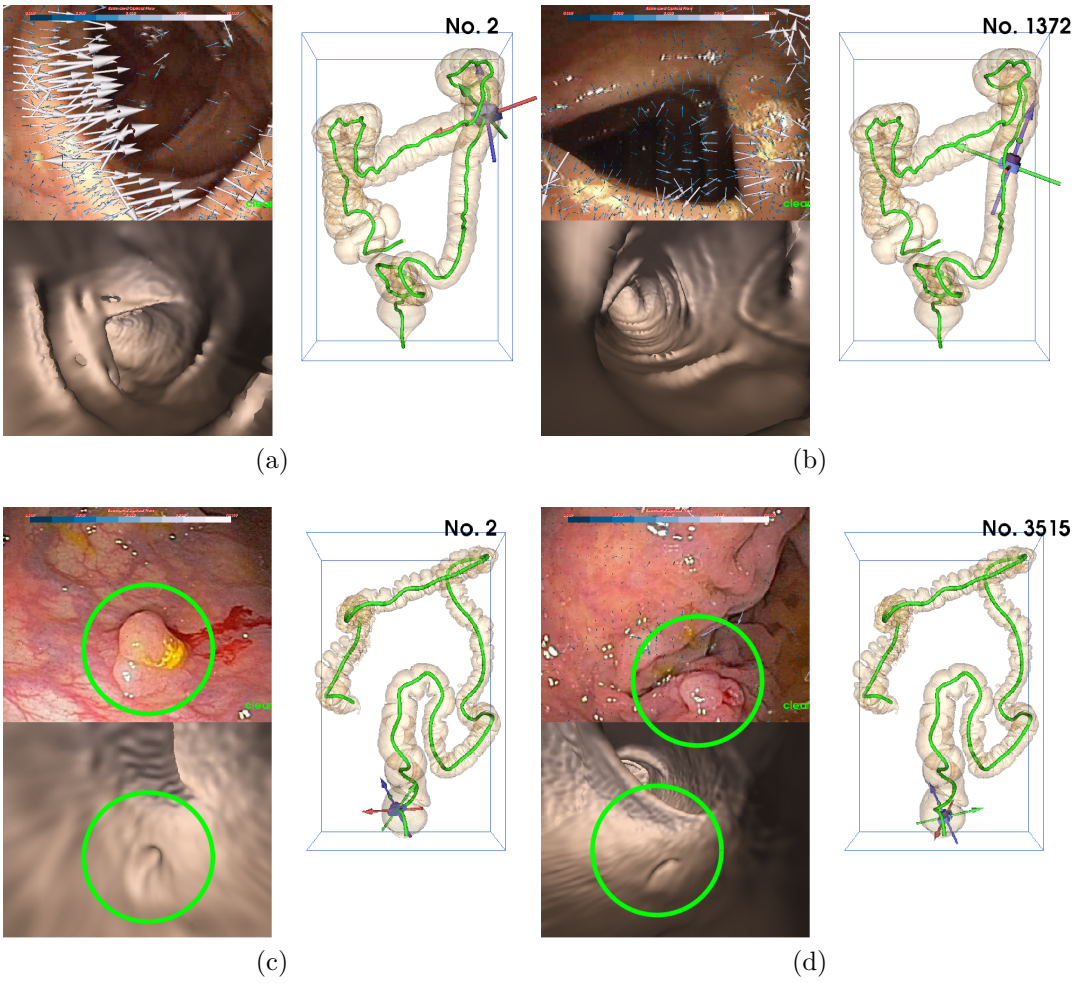


Figure 7.1: *Tracking results of the descending and sigmoid colonoscopy image sequences.* (a,b) tracking results in the descending colon; (c,d) the sigmoid colon. (a,c) the first co-aligned OC(top left image) and VC(bottom left image) images, as well as external view(right image); (b,d) the last co-aligned OC and VC images. My tracking algorithm can track the most portion of the descending colon(top row) and sigmoid colon(bottom row). Important features, the polyp, have been highlighted by green circles.

Chapters 4 through 6 presented my contributions to this dissertation work. In chapter 4, I described the problem of tracking consecutive optical colonoscopy images. I presented a colonoscopy tracking algorithm that combines sparse and dense optical flows. I then introduced a multi-scale optical flow algorithm to accurately

compute visual motion from consecutive colonoscopy images. I developed an FOE-based egomotion estimation strategy to compute camera motion parameters. Finally, I used both phantom and clinical experiments to validate the proposed multi-scale optical flow and the FOE-based egomotion estimation methods.

In Chapter 5, I described the problem of estimating large motion to continuously co-align OC and VC images when blurry images appear. I then presented an estimation algorithm which uses region flow computation and incremental egomotion estimation. Region flow densely matched all possible region pairs between two colonoscopy images interrupted by blurry images, and it measured significant image displacements. Accurate SIFT feature correspondences are generated by exploiting estimated image displacements to limit the search space of SIFT feature matching. Incremental egomotion estimation made use of SIFT feature correspondences to estimate large camera motion between the selected image pair. SIFT feature correspondences were integrated into a PDE-based scheme and artificially subdivided into a sequence of small optical flow fields. Small optical flow fields were employed to estimate camera motion parameters. Large camera motion can thus be incrementally recovered by accumulating all camera motion parameters. Continuous co-alignment of OC and VC images could be addressed by using substantial camera motion parameters to guide the virtual camera. Finally, I presented phantom validation as well as clinical demonstration of the proposed large motion estimation method.

In Chapter 6, I developed an enhancement to the region flow techniques in chapter 5, involving selection of an optimal image pair from before and after the blurry image sequence. This enhancement involves calculating temporal volume flow and using it to search for an image pair with sufficient similarity. I provided several examples demonstrating the effectiveness of this technique.

In the next part of this chapter, I discuss the results and challenges of my contributions and then conclude with areas for future work.

7.1 Discussion

In this section, I discuss the results and challenges of my work. My contributions are represented as a three-level visual motion processing framework, described in chapter 4 through 6. They include tracking of consecutive colonoscopy images based on multi-scale optical flow; large motion estimation based on region flow; and image pair search using temporal volume flow(TVF).

7.1.1 Results and Challenges of Multi-scale Optical Flow

Chapter 4 concentrates on the tracking of consecutive colonoscopy images, yielding two main contributions. The first contribution is multi-scale optical flow computation. A multi-scale selection strategy determines the optimized spatial-temporal scales, for sparse and dense optical flow computation. Sparse and dense optical flows can be computed accurately because spatial-temporal derivatives are calculated using the optimized scales. Another contribution is the determination of the FOE and its usage to sequentially compute the parameters of camera rotation and translation. Computations are stable because the FOE has the most stable camera motion information in the optical flow field.

Efficacy of the proposed colonoscopy tracking framework was demonstrated on both phantom and clinical colonoscopy image sequences. Straight and curved phantom image sequences were used to validate statistically the accuracy of tracking. Average estimated velocity error is less than $3mm/sec$ on the original and calibrated phantom image sequences at speeds of $10mm/sec$, $15mm/sec$, and $20mm/sec$. Average displacement error is less than $7mm$ as opposed to $288mm$, the actual translation distance of the colonoscope in the straight phantom, and $7mm$ as opposed to $286.56mm$ in the curved phantom. Several OC image sequences were used to demonstrate that the tracking algorithm based on optical flow was insensitive to the issues

of local deformation, artifacts, polyp removal, and multi-object motion.

However, there are several problems that limit the applicability of the algorithm to real-world clinical settings. First, the search for optimized spatial-temporal scales is a time-consuming process. The construction of a multi-scale image representation involves smoothing an image sequence several times by the Gaussian function. Therefore, the computational time of the tracking algorithm is $4sec/frame$, while the actual application requires the tracking time be at least $\frac{1}{30}sec/frame$. The computational cost must be reduced to fulfill the actual clinical requirements. Graphics processing unit(GPU) computing[152] and program optimization are two approaches that could accelerate the computation. Second, there is no metric to measure and control the accumulated tracking errors. As the errors accumulate, OC and VC images will vary significantly, causing a tracking failure. The third issue is that the first OC and VC images require manual co-alignment in order to start the tracking pipeline. However, manual co-alignment depends highly on the user's expertise. And the search for VC images to match with an OC image is nontrivial, because of shape variance between the real and virtual colons. Fourth, although insensitive to local deformation, the tracking algorithm is still sensitive to large deformation, such as stretching of the colon by a colonoscope. Resulting differences between the virtual and real colons cause the OC and VC images to be quite different, even if camera motion parameters are accurately estimated. Finally, the tracking algorithm needs a more realistic phantom to evaluate the accuracy, compared to simple Lego models.

7.1.2 Results and Challenges of Region Flow

Large motion caused by the appearance of blurry images is investigated in chapter 5. Similar to chapter 4, the main contributions can be summarized in two parts. The first contribution is the use of region flow to estimate visual motion. Because colonoscopy images fail to produce distinctive feature descriptors, many false feature

matches are generated. Region flow is developed by using a robust region descriptor to estimate all possible pixel shifts between colonoscopy images interrupted by a blurry sequence. It predefines sparse feature matching ranges, with significant improvement in the resulting accuracy. The second contribution is incremental egomotion estimation to compute large camera motion. The essential idea is the subdivision of large visual motion into a sequence of small optical flow fields, then incrementally recovering camera motion from optical flow fields.

Region flow method has been evaluated on both phantom and clinical colonoscopy image sequences. Straight and curved phantom image sequences, used in chapter 4, were again exploited to statistically evaluate accuracy of large motion estimation. Average velocity error is $3mm$ of $16mm$ traveled between two consecutive images in the straight phantom and also $3mm$ of $23.88mm$ traveled in the curved phantom. If the colonoscope is displaced at the the speed of $10mm/second$, the proposed strategy can accurately estimate large camera motion less than 12.6% relative error after excluding 72 blurry images. The region flow technique is combined with multi-scale optical flow to successfully track different colon segments and across multiple blurry images.

Unfortunately, like the tracking of consecutive images, the region flow technique cannot estimate large motion in real-time. It takes about *2 minutes*, which is far behind the needed real-time performance. It is still difficult to measure the accumulated tracking errors caused by region flow method. It is also difficult to propose a metric to compensate for tracking errors. Finally, significant colon deformation affects region flow method because of shape differences between the actual and virtual colons. For example, there may be a sharp turn in the virtual colon, but it may be deformed into a straight segment in the actual colon, due to insertion of the colonoscope. If camera motion parameters from optical images are used to directly drive the virtual camera, without accounting for deformation, then the virtual camera might jump out of the

virtual colon and result in tracking failure.

7.1.3 Results and Challenges of Temporal Volume Flow

In chapter 6, a TVF method was proposed by extending the intensity and gradient constancy models, used in optical flow computation, for use in the spatial-temporal domain and by integrating them into a variational function. A variational function was numerically computed by exploiting sequential linearization and multi-resolution temporal volume pyramids, which generates an accurate TVF. Temporal coherence was measured by counting the number of voxel correspondences connected by TVF vectors. An image pair with sufficient similarity was defined as two images with the maximum number of voxel correspondences. This intelligent image pair selection greatly enhanced region flow based large motion estimation.

Because the colonoscope's motion is unknown in OC images, the TVF algorithm was validated by visually inspecting the similarity of co-aligned OC and VC image pairs, with and without the TVF strategy. Three colonoscopy image sequences having blurry images were used to demonstrate accuracy and robustness of large motion estimation. Two image sequences demonstrated that the accuracy was improved using the TVF algorithm. The co-aligned OC and VC images were more similar than when not using TVF. The third image sequence showed that TVF also enhanced stability of large motion estimation. The original region flow algorithm had failed if two images were randomly selected.

However, like other techniques, the TVF calculation is computationally intensive, requiring approximately *2 minute* to complete. The biggest challenge in validating the TVF technique was that quantifiable metrics were difficult to develop, for objectively confirmation that the optimal image pair was selected. It was necessary to inspect visually the selected image pair.

7.2 Future Work

In the previous sections, I highlighted several challenges and limitations of my contributions. In this section, I will point out areas of future work to address them. Furthermore, I will discuss ways to generalize the results of my work.

Computational issue: The current tracking system cannot achieve real-time performance, especially when estimating large motion during blurry image interruption. There are two possible approaches to enhance performance. First, the current implementation should be optimized. All computations were performed in C++ on Linux, using ITK[235] and VTK[189] for image/volume processing and visualization. All these packages are good for algorithm development, but not optimized for software application. The performance of colonoscopy tracking can be improved by optimizing codes especially for colonoscopy tracking. Second, GPU computing[152] can be investigated to accelerate colonoscopy tracking.

Accumulated tracking errors: In this dissertation, I have not yet designed a metric to measure accumulated tracking errors of the tracking system. The accumulated errors are represented as the visual difference between co-aligned OC and VC images. Measuring the visual difference is a critical step in compensating for accumulated tracking errors. However, the current tracking system is able only to estimate camera motion parameters from the OC video stream, to drive the virtual camera and to co-align OC and VC images. A feedback strategy is needed to evaluate the difference between OC and VC images, so as to reduce accumulated tracking errors.

Matching OC and VC images can serve this purpose. We need to find some common attributes between OC and VC images and design a metric based on these attributes to measure image variance. There are two possible attribute candidates. One is the gradient distribution, which allows distinguishing between fold and non-fold image regions. The other is the depth value. Here, depth values of OC images can be

estimated though structure-from-motion techniques. Depth values of virtual images can be obtained from the Z-buffer of graphics cards. Consequently, the similarity between OC and VC images can be measured by comparing these two properties.

Manual alignment: The current tracking system assumes that the first OC image has been well co-aligned with a corresponding VC image. This manual adjustment severely limits its use in clinical applications. Automatic co-alignment of OC and VC images is critical in a clinical application.

Matching OC and VC images can also achieve this purpose. OC always starts from the cecum colon, which reduces the possible search space when performing image matching. Another solution is based on an external device. For example, careful location of the magnetic sensors can help reduce uncertainty of the VC camera location during image matching.

Colon deformation: Colon deformation is the main challenge that hinders the development of tracking OC images. A potential strategy is to reconstruct the trajectory of the OC camera, and to match this trajectory with local centerline segments in the VC, to understand the extent of deformation. Shape difference between real and virtual colons can therefore be measured and used to reduce the tracking errors.

Magnetic sensors can serve this purpose. We can bind multiple sensors with the colonoscope, and all these sensors would report the location information in real-time. Using a B-spline to fit the location information produces a profile of the colonoscope's movement. Assuming this profile is near the colon's centerline, it can be compared with the local centerline of a VC model to measure shape difference. After shape difference is accounted for, OC and VC image matching can be used to refine the OC and VC image co-alignment.

Realistic phantom design: The current phantom based on Lego bricks can produce accurate ground-truth values in the case of rigid motion. However, the colon is

a non-rigid organ. Therefore, a more realistic phantom should be designed, capable of undergoing deformation.

Quantifiable metrics for validating the search for an image pair: An image pair selected by TVF was validated by visual inspection in chapter 6. A quantifiable metric is needed to measure the selection results. There are two possible approaches to develop such a quantifiable metric. One strategy is to randomly select several pairs of temporal volumes. Then the most similar image pairs will be manually chosen from corresponding temporal volume pairs. These manually selected image pairs are used as the ground-truth. Next, the TVF algorithm is performed on these temporal volumes to select the image pairs. The comparison between the two sets of image pairs can be used to validate the image pair search.

An alternative approach is based on machine learning techniques[16], automatically analyzing the selected image pairs from the TVF algorithm. First, I select a set of temporal volume pairs as the training datasets. The TVF algorithm is performed on the training datasets to select a set of image pairs. Then SIFT feature correspondences are built on these selected image pairs, by using the region flow algorithm described in chapter 5. A good image pair contains a large amount of SIFT feature correspondences. Therefore, the evaluation of an image pair is equivalent to assess SIFT feature correspondences. A function that measures the number of accurate SIFT feature correspondences can be constructed[157]. The parameters of this function are determined by using machine learning techniques on the training datasets. After the parameters of the function are known, the image selection results can thus be automatically evaluated.

Generalization of the colonoscopy tracking system: Another important aspect of future work is to broaden the proposed colonoscopy tracking framework. For instance, multi-scale optical flow computation and FOE-based egomotion estimation can be used in many navigation applications. Examples include city and building nav-

igation and unmanned vehicle navigation. Both phantom and clinical experiments demonstrate that this egomotion estimation strategy can accurately estimate camera motion by using a simplified virtual model. This requirement can be easily fulfilled by using virtual building models in Google Earth. This technique can also be combined with GPS systems to enhance navigation accuracy.

Wide-baseline image matching[155, 147, 215] is an ad-hoc topic in the computer vision field. Traditional computer vision tasks, such as structure-from-motion[38], require many images to understand the current scene. These tasks now need only a few high-quality images, thanks to improvements in imaging devices. Most proposed algorithms are focusing on the development of robust feature descriptors, to avoid false feature matches. But many false feature matches still exist if the current image does not contain dominant feature points. The region flow algorithm concentrates on improving another important factor that affects feature matching accuracy—matching size. Based on my study in chapter 5, predefined matching size from region flow calculation can significantly improve accuracy of feature matching, even though the response of feature descriptors is indistinct. The application of region flow can improve feature matching accuracy on other types of images, such as natural images.

TVF has potential application to medical volume registration, because it estimates all voxel shifts. One important task in medical image analysis is the comparison between two CT volumes captured at different periods. These two CT volumes can be considered as two temporal volumes, and TVF can be applied to measure relative voxel movements between these two volumes. Thus, the progression of disease areas in CT data can be analyzed in terms of TVF vectors.

BIBLIOGRAPHY

[1] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):384–401, 1985.

[2] G. Adiv. Inherent ambiguities in recovering 3d motion and structure from a noisy flow field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):477–489, 1989.

[3] L. Alvarez, R. Deriche, J. Sanchez, and J. Weickert. Dense disparity map estimation respecting image derivatives. *Journal of Visual Communication and Image Representation*, 13(1/2):3–21, 2002.

[4] Luis Alvarez, Joachim Weickert, and Javier Snchez. Reliable estimation of dense optical flow fields with large displacements. *Intrernational Journal of Computer Vision*, 39(1):41–56, 2000.

[5] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transaction on Signal Processing*, 50:174–188, 2002.

[6] G. Aubert, R. Deriche, and P. Kornprobst. Computing optical flow via variational techniques. *SIAM Journal on Applied Mathematics*, 60:156–182, 1999.

[7] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *Intrernational Journal of Computer Vision*, 92:1–31, 2010.

[8] Robert L. Barclay, Joseph J. Vicari, Andrea S. Doughty, John F. Johanson, and Roger L. Greenlaw. Colonoscopic withdrawal times and adenoma detection during screening colonoscopy. *The New England Journal of Medicine*, 355:2533–2541, 2006.

[9] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow. *Intrernational Journal of Computer Vision*, 12(1):43–77, 1994.

[10] Dirk Bartz. Virtual endoscopy in research and clinical practice. *Computer Graphics Forum*, 24(1):1–17, 2005.

[11] A. Baumberg. Reliable feature matching across widely separated views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1774–1781, 2000.

[12] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110:346–359, 2008.

[13] S.S Beauchemin and J.L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3):433–467, 1995.

[14] Didier J.L.E. Bielen, Hilde T.C. Bosmans, Liesbeth L.I. De Wever, Frederik Maes, Sabine Tejpar, Dirk Vanbeckevoort, and Guy J.F. Marchal. Clinical validation of high-resolution fast spin-echo mr colonography after colon distention with air. *Journal of Magnetic Resonance Imaging*, 22(3):400–405, 2005.

[15] J. Bigun, G.H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, 1991.

[16] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2th edition, 2007.

[17] I. Bitter, A.E. Kaufman, and M. Sato. Penalized-distance volumetri skeleton algorithm. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):195–206, 2002.

[18] I. Bitter, M. Sato, , M. Bender, K.McDonnel, A.E. Kaufman, and M. Wan. Ceaser: A smooth, accurate and chi-square distribution centerline-extraction algorithm. In *Proceedings of IEEE Visualization 2000*, pages 45–52, Salt Lake City, UT, Oct. 8-13 2000.

