Robust and Effective Techniques for Multi-modal Image Registration

Guohua Lv

A dissertation submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy

Supervisors:
Professor Guojun Lu
Dr Shyh Wei Teng

Faculty of Information Technology
Monash University

May 2015

DECLARATION

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

[redacted]

February 6, 2015

Dedicated to my parents and brother.

Copyright Notices

Notice 1

Notice 2

Acknowledgements

First and foremost, I would like to thank my supervisors Professor Guojun Lu and Dr. Shyh Wei Teng. Without their great guidance, patience and encouragement, my PhD journey would not have been made it so far. They taught me the lessons of rigorous academic research, and working smartly and hard, which will be beneficial to the rest of my academic life.

With a very heavy heart, I would like to say "thank you" to Associate Professor Martin Lackmann for providing scholarship in my first year and image data sets used in my research. I just wish him peace in heaven.

I am really grateful to Monash University for the scholarships. Without any financial stress, I have been able to devote my energy into doing research.

I would like to express my heartfelt gratitude to an Aussie, Christian family: Stephen, Lucy, Philip, Laura and Hannah. Their great care and consideration relieved my loneliness and stress in the past few years. Also, I have been able to get an insight into cultural differences while chatting, camping or hanging out with them. It is hard for me to recall all wonderful and meaningful moments when I was with them. No doubt, God works in all things. Glory to God.

Last but not least, to my parents and brother, I am sure they are very proud of me. This justifies years of sacrifices and efforts that they have made in bringing me up and supporting my study.

Abstract

Image registration is the process of estimating the optimal transformation that aligns different imaging data into spatial correspondences. Multi-modal image registration is to register images which are captured by different types of imaging devices. This thesis aims to develop robust and effective techniques for multi-modal image registration. The challenge lies in the fact that the visual appearance may differ a lot between corresponding parts of multi-modal images. We have been exploring ways by investigating local image features. Two main contributions have been made in this thesis.

First, we have improved existing mono-modal and multi-modal image registration techniques by better utilizing gradient information. For a feature-based image registration technique, its effectiveness to a large extent relies on the discrimination power of local descriptors. In the existing techniques, gradient information is utilized in a number of ways for building local descriptors. We have analyzed the limitations of these techniques, and have proposed a technique for better utilizing gradient information. As a result, the discrimination power of local descriptors has been enhanced, leading to a better registration performance.

Second, we have developed a new multi-modal image registration technique, which has the following innovations:

1. We have proposed a technique to detect the intrinsic structural similarity in multi-modal microscopic images. This is achieved by exploiting the characteristics in intensity relationships between the Red-Green-Blue color channels.

2. To increase robustness to content differences, contour-based corners are used, instead of intensity-based keypoints in a state-of-the-art multi-modal image registration technique.

3. We have proposed a new local descriptor to better represent corners.

4. We have proposed a new way of scale estimation by making use of geometric relationships between corner triplets in two images.

The proposed multi-modal image registration technique achieves greater robustness in terms of both content differences and scale differences as compared to the state-of-the-art multi-modal image registration technique.

Abbreviations

BBF	Best Bin First
BRIEF	Binary Robust Independent Elementary Features
BRISK	Binary Robust Invariant Scalable Keypoints
COREG	COrner based REGistration
CPDA	Chord-to-Point Distance Accumulation
CSS	Curvature Scale-Space
CT	Computed Tomography
DEPAC	Distribution of Edge Pixels Along Contour
DoG	Difference of Gaussian
DSS	Detector of Structural Similarity
EO	Electro Optical
FAST	Features from Accelerated Segment Test
FREAK	Fast REtinA Keypoint
GDB-ICP	Generalized Dual-Bootstrap Iterative Closest Point
GLOH	Gradient Location-Orientation Histogram
GM	Gradient Magnitude
GO	Gradient Occurrence
GO-SSIFT	Gradient Occurrence with Symmetric SIFT
IS-SIFT	Improved Symmetric SIFT
LBP	Local Binary Pattern
LoG	Laplacian of Gaussian
LSS	Local Self-Similarity
LTP	Local Ternary Pattern
MI	Mutual Information
MI-SIFT	Mirror and Inversion Invariant SIFT

Abbreviations (continued)

MIND	Modality Independent Neighborhood Descriptor
MOG	Magnitudes and Occurrences of Gradient
MRI	Magnetic Resonance Imaging
MSER	Maximally Stable Extremal Region
NIR	Near Infra-Red
NLSD	Non-Local Shape Descriptor
NMI	Normalized Mutual Information
NNDR	Nearest Neighbor Distance Ratio
NOP	Number of Overlapped Pixels
ORB	Oriented and Rotated BRIEF
PCA	Principal Component Analysis
PDF	Probability Distribution Function
PET	Positron Emission Tomography
PIIFD	Partial Intensity Invariant Feature Descriptor
RANSAC	RANdom SAmple Consensus
SIFT	Scale Invariant Feature Transform
SOI	Structure of Interest
SPECT	Single Photon Emission Computed Tomography
SSIFT	Symmetric SIFT
SURF	Speeded-Up Robust Features
WLD	Weber Local Descriptor

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Image registration is an important process in computer vision and image processing applications, especially in medical imaging analysis. It is the process of finding the correct spatial alignment between images of the same scene that have been acquired in different imaging conditions [13, 76, 99, 118]. The difference in imaging conditions may be due to differences in time, viewpoint, illumination, capturing device, noise, cluttering and occlusion [118]. For example, Figure 1.1 shows two images which vary in scale, rotation and translation. In the domain of image registration, one image is usually referred to as the reference image, while the other is referred to as the target image, as shown in Figure 1.1 (a) and (b) respectively.