[19] Michael Julian Black. *Robust Incremental Optical Flow*. PhD thesis, Yale University, 1992.

[20] M.J. Black and P. Anandan. Robust dynamic motion estimation over time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 292–302, 1991.

[21] M.J. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996.

[22] G. Borgefors. Distance transformations in digital images. *Computer Vision and Image Understanding*, 34:344–371, 1986.

[23] Jean-Yves Bouguet. Camera calibration toolbox for matlab, 2010. www.vision.caltech.edu/bouguetj/calib_doc/index.html.

[24] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1222–1239, 2001.

[25] I. Bricault, G. Ferretti, and P. Cinquin. Multi-level strategy for computer-assisted transbronchial biopsy. In *Proceedings of 1th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 161–268, 1998.

[26] T. Brox. *From Pixels to Regions: Partial Differential Equations in Image Analysis*. PhD thesis, Saarland University, Germany, 2005.

[27] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proceedings of 8th European Conference on Computer Vision*, volume 4, pages 25–36, 2004.

[28] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[29] T. Brox and J. Weickert. Nonlinear matrix diffusion for optic flow estimation. In *L. Van Gool, editor, Pattern Recognition, volume 2449 of Lecture Notes in COmputer Science*, pages 446–453, 2002.

[30] A. Bruhn. *Variational Optical Flow Computation - Accurate Modeling and Efficient Numerics*. PhD thesis, Saarland University, Germany, 2006.

[31] Andres Bruhn and Joachim Weickert. Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 749–755, 2005.

[32] A. Brunh, J. Weickert, C. Feddern, and C. Schnorr. Variational optic flow computation in real-time. *Intrernational Journal of Computer Vision*, 14(5):608–615, 2005.

[33] A. Brunh, J. Weickert, T. Kohlberger, and C. Schnorr. Discontinuity-preserving variational optic flow computation in real-time. *International Journal of Computer Vision, special issue - International Conference on Scale space and PDE methods in Computer Vision, 2005*, 70(3):257–277, 2006.

[34] A. Brunh, J. Weickert, and C. Schnorr. Combining the advantages of local and global optic flow methods. In *L. Van Gool, editor, Pattern Recognition, volume 2449 of Lecture Notes in COmputer Science*, pages 454–462, 2002.

[35] A. Brunh, J. Weickert, and C. Schnorr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *Intrernational Journal of Computer Vision*, 61(3):211–231, 2005.

[36] Anna R. Bruss and Berthold K. P. Horn. Passive navigation. *Computer Vision, Graphics and Image Processing*, 21:3–20, 1983.

[37] Stadium Buzz. Sunflower power. Web, available at `http://www.nataliamatthews.com/large-view/Petal+Pushers/59912-4-0-4049/Photography/Color/Nature.html`, 2009.

[38] L. Van Gool C. Strecha, R. Fransens. Wide-baseline stereo from multiple views: a probabilistic account. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 552–559, 2004.

[39] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[40] Joyce E. Carpenter. Apollo 17 rover. In *Apollo Images*. National Aeronautics and Space Administration, Web, available at `http://ares.jsc.nasa.gov/HumanExplore/Exploration/EXLibrary/images/apollo.cfm`, 2011.

[41] Robert E. Carroll and Steven M. Seitz. Rectified surface mosaics. *Intrernational Journal of Computer Vision*, 85(3):1–8, 2009.

[42] S. Chandrasekaran and I. C. F. Ipsen. On the sensitivity of solution components in linear systems of equations. *SIAM Journal on Matrix Analysis and Applications*, 16(1):93–112, 1995.

[43] D. Chen, Hossam Abdelmunim, Aly A. Farag, Robert L. Falk, and Gerald W. Dryden. Segmentation of colon tissue in ct colonography using adaptive level sets method. In *MICCAI 2008 Workshop: Computational and Visualization Challenges in the New Era of Virtual Colonoscopy*, pages 108–115, 2008.

[44] D. Chen, B.Li, Z. Liang, M. Wan, A.E. Kaufman, and M. Wax. A tree-branch searching multiresolution approach to skeleteonization for virtual endoscopy. In *Proceedings of SPIE Medical Imaging*, volume 3979, pages 726–734, 2000.

[45] Dongqing Chen, Rachid Fahmi, Aly A. Farag, Robert L. Falk, and Gerald W. Dryden. Accurate and fast 3d colon segmentation in ct colonography. In *Proceedings of the Sixth IEEE international conference on Symposium on Biomedical Imaging: From Nano to Macro*, pages 490–493, 2009.

[46] Dongqing Chen, Zhengrong Liang, Mark R. Wax, Lihong Li, Bin Li, and Arie E. Kaufman. A novel approach to extract colon lumen from ct images for virtual colonoscopy. *IEEE Transactions on Medical Imaging*, 19(12):1220–1226, 2000.

[47] R. Chiou, A. Kaufman, Z. Liang, L. Hong, and M. Achniotou. Interactive fly-path planning using potential fields and cell decomposition for virtual endoscopy. *IEEE Trans Nuclear Sciences*, 46(4):1045–1049, 1999.

[48] chrischerabie. The autonomous car. In *Unmanned Cars: A Reality*. Web, available at `http://cherabie.blogspot.com/2008/10/unmanned-cars-reality.html`, October, 2008.

[49] I. Cohen. Nonlinear variational method for optical flow computation. In *Proceedings of 8th Scandinavian Conference on Image Analysis*, pages 523–530, 1993.

[50] Photius Coutsoukis. The large intestine. Web, available at `http://www.theodora.com/anatomy/the_large_intestine.html`, 2007.

[51] C.Schmid, R.Mohr, and C.Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[52] N. da Vitoria Lobo and J.K. Tsotsos. Using collinear points to compute egomotion and detect nonrigidity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 344–350, 1991.

[53] N. da Vitoria Lobo and J.K. Tsotsos. Computing egomotion and detecting independent motion from image motion using collinear points. *Computer Vision and Image Understanding*, 64:21–52, 1996.

[54] D. Deguchi, K. Mori, M. Feuerstein, T. Kitasaka, C.R. Maurer Jr., Y. Suenaga, H. Takabatake, M. Mori, and H. Natori. Selective image similarity measure for bronchoscope tracking based on image registration. *Medical Image Analysis*, 13:621–633, 2009.

[55] D. Deguchi, K. Mori, Y. Suenaga, J. Hasegawa, J. Toriwaki, H. T. Batake, and H. Natori. New image similarity measure for bronchoscope tracking based on image registration. In *Proceedings of 6th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 399–406, 2003.

[56] F. Deligianni, A. Chung, and G. Z. Yang. pq-space based 2d/3d registration for endoscope tracking. In *Proceedings of 6th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 311–318, 2003.

[57] F. Deligianni, A. Chung, and G. Z. Yang. Non-rigid 2d-3d registration with catheter tip em tracking for patient specific bronchoscope simulation. In *Proceedings of 9th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 281–288, 2006.

[58] Rachid Deriche, Pierre Kornprobst, and Gilles Aubert. Optical-flow estimation while preserving its discontinuities: A variational approach. In *In Proc. Second Asian Conference on Computer Vision*, pages 290–295, 1995.

[59] DiginfoTV. Tosy topio - table tennis playing robot. Web, available at `http://www.diginfo.tv/2007/12/05/07-0601-d.php`, 2007.

[60] Suzanne Dixon. What are the different types of colon polyps? colon polyps may or may not increase cancer risk, depending on type, 2009. http://coloncancer.about.com/od/coloncancerbasics/a/polyptypes.htm.

[61] Robert A. Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. In *Proceedings of SIGGRAPH '88 Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, volume 22, pages 65–74, 1988.

[62] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach. Ridges for image analysis. *Journal of Mathematical Imaging and Vision*, 4(4):353–373, 1994.

[63] L.D. Elsgolc. *Calculus of Variations*. Dover Publications, illustrated edition, 2007.

[64] G. Farneback. Fast and accurate motion estimation using orientation tensors and parametric motion models. In *Proceedings of the 15th IEEE Conference on Pattern Recognition*, volume 1, pages 135–139, 2000.

[65] Olivier Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, 1rd edition, 1993.

[66] Olivier Faugeras and Quang-Tuan Luong. *The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. The MIT Press, 1rd edition, 2004.

[67] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *Intrernational Journal of Computer Vision*, 70(1):41–54, 2006.

[68] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *Intrernational Journal of Computer Vision*, 70(1):41–54, 2006.

[69] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance Transforms of Sampled Functions. Cornell Computing and Information Science TR2004-1963, Cornell University, 2004.

[70] David J. Fleet and Allan D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1):77–104, 1990.

[71] D.J. Fleet and K. Langley. Recursive filters for optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):61–67, 1995.

[72] M. Franaszek, R.M. Summers, P.J. Pickhardt, and J.R. Choi. Hybrid segmentation of colon filled with air and opacified fluid for ct colonography. *IEEE Transactions on Medical Imaging*, 25(3):358–368, 2006.

[73] Robert Fusco. Colon cancer. Web, available at `http://www.gihealth.com/html/education/photo/colonCancer.html`, 2008.

[74] Robert Fusco. Large colon polyp. In *What Is A Colon Polyp?* Web, available at `http://www.gihealth.com/html/education/colonpolyps.html`, 2008.

[75] T. Gautama and MA. Van Hulle. A phase-based approach to the estimation of the optical flow field using spatial filtering. *IEEE Trans Neural Network*, 13(5):1127–1136, 2002.

[76] S. Haker, S. Angenent, A. Tannenbaum, and R. Kikinis. Nondistorting flattening maps and the 3d visualization of colon ct images. *IEEE Transactions on Medical Imaging*, 19(7):665–670, 2000.

[77] J.M. Hammersley and P. Clifford. Markov fields on finite graphs and lattices, 1971.

[78] Frank R. Hampel, Elvezio M. Ronchetti, Peter J. Rousseeuw, and Werner A. Stahel. *Robust Statistics - The Approach Based on Influence Functions*. Wiley, 1 edition, 1986.

[79] C. Harris and M. J. Stephens. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference*, pages 147–152, 1988.

[80] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2rd edition, 2004.

[81] H. Haussecker and D.J. Fleet. Computing optical flow with physical models of brightness variation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 760–767, 2000.

[82] H. Haussecker and D.J. Fleet. Estimating optical flow with physical models of brightness variation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):661–673, 2001.

[83] healthwise. Colonoscopy: Anatomy of the colon. Web, available at `http://www.revolutionhealth.com/articles/colonoscopy-anatomy-of-the-colon/zm2647`, September, 2006.

[84] D.J. Heeger and A.D. Jepson. Visual perception of three-dimensional motion. *Neural Computation*, 2(2):129–137, 1990.

[85] D.J. Heeger and A.D. Jepson. Linear subspace methods for recovering translational direction. In *Proceedings of the 1991 York conference on Spacial vision in humans and robots*, pages 39–62, 1991.

[86] D.J. Heeger and A.D. Jepson. Subspace methods for recovering rigid motion 1: Algorithm and implementation. *Intrernational Journal of Computer Vision*, 7(2):95–117, 1992.

[87] J. P. Helferty and W. E. Higgins. Combined endoscopic video tracking and virtual 3d ct registration for surgical guidance. In *Proceedings of IEEE Conference on Image Processing*, pages 961–964, 2002.

[88] J. P. Helferty, A. J. Sherbondy, A. P. Kiraly, and W. E. Higgins. System for live virtual-endoscopic guidance of bronchoscopy. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 68–68, 2005.

[89] C.R. Hema, Lam Chee Kiang, and Vivian Tang Sui Lot. Housekeeping robot: From concept to design. In *5th International Colloquium on Signal Processing & Its Applications (CSPA)*, pages 52–54, 2009.

[90] W. Hibbard and D. Santek. Interactivity is the key. In *Chapel Hill Workshop on Volume Visualization*, pages 39–43, 1989. University of North Carolina, Chapel Hill.

[91] William E Higgins, James P Helferty, Kongkuo Lu, Scott A Merritt, Lav Rai, and Kun-Chang Yu. 3d ct-video fusion for image-guided bronchoscopy. *Computerized Medical Imaging and Graphics*, 32:159–173, 2007.

[92] E.C. Hildreth. Computations underlying the measurement of visual motion. *Artificial Intelligence*, 23:309–354, 1984.

[93] Ellen C. Hildreth. Recovering heading for visually-guided navigation. *Vision Research*, 32:1177–1192, 1991.

[94] Ellen C. Hildreth, H.B. Barlow, and H.C. Longuet-Higgins. Recovering heading for visually guided navigation in the presence of self-moving objects. *Philosophical Transactions: Biological Sciences*, 337(1281):305–313, 1992.

[95] L. Hong, A. Kaufman, Y. Wei, A. Viswambharan, M. Wax, and Z. Liang. 3d virtual colonoscopy. In *Proceedings of the IEEE Symposium on Biomedical Visualization*, pages 26–32, 1995.

[96] L. Hong, Z. Liang, A. Viswambharan, A. Kaufman, and M. Wax. Virtual voyage: Interactive navigation in the human colon. *ACM Transactions on Graphics*, pages 27–34, 1997.

[97] Wei Hong, Xianfeng Gu, Feng Qiu, Miao Jin, and Arie Kaufman. Conformal virtual colon flattening. In *Proceedings of the 2006 ACM symposium on Solid and physical modeling*, pages 85–93, 2006.

[98] Berthold K.P. Horn and E.J. Weldon Jr. Direct methods for recovering motion. *Intrernational Journal of Computer Vision*, 2:51–76, 1988.

[99] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(3):185–203, 1981.

[100] A. Huang, D. Roy, M. Franaszek, and R.M. Summers. Teniae coli guided navigation and registration for virtual colonoscopy. In *IEEE Visualization 2005*, pages 279–285, 2005.

[101] Peter. J. Huber. *Robust Statistics*. Wiley-Interscience, 1 edition, 1981.

[102] Robert Hummel and V. Sundareswaran. Motion parameter estimation from global flow field data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):449–476, 1993.

[103] Fabrizio Hura and Isao Shimoyama. Visual guidance of a small mobile robot using active, biologically-inspired, eye movements. In *IEEE International Conference on Robotics & Automation*, pages 1859–1864, 1998. Leuven, Belgium.

[104] Daniel P. Huttenlocher, Michael E. Leventon, and William J. Rucklidge. Visually-Guided Navigation by Comparing Two-Dimensional Edge Images. Computing Science Technical Report TR94-1407, Cornell University, 1994.

[105] National Cancer Institute. Colon and rectal cancer. Web, available at `http://www.cancer.gov/cancertopics/types/colon-and-rectal`, 2010.

[106] A.D. Jepson and D.J. Heeger. A fast subspace algorithm for recovering rigid motion. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 124–131, 1991.

[107] J.Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 384–393, 2002.

[108] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *Proceedings of the European Conference on Computer Vision*, pages 404–416, 2004.

[109] A. Kaufman, S. Lakare, K. Kreeger, and I. Bitter. Virtual colonoscopy. *Communications of the ACM*, 48(2):37–41, 2005.

[110] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 511–517, 2004.

[111] Yan Ke, Rahul Sukthankar, and Martial Hebert. Efficient visual event detection using volumetric features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 166–173, 2005.

[112] Cynthia E. Keen. Clinical trial validates ct virtual colonoscopy. European Hospital, available at `http://www.european-hospital.com/en/article/3413-Clinical_trial_validates_CT_virtual_colonoscopy.html`, 2008.

[113] David H. Kim, Perry J. Pickhardt, Andrew J. Taylor, Winifred K. Leung, Thomas C. Winter, J. Louis Hinshaw, Deepak V. Gopal, Mark Reichelderfer, Richard H. Hsu, and Patrick R. Pfau. Ct colonography versus colonoscopy for the detection of advanced neoplasia. *The New England Journal of Medicine*, 357:1403–1412, 2007.

[114] Se Hyung Kim, Jeong Min Lee, Joon-Goo Lee, Jong Hyo Kim, Philippe A. Lefere, Joon Koo Han, and Byung Ihn Choi. Computer-aided detection of colonic polyps at ct colonography using a hessian matrixbased algorithm: Preliminary study. *American Journal of Roentgenology*, 189:41–51, 2007.

[115] Gina Kloata. Colonoscopies miss many cancers, study finds. The New York Times, Web, available at `http://query.nytimes.com/gst/fullpage.html?res=9C04E4DA1E3AF935A25751C1A96E9C8B63&sec=&spon=&pagewanted=`, 2008.

[116] Jana Kosecka. Visually guided navigation, 1996.

[117] Arun Kumar, Allen R. Tannenbaum, and Gary J. Balas. Optical flow: A curve evolution approach. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 5:598–610, 1996.

[118] Philippe Lacroute and Marc Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *Proceedings of SIGGRAPH '94*, pages 451–458, 1994.

[119] Shang-Hong Lai and Baba C. Vemuri. Reliable and efficient computation of optical flow. *Intrernational Journal of Computer Vision*, 29(2):87–105, 1998.

[120] S. Lakare, M. Wan, M. Sato, and A. Kaufman. 3d digital cleansing using segmentation rays. In *Visualization 2000*, pages 37–44, 2000.

[121] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2):107–123, 2005.

[122] I. Laptev and T. Lindeberg. Space-time interest points. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, pages 432–439, 2003.

[123] Taek-Hee Lee, Ho Lee Jeongjin Lee and, Heewon Kye, Yeong Gil Shin, and Soo Hong Kim. Fast perspective volume ray casting method using gpu-based acceleration techniques for translucency rendering in 3d endoluminal ct colonography. *Computers in Biology and Medicine*, 39:657–666, 2009.

[124] Jenifer K. Lehrer. Normal anatomy. Colon Diverticula Image, Web, available at `http://healthguide.howstuffworks.com/colon-diverticula-picture.htm`, 2006.

[125] Marc Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8:29–37, 1988.

[126] Marc Levoy. Efficient ray tracing of volume data. *ACM Transactions on Graphics*, 9:245–261, 1990.

[127] Marc Levoy. A hybrid ray tracer for rendering polygon and volume data. *IEEE Computer Graphics and Applications*, pages 33–40, 1990.

[128] Jiang Li, Jianhua Yao, Ronald M. Summers, and Amy K. Hara. An efficient feature selection algorithm for computer-aided polyp detection. In *FLAIRS Conference 2005*, pages 381–386, 2005.

[129] L. Li, D. Chen, S. Lakare, K. Kreeger, I. Bitter, A. Kaufman, M. Wax, P. Djuric, and Z. Liang. An image segmentation approach to extract colon lumen through colonic material tagging and hidden markov random field model for virtual colonoscopy. In *SPIE Medical Imaging*, pages 406–411, 2002.

[130] Stan Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer, 2 edition, 2001.

[131] W. Li, A. Kaufman, and K. Kreeger. Real-time volume rendering for virtual colonoscopy. In *Proceedings of Volume Graphics*, pages 363–375, 2001.

[132] D. Lieberman. Quality and colonoscopy: a new imperative. *Gastrointestinal endoscopy*, 61(3):392–394, 2005.

[133] John Lim and Nick Barnes. Directions of egomotion from antipodal points. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008. June 27-28, Anchorage, Alaska.

[134] John Lim and Nick Barnes. Estimation of the epipole using optical flow at antipodal points. *Computer Vision and Image Understanding*, 114(2):245–253, 2009.

[135] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Springer, 1rd edition, 1993.

[136] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. *Intrernational Journal of Computer Vision*, 30(2):117–154, 1998.

[137] T. Lindeberg. Feature detection with automatic scale selection. *Intrernational Journal of Computer Vision*, 30(2):77–116, 1998.

[138] T. Lindeberg and J. Garding. Shape-adapted smoothing in estimation of 3d depth cues from affine distortions of local 2d structure. In *Proceedings of the 3th European Conference on Computer Vision*, pages A:389–400, 1994.

[139] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: dense correspondence across different scenes. In *Proceedings of European Conference on Computer Vision (ECCV)*, Marseille, France, 2008.

[140] Haiying Liu, Rama Chellappa, and Azriel Rosenfeld. Accurate dense optical flow estimation using adaptive structure tensors and a parametric model. *IEEE Transactions on Image Processing*, 12(10):1170–1180, 2003.

[141] J. Liu, K.R. Subramanian, T. Yoo, and R. Uitert. A stable optic-flow based method for tracking colonoscopy images. In *Proc. of Mathematical Methods in Biomedical Image Analysis*, pages 1–8, 2008. June 27-28, Anchorage, Alaska.

[142] Jianfei Liu and K. R. Subrmanian. Robust centerline extraction from tubular structures in medical images. In *Proceedings of SPIE Medical Imaging*, volume 6509, page 65092V, 2007.

[143] Jianfei Liu and Xiaopeng Zhang. Enclosure sphere based cell visibility for virtual endoscopy. In *Proceedings of IEEE Visualization 2005*, page 98, 2005.

[144] Renting Liu, Zhaorong Li, and Jiaya Jia. Image partial blur detection and classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008. June 27-28, Anchorage, Alaska.

[145] H.C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. In *Proceedings of the Royal Society of London*, pages 385–397, 1980.

[146] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proc. ACM SIGGRAPH Computer Graphics*, volume 21, pages 163–169, 1987.

[147] D. Lowe. Distinctive image features from scale-invariant keypoints. *Intrernational Journal of Computer Vision*, 60(2):91–110, 2004.

[148] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 281–288, 1981.

[149] D. Marr. *Vision*. W.H.Freeman and Co Ltd, 1rd edition, 1982.

[150] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[151] Natalia Matthews. Seattle market sunflower. In *A collection of various images from around this glorious world. California*. Gallery, Web, available at `http://www.nataliamatthews.com/large-view/Petal+Pushers/59912-4-0-4049/Photography/Color/Nature.html`, 2011.

[152] Wen mei W. Hwu. *GPU Computing Gems Emerald Edition (Applications of GPU Computing Series)*. Morgan Kaufmann, 1rd edition, 2011.

[153] E. Memin and P. Perez. Robust discontinuity-preserving model for estimating optical flow. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 1, pages 920 –924 vol.1, 25-29 1996.

[154] K. Mikolajczyk. *Detection of local features invariant to affine transformations*. PhD thesis, INPG Grenoble, French, 2002.

[155] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *Intrernational Journal of Computer Vision*, 60(1):63–86, 2004.

[156] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[157] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Intrernational Journal of Computer Vision*, 65(1):43–72, 2005.

[158] Venkat Mohan. Colonoscopy. In *The Basics of Colonoscopy*. WebMD, Web, available at `http://www.webmd.com/digestive-disorders/colonoscopy`, 2010.

[159] K. Mori, D. Deguchi, K. Akiyama, T. Kitasaka, C. R. Maurer Jr., Y. Suenaga, H. Takabatake, M. Mori, and H. Natori. Hybrid bronchoscope tracking using a magnetic tracking sensor and image registration. In *Proceedings of 8th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 543–555, 2005.

[160] K. Mori, D. Deguchi, J. Sugiyama, Y. Suenaga, J. Toriwaki, C.R. Maurer Jr., H. Takabatake, and H. Natori. Tracking of a bronchoscope using epipolar geometry analysis and intensity-based image registration of real and virtual endoscopic images. *Medical Image Analysis*, 6(3):321–336, 2002.

[161] K. Mori, J. Hasegawa, J. Toriwaki, S. Yokoi, H. Anno, and K. Katada. A method to extract pipe structured components in three dimensional medical images and simulation of bronchus endoscope images. In *Proceedings of the 3D Image Conference*, pages 269–274, 1994.

[162] Peter Mountney, Danail Stoyanov, and Guang-Zhong Yang. Three-dimensional tissue deformation recovery and tracking: Introducing techniques based on laparoscopic or endoscopic images. *IEEE Signal Processing Magazine*, 27:14–24, 2010.

[163] J. Nagao, K. Mori, T. Enjouji, and D. Deguchi. Fast and accurate bronchoscope tracking using image registration and motion prediction. In *Proceedings of 7th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 551–558, 2004.

[164] H. H. Nagel and A. Gehrke. Spatiotemporally adaptive estimation and segmentation of of-fields. In *Proceedings of the first European Conference on Computer Vision*, pages 86–102, 1998.

[165] Hans-Hellmut Nagel. Constraints for the estimation of displacement vector fields from image sequences. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 945–951, 1983.

[166] Hans-Hellmut Nagel. Extending the 'oriented smoothness constraint' into the temporal domain and the estimation of derivatives of optical flow. In *Proceedings of the first European Conference on Computer Vision*, pages 139–148, 1990.

[167] Delphine Nain, Steven Haker, W. Eric L. Grimson, Eric Cosman Jr, William W. Wells, Hoon Ji, Ron Kikinis, and Carl-Fredrik Westin. Intra-patient prone to supine colon registration for synchronized virtual colonoscopy. In *Proceedings of MICCAI 2002*, pages 573 –580, 2002.

[168] J. Nappi, H. Frimmel, A. Okamura, A. H. Dachman, and H. Yoshida. Region-based supine-prone correspondence for reduction of false positives in cad of ct colonography. In *Proceedings of the 18th International Congress and Exhibition on Computer Assisted Radiology and Surgery*, pages 993–998, 2004.

[169] S. Negahdaripour, A. Shokrollahi, J. Fox, and S. Arora. Improved methods for undersea optical stationkeeping. In *IEEE International Conference on Robotics & Automation*, pages 2752–2758, 1991. Sacramento, CA , USA.

[170] S.B. Nickerson, M. Jenkin, E. Milios, B. Down, P. Jasiobedzki, A. Jepson, D. Terzopoulos, J. Tsotsos, D. Wilkes, N. Bains, and K. Tran. Design of ark, a sensor-based mobile robot for industrial environments. In *Intelligent Vehicles 93*, pages 252–257, 1993. Yokohama, Japan.

[171] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 652–659, 2004.

[172] The Future of Things. The rise and fall of asimo. Web, available at `http://thefutureofthings.com/pod/121/the-rise-and-fall-of-asimo.html`, 2010.

[173] JungHwan Oh, Sae Hwang, Yu Cao, W. Tavanapong, Danyu Liu, J. Wong, and P.C. de Groen. Measuring objective quality of colonoscopy. *IEEE Transactions on Biomedical Engineering*, 56:2190–2196, 2009.

[174] A. Pabby, R. E. Schoen, J. L. Weissfeld, R. Burt, J. W. Kikendall, P. Lance, E. Lance, and A. Schatzkin. Analysis of colorectal cancer occurence during surveillance colonoscopy in the dietary prevention trial. *Gastrointestinal endoscopy*, 21:392–394, 1983.

[175] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *Intrernational Journal of Computer Vision*, 67(2):141–158, 2006.

[176] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.

[177] P.J. Pickhardt, P.A. Nugent, P.A. Mysliwiec, J.R. Choi, and W.R. Schindler. Location of adenomas missed by optical colonoscopy. *Annals of Internal Medicine*, 141:42–42, 2004.

[178] K. Prazdny. Egomotion and relative depth map from optical flow. *Biological Cybernetics*, 36:87–102, 1980.

[179] K. Prazdny. Determining the instantaneous direction of motion from optical flow generated by a curvilinearly moving observer. *Computer Graphics and Image Processing*, 17:238–248, 1981.

[180] L. Rai, S. A. Merritt, and W. E. Higgins. Real-time image-based guidance method for lung-cancer assessment. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2437–2444, 2006.

[181] J.H. Reiger and D.T. Lawton. Processing differential image motion. *Journal of the Optical Society of America A*, 2(2):354–359, 1985.

[182] Douglas K. Rex. Colonoscopy withdrawal time predicts adenoma detection rates. *Journal Watch Gastroenterology*, 2006.

[183] R. D. Richtmyer and K. W. Morton. *Difference Methods for Initial-Value Problems*. Krieger Pub Co, 2rd edition, 1994.

[184] P. J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984.

[185] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. Wiley, first edition, 1987.

[186] T. W. Ryan. *The Prediction of Cross-Correlation Accuracy in Digital Stereo-Pair Images*. PhD thesis, University of Arizona, 1981.

[187] Mie Sato, Sarang Lakare, Ming Wan, Arie Kaufman, Zhengrong Liang, and Mark Wax. An automatic colon segmentation for 3d virtual colonoscopy. *IEICE Trans. Information and Systems*, 84(1):201–208, 2001.

[188] Yoshinobu Sato, Shin Nakajima, Nobuyuki Shiraga, Hideki Atsumi, Shigeyuki Yoshida, Thomas Koller, Guido Gerig, and Ron Kikinis. Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical Image Analysis*, 2(2):143–168, 1998.

[189] W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Prentice Hall Inc., 4th edition, 2006. www.vtk.org.

[190] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th international conference on Multimedia*, pages 357–360, 2007.

[191] S. Se, D. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2051–2058, 2001.

[192] I. Serlie, R. Truyen, J. Florie, F. H. Post, L. J. van Vliet, and F. Vos. Computed cleansing for virtual colonoscopy using a three-material transition model. In *Proceedings of MICCAI 2003*, pages 175–183, 2003.

[193] S. Seshamani, W. Lau, and G. hager. Real-time endoscopic mosaicking. In *Proceedings of MICCAI 2006*, pages 355 –363, 2006.

[194] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[195] D. Shulman and J. Y. Aloimonos. (non-)rigid motion interpretation: A regularized approach. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 233:217–234, 1988.

[196] A. Singh. Optic flow computation: A united perspective. In *IEEE Computer Society Press*, 1992.

[197] Ajit Singh. An estimation-theoretic framework for image-flow computation. In *Proceedings of the Third IEEE International Conference on Computer Vision*, pages 168–177, 1990.

[198] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *Proceedings of 9th IEEE International Conference on Computer Vision*, volume 2, pages 1470–1477, 2003.

[199] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *The International Journal of Robotics Research*, 5(4):56–68, 1986.

[200] R. C. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Proceedings of the Second Annual Conference on Uncertainty in Artificial Intelligence*, pages 435–461, 1986. University of Pennsylvania, Philadelphia, PA, USA.

[201] G. Stewart and J. Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

[202] Ronald M. Summers. Current concepts and future directions in computer-aided diagnosis for ct colonography. In *Computer Assisted Radiology and Surgery (CARS)*, 2002.

[203] V. Sundareswaran. Egomotion from global flow field data. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 140–145, 1991.

[204] M. Tabb and N. Ahuja. Multiscale image segmentation by integrated edge and region detection. *IEEE Transactions on Imaging Processing*, 6:642–655, 1997.

[205] C. J. Taylor and D. J. Kriegman. Vision-based motion planning and exploration algorithms for mobile robots. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, 1991.

[206] Jonathan Taylor, Allan D. Jepson, and Kiriakos N. Kutulakos. Non-rigid structure from locally-rigid motion. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2761–2768, 2010.

[207] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *pami*, 8:413–424, 1986.

[208] ThunderBolt. Rq-4a global hawk, unmanned aerial vehicle (uav). In *U.S. Air Force and Navy see savings in joint drone work*. Web, available at `http://www.armybase.us/2010/07/u-s-air-force-and-navy-see-savings-in-joint-drone-work/`, July, 2010.

[209] T. Tian, C. Tomasi, and D. Heeger. Comparison of approaches to egomotion computation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 315–320, 1996.

[210] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill Posed Problems*. Vh Winston, 1rd edition, 1977.

[211] C. Tomasi and J. Shi. Direction of heading from image deformations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 422–427, 1993.

[212] C. Tomasi and J. Shi. Image deformations are better than optical flow. *Mathematical and Computer Modelling Journal*, 24(5/6):165–175, 1996.

[213] T.Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *Proceedings of the British Machine Vision Conference*, pages 412–425, 2000.

[214] T.Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *Intrernational Journal of Computer Vision*, 59(1):61–85, 2004.

[215] T. Tuytelaars and K. Mikolajczyk. *Local Invariant Feature Detectors: A Survey*. Now Publishers Inc, 1rd edition, 2008.

[216] Robert L. Van Uitert and Ronald M. Summers. Automatic correction of level set based subvoxel precise centerlines for virtual colonoscopy using the colon outer wall. *IEEE Transactions on Medical Imaging*, 26(8):1069–1078, 2007.

[217] Shimon Ullman. *The Interpretation of Visual Motion*. The MIT Press, 1rd edition, 1979.

[218] A. Verri, F. Girosi, and V. Torre. Mathematical properties of the two-dimensional motion field: from singular points to motion parameters. *J. Opt. Soc. Am. A*, 6:698–712, 1989.

[219] viatronix. V3d-colon, 2011.

[220] Anna Vilanova, Rainer Wegenkittl, Andreas Knig, and Eduard Grller. Nonlinear virtual colon unfolding. In *Proceedings of Visualization '01*, pages 411–420, 2001.

[221] DJ Vining. Virtual endoscopy: is it reality? *Radiology*, 200:30–31, 1996.

[222] DJ Vining and DW Gelfand. Noninvasive colonoscopy using helical ct scanning, 3d reconstruction, and virtual reality. In *the 1994 Meeting of the Society of Gastrointestinal Radiologists*, pages 16–18, Maui, Hawaii, 1994.

[223] Andrew Walbank. Ct scan or cat scan (computed tomography imaging). Web, available at `http://www.virtualmedicalcentre.com/healthinvestigations.asp?sid=2`, 2008.

[224] M. Wan, F. Dachille, and A. Kaufman. Distance-field based skeletons for virtual navigation. In *Visualization 2001*, pages 239 – 246, 2001.

[225] M. Wan, Z. Liang, I. Bitter, and A.E. Kaufman. Automatic centerline extraction for virtual colonoscopy. *IEEE Transactions on Medical Imaging*, 21(12):1450–1460, 2002.

[226] J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner, 1rd edition, 1998.

[227] J. Weickert and C. Schnorr. A theoretical framework for convex regularizers in pde-based computation of image motion. *Intrernational Journal of Computer Vision*, 45(3):245–264, 2001.

[228] J. Weickert and C. Schnorr. Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision*, 14(3):245–255, 2001.

[229] Martin Welk, Joachim Weickert, and Gabriele Steidl. A four-pixel scheme for singular differential equations. In *Lecture Notes in Computer Science, Scale Space and PDE Methods in Computer Vision*, pages 610–621, 2005.

[230] Lee. A Westover. *Splatting: A Parallel, Feed-Forward Volume Rendering Algorithm*. PhD thesis, University of North Carolina at Chapel Hill Chapel Hill, 1991.

[231] Geert Willems, Tinne Tuytelaars, and Luc Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Proceedings of the 10th European Conference on Computer Vision*, pages 650–663, 2008.

[232] A. P. Witkin. Scale-space filtering. In *Proceedings of the Eight Int. Joint Conf. on Artificial Intelligence*, volume 57, pages 99–110, 1993.

[233] Jianhua Yao, Meghan Miller, Marek Franaszek, and Ronald M. Summers. Colonic polyp segmentation in ct colonography-based on fuzzy clustering and deformable models. *IEEE Transactions on Medical Imaging*, 23(11):1344–1352, 2004.

[234] Ming Ye and Robert M. Haralick. Two-stage robust optical flow estimation. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:2623, 2000.

[235] T.S. Yoo, editor. *Insight into Images: Principles and Practice for Segmentation, Registration, and Image Analysis*. A.K. Peters, 2004. http://www.itk.org.

[236] David M. Young. *Iterative Solution of Large Linear Systems (Computer Science and Applied Mathematics)*. Academic Pr, 1rd edition, 1971.

[237] J. Yuh and S. Negahdaripour. Adaptive control with visual sensing for an rov in unstructured undersea environments. *International Journal of Robotics and Automation*, 5:32–39, 1990.

[238] Junku YuH. A neural net controller for underwater robotic vehicles. *IEEE JOURNAL OF OCEANIC ENGINEERING*, 15:161–166, 1990.

[239] John Zhang. *Computing camera heading: a study*. PhD thesis, Stanford University, 1999.

[240] Lingxiao Zhao, Charl Botha, Javier Bescos, Roel Truyen, Frans Vos, and Frits Post. Lines of curvature for polyp detection in virtual colonoscopy. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):885–892, 2006.

[241] Yong Zhou and Arthur W. Toga. Efficient skeletonization of volumetric objects. *IEEE Transactions on Visualization and Computer Graphics*, 5(3):196–209, 1999.

APPENDIX A: CALCULUS OF VARIATIONS

Here, I describe the minimization of variational function through calculus of variations. Consider the following energy function,

$$E(\overrightarrow{u}(\boldsymbol{p})) = \int_{\Omega} (\underbrace{M(D^k F, \overrightarrow{u})}_{\text{Data constraint}} + \underbrace{\alpha S(\nabla F, \nabla \overrightarrow{u})}_{\text{Smoothness constraint}}) d\boldsymbol{p} \tag{A.1}$$

where $\boldsymbol{p} = (x_1, x_2, \cdots, x_n)$ denotes a n-coordinate point and $\overrightarrow{u} = (u_1, u_2, \cdots, u_m)$ is a m-tuple to be estimated. Assume $D^k F$ is the set of all partial derivatives of F of order k. $M(D^k F, \overrightarrow{u})$ is a data assumption and $S(\nabla F, \nabla \overrightarrow{u})$ is a data smoothness constraint. α is a constant to balance data and smoothness terms.

Let me simplify Eq. A.1 into

$$E(y(x)) = \int_a^b F(x, y(x), y'(x)) dx \tag{A.2}$$

and derive its *Euler-Lagrange equations* to really understand the property of minimization through calculus of variations. Here, the end points $y(x_0) = y_0$ and $y(x_1) = y_1$ of admissible curves are fixed. Assuming that an extremum occurs along a curve $y = y(x)$, we take any admissible curve $y = y^*(x)$, neighboring to $y = y(x)$, and set up a one-parameter family of curves

$$\begin{aligned} y(x, \alpha) &= y(x) + \alpha(y^*(x) - y(x)) \\ \delta y &= y^*(x) - y(x) \\ (\delta y)' &= y^{*\prime}(x) - y'(x) = \delta y' \end{aligned}$$

Assume

$$E(y(x, \alpha)) = \varphi(\alpha) \tag{A.3}$$

we have

$$\varphi'(0) = 0 \tag{A.4}$$

Since

$$\varphi(\alpha) = \int_{x_0}^{x_1} F(x, y(x,\alpha), y'(x,\alpha))dx \tag{A.5}$$

we have

$$\varphi'(\alpha) = \int_{x_0}^{x_1} \left[\partial_y F \partial_\alpha y(x,\alpha) + \partial_{y'} F \partial_\alpha y'(x,\alpha)\right] dx \tag{A.6}$$

where

$$\begin{aligned} \partial_y F &= \frac{\partial}{\partial y} F(x, y(x,\alpha), y'(x,\alpha)) \\ \partial_{y'} F &= \frac{\partial}{\partial y'} F(x, y(x,\alpha), y'(x,\alpha)) \end{aligned} \tag{A.7}$$

Because of the relations

$$\begin{aligned} \partial_\alpha y(x,\alpha) &= \frac{\partial}{\partial \alpha}(y(x) + \alpha\delta y) = \delta y \\ \partial_\alpha y'(x,\alpha) &= \frac{\partial}{\partial \alpha}(y'(x) + \alpha\delta y') = \delta y' \end{aligned} \tag{A.8}$$

it follows that

$$\begin{aligned} \varphi'(\alpha) &= \int_{x_0}^{x_1} \left[\partial_y F(x, y(x,\alpha), y'(x,\alpha))\delta y + \partial_{y'} F(x, y(x,\alpha), y'(x,\alpha))\delta y'\right] dx \\ \varphi'(0) &= \int_{x_0}^{x_1} \left[\partial_y F(x, y(x), y'(x))\delta y + \partial_{y'} F(x, y(x), y'(x))\delta y'\right] dx \end{aligned} \tag{A.9}$$

The condition of the extremum is therefore

$$\int_{x_0}^{x_1} (\partial_y F \delta y + \partial_{y'} F \delta y')dx = 0 \tag{A.10}$$

As $\delta y' = (\delta y)'$, we have

$$\delta E = [\partial_{y'} F \delta y]_{x_0}^{x_1} + \int_{x_0}^{x_1} (\partial_y F - \frac{d}{dx}\partial_{y'} F)\delta y dx \tag{A.11}$$

Because

$$\begin{aligned} \delta y|_{x=x_0} &= y^*(x_0) - y(x_0) = 0 \\ \delta y|_{x=x_1} &= y^*(x_1) - y(x_1) = 0 \end{aligned} \tag{A.12}$$

we can obtain

$$\delta E = \int_{x_0}^{x_1} (\partial_y F - \frac{d}{dx}\partial_{y'} F)\delta y dx \tag{A.13}$$

Therefore, the necessary condition for an extremum takes the following form

$$\int_{x_0}^{x_1} (\partial_y F - \frac{d}{dx}\partial_{y'} F)\delta y dx = 0 \tag{A.14}$$

As $\delta y \equiv 0$ only occurs in the boundary points in Eq. A.12,

$$\partial_y F - \frac{d}{dx}\partial_{y'} F = 0 \tag{A.15}$$

This is the *Euler-Lagrange equation* of Eq. A.2. High order energy function can be derived based on the same procedure.