(a) Reference Image (b) Target Image

Figure 1.1: An Example of Transformations between Images

As pointed out in [13], a new image registration technique is developed by aiming

to address different combinations of problems in the following four areas:

1. feature space, involving the attributes of image data that are to be utilized for matching,
2. search space, associated with transformations between images to be registered,
3. search strategy, defining the approach of going through the transformations for the one that suits best, and
4. similarity metric, used to evaluate the merit of any possible solution.

Based on this analysis, the registration of two images I_r and I_t can be formulated as

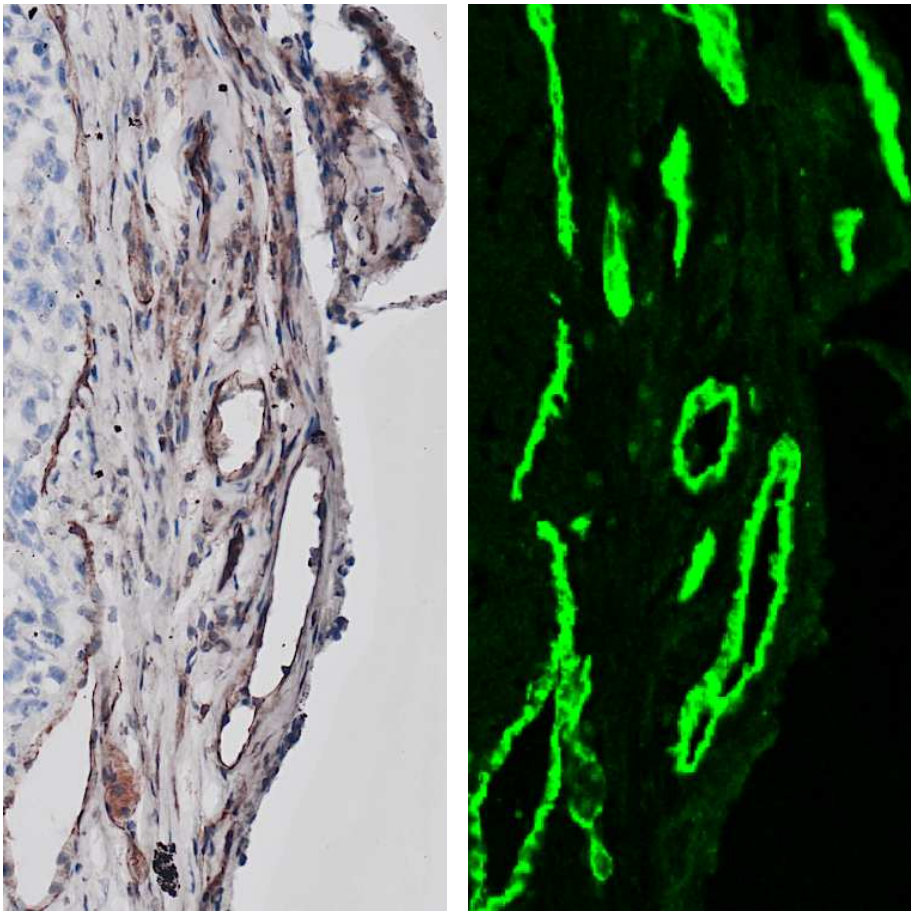
$$\widehat{T} = \arg\max_{T \in \mathcal{T}} \mathcal{S}(I_r, I_t(T)), \tag{1.1}$$

where I_r and I_t are the reference and target images, $\mathcal{T}$ and $\mathcal{S}$ denote the space of transformations and the similarity measure respectively. As Equation 1.1 states, image registration is a process of finding such a transformation that maximizes the similarity between the reference and target images.

Multi-modal image registration is to register images which are captured by different types of devices and has applications in remote sensing, robot navigation, security surveillance and medical image analysis, etc. Particularly in medical imaging, it is very common that images are captured by different types of scanners such as CT (computed tomography), MRI (magnetic resonance imaging), PET (positron emission tomography), SPECT (single-photon emission computed tomography), just to name a few [60]. In a pair of multi-modal images, intensity variations between corresponding parts might be very substantial [32, 106]. Thus, it is a challenging task to effectively register multi-modal images.

Multi-modal image registration can be viewed as the task of integrating information from different types of sources [46] and is especially important in medical imaging. An accurate registration is helpful to image-guided therapy, diagnoses of various diseases, patient monitoring, planning and assessing the quality of treatment, etc [8].