APPENDIX B: LINEAR ISOTROPIC SCALE SPACE

This appendix describes some important properties of linear isotropic scale space. They are important for early computer vision problems. Linear isotropic scale space is defined as

$$L(\boldsymbol{p};\tau) = I(\boldsymbol{p}) * G(\boldsymbol{p};\sigma^2) \tag{B.1}$$

where $\boldsymbol{p} = (x_1, x_2, \cdots, x_n)$ denotes a n-coordinate point. Linear isotropic scale space has several useful properties.

Causality: No new level surfaces are created when the scale parameter σ^2 is increased.

Isotropy and Homogeneity: Spatial positions and scale levels can be treated in a similar manner.

Semi-group:

$$G(\cdot;\sigma_1^2) * G(\cdot;\sigma_2^2) * I(\cdot) = G(\cdot;\sigma_1^2 + \sigma_2^2) * I(\cdot) \tag{B.2}$$

Scaling:

$$L(\boldsymbol{p};\sigma^2) = \tilde{L}(s\boldsymbol{p};s^2\sigma^2) \tag{B.3}$$
$$\tilde{L}(s\boldsymbol{p};s^2\sigma^2) = \tilde{I}(s\boldsymbol{p}) * G(s\boldsymbol{p};s^2\sigma^2), \quad \tilde{I}(s\boldsymbol{p}) = I(\boldsymbol{p})$$

Semi-group and scaling properties express that scale response $L(\boldsymbol{p};\sigma^2)$ is linearly dependent on the scale parameter σ^2. Thus, the Gaussian scale space is also called linear scale space.

Commutative:

$$(\frac{\partial^n I}{\partial x^n}) * G = \frac{\partial^n (I * G)}{\partial x^n} = (\frac{\partial^n G}{\partial x^n}) * I \tag{B.4}$$

Instead of computing derivatives of a possibly discontinuous I, we can differentiate the continuous Gaussian function and convolve I by the Gaussian derivative. The *semi-group* property is reformulated as

$$\frac{\partial^m G(\cdot;\sigma_1^2)}{\partial x^m} * \frac{\partial^n G(\cdot;\sigma_2^2)}{\partial y^n} * I(\cdot) = \frac{\partial^{n+m} G(\cdot;\sigma_1^2+\sigma_2^2)}{\partial x^m \partial y^n} * I(\cdot), \tag{B.5}$$

and the *scaling property*

$$\begin{gathered} \frac{\partial^n L(\boldsymbol{p};\sigma^2)}{\partial x^n} = s^n \frac{\partial^n \tilde{L}(\tilde{\boldsymbol{p}};\tilde{\sigma}^2)}{\partial \tilde{x}^n} \\ \tilde{\boldsymbol{p}} = s\boldsymbol{p}, \qquad \tilde{\sigma} = s\sigma \end{gathered} \tag{B.6}$$

where n, m denote the order of differentiation. As we can see, the scaling property for scale-space derivatives of Eq. B.6 differs from Eq. B.3 by a factor s^n in the amplitude.

Scale-normalized derivative operators: In order to compensate for this variation, Lindeberg[135] introduce dimensionless coordinates $\bar{\boldsymbol{p}} = (\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_n)$ and $\bar{\boldsymbol{q}} = (\bar{y}_1, \bar{y}_2, \cdots, \bar{y}_n)$

$$\begin{gathered} \bar{x}_1 = x_1/\sigma, \cdots \bar{x}_n = x_n/\sigma, \\ \bar{y}_1 = \tilde{x}_1/\tilde{\sigma}, \cdots \bar{y}_n = \tilde{x}_n/\tilde{\sigma} \end{gathered} \tag{B.7}$$

and correspondingly

$$\begin{gathered} \partial_{\bar{x}_1^m} = \sigma^m \partial_{x_1^m}, \cdots, \partial_{\bar{x}_n^m} = \sigma^m \partial_{x_n^m}, \\ \partial_{\bar{y}_1^m} = \tilde{\sigma}^m \partial_{\tilde{x}_1^m}, \cdots, \partial_{\bar{y}_n^m} = \tilde{\sigma}^m \partial_{\tilde{x}_n^m} \end{gathered} \tag{B.8}$$

The scaling property of scale-space derivatives is thereby exactly the same as

the scaling property of the scale-space representation L.

$$\partial_{\bar{x}_1^m} L(\boldsymbol{p}; \sigma) = \partial_{\bar{y}_1^m} \tilde{L}(\tilde{\boldsymbol{p}}; \tilde{\sigma}), \cdots, \partial_{\bar{x}_n^m} L(\boldsymbol{p}; \sigma) = \partial_{\bar{y}_n^m} \tilde{L}(\tilde{\boldsymbol{p}}; \tilde{\sigma}) \tag{B.9}$$

APPENDIX C: NONLINEAR ANISOTROPIC SCALE SPACE

In order to better retain local image structures, anisotropic Gaussian scale space is used to detect image features. The scale parameter $\mathbf{\Sigma}$ should be related to the anisotropy of the local image structure. In the two dimensional image domain, the anisotropy is measured in terms of the *structure tensor* $\mathbf{J}(x, y; \Sigma_s, \Sigma_w)$[138, 226].

$$\begin{aligned}\mathbf{J}(\cdot; \Sigma_s, \Sigma_w) &= G(\cdot; \Sigma_w) * (\nabla L \otimes \nabla L) = G(\cdot; \Sigma_s) * (\nabla L \nabla L^T) \\ &= \iint_{(\xi,\zeta)\in\mathbb{R}^2} (\nabla L(\xi, \zeta; \mathbf{\Sigma}_s))(\nabla L(\xi, \zeta; \mathbf{\Sigma}_s))^T G(x - \xi, y - \zeta; \mathbf{\Sigma}_w) d\xi d\zeta \\ &= \iint_{(\xi,\zeta)\in\mathbb{R}^2} \begin{pmatrix} (\partial_\xi L)^2 & (\partial_\xi L)(\partial_\zeta L) \\ (\partial_\xi L)(\partial_\zeta L) & (\partial_\zeta L)^2 \end{pmatrix} G(x - \xi, y - \zeta; \mathbf{\Sigma}_w) d\xi d\zeta\end{aligned} \tag{C.1}$$

The gradient derivatives are smoothed by a Gaussian function with a local spatial scale Σ_s (derivative scale). All derivatives are also averaged in a neighborhood by a Gaussian window function with variance Σ_w (integration scale).

An important property of anisotropic-Gaussian scale space is the commutative property. Let $I_L, I_R : \mathbb{R}^2 \to \mathbb{R}$ denote two intensity patterns related by an invertible linear transformation $\boldsymbol{q} = \mathbf{B}\boldsymbol{p}$, i.e.,

$$I_L(\boldsymbol{p}) = I_R(\mathbf{B}\boldsymbol{p}) \tag{C.2}$$

and define the anisotropic-Gaussian scale-space representations by

$$L(\cdot; \mathbf{\Sigma}^L) = G(\cdot; \mathbf{\Sigma}^L) * I_L(\cdot) \tag{C.3}$$

$$R(\cdot; \mathbf{\Sigma}^R) = G(\cdot; \mathbf{\Sigma}^R) * I_R(\cdot) \tag{C.4}$$

where $\mathbf{\Sigma}^L, \mathbf{\Sigma}^R$ are symmetric positive semi-definite matrices. Then L and R are

related by

$$L(\boldsymbol{p};\boldsymbol{\Sigma}^L) = R(\boldsymbol{q};\boldsymbol{\Sigma}^R) \tag{C.5}$$

where

$$\boldsymbol{\Sigma}^R = \mathbf{B}\boldsymbol{\Sigma}^L\mathbf{B}^\mathsf{T} \tag{C.6}$$

Hence, for any matrix $\boldsymbol{\Sigma}^L$ there exists a matrix $\boldsymbol{\Sigma}^R$ such that anisotropic-Gaussian scale-space representations of I_L and I_R are equal. Structure tensors $\mathbf{J}_L$ and $\mathbf{J}_R$ are related according to Eq. C.6

$$\mathbf{J}_R(\boldsymbol{q};\boldsymbol{\Sigma}_s^R,\boldsymbol{\Sigma}_w^R) = \mathbf{B}\mathbf{J}_L(\boldsymbol{p};\boldsymbol{\Sigma}_s^L,\boldsymbol{\Sigma}_w^L)\mathbf{B}^\mathsf{T} \tag{C.7}$$

The deduction details of Eq. C.7 can be referred to in Lindeberg's work[135, 138].

However, it is difficult to construct anisotropic-Gaussian scale space by convolving anisotropic Gaussian function with an image using Eq. 3.8. An alternative method is to first remove image's affinities by transforming an original image into a normalized image. Linear Gaussian function is then executed on the normalized image to achieve the same purpose. This property is very useful and has been used in many applications, such as corner detection[11, 155, 156] and shape from shading[138]. Let me mathematically derive this transformation process. Assume

$$\boldsymbol{\Sigma}_s^L = \sigma_s^2\mathbf{J}_L^{-1} \qquad \boldsymbol{\Sigma}_w^L = \sigma_w^2\mathbf{J}_L^{-1} \tag{C.8}$$

where the scalars σ_s^2 and σ_w^2 are differentiation and integration constants, respectively.

$$\begin{aligned}\boldsymbol{\Sigma}_s^R &= \mathbf{B}\boldsymbol{\Sigma}_s^L\mathbf{B}^\mathsf{T} = \sigma_s^2(\mathbf{B}\mathbf{J}_L^{-1}\mathbf{B}^\mathsf{T}) = \sigma_s^2(\mathbf{B}^{-\mathsf{T}}\mathbf{J}_L\mathbf{B}^{-1})^{-1} = \sigma_s^2\mathbf{J}_R^{-1}\\ \boldsymbol{\Sigma}_w^R &= \mathbf{B}\boldsymbol{\Sigma}_w^L\mathbf{B}^\mathsf{T} = \sigma_w^2(\mathbf{B}\mathbf{J}_L^{-1}\mathbf{B}^\mathsf{T}) = \sigma_w^2(\mathbf{B}^{-\mathsf{T}}\mathbf{J}_L\mathbf{B}^{-1})^{-1} = \sigma_w^2\mathbf{J}_R^{-1}\end{aligned} \tag{C.9}$$

We can obtain,

$$\mathbf{J}_R^{-1} = \mathbf{B}\mathbf{J}_L^{-1}\mathbf{B}^\mathsf{T} \Longrightarrow \mathbf{B} = \mathbf{J}_R^{-1/2}\mathbf{M}\mathbf{J}_L^{1/2} \tag{C.10}$$

where $\mathbf{M}$ is an arbitrary rotation matrix. We can derive

$$\begin{aligned} \boldsymbol{q} &= \mathbf{B}\boldsymbol{p} = \mathbf{J}_R^{-1/2}\mathbf{M}\mathbf{J}_L^{1/2}\boldsymbol{p} \\ \Rightarrow \mathbf{J}_R^{1/2}\boldsymbol{q} &= \mathbf{M}\mathbf{J}_L^{1/2}\boldsymbol{p} \end{aligned} \tag{C.11}$$

APPENDIX D: EGOMOTION ESTIMATION SENSITIVITY

In this appendix I analyze the stability of egomotion estimation approach that simultaneously computes camera translation and rotation velocities. Bruss and Horn's method was considered to be one of the superior methods[209] for egomotion estimation if optical flow is accurately estimated; it is a typically representative egomotion estimation method that minimizes estimated optical flow and visual motion flow to simultaneously estimate camera translation and rotation parameters, as expressed in Eq. 3.27. Its computation is independent on Focus of Expansion(FOE). I reimplemented this algorithm in C++ with the help of Matlab codes provided by Heeger[209].[1] I compared my FOE based approach to this method. Fig. D.1 illustrates the application of both methods on a 750 frame virtual colonoscopy (CT) image sequence. Because the actual camera motion parameters are known in this VC sequence, the sensitivity of this non-FOE method can be quantitatively analyzed by comparing estimated and known camera motion parameters. It can be seen that the first 127 frames produce very little error. After 127 frames, the error starts increasing (magenta curve) to about 80mm at the end of the sequence, while the error using my method (blue curve) remains at around 10mm. The first 150 VC images are chosen to perform error analysis. Fig. D.2a shows the absolute translation error with respect to X, Y, and Z axes in the world coordinate. Camera translation error is significantly large at point A (frame 129), which means that the non-FOE based method is very sensitive to optical flow errors.

Let me mathematically analyze the difficulties of Bruss and Horn's method in estimating camera motion parameters. Basically, their method attempts to estimate camera motion parameters in a 6×6 linear system, $\mathbf{A}\overrightarrow{x} = \overrightarrow{b}$, where $\overrightarrow{x} =$

[1] `http://www.cns.nyu.edu/heegerlab/index.php?page=software&id=egomotionalgo`

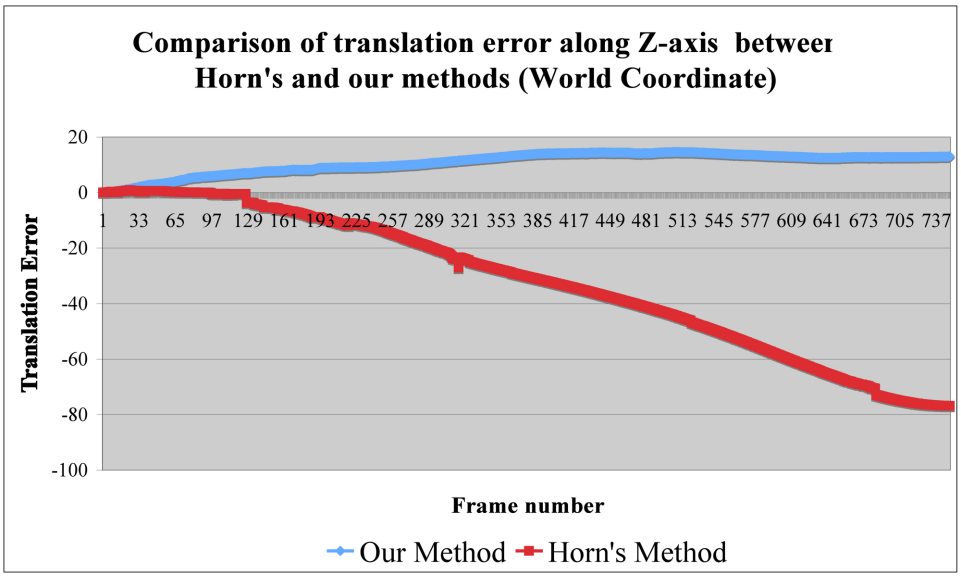


Figure D.1: *Comparison to Bruss and Horn's method on a 750 frame virtual colonoscopy sequence.* Absolute error along Z axis. My method results in an accumulated error of about 10mm, while Horn's method is nearly 80mm.

$(T_X, T_Y, T_Z, R_X, R_Y, R_Z)$. The estimation errors can be modeled as

$$\mathbf{A}\overrightarrow{x} = \overrightarrow{b} + \overrightarrow{\delta} \tag{D.1}$$

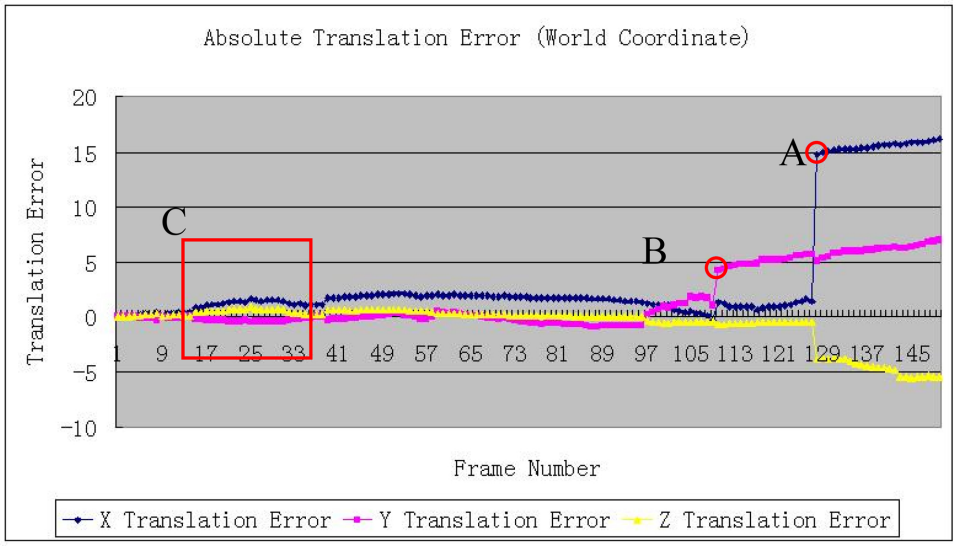
where $\overrightarrow{\delta}$ is a perturbation vector. In terms of matrix perturbation theory[201, 42], the bound of the relative error is

$$\frac{1}{n}\kappa(\mathbf{A})\frac{\|\overrightarrow{b}\|}{\|\mathbf{A}\|\|\overrightarrow{x}\|}\epsilon_b\mu \leq \frac{\|\overline{x} - \overrightarrow{x}\|}{\|\overrightarrow{x}\|} \leq \sqrt{n}\kappa(\mathbf{A})\frac{\|\overrightarrow{b}\|}{\|\mathbf{A}\|\|\overrightarrow{x}\|}\epsilon_b \tag{D.2}$$

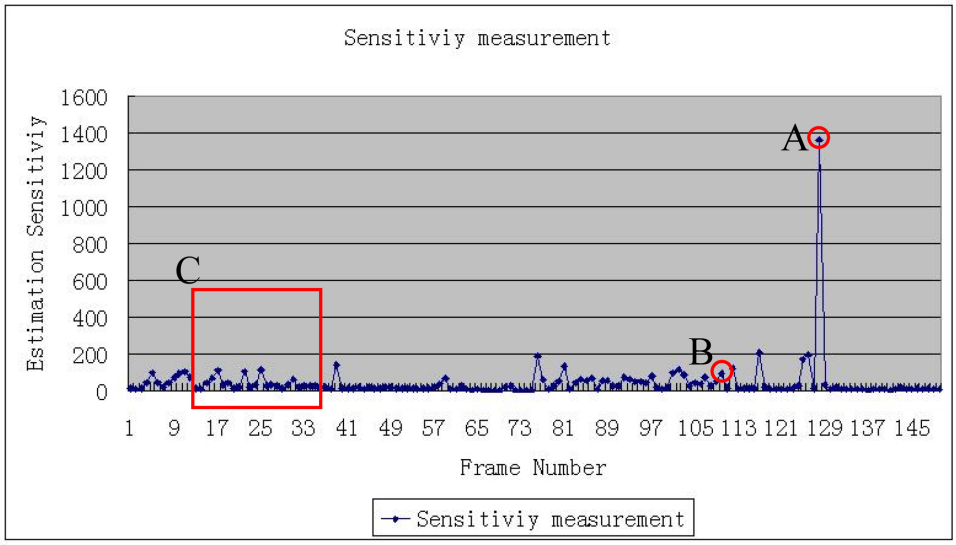
where $\kappa(\mathbf{A})$ is the condition number and $\epsilon_b = \|\overrightarrow{\delta}\|/\|\overrightarrow{b}\|$. $\overline{x}$ is the estimated value of $\overrightarrow{x}$. Assume $\mathbf{A}^{-1} = (\overrightarrow{r}_1 \ldots \overrightarrow{r}_n)$ and ψ_i is the angle between $\overrightarrow{r}_i$ and $\overrightarrow{\delta}$, then $\mu = \frac{\max_i\{\|\overrightarrow{r}_i\||cos\psi_i|\}}{\max_k\|\overrightarrow{r}_k\|}$.

Let $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ be the eigenvalues of $\mathbf{A}$, the lower bound of Eq. D.2 can be converted into

$$\frac{1}{n}\kappa(\mathbf{A})\frac{\|\overrightarrow{b}\|}{\|\mathbf{A}\|\|\overrightarrow{x}\|}\epsilon_b\mu = \frac{1}{n}\frac{|\lambda_n|}{|\lambda_1|}\frac{\|\overrightarrow{b}\|}{|\lambda_n|\|\overrightarrow{x}\|}\epsilon_b\mu = \frac{1}{n}\frac{\mu}{|\lambda_1|}\frac{\|\overrightarrow{b}\|}{\|\overrightarrow{x}\|}\epsilon_b$$



(a)



(b)

Figure D.2: *Relationship between absolute translation errors and the sensitivity measurement* $\zeta = \frac{\mu}{|\lambda_1|}$ *of the estimation system.* (a) The absolute estimated translation errors of 150 frames of a CT colonoscopy sequence. (b) Corresponding sensitivity measurement .

as $\kappa(\mathbf{A}) = \frac{|\lambda_n|}{|\lambda_1|}$ and $\|\mathbf{A}\| = |\lambda_n|$[201]. $\frac{\|\vec{b}\|}{\|\vec{x}\|}$ can be treated as a constant since $\vec{x}$ and $\vec{b}$ are the actual input and output signals, and do not affect the estimation process. ϵ_b relies on the measured signal and the output signal. Therefore, $\zeta = \frac{\mu}{|\lambda_1|}$ is solely related to the linear system. If it is stable, the estimated error will be small even if

the perturbation ratio ϵ_b is high. Fig. D.2 shows the relationship between the absolute translation error of the selected VC image sequence and ζ. Large translation errors are seen in X and Z on the frames close to the point marked A because $\lambda_1 \approx 3 \times 10^{-4}$. λ_1 increases the lower bound, although $\epsilon_b = 0.31$ is small. At the point B, a small error along Y-axis is due to the perturbation error, $\epsilon_b = 0.85$ and $\zeta = 89$. Although it might be possible to model the optical flow estimation error, it is considerably harder to model ζ. ζ is challenging to model because it is dependent on the distribution of the feature points as well as the relationship between the perturbation vector and the estimation matrix. In addition, the perturbation effect is difficult to predict, as seen in point C, where the X translation error initially increases, then decreases near B.

APPENDIX E: LMS BASED EGOMOTION ESTIMATION

Here, I describe the least median of squares(LMS) estimator to estimate camera translation and rotation parameters. Let me first introduce some basic concepts of LMS estimation, as described in Rousseeuw[185]. The classical linear model is represented as

$$y_i = x_{i1}\theta_1 + \cdots + x_{ip}\theta_p + e_i \text{ for } i = 1, \ldots, m, \tag{E.1}$$

where m is the sample size. The variables $x_{i1}, \ldots, x_{ip}$ are called the explanatory variables, whereas the variable y_i is called the response variable. e_i is the error term when a normal distribution is assumed. The aim of linear regression is to estimate $\theta = [\theta_1, \cdots, \theta_p]^T$. Applying a regression technique like the least sum of squares to yield $\hat{\theta} = [\hat{\theta}_1, \cdots, \hat{\theta}_p]^T$, the following formula can be obtained

$$\hat{y}_i = x_{i1}\hat{\theta}_1 + \cdots + x_{ip}\hat{\theta}_p, \tag{E.2}$$

The residual r_i of the ith case is the difference between what is actually observed and what is estimated:

$$r_i = y_i - \hat{y}_i \tag{E.3}$$

In terms of Eq. E.3, the LMS is defined as

$$\min_{\hat{\theta}} \text{med}_i r_i^2 \tag{E.4}$$

Next, I elaborate the application of the LMS estimator to compute camera rotation parameters. It consists of 6 steps:

1. Randomly choose three linear equations from Eq. 4.16 to compute rotation parameters, $\overrightarrow{R}$.

2. Substitute $\overrightarrow{R}$ for Eq. 4.16 and determine the median of the squared residuals,

M_k, where k is the current iteration number.

$$M_k = \text{med}_{i\in[1,n]}(r_i^R)^2 \tag{E.5}$$

n is the number of sparse optical flow vectors, and residual r_i^R is the difference between what is actually observed and what is estimated in ith equation of Eq. 4.16.

3. Iterate previous two steps K times until all possible input data distributions are investigated and choose the minimum $M_k, k \in (1, \mathsf{K})$. Report the corresponding rotation, $\overrightarrow{R}$.

4. Compute the scale of the minimum residual[185] as

$$s = 1.4826(1 + \frac{5}{n-3})\sqrt{\min_{\overrightarrow{R}} \text{med}_i (r_i^R)^2} \tag{E.6}$$

where $1/\Phi^{-1}(0.75) \approx 1.4826$ is an asymptotic correction factor for the case of normal errors. Φ denotes the cumulative distribution function of the standard normal distribution. More details can be found in Rousseeuw[184, 185].

5. The residuals are next normalized, $(r_i^R)/s$, and used to weight the ith observation,

$$w_i = \begin{cases} 1 & \text{if } |r_i^R| \leq 2.5|s| \\ 0 & \text{otherwise} \end{cases} \tag{E.7}$$

6. The final rotation parameters $\overrightarrow{R}$ can be recomputed according to reweighed least squares (RLS) regression.

$$\min_{\overrightarrow{R}} \sum_{i=1}^{n} w_i (r_i^R)^2 \tag{E.8}$$

LMS estimator is also used in the camera translation computation. Camera translation estimation is similar to camera rotation computation, except that the squared residual $(r_i^T)^2$ is not from a single equation but from the sum of the residuals resulting from three equations in each group in Eq. 4.18. Assuming $\hat{T} = (\hat{T}_X, \hat{T}_Y, \hat{T}_Z)$ be the estimated translation velocities at a particular iteration, then $(r_i^T)^2$ is given by

$$\begin{aligned}(r_i^T)^2 = (u_{xi} - u_{xi}^R - (-\frac{1}{Z_i}\hat{T}_X + \frac{x_i}{Z_i}\hat{T}_Z))^2 + (u_{yi} - u_{yi}^R - (-\frac{1}{Z_i}\hat{T}_Y + \frac{y_i}{Z_i}\hat{T}_Z))^2 \\ + (x_i(u_{xi} - u_{xi}^R + y_i(u_{yi} - u_{yi}^R) - (-\frac{x_i}{Z_i}\hat{T}_X - \frac{y_i}{Z_i}\hat{T}_Y + \frac{x_i^2 + y_i^2}{Z_i}\hat{T}_Z))^2\end{aligned} \tag{E.9}$$

APPENDIX F: CAMERA TRANSFORMATION MATRIX

This appendix derives the mathematical relationship between camera translation velocity in the camera coordinate and world coordinate systems. Let $\overrightarrow{T}$ be the velocity in camera coordinate and $\overrightarrow{T}^W$ in world coordinate. It consists of sequential rotations around X, Y, and Z axes. Assuming the camera orientation at t is $\Theta = (\Theta_X, \Theta_Y, \Theta_Z)$,

$$\mathbf{M}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C(\Theta_X) & -S(\Theta_X) \\ 0 & -S(\Theta_X) & C(\Theta_X) \end{bmatrix} \begin{bmatrix} C(\Theta_Y) & 0 & S(\Theta_Y) \\ 0 & 1 & 0 \\ -S(\Theta_Y) & 0 & C(\Theta_Y) \end{bmatrix} \begin{bmatrix} C(\Theta_Z) & -S(\Theta_Z) & 0 \\ S(\Theta_Z) & C(\Theta_Z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{F.1}$$

where $C = \cos$ and, $S = \sin$.

The transformation between camera rotation velocities $\overrightarrow{R}$ and $\overrightarrow{R}^W$ can be performed by first align Z axis of both camera and world coordinates followed by rotating the XY plane.

$$\begin{aligned} \overrightarrow{R}^W &= \begin{bmatrix} C(\Theta_Z) & S(\Theta_Z) & 0 \\ -S(\Theta_Z) & C(\Theta_Z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C(\Theta_Y) & 0 & -S(\Theta_Y) \\ 0 & 1 & 0 \\ S(\Theta_Y) & 0 & C(\Theta_Y) \end{bmatrix} \begin{bmatrix} R_X \\ 0 \\ 0 \end{bmatrix} \\ &+ \begin{bmatrix} C(\Theta_Z) & S(\Theta_Z) & 0 \\ -S(\Theta_Z) & C(\Theta_Z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ R_Y \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ R_Z \end{bmatrix} \\ &= \begin{bmatrix} C(\Theta_Y)C(\Theta_Z) & S(\Theta_Z) & 0 \\ -C(\Theta_Y)S(\Theta_Z) & C(\Theta_Z) & 0 \\ S(\Theta_Y) & 0 & 1 \end{bmatrix} \begin{bmatrix} R_X \\ R_Y \\ R_Z \end{bmatrix} \end{aligned} \tag{F.2}$$

Thus,

$$\mathbf{M}^R = \begin{bmatrix} C(\Theta_Y)C(\Theta_Z) & S(\Theta_Z) & 0 \\ -C(\Theta_Y)S(\Theta_Z) & C(\Theta_Z) & 0 \\ S(\Theta_Y) & 0 & 1 \end{bmatrix} \tag{F.3}$$

APPENDIX G: EGOMOTION ESTIMATION CONDITION

Here, I derive the equation relating egomotion and optical flow, and analyze why egomotion estimation fails in the case of significant camera motion.

Assume an instantaneous point $\boldsymbol{P} = (X, Y, Z)$ in the camera coordinate. Because of camera motion, this point moves into $\boldsymbol{P}_1 = (X_1, Y_1, Z_1)$. So the 3D rigid motion of $\boldsymbol{P}$ can be represented as

$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - \overrightarrow{T} = \mathbf{M}_Z \mathbf{M}_Y \mathbf{M}_X \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - \overrightarrow{T} \tag{G.1}$$

Where $\mathbf{M}_X, \mathbf{M}_Y$ and $\mathbf{M_Z}$ are rotation matrices around $X-, Y-$ and $Z-$ axes. The rotation matrix $\mathbf{M}$ is defined as

$$\begin{bmatrix} C(R_Y)C(R_Z) & S(R_X)S(R_Y)C(R_Z) + C(R_X)S(R_Z) & -C(R_X)S(R_Y)C(R_Z) + S(R_X)S(R_Z) \\ -C(R_Y)S(R_Z) & -S(R_X)S(R_Y)S(R_Z) + C(R_X)C(R_Z) & C(R_X)S(R_Y)S(R_Z) + S(R_X)C(R_Z) \\ S(R_Y) & -S(R_X)C(R_Y) & C(R_X)C(R_Y) \end{bmatrix} \tag{G.2}$$

Here, $C(\theta) = \cos(\theta)$ and $S(\theta) = \sin(\theta)$.

If θ is small (usually less than 10^0), then $\cos(\theta) \approx 1$ and $\sin(\theta) \approx \theta$. Simultaneously, I disregarded the higher order terms in Eq. G.1 and derived the following equation.

$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} \approx \begin{bmatrix} 1 & R_Z & -R_Y \\ -R_Z & 1 & R_X \\ R_Y & -R_X & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$
$$= \left(\begin{bmatrix} 0 & R_Z & -R_Y \\ -R_Z & 0 & R_X \\ R_Y & -R_X & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}$$

Then the velocity $\overrightarrow{V}$ of the object point $\boldsymbol{P}$ with respect to the X, Y, Z coordinates is

$$\overrightarrow{V} = \begin{bmatrix} X_1 - X \\ Y_1 - Y \\ Z_1 - Z \end{bmatrix} = \begin{bmatrix} 0 & R_Z & -R_Y \\ -R_Z & 0 & R_X \\ R_Y & -R_X & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix} = -\overrightarrow{T} - \overrightarrow{R} \times \boldsymbol{P} \tag{G.3}$$

Rewrite Eq. G.3 in the component form:

$$\begin{aligned} X' &= -T_X - R_Y Z + R_Z Y \\ Y' &= -T_Y - R_Z X + R_X Z \\ Z' &= -T_Z - R_X Y + R_Y X \end{aligned} \tag{G.4}$$

Let $\boldsymbol{p} = (x, y)$ denote the coordinate of a projection point of $\boldsymbol{P}$ in the image plane. Their spatial relationship is

$$x = f\frac{X}{Z}, y = f\frac{Y}{Z} \tag{G.5}$$

where f is the focal length. Differentiating Eq. G.5 with respect to time and using

Eq. G.4, a visual motion vector, denoted by $\overrightarrow{v} = (v_x, v_y)$, at $\boldsymbol{p}$ is given by

$$\begin{aligned} v_x = x' &= f(\frac{X'}{Z} - \frac{XZ'}{Z^2}) = \frac{-T_X f + T_Z x}{Z} + R_X \frac{xy}{f} - R_Y(f + \frac{x^2}{f}) + R_Z y \\ v_y = y' &= f(\frac{Y'}{Z} - \frac{YZ'}{Z^2}) = \frac{-T_Y f + T_Z y}{Z} + R_X(f + \frac{y^2}{f}) - R_Y \frac{xy}{f} - R_Z x \end{aligned} \tag{G.6}$$

This derivation clearly illustrates that the geometrical representation of visual motion flow has been greatly simplified. The accuracy of this equation is governed by the amount of translation and rotation velocities. The smaller the camera motion, the less the simulation error. Because camera motion is usually small in the case of consecutive frames, egomotion estimation generates accurate results based on Eq. G.6. Unfortunately, significant visual motion selected from region flow generates substantial camera motion, and Eq. G.6 is being violated. It causes significant estimation errors.

APPENDIX H: SEQUENTIAL LINEARIZATION

In this appendix, I will discuss sequential linearization used in incremental egomotion estimation as well as temporal volume flow computation.

H.1 Sequential Linearization in Incremental Egomotion Estimation

After removing the SIFT constraint found in Eq. 5.11, let me rewrite the energy function.

$$\begin{aligned} E(u_x, u_y) = \iint & \underbrace{\Psi((I(x+u_x, y+u_y, t+1) - I(x,y,t))^2)}_{\text{Intensity constraint}} \\ & + \gamma \underbrace{\Psi((\nabla I(x+u_x, y+u_y, t+1) - \nabla I(x,y,t))^2)}_{\text{Gradient constraint}} \\ & + \underbrace{\alpha\Psi((\nabla u_x)^2 + (\nabla u_y)^2) + \beta\Psi(|u_x - g_x|^2 + |u_y - g_y|^2)}_{\text{Smoothness constraint}} dxdy \end{aligned} \tag{H.1}$$

The Euler-Lagrane equations for Eq. H.1, with respect to $\overrightarrow{u}$, read

$$\begin{aligned} & \Psi'((\partial_t I)^2)\partial_t I \partial_x I + \gamma\Psi'((\partial_{xt} I)^2 + (\partial_{yt} I)^2)(\partial_{xx} I \partial_{xt} I + \partial_{xy} I \partial_{yt} I) \\ & + \beta\Psi'(|u_x - g_x|^2 + |u_y - g_y|^2)(u_x - g_x) - \alpha \text{div}(\Psi'(|\nabla u_x|^2 + |\nabla u_y|^2)\nabla u_x) = 0 \\ & \Psi'((\partial_t I)^2)\partial_t I \partial_y I + \gamma\Psi'((\partial_{xt} I)^2 + (\partial_{yt} I)^2)(\partial_{xy} I \partial_{xt} I + \partial_{yy} I \partial_{yt} I) \\ & + \beta\Psi'(|u_x - g_x|^2 + |u_y - g_y|^2)(u_y - g_y) - \alpha \text{div}(\Psi'(|\nabla u_x|^2 + |\nabla u_y|^2)\nabla u_y) = 0 \end{aligned} \tag{H.2}$$

where

$$\begin{aligned} \Psi'(x^2) &= \frac{1}{2\sqrt{x^2 + \epsilon^2}} \qquad \partial_x I = \frac{\partial I_2(x+u_x, y+u_y, t+1)}{\partial x} \\ \partial_{xy} I &= \frac{\partial^2 I_2(x+u_x, y+u_y, t+1)}{\partial x \partial y} \qquad \partial_y I = \frac{\partial I_2(x+u_x, y+u_y)}{\partial y} \\ \partial_{xx} I &= \frac{\partial^2 I_2(x+u_x, y+u_y, t+1)}{\partial x^2} \qquad \partial_{yy} I = \frac{\partial^2 I_2(x+u_x, y+u_y, t+1)}{\partial y^2} \end{aligned}$$

$$\partial_t I = I_2(x+u_x, y+u_y, t+1) - I_1(x,y,t)$$
$$\partial_{xt} I = \frac{\partial I_2(x+u_x, y+u_y, t+1)}{\partial x} - \frac{\partial I_1(x,y,t)}{\partial x}$$
$$\partial_{yt} I = \frac{\partial I_2(x+u_x, y+u_y, t+1)}{\partial y} - \frac{\partial I_1(x,y,t)}{\partial y}$$

Sequential linearization is used to remove non-linearity from the Euler-Lagrange equations. It is represented as two nested fixed point iterations to gradually remove non-linearity in Eq. H.2. Let l denote the outer iteration index and k the current image scale level. Eq. H.2 is rewritten as

$$\begin{aligned}
&\Psi'((\partial_t I^{k,l+1})^2)\partial_x I^{k,l}\partial_t I^{k,l+1} + \gamma\Psi'((\partial_{xt} I^{k,l+1})^2 + (\partial_{yt} I^{k,l+1})^2)(\partial_{xx} I^{k,l}\partial_{xt} I^{k,l+1} + \partial_{xy} I^{k,l}\partial_{yt} I^{k,l+1}) \\
&+ \beta\Psi'(|u_x^{k,l+1} - g_x|^2 + |u_y^{k,l+1} - g_y|^2)(u_x^{k,l+1} - g_x) - \alpha \mathrm{div}(\Psi'(|\nabla u_x^{k,l+1}|^2 + |\nabla u_y^{k,l+1}|^2)\nabla u_x^{k,l+1}) = 0 \\
&\Psi'((\partial_t I^{k,l+1})^2)\partial_y I^{k,l}\partial_t I^{k,l+1} + \gamma\Psi'((\partial_{xt} I^{k,l+1})^2 + (\partial_{yt} I^{k,l+1})^2)(\partial_{xy} I^{k,l}\partial_{xt} I^{k,l+1} + \partial_{yy} I^{k,l}\partial_{yt} I^{k,l+1}) \\
&+ \beta\Psi'(|u_x^{k,l+1} - g_x|^2 + |u_y^{k,l+1} - g_y|^2)(u_y^{k,l+1} - g_y) - \alpha \mathrm{div}(\Psi'(|\nabla u_x^{k,l+1}|^2 + |\nabla u_y^{k,l+1}|^2)\nabla u_y^{k,l+1}) = 0
\end{aligned} \tag{H.3}$$

At iteration $l+1$, we can approximate, through Talyor expansion, the following:

$$\begin{aligned}
\partial_t I^{k,l+1} &\approx \partial_t I^{k,l} + \partial_x I^{k,l} du_x^{k,l} + \partial_y I^{k,l} du_y^{k,l} \\
\partial_{xt} I^{k,l+1} &\approx \partial_{xt} I^{k,l} + \partial_{xx} I^{k,l} du_x^{k,l} + \partial_{xy} I^{k,l} du_y^{k,l} \\
\partial_{yt} I^{k,l+1} &\approx \partial_{yt} I^{k,l} + \partial_{xy} I^{k,l} du_x^{k,l} + \partial_{yy} I^{k,l} du_y^{k,l}
\end{aligned} \tag{H.4}$$