1.2 Motivation



(a) Color (b) Confocal

Figure 1.2: An Example of Multimodal Microscopic Images

The focus of the research in this thesis is on developing a robust technique for registering multi-modal images. Among the different types of multi-modal images we have tested, microscopic images are the most challenging. A sample pair of such images is given in Figure 1.2. The two images in Figure 1.2(a) and (b) are called color image and confocal image respectively in this thesis. Clearly, visual differences between color and confocal images are very large, which is caused by different staining and capturing techniques. From the perspective of image registration, we are motivated by a few potential problems as follows.

i. In the two types of microscopic images, many image structures in the color image

do not appear at the corresponding parts of the confocal image. Thus, the intrinsic structural similarity needs to be detected.

ii. Even at corresponding parts where image structures are clear in both of the two types of images, their contents are largely different. It would be challenging to deal with the large content differences from the perspective of image registration.

iii. It might be more and more challenging to effectively register color and confocal images as their scale difference increases.

1.3 Research Objectives

Overall, this research aims to propose multi-modal image registration techniques. As our objectives, a proposed technique should

i. be able to effectively register multi-modal microscopic images;

ii. be invariant to different types of imaging modalities;

iii. be robust to differences in image contents;

iv. be robust to differences in scale.

1.4 Contributions of the Thesis

A summary of contributions of the thesis is given as follows.

i. The SIFT (Scale Invariant Feature Transform) descriptor [56] is improved by better utilizing gradient information in building and matching descriptors. By pointing out the limitations of only using either Gradient Magnitudes (GM) or Gradient Occurrences (GO) in building descriptors, we observe that both GM and GO are important gradient information in building image descriptors. Thus, we will propose to utilize both of the two types of gradient information. The proposed technique improves the registration performance in both mono-modal and multi-modal cases as compared to only using either GM or GO. More

generally, the proposed technique can be applied or extended to SIFT-like descriptors to improve the registration performance. (See Chapter 3)

ii. We will propose a multi-modal image registration technique based on corners, which has the following four components.

   a. In multi-modal microscopic images, the intrinsic structural similarity is detected. Due to the low structural similarity in these images, it is very challenging to achieve an effective registration. The structural similarity is detected by utilizing intensity relationships between the Red-Green-Blue color channels. (See Chapter 4)

   b. Rather than using intensity-based keypoints in existing multi-modal image registration techniques, contour-based corners are used. In order to achieve robustness to content differences, curvatures are used to represent corners. (See Chapter 5)

   c. To increase the discrimination in representing corners, we will propose a novel corner descriptor called Distribution of Edge Pixels Along Contour (DEPAC). Due to the fact that the number of edges between corresponding parts of multi-modal images may differ a lot, a DEPAC descriptor only represents the edges along the contour where a corner is located. (See Chapter 5)

   d. We will propose a new way of estimating the scale difference in an image pair. This is achieved by making use of geometric relationships between corner triplets in two images. The estimated scale difference is very close to the ground-truth scale difference. (See Chapter 5)

   Compared with the latest multi-modal image registration technique in [49], the proposed technique achieves greater robustness to both content differences and scale differences.

1.5 Structure of the Thesis

The rest of the thesis is structured as follows. In Chapter 2, a review of existing image registration techniques is given and a few promising registration techniques

are highlighted. In Chapter 3, a study is made on gradient utilizations for building and matching SIFT-like descriptors, and a new way of utilizing gradients is proposed. Chapter 4 addresses the low structural similarity in multi-modal microscopic images and elaborates our proposed detector of structural similarity. In Chapter 5, we will analyze the problem in multi-modal microscopic images after applying the proposed technique in Chapter 4, and propose a multi-modal image registration technique based on corners. Chapter 6 concludes the thesis and points out directions for our future work.

Chapter 2

Literature Review

In this chapter we will review the existing image registration techniques. The entire chapter is organized as follows. In Section 2.1, geometric transformations in image registration are categorized and introduced, where it is pointed out which transformations we focus on. Next, we describe the general approach of intensity-based and feature-based registration techniques in Section 2.2, where it is highlighted that research in this thesis is based on feature-based registration techniques. From Section 2.3 to Section 2.7, we will review the existing feature-based registration techniques. In Section 2.3, techniques for detecting local features will be reviewed. Sections 2.4 and 2.5 summarize popular mono-modal and multi-modal image registration techniques. Techniques for feature matching and refining matches are discussed in Sections 2.6 and 2.7 respectively. Finally, Section 2.8 summarizes the chapter.

2.1 Models of Geometric Transformations

As mentioned in Chapter 1, image registration aims to estimate the optimal transformation which is used to align two images. Geometric transformations for image registration can be divided into four categories: rigid, affine, projective and curved [60, 77, 110]. The four categories of transformations are summarized as follows. Let (x_1, y_1) and (x_2, y_2) denote two points from the two images that are being registered. Under a particular geometric transformation, (x_1, y_1) is transformed to (x_2, y_2).

2.1.1 Rigid Transformation

A rigid transformation only involves changes in rotation and translation [20,110] as

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}, \tag{2.1}$$

where θ is the rotation angle, t_x and t_y are translations in the x and y directions respectively. Rigid transformation is also known as rigid-body transformation [20,34] as this type of transformation preserves the distance between any two points in the reference image.

2.1.2 Affine Transformation

In addition to rotation and translation changes, transformations allow for a global change of scale and/or shear [108] are referred to as affine transformations [34]. Under an affine transformation, (x_1, y_1) is transformed to (x_2, y_2) [49,98,110] by

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix}. \tag{2.2}$$

It is also common for Equation 2.2 to be re-formulated as

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}, \tag{2.3}$$

where $\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$ is known as the transformation matrix for an affine transformation.

An affine transformation has three properties as follows [98]. First, the collinearity relation between points is preserved, meaning that all points lying on the same line are still collinear after an affine transformation. Second, the distance ratio between line segments remains unchanged. For different collinear points p_1, p_2 and p_3, the distance

ratio between $|\overrightarrow{p_1p_2}|$ and $|\overrightarrow{p_2p_3}|$ does not change after an affine transformation. Third, parallel lines continue to be parallel after the transformation.

2.1.3 Projective Transformation

Projective transformation is also known as homography as this transformation is operated on homogeneous coordinates [101]. Under a projective transformation, (x_1, y_1) is transformed to (x_2', y_2') on homogeneous coordinates and is then normalized onto Cartesian coordinates to obtain (x_2, y_2) as

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = H \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} x_2' \\ y_2' \\ \omega \end{bmatrix} = \begin{bmatrix} \omega x_2 \\ \omega y_2 \\ \omega \end{bmatrix}, \tag{2.4}$$

where H is known as the homography matrix, and ω is a coefficient for the normalization between homogeneous coordinates and Cartesian coordinates. From Equation 2.4, (x_2, y_2) can be computed by

$$x_2 = \frac{h_{11}x_1 + h_{12}y_1 + h_{13}}{h_{31}x_1 + h_{32}y_1 + h_{33}} \tag{2.5}$$

and

$$y_2 = \frac{h_{21}x_1 + h_{22}y_1 + h_{23}}{h_{31}x_1 + h_{32}y_1 + h_{33}}. \tag{2.6}$$

In contrast to the properties of affine transformation stated in Section 2.1.2, the first and second properties apply to projective transformation, but the third property does not. Only straight lines are preserved after a projective transformation.

2.1.4 Curved Transformation

Curved transformation ia also known as elastic or deformable transformation [60,77]. Under a curved transformation, straight lines may be mapped to curves [34], which is more challenging than rigid, affine and projective transformations to achieve an effective registration. There are many different deformation models [34, 99]. In

general, deformation models can be divided into two categories: physical models and function representations [34]. A detailed introduction can be found in [99].

Note that curved transformations are not involved in our multi-modal microscopic images and other tested multi-modal images. Thus, dealing with curved transformations is outside the focus of this thesis. The tested multi-modal images involve affine or projective transformations.

2.2 Intensity-based vs Feature-based Techniques for Image Registration

Image registration techniques can be categorized into intensity-based and feature-based techniques. In this section, we will summarize the general approach for each of the two categories.

2.2.1 Intensity-based Techniques

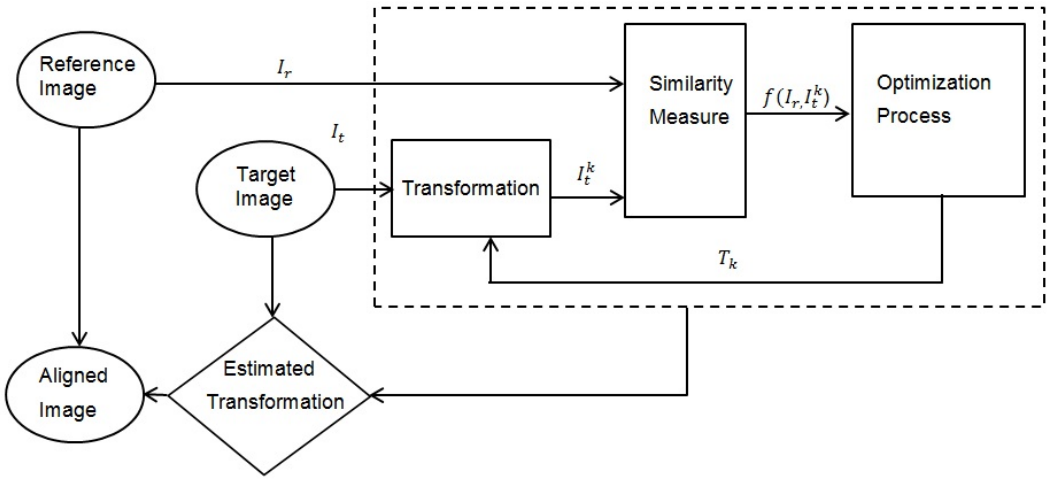


Figure 2.1: General Approach of Intensity-based Image Registration Techniques

In general, an intensity-based image registration technique estimates a transformation between the reference and target images by directly comparing their intensity patterns [99]. In recent years, popular intensity-based image registration techniques include [45, 71–73, 78, 100]. Figure 2.1 illustrates the general approach of