In Eq. H.4, $d\overrightarrow{u}^{k,l} = (du_x^{k,l}, du_y^{k,l})$ is an incremental optical flow vector, and $u_x^{k,l+1} = u_x^{k,l} + du_x^{k,l}$ and $u_y^{k,l+1} = u_y^{k,l} + du_y^{k,l}$. Let me define four terms to abbreviate the descriptions of the Euler-Lagrange equations

$$\begin{aligned}
(\Psi'_I)^{k,l} &= \Psi'((\partial_t I^{k,l} + \partial_x I^{k,l} du_x^{k,l} + \partial_y I^{k,l} du_y^{k,l})^2) \\
(\Psi'_G)^{k,l} &= \Psi'((\partial_{xt} I^{k,l} + \partial_{xx} I^{k,l} du_x^{k,l} + \partial_{xy} I^{k,l} du_y^{k,l})^2 \\
&\quad + (\partial_{yt} I^{k,l} + \partial_{xy} I^{k,l} du_x^{k,l} + \partial_{yy} I^{k,l} du_y^{k,l})^2)
\end{aligned}$$

$$
\begin{aligned}
(\Psi'_M)^{k,l} &= \Psi'((u_x^{k,l} - g_x)^2 + (u_y^{k,l} - g_y)^2) \\
(\Psi'_S)^{k,l} &= \Psi'(|\nabla u_x^{k,l}|^2 + |\nabla u_y^{k,l}|^2)
\end{aligned} \tag{H.5}
$$

Therefore, Eq. H.3 is derived as

$$
\begin{aligned}
&(\Psi'_I)^{k,l}\partial_x I^{k,l}(\partial_t I^{k,l} + \partial_x I^{k,l} du_x^{k,l} + \partial_y I^{k,l} du_y^{k,l}) + \beta(\Psi'_M)^{k,l}(u_x^{k,l} + du_x^{k,l} - g_x) \\
&\quad + \gamma(\Psi'_G)^{k,l}\partial_{xx} I^{k,l}(\partial_{xt} I^{k,l} + \partial_{xx} I^{k,l} du_x^{k,l} + \partial_{xy} I^{k,l} du_y^{k,l}) \\
&\quad + \gamma(\Psi'_G)^{k,l}\partial_{xy} I^{k,l}(\partial_{yt} I^{k,l} + \partial_{xy} I^{k,l} du_x^{k,l} + \partial_{yy} I^{k,l} du_y^{k,l}) \\
&\quad - \alpha \mathrm{div}((\Psi'_S)^{k,l}\nabla(u_x^{k,l} + du_x^{k,l})) = 0 \\
&(\Psi'_I)^{k,l}\partial_y I^{k,l}(\partial_t I^{k,l} + \partial_x I^{k,l} du_x^{k,l} + \partial_y I^{k,l} du_y^{k,l}) + \beta(\Psi'_M)^{k,l}(u_y^{k,l} + du_y^{k,l} - g_y) \\
&\quad + \gamma(\Psi'_G)^{k,l}\partial_{xy} I^{k,l}(\partial_{xt} I^{k,l} + \partial_{xx} I^{k,l} du_x^{k,l} + \partial_{xy} I^{k,l} du_y^{k,l}) \\
&\quad + \gamma(\Psi'_G)^{k,l}\partial_{yy} I^{k,l}(\partial_{yt} I^{k,l} + \partial_{xy} I^{k,l} du_x^{k,l} + \partial_{yy} I^{k,l} du_y^{k,l}) \\
&\quad - \alpha \mathrm{div}((\Psi'_S)^{k,l}\nabla(u_y^{k,l} + du_y^{k,l})) = 0
\end{aligned} \tag{H.6}
$$

Another inner iteration is introduced to remove non-linearity in $(\Psi'_I)^{k,l}$, $(\Psi'_G)^{k,l}$, $(\Psi'_M)^{k,l}$ and $(\Psi'_S)^{k,l}$. Assuming m be the iteration index and incremental optical flow vector $d\overrightarrow{u}^{k,l}$ rewritten as $d\overrightarrow{u}^{k,l,m} = (du_x^{k,l,m}, du_y^{k,l,m})$, Eq. H.6 is linearized as

$$
\begin{aligned}
&(\Psi'_I)^{k,l,m}\partial_x I^{k,l,m}(\partial_t I^{k,l,m} + \partial_x I^{k,l,m} du_x^{k,l,m+1} + \partial_y I^{k,l,m} du_y^{k,l,m+1}) \\
&+ \gamma(\Psi'_G)^{k,l,m}\partial_{xx} I^{k,l,m}(\partial_{xt} I^{k,l,m} + \partial_{xx} I^{k,l,m} du_x^{k,l,m+1} + \partial_{xy} I^{k,l,m} du_y^{k,l,m+1}) \\
&+ \gamma(\Psi'_G)^{k,l,m}\partial_{xy} I^{k,l,m}(\partial_{yt} I^{k,l,m} + \partial_{xy} I^{k,l,m} du_x^{k,l,m+1} + \partial_{yy} I^{k,l,m} du_y^{k,l,m+1}) \\
&+ \beta(\Psi'_M)^{k,l,m}(u_x^{k,l,m} + du_x^{k,l,m+1} - g_x) - \alpha \mathrm{div}((\Psi'_S)^{k,l,m}\nabla(u_x^{k,l,m} + du_x^{k,l,m+1})) = 0 \\
&(\Psi'_I)^{k,l,m}\partial_y I^{k,l,m}(\partial_t I^{k,l,m} + \partial_x I^{k,l,m} du_x^{k,l,m+1} + \partial_y I^{k,l,m} du_y^{k,l,m+1}) \\
&+ \gamma(\Psi'_G)^{k,l,m}\partial_{xy} I^{k,l,m}(\partial_{xt} I^{k,l,m} + \partial_{xx} I^{k,l,m} du_x^{k,l,m+1} + \partial_{xy} I^{k,l,m} du_y^{k,l,m+1}) \\
&+ \gamma(\Psi'_G)^{k,l,m}\partial_{yy} I^{k,l,m}(\partial_{yt} I^{k,l,m} + \partial_{xy} I^{k,l,m} du_x^{k,l,m+1} + \partial_{yy} I^{k,l,m} du_y^{k,l,m+1}) \\
&+ \beta(\Psi'_M)^{k,l,m}(u_y^{k,l,m} + du_y^{k,l,m+1} - g_y) - \alpha \mathrm{div}((\Psi'_S)^{k,l,m}\nabla(u_y^{k,l,m} + du_y^{k,l,m+1})) = 0
\end{aligned} \tag{H.7}
$$

where $(du_x^{k,l,0}, du_y^{k,l,0}) = (0,0)$. Note that $(\Psi'_I)^{k,l,m}$, $(\Psi'_G)^{k,l,m}$, $(\Psi'_M)^{k,l,m}$ and $(\Psi'_S)^{k,l,m}$ are related to incremental optical flow $(du_x^{k,l,m}, du_y^{k,l,m})$ at previous iteration m, while Eq. H.7 is estimating the current incremental optical flow at iteration $m+1$. For this reason, Eq. H.7 is a linear equation with respect to $(du_x^{k,l,m+1}, du_y^{k,l,m+1})$.

However, the relationship between diffusion terms and incremental optical flow vectors is still implicitly linear. In order to eliminate this ambiguity, a four-pixel scheme[229, 26] is described to serve this purpose, which is shown in Fig. H.1.

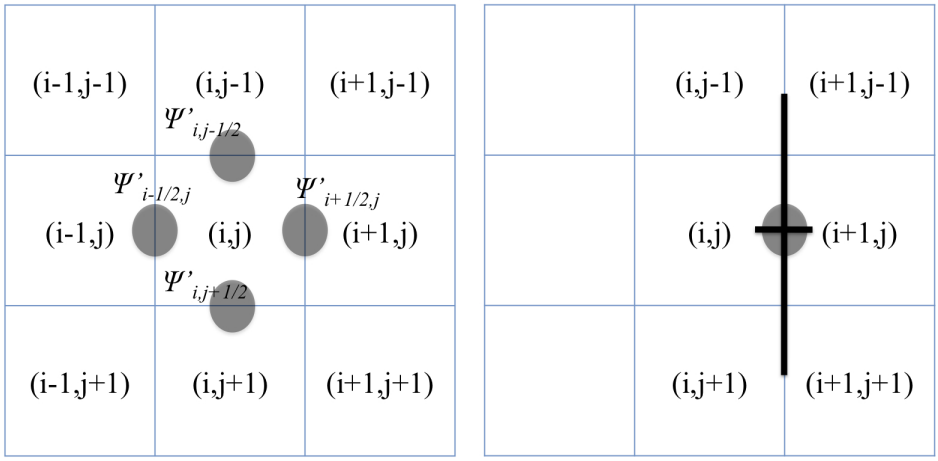


Figure H.1: *Numerical computation using 4-neighborhood.* This figure is reproduced from Brox[26].

Fig. H.1a indicates that diffusion only takes effect between the current pixel at (i,j) and its four-neighbor pixels at $(i,j-1)$, $(i-1,j)$, $(i+1,j)$, and $(i,j+1)$. They are represented as four dark plates in Fig. H.1a, and defined as $\Psi'_S(i,j-1/2)$, $\Psi'_S(i-1/2,j)$, $\Psi'_S(i+1/2,j)$, and $\Psi'_S(i,j+1/2)$. Fig. H.1b shows an example of $\Psi'_S(i+1/2,j)$. Ψ'_S defined in Eq. H.5 expresses that we need to define ∇u_x and ∇u_y in four locations illustrated as black plates in Fig. H.1a, such as $(i+1/2,j)$. Let u represent either u_x or u_y, and $\nabla u(i+1/2,j)$ is calculated as

$$|\nabla u(i+\frac{1}{2},j)| \approx \sqrt{(u(i+1,j)-u(i,j))^2 + \left(\frac{1}{2}\left(\frac{u(i+1,j+1)-u(i+1,j-1)}{2} + \frac{u(i,j+1)-u(i,j-1)}{2}\right)\right)^2} \tag{H.8}$$

Similarly, ∇u is also computed at $(i-1/2,j),(i,j-1/2)$ and $(i,j+1/2)$,

$$
\begin{aligned}
|\nabla u(i-\frac{1}{2},j)| &\approx \sqrt{(u(i,j)-u(i-1,j))^2+\left(\tfrac{1}{2}\left(\tfrac{u(i-1,j+1)-u(i-1,j-1)}{2}+\tfrac{u(i,j+1)-u(i,j-1)}{2}\right)\right)^2}\\
|\nabla u(i,j+\frac{1}{2})| &\approx \sqrt{(u(i,j+1)-u(i,j))^2+\left(\tfrac{1}{2}\left(\tfrac{u(i+1,j+1)-u(i-1,j+1)}{2}+\tfrac{u(i+1,j)-u(i-1,j)}{2}\right)\right)^2}\\
|\nabla u(i,j-\frac{1}{2})| &\approx \sqrt{(u(i,j)-u(i,j-1))^2+\left(\tfrac{1}{2}\left(\tfrac{u(i+1,j-1)-u(i-1,j-1)}{2}+\tfrac{u(i+1,j)-u(i-1,j)}{2}\right)\right)^2}
\end{aligned}
\tag{H.9}
$$

In order to clarify the diffusion relationship between the current point and its four neighbor points, Ψ'_S is alternatively rewritten as $(\Psi'_S)_{(i,j)\sim(i-1,j)}$, $(\Psi'_S)_{(i,j)\sim(i+1,j)}$, $(\Psi'_S)_{(i,j)\sim(i,j-1)}$, and $(\Psi'_S)_{(i,j)\sim(i,j+1)}$. They are generalized as $(\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}$, where $\boldsymbol{p}=(i,j)$ and $\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})$ are two image points, and $\mathcal{N}(\boldsymbol{p})$ is $\boldsymbol{p}$'s 4-neighborhood. $\mathrm{div}(\Psi'_S(|\nabla u_x|^2+|\nabla u_y|^2)\nabla u_x)$ is derived as

$$
\begin{aligned}
&\mathrm{div}(\Psi'_S(|\nabla u_x|^2+|\nabla u_y|^2)\nabla u_x)\\
&\quad=\Psi'_S(|\nabla u_x(i+\frac{1}{2},j)|^2+|\nabla u_y(i+\frac{1}{2},j)|^2)(u_x(i+1,j)-u_x(i,j))\\
&\quad-\Psi'_S(|\nabla u_x(i-\frac{1}{2},j)|^2+|\nabla u_y(i-\frac{1}{2},j)|^2)(u_x(i,j)-u_x(i-1,j))\\
&\quad+\Psi'_S(|\nabla u_x(i,j+\frac{1}{2})|^2+|\nabla u_y(i,j+\frac{1}{2})|^2)(u_x(i,j+1)-u_x(i,j))\\
&\quad-\Psi'_S(|\nabla u_x(i,j-\frac{1}{2})|^2+|\nabla u_y(i,j-\frac{1}{2})|^2)(u_x(i,j)-u_x(i,j-1))\\
&\quad=(\Psi'_S)_{(i,j)\sim(i-1,j)}u_x(i-1,j)+(\Psi'_S)_{(i,j)\sim(i,j-1)}u_x(i,j-1)\\
&\quad+(\Psi'_S)_{(i,j)\sim(i+1,j)}u_x(i+1,y)+(\Psi'_S)_{(i,j)\sim(i,j+1)}u_x(i,j+1)\\
&\quad-((\Psi'_S)_{(i,j)\sim(i-1,j)}+(\Psi'_S)_{(i,j)\sim(i,j-1)}+(\Psi'_S)_{(i,j)\sim(i+1,j)}+(\Psi'_S)_{(i,j)\sim(i,j+1)})u_x(i,j)\\
&\quad=\sum_{q\in\mathcal{N}(\boldsymbol{p})}(\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}u_x(\boldsymbol{q})-\sum_{q\in\mathcal{N}(\boldsymbol{p})}(\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}u_x(\boldsymbol{p})
\end{aligned}
\tag{H.10}
$$

The same deduction can be used to compute $\mathrm{div}(\Psi'_S(|\nabla u_x|^2+|\nabla u_y|^2)\nabla u_y)$.

$$
\mathrm{div}(\Psi'_S(|\nabla u_x|^2+|\nabla u_y|^2)\nabla u_y)=\sum_{q\in\mathcal{N}(\boldsymbol{p})}(\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}u_y(\boldsymbol{q})-\sum_{q\in\mathcal{N}(\boldsymbol{p})}(\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}u_y(\boldsymbol{p}) \tag{H.11}
$$

Substituting Eq. H.10 and Eq. H.11 for Eq. H.7, we obtain the following linear Euler-

Lagrange equation at $\boldsymbol{p}$.

$$
\begin{aligned}
&(\Psi'_I)^{k,l,m}_{\boldsymbol{p}}(\partial_x I^{k,l,m})_{\boldsymbol{p}}(\partial_t I^{k,l,m})_{\boldsymbol{p}} + \gamma(\Psi'_G)^{k,l,m}_{\boldsymbol{p}}(\partial_{xx} I^{k,l,m})_{\boldsymbol{p}}(\partial_{xt} I^{k,l,m})_{\boldsymbol{p}} \\
&+ \gamma(\Psi'_G)^{k,l,m}_{\boldsymbol{p}}(\partial_{xy} I^{k,l,m})_{\boldsymbol{p}}(\partial_{yt} I^{k,l,m})_{\boldsymbol{p}} + \beta(\Psi'_M)^{k,l,m}_{\boldsymbol{p}}((u_x^{k,l,m})_{\boldsymbol{p}} - (g_x)_{\boldsymbol{p}}) \\
&+ (\Psi'_I)^{k,l,m}_{\boldsymbol{p}}(\partial_x I^{k,l,m})_{\boldsymbol{p}}(\partial_y I^{k,l,m})_{\boldsymbol{p}}(du_y)^{k,l,m+1}_{\boldsymbol{p}} + \gamma(\Psi'_G)^{k,l,m}_{\boldsymbol{p}}(\partial_{xx} I^{k,l,m})_{\boldsymbol{p}}(\partial_{xy} I^{k,l,m})_{\boldsymbol{p}}(du_y)^{k,l,m+1}_{\boldsymbol{p}} \\
&+ \gamma(\Psi'_G)^{k,l,m}_{\boldsymbol{p}}(\partial_{xy} I^{k,l,m})_{\boldsymbol{p}}(\partial_{yy} I^{k,l,m})_{\boldsymbol{p}}(du_y)^{k,l,m+1}_{\boldsymbol{p}} + (\Psi'_I)^{k,l,m}_{\boldsymbol{p}}(\partial_x I^{k,l,m})^2_{\boldsymbol{p}}(du_x)^{k,l,m+1}_{\boldsymbol{p}} \\
&+ \beta(\Phi'_M)^{k,l,m}_{\boldsymbol{p}}(du_x)^{k,l,m+1}_{\boldsymbol{p}} + \gamma(\Psi'_G)^{k,l,m}_{\boldsymbol{p}}(\partial_{xx} I^{k,l,m})^2_{\boldsymbol{p}}(du_x)^{k,l,m+1}_{\boldsymbol{p}} \\
&+ \gamma(\Psi'_G)^{k,l,m}_{\boldsymbol{p}}(\partial_{xy} I^{k,l,m})^2_{\boldsymbol{p}}(du_x)^{k,l,m+1}_{\boldsymbol{p}} - \alpha \sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)^{k,l,m}_{\boldsymbol{p}\sim\boldsymbol{q}}((u_x)^{k,l,m}_{\boldsymbol{q}} + (du_x)^{k,l,m+1}_{\boldsymbol{q}}) \\
&+ \alpha \sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)^{k,l,m}_{\boldsymbol{p}\sim\boldsymbol{q}}((u_x)^{k,l,m}_{\boldsymbol{p}} + (du_x)^{k,l,m+1}_{\boldsymbol{p}}) = 0 \\
&(\Psi'_I)^{k,l,m}_{\boldsymbol{p}}(\partial_y I^{k,l,m})_{\boldsymbol{p}}(\partial_t I^{k,l,m})_{\boldsymbol{p}} + \gamma(\Psi'_G)^{k,l,m}_{\boldsymbol{p}}(\partial_{xy} I^{k,l,m})_{\boldsymbol{p}}(\partial_{xt} I^{k,l,m})_{\boldsymbol{p}} \\
&+ \gamma(\Psi'_G)^{k,l,m}_{\boldsymbol{p}}(\partial_{yy} I^{k,l,m})_{\boldsymbol{p}}(\partial_{yt} I^{k,l,m})_{\boldsymbol{p}} + \beta(\Psi'_M)^{k,l,m}_{\boldsymbol{p}}((u_y^{k,l,m})_{\boldsymbol{p}} - (g_y)_{\boldsymbol{p}}) \\
&+ (\Psi'_I)^{k,l,m}_{\boldsymbol{p}}(\partial_y I^{k,l,m})_{\boldsymbol{p}}(\partial_x I^{k,l,m})_{\boldsymbol{p}}(du_x)^{k,l,m+1}_{\boldsymbol{p}} + \gamma(\Psi'_G)^{k,l,m}_{\boldsymbol{p}}(\partial_{xy} I^{k,l,m})_{\boldsymbol{p}}(\partial_{xx} I^{k,l,m})_{\boldsymbol{p}}(du_x)^{k,l,m+1}_{\boldsymbol{p}} \\
&+ \gamma(\Psi'_G)^{k,l,m}_{\boldsymbol{p}}(\partial_{yy} I^{k,l,m})_{\boldsymbol{p}}(\partial_{xy} I^{k,l,m})_{\boldsymbol{p}}(du_x)^{k,l,m+1}_{\boldsymbol{p}} + (\Psi'_I)^{k,l,m}_{\boldsymbol{p}}(\partial_y I^{k,l,m})^2_{\boldsymbol{p}}(du_y)^{k,l,m+1}_{\boldsymbol{p}} \\
&+ \beta(\Phi'_M)^{k,l,m}_{\boldsymbol{p}}(du_y)^{k,l,m+1}_{\boldsymbol{p}} + \gamma(\Psi'_G)^{k,l,m}_{\boldsymbol{p}}(\partial_{xy} I^{k,l,m})^2_{\boldsymbol{p}}(du_y)^{k,l,m+1}_{\boldsymbol{p}} \\
&+ \gamma(\Psi'_G)^{k,l,m}_{\boldsymbol{p}}(\partial_{yy} I^{k,l,m})^2_{\boldsymbol{p}}(du_y)^{k,l,m+1}_{\boldsymbol{p}} - \alpha \sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)^{k,l,m}_{\boldsymbol{p}\sim\boldsymbol{q}}((u_y)^{k,l,m}_{\boldsymbol{q}} + (du_y)^{k,l,m+1}_{\boldsymbol{q}}) \\
&+ \alpha \sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)^{k,l,m}_{\boldsymbol{p}\sim\boldsymbol{q}}((u_y)^{k}_{\boldsymbol{p}} + (du_y)^{k,l,m+1}_{\boldsymbol{p}}) = 0
\end{aligned}
\tag{H.12}
$$