intensity-based image registration techniques. This approach can be summarized as follows. The reference image I_r and the target image I_t are the two images to be registered. A geometrical transformation T_k is applied to transform the target image, where k represents the k^{th} iteration of the optimization process. The transformed target image is denoted as I_t^k. The transformed target image and the reference images are overlapped. The overlapped area is measured by a similarity metric $f(I_r, I_t^k)$. The optimization process adaptively adjusts geometrical transformations until the similarity between the reference image and the transformed target image is maximized. With an estimated transformation that leads to the optimal similarity, the reference and target images are aligned.

2.2.2 Feature-based Techniques

A feature-based image registration technique establishes correspondences between interest points in the reference and target images [66]. With correspondences, a geometrical transformation is estimated and then used to align two images [66]. This category of image registration techniques generally includes the following four steps.

a. Detecting feature points in the reference and target images

   Generally, image points which differ from their neighborhood in a specific way are detected. In Section 2.3, two categories of feature points, keypoints and corners, will be discussed.

b. Representing feature points

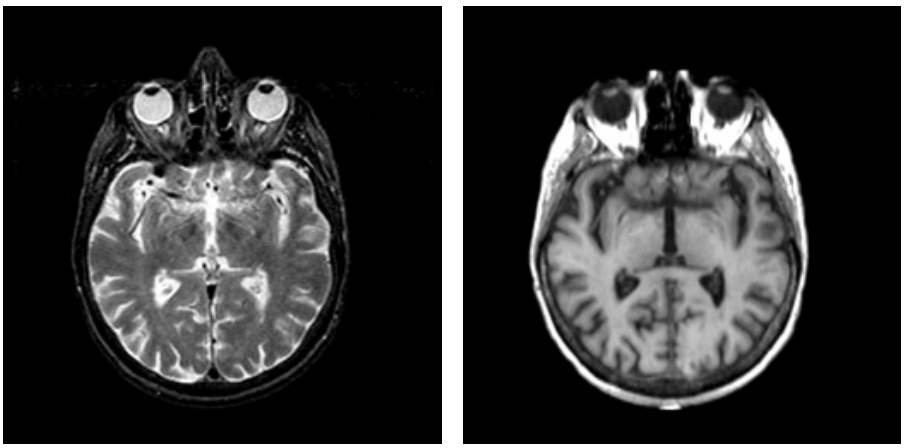
   A feature point is represented using image information within its neighborhood, so the representation is commonly called a local descriptor. In Sections 2.4 and 2.5, a number of popular techniques will be introduced.

c. Matching feature points

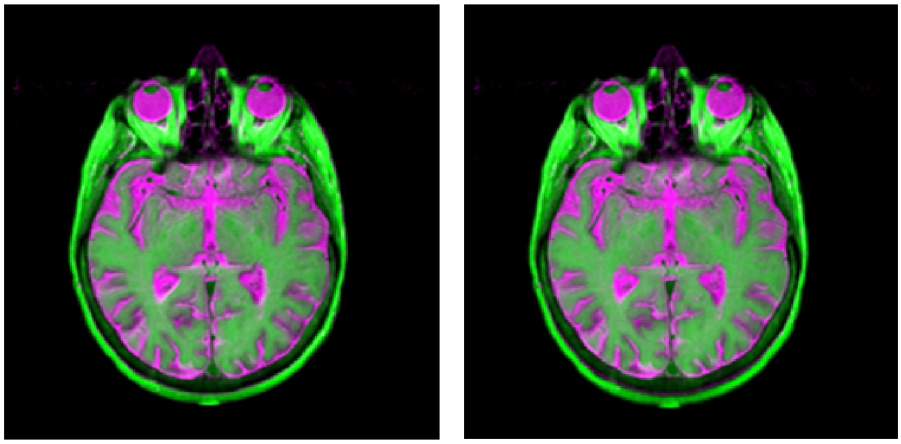
   With local descriptors for any two feature points, the distance between them can be computed. In accordance with a specific matching criterion, all the feature points are matched. In Section 2.6, techniques for feature matching will be summarized.

d. Refining matches and estimating an image transformation

To estimate an image transformation, the refinement of matches or outlier removal is necessary. The refined matches are used to estimate an image transformation, thereby aligning the reference and target images. Refining matches and estimating a transformation will be elaborated in Section 2.7.