H.2 Sequential Linearization in Temporal Volume Flow Computation

The previous section illustrates sequential linearizaiton used by incremental ego-motion estimation in the image domain. In this section, I present sequential linearization in computing temporal volume flow(TVF) in the temporal volume domain. Let

me rewrite the PDE metric shown in Eq. 6.5 that estimates TVF.

$$
\begin{aligned}
E(\overrightarrow{w}) = \iint_{(x,y,t)\in\mathbb{R}^2\times\mathbb{R}_+} & (\Psi((V(x+w_x, y+w_y, t+w_t, \rho+1) - V(x,y,t,\rho))^2 \\
& + \gamma(\nabla_3 V(x+w_x, y+w_y, t+w_t, \rho+1) - \nabla_3 V(x,y,t,\rho))^2) \\
& + \alpha\Psi(|\nabla_3 w_x|^2 + |\nabla_3 w_y|^2 + |\nabla_3 w_t|^2))dxdydt
\end{aligned}
\tag{H.13}
$$

Minimizing Eq. H.13 can be mathematically solved through the Euler-Lagrange equations. With respect to x, y and t components, it reads

$$
\begin{aligned}
& \Psi'((\partial_\rho V)^2 + \gamma((\partial_{x\rho}V)^2 + (\partial_{y\rho}V)^2 + (\partial_{t\rho}V)^2))(\partial_x V\partial_\rho V + \gamma(\partial_{xx}V\partial_{x\rho}V + \partial_{xy}V\partial_{y\rho}V \\
& + \partial_{xt}V\partial_{t\rho}V)) - \alpha\mathrm{div}_3\left(\Psi'(|\nabla_3 w_x|^2 + |\nabla_3 w_y|^2 + |\nabla_3 w_t|^2)\nabla_3 w_x\right) = 0 \\
& \Psi'((\partial_\rho V)^2 + \gamma((\partial_{x\rho}V)^2 + (\partial_{y\rho}V)^2 + (\partial_{t\rho}V)^2))(\partial_y V\partial_\rho V + \gamma(\partial_{xy}V\partial_{x\rho}V + \partial_{yy}V\partial_{y\rho}V \\
& + \partial_{yt}V\partial_{t\rho}V)) - \alpha\mathrm{div}_3\left(\Psi'(|\nabla_3 w_x|^2 + |\nabla_3 w_y|^2 + |\nabla_3 w_t|^2)\nabla_3 w_y\right) = 0 \\
& \Psi'((\partial_\rho V)^2 + \gamma((\partial_{x\rho}V)^2 + (\partial_{y\rho}V)^2 + (\partial_{t\rho}V)^2))(\partial_t V\partial_\rho V + \gamma(\partial_{xt}V\partial_{x\rho}V + \partial_{yt}V\partial_{y\rho}V \\
& + \partial_{tt}V\partial_{t\rho}V)) - \alpha\mathrm{div}_3\left(\Psi'(|\nabla_3 w_x|^2 + |\nabla_3 w_y|^2 + |\nabla_3 w_t|^2)\nabla_3 w_t\right) = 0
\end{aligned}
\tag{H.14}
$$

where the derivatives related to $\partial_{*\rho}V$ are defined as temporal difference.

$$
\begin{aligned}
\partial_\rho V &= V(x+w_x, y+w_y, t+w_t, \rho+1) - V(x,y,t,\rho) \\
\partial_{x\rho}V &= \partial_x V(x+w_x, y+w_y, t+w_t, \rho+1) - \partial_x V(x,y,t,\rho) \\
\partial_{y\rho}V &= \partial_y V(x+w_x, y+w_y, t+w_t, \rho+1) - \partial_y V(x,y,t,\rho) \\
\partial_{t\rho}V &= \partial_t V(x+w_x, y+w_y, t+w_t, \rho+1) - \partial_t V(x,y,t,\rho)
\end{aligned}
\tag{H.15}
$$

Let me abbreviate data and smoothness terms in Eq. H.13 to simplify the descrip-

tion,

$$\begin{aligned}\Psi_D &= \Psi((V(x+w_x, y+w_y, t+w_t, \rho+1) - V(x,y,t,\rho))^2 \\ &\quad + \gamma(\nabla_3 V(x+w_x, y+w_y, t+w_t, \rho+1) - \nabla_3 V(x,y,t,\rho))^2) \\ \Psi_S &= \Psi(|\nabla_3 w_x|^2 + |\nabla_3 w_y|^2 + |\nabla_3 w_t|^2)\end{aligned} \tag{H.16}$$

Non-linearity happens in Ψ'_D and Ψ'_S. Sequential linearization is an efficient strategy to remove non-linearity in Eq. H.14, so as to easily minimize Eq. H.13. Sequential linearization involves two nested fixed point iterations. Let l denote the outer iteration index at temporal volume pyramid level k, and define $\overrightarrow{w}^{k,l} = (w_x^{k,l}, w_y^{k,l}, w_t^{k,l}, 1)$. This iteration is performed to remove non-linearity from intensity and gradient constancy constraints. $\overrightarrow{w}^{k,l}$ is computed through

$$\begin{aligned}&\Psi'((\partial_\rho V^{k,l+1})^2 + \gamma((\partial_{x\rho} V^{k,l+1})^2 + (\partial_{y\rho} V^{k,l+1})^2 + (\partial_{t\rho} V^{k,l+1})^2)) \\ &(\partial_x V^{k,l} \partial_\rho V^{k,l+1} + \gamma(\partial_{xx} V^{k,l} \partial_{x\rho} V^{k,l+1} + \partial_{xy} V^{k,l} \partial_{y\rho} V^{k,l+1} + \partial_{xt} V^{k,l} \partial_{t\rho} V^{k,l+1})) \\ &- \alpha \mathrm{div}_3 \left(\Psi'(|\nabla_3 w_x^{k,l+1}|^2 + |\nabla_3 w_y^{k,l+1}|^2 + |\nabla_3 w_t^{k,l+1}|^2) \nabla_3 w_x^{k,l+1}\right) = 0 \\ &\Psi'((\partial_\rho V^{k,l+1})^2 + \gamma((\partial_{x\rho} V^{k,l+1})^2 + (\partial_{y\rho} V^{k,l+1})^2 + (\partial_{t\rho} V^{k,l+1})^2)) \\ &(\partial_y V^{k,l} \partial_\rho V^{k,l+1} + \gamma(\partial_{xy} V^{k,l} \partial_{x\rho} V^{k,l+1} + \partial_{yy} V^{k,l} \partial_{y\rho} V^{k,l+1} + \partial_{yt} V^{k,l} \partial_{t\rho} V^{k,l+1})) \\ &- \alpha \mathrm{div}_3 \left(\Psi'(|\nabla_3 w_x^{k,l+1}|^2 + |\nabla_3 w_y^{k,l+1}|^2 + |\nabla_3 w_t^{k,l+1}|^2) \nabla_3 w_y^{k,l+1}\right) = 0 \\ &\Psi'((\partial_\rho V^{k,l+1})^2 + \gamma((\partial_{x\rho} V^{k,l+1})^2 + (\partial_{y\rho} V^{k,l+1})^2 + (\partial_{t\rho} V^{k,l+1})^2)) \\ &(\partial_t V^{k,l} \partial_\rho V^{k,l+1} + \gamma(\partial_{xt} V^{k,l} \partial_{x\rho} V^{k,l+1} + \partial_{yt} V^{k,l} \partial_{y\rho} V^{k,l+1} + \partial_{tt} V^{k,l} \partial_{t\rho} V^{k,l+1})) \\ &- \alpha \mathrm{div}_3 \left(\Psi'(|\nabla_3 w_x^{k,l+1}|^2 + |\nabla_3 w_y^{k,l+1}|^2 + |\nabla_3 w_t^{k,l+1}|^2) \nabla_3 w_t^{k,l+1}\right) = 0\end{aligned} \tag{H.17}$$

Notice that Eq. H.17 is fully implicit non-linearity in the smoothness term and semi-implicit non-linearity in the data term. At iteration $l+1$, temporal derivatives can

be approximated through Taylor expansions,

$$\begin{aligned}
\partial_\rho V^{k,l+1} &\approx \partial_\rho V^{k,l} + \partial_x V^{k,l} dw_x^{k,l} + \partial_y V^{k,l} dw_y^{k,l} + \partial_t V^{k,l} dw_t^{k,l} \\
\partial_{x\rho} V^{k,l+1} &\approx \partial_{x\rho} V^{k,l} + \partial_{xx} V^{k,l} dw_x^{k,l} + \partial_{xy} V^{k,l} dw_y^{k,l} + \partial_{xt} V^{k,l} dw_t^{k,l} \\
\partial_{y\rho} V^{k,l+1} &\approx \partial_{y\rho} V^{k,l} + \partial_{xy} V^{k,l} dw_x^{k,l} + \partial_{yy} V^{k,l} dw_y^{k,l} + \partial_{yt} V^{k,l} dw_t^{k,l} \\
\partial_{t\rho} V^{k,l+1} &\approx \partial_{t\rho} V^{k,l} + \partial_{xt} V^{k,l} dw_x^{k,l} + \partial_{yt} V^{k,l} dw_y^{k,l} + \partial_{tt} V^{k,l} dw_t^{k,l}
\end{aligned} \tag{H.18}$$

Here, let $d\overrightarrow{w}^{k,l} = (dw_x^{k,l}, dw_y^{k,l}, dw_t^{k,l})$, $w_x^{k,l+1} = w_x^{k,l} + dw_x^{k,l}$, $w_y^{k,l+1} = w_y^{k,l} + dw_y^{k,l}$, and $w_t^{k,l+1} = w_t^{k,l} + dw_t^{k,l}$. Note that this linearization is performed when a fixed iteration is done, instead of explicitly putting Eq. 6.2 into computation. Therefore, Eq. 6.1 is imitated by a sequence of linear approximations replacing one single linear approximation. This strategy is thereby called sequential linearization.

Ψ'_D and Ψ'_S are rewritten as

$$\begin{aligned}
\Psi'^{k,l}_D &= \Psi'((\partial_\rho V^{k,l} + \partial_x V^{k,l} dw_x^{k,l} + \partial_y V^{k,l} dw_y^{k,l} + \partial_t V^{k,l} dw_t^{k,l})^2 \\
&\quad + \gamma((\partial_{x\rho} V^{k,l} + \partial_{xx} V^{k,l} dw_x^{k,l} + \partial_{xy} V^{k,l} dw_y^{k,l} + \partial_{xt} V^{k,l} dw_t^{k,l})^2 \\
&\quad + (\partial_{y\rho} V^{k,l} + \partial_{xy} V^{k,l} dw_x^{k,l} + \partial_{yy} V^{k,l} dw_y^{k,l} + \partial_{yt} V^{k,l} dw_t^{k,l})^2 \\
&\quad + (\partial_{t\rho} V^{k,l} + \partial_{xt} V^{k,l} dw_x^{k,l} + \partial_{yt} V^{k,l} dw_y^{k,l} + \partial_{tt} V^{k,l} dw_t^{k,l})^2)) \\
\Psi'^{k,l}_S &= \Psi'(|\nabla_3(w_x^{k,l} + dw_x^{k,l})|^2 + |\nabla_3(w_y^{k,l} + dw_y^{k,l})|^2 + |\nabla_3(w_t^{k,l} + dw_t^{k,l})|^2)
\end{aligned} \tag{H.19}$$

So Eq. H.17 reads

$$\begin{aligned}
&\Psi'^{k,l}_D(\partial_x V^{k,l}(\partial_\rho V^{k,l} + \partial_x V^{k,l} dw_x^{k,l} + \partial_y V^{k,l} dw_y^{k,l} + \partial_t V^{k,l} dw_t^{k,l}) \\
&+ \gamma(\partial_{xx} V^{k,l}(\partial_{x\rho} V^{k,l} + \partial_{xx} V^{k,l} dw_x^{k,l} + \partial_{xy} V^{k,l} dw_y^{k,l} + \partial_{xt} V^{k,l} dw_t^{k,l}) \\
&+ \partial_{xy} V^{k,l}(\partial_{y\rho} V^{k,l} + \partial_{xy} V^{k,l} dw_x^{k,l} + \partial_{yy} V^{k,l} dw_y^{k,l} + \partial_{yt} V^{k,l} dw_t^{k,l}) \\
&+ \partial_{xt} V^{k,l}(\partial_{t\rho} V^{k,l} + \partial_{xt} V^{k,l} dw_x^{k,l} + \partial_{yt} V^{k,l} dw_y^{k,l} + \partial_{tt} V^{k,l} dw_t^{k,l}))) \\
&- \alpha \mathrm{div}_3 \left(\Psi'^{k,l}_S \nabla_3(w_x^{k,l} + dw_x^{k,l}) \right) = 0
\end{aligned}$$

$$
\begin{aligned}
&\Psi_D'^{k,l}(\partial_y V^{k,l}(\partial_\rho V^{k,l} + \partial_x V^{k,l} dw_x^{k,l} + \partial_y V^{k,l} dw_y^{k,l} + \partial_t V^{k,l} dw_t^{k,l}) \\
&+ \gamma(\partial_{xy} V^{k,l}(\partial_{x\rho} V^{k,l} + \partial_{xx} V^{k,l} dw_x^{k,l} + \partial_{xy} V^{k,l} dw_y^{k,l} + \partial_{xt} V^{k,l} dw_t^{k,l}) \\
&+ \partial_{yy} V^{k,l}(\partial_{y\rho} V^{k,l} + \partial_{xy} V^{k,l} dw_x^{k,l} + \partial_{yy} V^{k,l} dw_y^{k,l} + \partial_{yt} V^{k,l} dw_t^{k,l}) \\
&+ \partial_{yt} V^{k,l}(\partial_{t\rho} V^{k,l} + \partial_{xt} V^{k,l} dw_x^{k,l} + \partial_{yt} V^{k,l} dw_y^{k,l} + \partial_{tt} V^{k,l} dw_t^{k,l}))) \\
&- \alpha \mathrm{div}_3 \left(\Psi_S'^{k,l} \nabla_3 (w_y^{k,l} + dw_y^{k,l}) \right) = 0 \\
&\Psi_D'^{k,l}(\partial_t V^{k,l}(\partial_\rho V^{k,l} + \partial_x V^{k,l} dw_x^{k,l} + \partial_y V^{k,l} dw_y^{k,l} + \partial_t V^{k,l} dw_t^{k,l}) \\
&+ \gamma(\partial_{xt} V^{k,l}(\partial_{x\rho} V^{k,l} + \partial_{xx} V^{k,l} dw_x^{k,l} + \partial_{xy} V^{k,l} dw_y^{k,l} + \partial_{xt} V^{k,l} dw_t^{k,l}) \\
&+ \partial_{yt} V^{k,l}(\partial_{y\rho} V^{k,l} + \partial_{xy} V^{k,l} dw_x^{k,l} + \partial_{yy} V^{k,l} dw_y^{k,l} + \partial_{yt} V^{k,l} dw_t^{k,l}) \\
&+ \partial_{tt} V^{k,l}(\partial_{t\rho} V^{k,l} + \partial_{xt} V^{k,l} dw_x^{k,l} + \partial_{yt} V^{k,l} dw_y^{k,l} + \partial_{tt} V^{k,l} dw_t^{k,l}))) \\
&- \alpha \mathrm{div}_3 \left(\Psi_S'^{k,l} \nabla_3 (w_t^{k,l} + dw_t^{k,l}) \right) = 0
\end{aligned} \tag{H.20}
$$

However, Eq. H.20 still remains nonlinear with respect to incremental TVF vectors $d\overrightarrow{w}^{k,l}$, which is caused by $\Psi_D'^{k,l}$ and $\Psi_S'^{k,l}$. Another inner iteration is introduced to remove their non-linearity. Assume m be the iteration index and let $\Psi_D'^{k,l,m}$ and $\Psi_S'^{k,l,m}$ denote the updated abbreviation parameters. Assuming $d\overrightarrow{w}^{k,l,m} = (dw_x^{k,l,m}, dw_y^{k,l,m}, dw_t^{k,l,m})$, it becomes the updated incremental temporal volume flow vector. Eq. H.20 is linearized as

$$
\begin{aligned}
&\Psi_D'^{k,l,m}(\partial_x V^{k,l}(\partial_\rho V^{k,l} + \partial_x V^{k,l} dw_x^{k,l,m+1} + \partial_y V^{k,l} dw_y^{k,l,m+1} + \partial_t V^{k,l} dw_t^{k,l,m+1}) \\
&+ \gamma(\partial_{xx} V^{k,l}(\partial_{x\rho} V^{k,l} + \partial_{xx} V^{k,l} dw_x^{k,l,m+1} + \partial_{xy} V^{k,l} dw_y^{k,l,m+1} + \partial_{xt} V^{k,l} dw_t^{k,l,m+1}) \\
&+ \partial_{xy} V^{k,l}(\partial_{y\rho} V^{k,l} + \partial_{xy} V^{k,l} dw_x^{k,l,m+1} + \partial_{yy} V^{k,l} dw_y^{k,l,m+1} + \partial_{yt} V^{k,l} dw_t^{k,l,m+1}) \\
&+ \partial_{xt} V^{k,l}(\partial_{t\rho} V^{k,l} + \partial_{xt} V^{k,l} dw_x^{k,l,m+1} + \partial_{yt} V^{k,l} dw_y^{k,l,m+1} + \partial_{tt} V^{k,l} dw_t^{k,l,m+1}))) \\
&- \alpha \mathrm{div}_3 \left(\Psi_S'^{k,l,m} \nabla_3 (w_x^{k,l} + dw_x^{k,l,m+1}) \right) = 0 \\
&\Psi_D'^{k,l,m}(\partial_y V^{k,l}(\partial_\rho V^{k,l} + \partial_x V^{k,l} dw_x^{k,l,m+1} + \partial_y V^{k,l} dw_y^{k,l,m+1} + \partial_t V^{k,l} dw_t^{k,l,m+1}) \\
&+ \alpha(\partial_{xy} V^{k,l}(\partial_{x\rho} V^{k,l} + \partial_{xx} V^{k,l} dw_x^{k,l,m+1} + \partial_{xy} V^{k,l} dw_y^{k,l,m+1} + \partial_{xt} V^{k,l} dw_t^{k,l,m+1})
\end{aligned}
$$

$$
\begin{aligned}
&+ \partial_{yy}V^{k,l}(\partial_{y\rho}V^{k,l} + \partial_{xy}V^{k,l}dw_x^{k,l,m+1} + \partial_{yy}V^{k,l}dw_y^{k,l,m+1} + \partial_{yt}V^{k,l}dw_t^{k,l,m+1}) \\
&+ \partial_{yt}V^{k,l}(\partial_{t\rho}V^{k,l} + \partial_{xt}V^{k,l}dw_x^{k,l,m+1} + \partial_{yt}V^{k,l}dw_y^{k,l,m+1} + \partial_{tt}V^{k,l}dw_t^{k,l,m+1}))) \\
&- \alpha \mathrm{div}_3 \left(\Psi_S'^{k,l,m} \nabla_3 (w_y^{k,l} + dw_y^{k,l,m+1}) \right) = 0 \\
&\Psi_D'^{k,l,m}(\partial_t V^{k,l}(\partial_\rho V^{k,l} + \partial_x V^{k,l}dw_x^{k,l,m+1} + \partial_y V^{k,l}dw_y^{k,l,m+1} + \partial_t V^{k,l}dw_t^{k,l,m+1}) \\
&+ \gamma(\partial_{xt}V^{k,l}(V_{x\rho}^{k} + V_{xx}^{k,l}dw_x^{k,l,m+1} + V_{xy}^{k,l}dw_y^{k,l,m+1} + V_{xt}^{k,l}dw_t^{k,l,m+1}) \\
&+ \partial_{yt}V^{k,l}(\partial_{y\rho}V^{k,l} + \partial_{xy}V^{k,l}dw_x^{k,l,m+1} + \partial_{yy}V^{k,l}dw_y^{k,l,m+1} + \partial_{yt}V^{k,l}dw_t^{k,l,m+1}) \\
&+ \partial_{tt}V^{k,l}(\partial_{t\rho}V^{k,l} + \partial_{xt}V^{k,l}dw_x^{k,l,m+1} + \partial_{yt}V^{k,l}dw_y^{k,l,m+1} + \partial_{tt}V^{k,l}dw_t^{k,l,m+1}))) \\
&- \alpha \mathrm{div}_3 \left(\Psi_S'^{k,l,m} \nabla_3 (w_t^{k,l} + dw_t^{k,l,m+1}) \right) = 0
\end{aligned}
\tag{H.21}
$$

where $(dw_x^{k,l,0}, dw_y^{k,l,0}, dw_t^{k,l,0}) = (0,0,0)$. $dw_x^{k,l,m+1}$, $dw_y^{k,l,m+1}$ and $dw_t^{k,l,m+1}$ have been removed from $\Psi_D'^{k,l,m}$ and $\Psi_S'^{k,l,m}$. Eq. H.21 is a linear equation with respect to $dw_x^{k,l,m+1}$, $dw_y^{k,l,m+1}$ and $dw_t^{k,l,m+1}$.

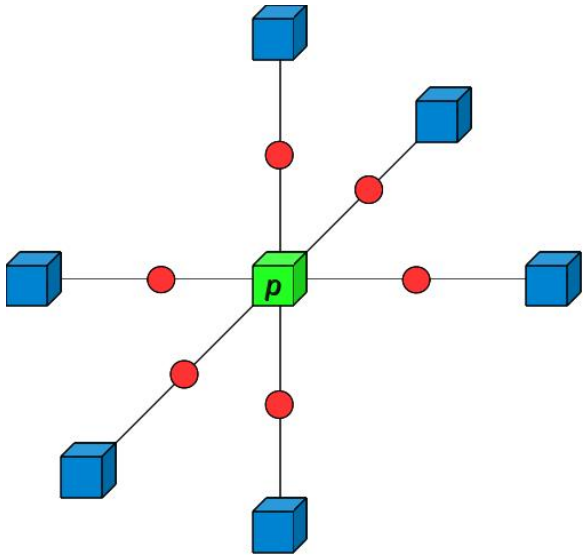


Figure H.2: *Numerical divergence computation between a voxel* $\boldsymbol{p}$ *shown in a green cube and its 6-neighborhood voxels illustrated in blue cubes.* The divergence term between $\boldsymbol{p}$ and its neighborhood voxels are defined at locations indicated by red spheres.

Next, let me investigate how to compute divergence terms at a voxel $\boldsymbol{p}$ in Eq. H.21.

$$\begin{aligned} &\mathrm{div}_3\left(\Psi_S'^{k,l,m}\nabla_3(w_x^{k,l}+dw_x^{k,l,m+1})\right)\\ &\mathrm{div}_3\left(\Psi_S'^{k,l,m}\nabla_3(w_y^{k,l}+dw_y^{k,l,m+1})\right)\\ &\mathrm{div}_3\left(\Psi_S'^{k,l,m}\nabla_3(w_t^{k,l}+dw_t^{k,l,m+1})\right) \end{aligned} \tag{H.22}$$

Similar to the 4-neighborhood used in incremental egomotion estimation, the 6-neighborhood is employed to compute divergence terms in the temporal volume domain, as illustrated in Fig. H.2. Assuming the indices of the current voxel $\boldsymbol{p}$ be (x, y, z), indicated by a green cube in Fig. H.2, the divergence terms between this voxel and its 6-neighborhood voxels in blue cubes($(x+\frac{1}{2}, y, z), (x-\frac{1}{2}, y, z), (x, y+\frac{1}{2}, z), (x, y-\frac{1}{2}, z), (x, y, z+\frac{1}{2})$, and $(x, y, z-\frac{1}{2})$) are represented as red spheres. Because the divergence terms in Eq. H.22 involve TVF vector differences, let me investigate their computations. Assuming w is the abbreviation of w_x, w_y or w_t, TVF vector difference is defined as

$$|\nabla w(x+\frac{1}{2},y,z)| \approx \sqrt{\begin{array}{c}(w(x+1,y,z)-w(x,y,z))^2\\ +\left(\frac{1}{2}\left(\frac{w(x+1,y+1,z)-w(x+1,y-1,z)}{2}+\frac{w(x,y+1,z)-w(x,y-1,z)}{2}\right)\right)^2\\ +\left(\frac{1}{2}\left(\frac{w(x+1,y,z+1)-w(x+1,y,z-1)}{2}+\frac{w(x,y,z+1)-w(x,y,z-1)}{2}\right)\right)^2\end{array}}$$

$$|\nabla w(x-\frac{1}{2},y,z)| \approx \sqrt{\begin{array}{c}(w(x,y,z)-w(x-1,y,z))^2\\ +\left(\frac{1}{2}\left(\frac{w(x-1,y+1,z)-w(x-1,y-1,z)}{2}+\frac{w(x,y+1,z)-w(x,y-1,z)}{2}\right)\right)^2\\ +\left(\frac{1}{2}\left(\frac{w(x-1,y,z+1)-w(x-1,y,z-1)}{2}+\frac{w(x,y,z+1)-w(x,y,z-1)}{2}\right)\right)^2\end{array}}$$

$$|\nabla w(x,y+\frac{1}{2},z)| \approx \sqrt{\begin{array}{c}(w(x,y+1,z)-w(x,y,z))^2\\ +\left(\frac{1}{2}\left(\frac{w(x+1,y+1,z)-w(x-1,y+1,z)}{2}+\frac{w(x+1,y,z)-w(x-1,y,z)}{2}\right)\right)^2\\ +\left(\frac{1}{2}\left(\frac{w(x,y+1,z+1)-w(x,y+1,z-1)}{2}+\frac{w(x,y,z+1)-w(x,y,z-1)}{2}\right)\right)^2\end{array}}$$

$$
\begin{aligned}
|\nabla w(x, y-\frac{1}{2}, z)| \approx & \sqrt{\begin{array}{c}(w(x,y,z)-w(x,y-1,z))^2 \\ +\left(\frac{1}{2}\left(\frac{w(x+1,y-1,z)-w(x-1,y-1,z)}{2}+\frac{w(x+1,y,z)-w(x-1,y,z)}{2}\right)\right)^2 \\ +\left(\frac{1}{2}\left(\frac{w(x,y-1,z+1)-w(x,y-1,z-1)}{2}+\frac{w(x,y,z+1)-w(x,y,z-1)}{2}\right)\right)^2\end{array}} \\
|\nabla w(x, y, z+\frac{1}{2})| \approx & \sqrt{\begin{array}{c}(w(x,y,z+1)-w(x,y,z))^2 \\ +\left(\frac{1}{2}\left(\frac{w(x+1,y,z+1)-w(x-1,y,z+1)}{2}+\frac{w(x+1,y,z)-w(x-1,y,z)}{2}\right)\right)^2 \\ +\left(\frac{1}{2}\left(\frac{w(x,y+1,z)-w(x,y-1,z)}{2}+\frac{w(x,y+1,z+1)-w(x,y-1,z+1)}{2}\right)\right)^2\end{array}} \\
|\nabla w(x, y, z-\frac{1}{2})| \approx & \sqrt{\begin{array}{c}(w(x,y,z)-w(x,y,z-1))^2 \\ +\left(\frac{1}{2}\left(\frac{w(x+1,y,z)-w(x-1,y,z)}{2}+\frac{w(x+1,y,z-1)-w(x-1,y,z-1)}{2}\right)\right)^2 \\ +\left(\frac{1}{2}\left(\frac{w(x,y+1,z)-w(x,y-1,z)}{2}+\frac{w(x,y+1,z-1)-w(x,y-1,z-1)}{2}\right)\right)^2\end{array}}
\end{aligned}
\tag{H.23}
$$

Next, I use $\mathrm{div}_3(\Psi'_S(|\nabla w_x|^2+|\nabla w_y|^2+|\nabla w_t|^2)\nabla w_x)$ as an example to illustrate the divergence computation.

$$
\begin{aligned}
&\mathrm{div}_3(\Psi'_S(|\nabla_3 w_x|^2+|\nabla_3 w_y|^2+|\nabla_3 w_t|^2)\nabla w_x) \\
&= \Psi'_S(|\nabla_3 w_x(x+\frac{1}{2},y,z)|^2+|\nabla_3 w_y(x+\frac{1}{2},y,z)|^2+|\nabla_3 w_t(x+\frac{1}{2},y,z)|^2)(w_x(x+1,y,z)-w_x(x,y,z)) \\
&- \Psi'_S(|\nabla_3 w_x(x-\frac{1}{2},y,z)|^2+|\nabla_3 w_y(x-\frac{1}{2},y,z)|^2+|\nabla_3 w_t(x-\frac{1}{2},y,z)|^2)(w_x(x,y,z)-w_x(x-1,y,z)) \\
&+ \Psi'_S(|\nabla_3 w_x(x,y+\frac{1}{2},z)|^2+|\nabla_3 w_y(x,y+\frac{1}{2},z)|^2+|\nabla_3 w_t(x,y+\frac{1}{2},z)|^2)(w_x(x,y+1,z)-w_x(x,y,z)) \\
&- \Psi'_S(|\nabla_3 w_x(x,y-\frac{1}{2},z)|^2+|\nabla_3 w_y(x,y-\frac{1}{2},z)|^2+|\nabla_3 w_t(x,y-\frac{1}{2},z)|^2)(w_x(x,y,z)-w_x(x,y-1,z)) \\
&+ \Psi'_S(|\nabla_3 w_x(x,y,z+\frac{1}{2})|^2+|\nabla_3 w_y(x,y,z+\frac{1}{2})|^2+|\nabla_3 w_t(x,y,z+\frac{1}{2})|^2)(w_x(x,y,z+1)-w_x(x,y,z)) \\
&- \Psi'_S(|\nabla_3 w_x(x,y,z-\frac{1}{2})|^2+|\nabla_3 w_y(x,y,z-\frac{1}{2})|^2+|\nabla_3 w_t(x,y,z-\frac{1}{2})|^2)(w_x(x,y,z)-w_x(x,y,z-1)) \\
&= (\Psi'_S)_{(x,y,z)\sim(x-1,y,z)}w_x(x-1,y,z)+(\Psi'_S)_{(x,y,z)\sim(x,y-1,z)}w_x(x,y-1,z) \\
&+ (\Psi'_S)_{(x,y,z)\sim(x+1,y,z)}w_x(x+1,y,z)+(\Psi'_S)_{(x,y,z)\sim(x,y+1,z)}w_x(x,y+1,z) \\
&+ (\Psi'_S)_{(x,y,z)\sim(x,y,z+1)}w_x(x,y,z+1)+(\Psi'_S)_{(x,y,z)\sim(x,y,z-1)}w_x(x,y,z-1) \\
&- ((\Psi'_S)_{(x,y,z)\sim(x-1,y,z)}+(\Psi'_S)_{(x,y,z)\sim(x,y-1,z)}+(\Psi'_S)_{(x,y,z)\sim(x+1,y,z)} \\
&+ (\Psi'_S)_{(x,y,z)\sim(x,y+1,z)}+(\Psi'_S)_{(x,y,z)\sim(x,y,z-1)}+(\Psi'_S)_{(x,y,z)\sim(x,y,z+1)})w_x(x,y,z) \\
&= \sum_{q\in\mathcal{N}(\boldsymbol{p})}(\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}w_x(\boldsymbol{q})-\sum_{q\in\mathcal{N}(\boldsymbol{p})}(\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}w_x(\boldsymbol{p})
\end{aligned}
\tag{H.24}
$$

where $\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})$ is a 6-neighborhood voxel of $\boldsymbol{p}$. $(\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}$ denotes the diffusivity between points $\boldsymbol{p}$ and $\boldsymbol{q}$. The same derivation can be used to compute the other two

divergence terms in Eq. H.22. Substituting Eq. H.24 for Eq. H.21, we can explicitly express $(dw_x^{k,l,m+1}, dw_y^{k,l,m+1}, dw_t^{k,l,m+1})$ at $\boldsymbol{p}$ as

$$
\begin{aligned}
(dw_x)_{\boldsymbol{p}}^{k,l,m+1} = & \left(\sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} \left((w_x)_{\boldsymbol{q}}^{k,l} + (dw_x)_{\boldsymbol{q}}^{k,l,m+1} \right) - \sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} (w_x)_{\boldsymbol{p}}^{k,l} \right. \\
& - \frac{1}{\alpha} (\Psi'_D)_{\boldsymbol{p}}^{k,l,m} \left((\partial_x V^{k,l})_{\boldsymbol{p}} \left((\partial_y V^{k,l})_{\boldsymbol{p}} (dw_y)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_t V^{k,l})_{\boldsymbol{p}} (dw_t)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_\rho V^{k,l})_{\boldsymbol{p}} \right) \right. \\
& + \gamma (\partial_{xx} V^{k,l})_{\boldsymbol{p}} \left((\partial_{xy} V^{k,l})_{\boldsymbol{p}} (dw_y)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_{xt} V^{k,l})_{\boldsymbol{p}} (dw_t)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_{x\rho} V^{k,l})_{\boldsymbol{p}} \right) \\
& + \gamma (\partial_{xy} V^{k,l})_{\boldsymbol{p}} \left((\partial_{yy} V^{k,l})_{\boldsymbol{p}} (dw_y)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_{yt} V^{k,l})_{\boldsymbol{p}} (dw_t)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_{y\rho} V^{k,l})_{\boldsymbol{p}} \right) \\
& \left. + \gamma (\partial_{xt} V^{k,l})_{\boldsymbol{p}} \left((\partial_{yt} V^{k,l})_{\boldsymbol{p}} (dw_y)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_{tt} V^{k,l})_{\boldsymbol{p}} (dw_t)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_{t\rho} V^{k,l})_{\boldsymbol{p}} \right) \right) \\
& \left. / \left(\sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} + \frac{1}{\alpha} (\Psi'_D)_{\boldsymbol{p}}^{k,l,m} \left((\partial_x V^{k,l})_{\boldsymbol{p}}^2 + \gamma \left((\partial_{xx} V^{k,l})_{\boldsymbol{p}}^2 + (\partial_{xy} V^{k,l})_{\boldsymbol{p}}^2 + (\partial_{xt} V^{k,l})_{\boldsymbol{p}}^2 \right) \right) \right) \right. \\
(dw_y)_{\boldsymbol{p}}^{k,l,m+1} = & \left(\sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} \left((w_y)_{\boldsymbol{q}}^{k,l} + (dw_y)_{\boldsymbol{q}}^{k,l,m+1} \right) - \sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} (w_y)_{\boldsymbol{p}}^{k,l} \right. \\
& - \frac{1}{\alpha} (\Psi'_D)_{\boldsymbol{p}}^{k,l,m} \left((\partial_y V^{k,l})_{\boldsymbol{p}} \left((\partial_x V^{k,l})_{\boldsymbol{p}} (dw_x)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_t V^{k,l})_{\boldsymbol{p}} (dw_t)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_\rho V^{k,l})_{\boldsymbol{p}} \right) \right. \\
& + \gamma (\partial_{xy} V^{k,l})_{\boldsymbol{p}} \left((\partial_{xx} V^{k,l})_{\boldsymbol{p}} (dw_x)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_{xt} V^{k,l})_{\boldsymbol{p}} (dw_t)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_{x\rho} V^{k,l})_{\boldsymbol{p}} \right) \\
& + \gamma (\partial_{yy} V^{k,l})_{\boldsymbol{p}} \left((\partial_{xy} V^{k,l})_{\boldsymbol{p}} (dw_x)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_{yt} V^{k,l})_{\boldsymbol{p}} (dw_t)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_{y\rho} V^{k,l})_{\boldsymbol{p}} \right) \\
& \left. + \gamma (\partial_{yt} V^{k,l})_{\boldsymbol{p}} \left((\partial_{xt} V^{k,l})_{\boldsymbol{p}} (dw_x)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_{tt} V^{k,l})_{\boldsymbol{p}} (dw_t)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_{t\rho} V^{k,l})_{\boldsymbol{p}} \right) \right) \\
& \left. / \left(\sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} + \frac{1}{\alpha} (\Psi'_D)_{\boldsymbol{p}}^{k,l,m} \left((\partial_y V^{k,l})_{\boldsymbol{p}}^2 + \gamma \left((\partial_{xy} V^{k,l})_{\boldsymbol{p}}^2 + (\partial_{yy} V^{k,l})_{\boldsymbol{p}}^2 + (\partial_{yt} V^{k,l})_{\boldsymbol{p}}^2 \right) \right) \right) \right. \\
(dw_t)_{\boldsymbol{p}}^{k,l,m+1} = & \left(\sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} \left((w_t)_{\boldsymbol{q}}^{k,l} + (dw_t)_{\boldsymbol{q}}^{k,l,m+1} \right) - \sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} (w_t)_{\boldsymbol{p}}^{k,l} \right. \\
& - \frac{1}{\alpha} (\Psi'_D)_{\boldsymbol{p}}^{k,l,m} \left((\partial_t V^{k,l})_{\boldsymbol{p}} \left((\partial_x V^{k,l})_{\boldsymbol{p}} (dw_x)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_y V^{k,l})_{\boldsymbol{p}} (dw_y)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_\rho V^{k,l})_{\boldsymbol{p}} \right) \right. \\
& + \gamma (\partial_{xt} V^{k,l})_{\boldsymbol{p}} \left((\partial_{xx} V^{k,l})_{\boldsymbol{p}} (dw_x)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_{xy} V^{k,l})_{\boldsymbol{p}} (dw_y)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_{x\rho} V^{k,l})_{\boldsymbol{p}} \right) \\
& + \gamma (\partial_{yt} V^{k,l})_{\boldsymbol{p}} \left((\partial_{xy} V^{k,l})_{\boldsymbol{p}} (dw_x)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_{yy} V^{k,l})_{\boldsymbol{p}} (dw_y)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_{y\rho} V^{k,l})_{\boldsymbol{p}} \right) \\
& \left. + \gamma (\partial_{tt} V^{k,l})_{\boldsymbol{p}} \left((\partial_{xt} V^{k,l})_{\boldsymbol{p}} (dw_x)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_{yt} V^{k,l})_{\boldsymbol{p}} (dw_y)_{\boldsymbol{p}}^{k,l,m+1} + (\partial_{t\rho} V^{k,l})_{\boldsymbol{p}} \right) \right) \\
& \left. / \left(\sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} (\Psi'_S)_{\boldsymbol{p}\sim\boldsymbol{q}}^{k,l,m} + \frac{1}{\alpha} (\Psi'_D)_{\boldsymbol{p}}^{k,l,m} \left((\partial_t V^{k,l})_{\boldsymbol{p}}^2 + \gamma \left((\partial_{xt} V^{k,l})_{\boldsymbol{p}}^2 + (\partial_{yt} V^{k,l})_{\boldsymbol{p}}^2 + (\partial_{tt} V^{k,l})_{\boldsymbol{p}}^2 \right) \right) \right) \right. \qquad \text{(H.25)}
\end{aligned}
$$