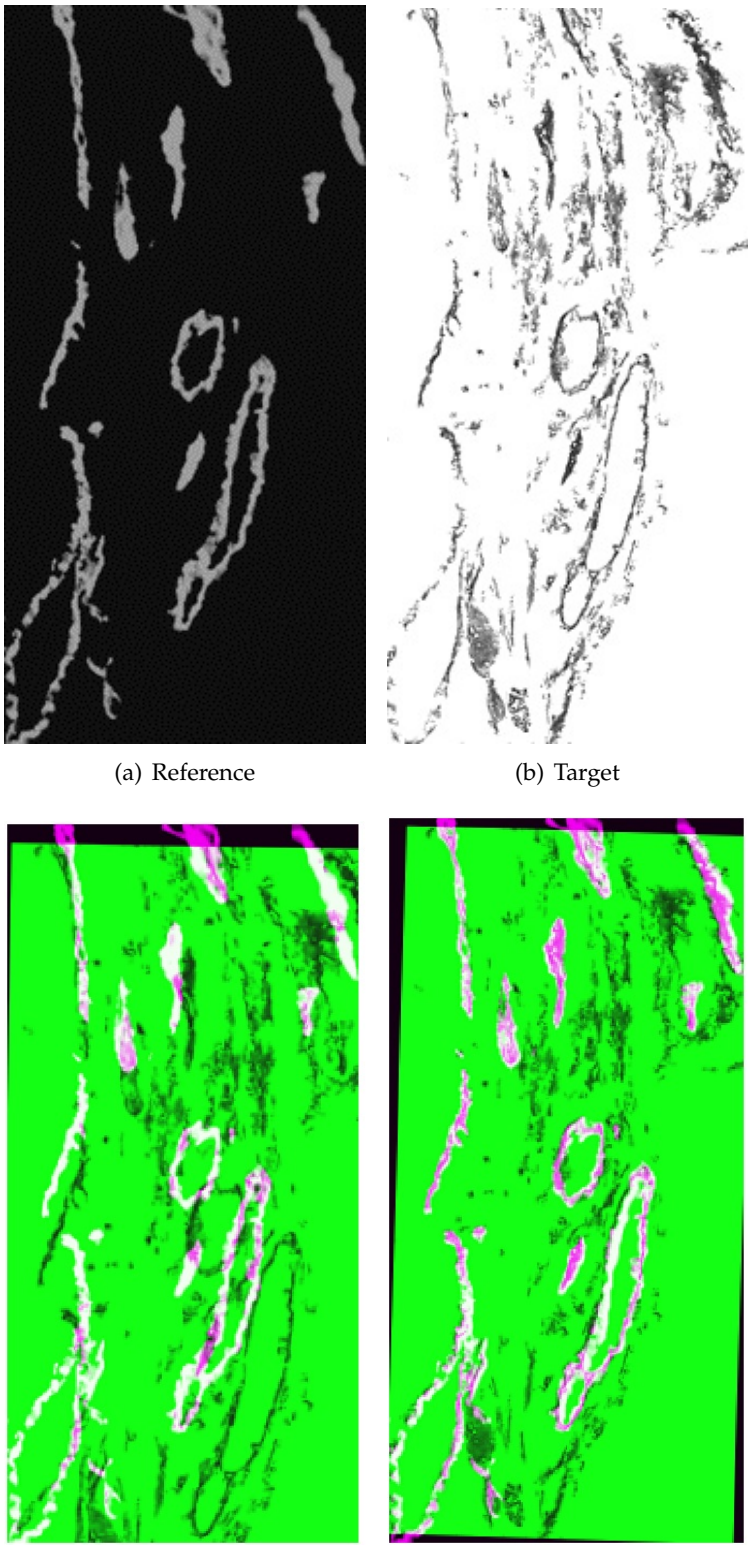
(a) Reference (b) Target



(c) Alignment Using Intensity-based Technique [45] (d) Alignment Using Feature-based Technique [49]

Figure 2.2: Comparing Alignment from Intensity-based and Feature-based Image Registration Techniques (First Example)

Here, we preliminarily compare intensity-based and feature-based registration techniques. As a benchmark intensity-based image registration technique, elastix [45] is used for performance comparison. For the feature-based registration technique, we use the latest one which was proposed in [49]. Please note that the feature-based tech-



(a) Reference (b) Target

(c) Alignment Using Intensity-based Technique [45] (d) Alignment Using Feature-based Technique [49]

Figure 2.3: Comparing Alignment from Intensity-based and Feature-based Image Registration Techniques (Second Example)

nique [49] will be elaborated in Section 2.5.3. In Figure 2.2 and Figure 2.3, two examples are shown to compare the alignment results between elastix [45] and the feature-based registration technique [49]. In the first example shown in Figure 2.2, both [45] and [49] perform well in registering the two images shown in Figure 2.2 (a) and (b) as the reference and target images are correctly aligned. In the second example shown in Figure 2.3, elastix [45] fails to align the two images shown in Figure 2.3 (a) and (b), as shown in Figure 2.3 (c). In contrast, the alignment error in Figure 2.3 (d) is much smaller, which is achieved by the feature-based registration technique in [49]. In the two examples, elastix is more sensitive to differences in image characteristics as compared to [49]. Compared with the two images in Figure 2.2 (a) and (b), the two images in Figure 2.3 (a) and (b) present larger content differences, which increases the difficulty in achieving effective registration. The larger content differences present in Figure 2.3 (a) and (b) lie in two aspects. First, the pixels in the reference image are spatially close one another, whereas many pixels in the target image are unconnected. Second, the target image presents more intensity variations as compared to the reference image. Note that, a detailed performance comparison between elastix [45] and our proposed feature-based technique will be presented in Section 5.6 of Chapter 5.

Compared with feature-based registration techniques, the limitations of intensity-based registration techniques are as follows. First, intensity-based registration techniques are likely to fail in registering images with large content differences such as Figure 2.3 (a) and (b). Second, many intensity-based image registration techniques are sensitive to local minima of mutual information, and this creates problems in registering image pairs that have low-overlapping [111]. Third, an intensity-based registration technique requires an optimization process of searching for an optimal transformation, as stated in Section 2.2.1. If an initialized transformation is far from the ground-truth one, the computational cost caused by the optimization process can be huge [37]. Thus in this thesis we will focus on feature-based image registration techniques for registering multi-modal images where contents differ a lot. In Sections 2.3 to 2.7, we will review techniques of the four main steps in feature-based image registration.

2.3 Detection of Local Features

In this section, we first review local feature detectors which have been popular in recent years. Second, two typical detectors are elaborated as these two detectors will be used in our proposed techniques in Chapters 3, 4 and 5.

2.3.1 A Review of Local Feature Detectors

Here, we review local feature detectors in three categories: corner detectors, blob detectors and region detectors, similar to the categorization in [105].

i. Corner Detectors

Corner detectors can be classified into three categories: intensity-based, contour-based and model-based detectors [6, 25, 92, 105]. In general, an intensity-based corner detector uses a measure based on intensities or gradients in a neighborhood of an image point to decide whether it is regarded as a corner; a contour-based corner detector extracts contours after detecting edges using an edge detector such as [16] and then searches for curvature maxima; and a model-based corner detector determines corners by comparing image information with a pre-defined model.

Among intensity-based corner detectors, the Harris corner [28] is a well-known one. A Harris corner is an image point where there are significant changes in all directions. In [24], image pixels are classified into three categories: region, contour and interest point by using an auto-correlation matrix. The detector proposed in [95] is based on a tracking algorithm in order to handle changes of interest points over time. Thus, this detector is commonly used in the domain of object tracking. The Harris corner detector [28] was improved to achieve scale invariance in [64]. Another popular corner detector is the Susan detector [97]. The main idea of the Susan corner detector is as follows. For each pixel in an image, a circular neighborhood with a fixed radius is partitioned into *similar* and *dissimilar* categories. A *similar* or *dissimilar* category is dependent on whether a neighboring pixel has similar intensity values with the centered pixel of the

circular neighborhood. Corners are detected at the locations where the number of pixels having similar intensity values with the centered pixel in a neighborhood reaches a local minimum and is below a pre-defined threshold. The FAST corner detector [88, 89] is built on the Susan detector, aiming at improving efficiency. Compared to the Susan detector, FAST only compares pixels within a circle of fixed radius around a point. A second difference is that FAST classifies the compared pixels into *dark*, *similar* and *brighter* subsets, instead of the *similar* and *dissimilar* categories in the Susan detector. With regard to efficiency, FAST is up to 30 times faster than the DoG detector which will be detailed in Section 2.3.2.

With regard to contour-based corner detectors, performance comparisons were carried out in [5]. In [5], 11 contour-based corner detectors are compared, including RJ [87], CSS [67], He & Yung [30], MSCP [116], ARCSS [3], AD [83], Eigenvalue [102], Zhang [114], GCM [115], CPDA [4] and Fast-CPDA [6]. Conclusions made in [5] are summarized as follows. First, Zhang [114], CPDA [4] and Fast-CPDA [6] corner detectors perform better than the others in terms of both accuracy and robustness. Second, in terms of efficiency, the Fast-CPDA corner detector [6] is the fastest of all the corner detectors compared. This detector will be elaborated in Section 2.3.3. As the Fast-CPDA corner detector [6] performs relatively better in accuracy, robustness and efficiency, we will use this detector to detect corners for our proposed technique in Chapter 5.

A model-based corner detector such as [75, 96] fits a region around an image point to a pre-defined model, thereby deciding whether this point is a corner. In [75], the proposed model is based on a function that defines a straight-line edge. A corner is determined if two straight-line edges merge into a single point that creates two homogeneous gray regions with different intensities. A junction model is used in [96]. Note that an L junction in [96] is regarded as a corner.

ii. Blob Detectors

In [51], blobs are defined as bright regions on dark backgrounds, or vice versa. In other words, a blob is an image region in which all points are in a sense similar to each other. There are five popular blob detectors: Hessian [91], Hessian-Laplace/Affine [65], Salient Regions [41], DoG (Difference-of-Gaussians)

[56] and SURF [9, 10].

The main ideas in each of the five blob detectors [9, 41, 51, 56, 65] are briefly summarized as follows. The Hessian detector [91] detects blob-like structures based on the determinant of the Hessian matrix:

$$H = \begin{bmatrix} I_{xx}(x,y,\sigma) & I_{xy}(x,y,\sigma) \\ I_{xy}(x,y,\sigma) & I_{yy}(x,y,\sigma) \end{bmatrix}, \tag{2.7}$$

where I_{xx}, I_{xy} and I_{yy} are second-order Gaussian-smoothed image derivatives, and σ is a factor for Gaussian smoothing. Hessian-Laplace and Hessian-Affine [65] aim to achieve scale and affine invariance respectively. Rather than using image derivatives, the Salient Regions detector [41] is motivated by information theory. In the Salient Regions detector, blobs are detected in terms of saliency that is measured using the entropy of PDF (Probability Distribution Function) of intensity values within a local region. To achieve scale invariance, an additional criterion called *self-dissimilarity* is proposed, which is defined as the derivative of probability distribution with regard to scale. The entropy of probability distribution function and the *self-dissimilarity* function are then multiplied as the saliency measurement of an image region. The three blob detectors [41, 51, 65] discussed above are computationally expensive due to computations of derivatives or entropy at each image location. DoG [56] and SURF [9, 10] have been developed mainly for improving efficiency. DoG [56] approximates the Laplacian and SURF [9, 10] approximates the Hessian matrix by using integral images. More details for DoG and SURF can be found in Sections 2.3.2 and 2.4.1.1 respectively.

iii. Region Detectors

Compared to blob detectors, region detectors determine feature windows based on their boundary, and are therefore related to image segmentation techniques [21]. There exist three popular region detectors: Maximally Stable Extremal Region (MSER) [62], Intensity-Based Regions (IBR) [103, 104] and superpixels [69, 86]. MSER was proposed in [62] to establish correspondences between a pair of images taken from different viewpoints. A MSER is a connected component of

a thresholded image. The *Extremal* in MSER means that all the pixels inside the MSER have either higher or lower intensity than those pixels outside the MSER. The *Maximally Stable* in MSER refers to the optimization in the process of selecting an appropriate threshold. The MSER detector has been widely used in object recognition. One limitation of MSER is, as pointed out in [63], that MSER is very sensitive to changes in blur, which is because the segmentation process is less accurate as region boundaries become smooth.

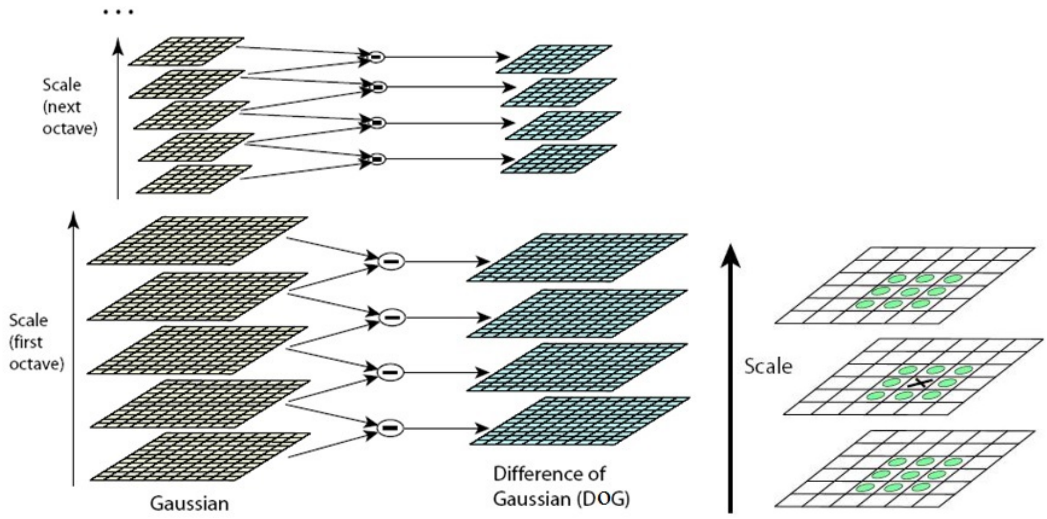
IBR [103, 104] detects image regions as follows. First, local extrema of image intensities are detected over multiple scales. Second, given a local extremum in intensity, an intensity function is defined to evaluate intensity changes along rays radially emanating from the extremum. A maximum of the intensity function is reached at positions where image intensity suddenly increases or decreases. Accordingly, a maximum is determined along each ray. All points corresponding to maxima of the intensity function are linked to enclose an irregularly-shaped region. Third, the irregularly-shaped region is fitted to an elliptical region.

Superpixels [103, 104] is a segmentation-based technique for detecting regions. Superpixels are produced by applying Normalized Cuts [94] which is a classical image segmentation technique. The segmented regions are uniform or very similar in intensities. Superpixels has been used successfully for modeling and exploiting mid-level visual cues [105]. However, superpixels are not suited for the purpose of image registration as the uniform regions are far from discriminative.

2.3.2 DoG Keypoints

To detect image locations that are invariant to scale changes, stable local features are searched across various scales using a continuous function of scale which is known as scale space. The scale space of an image is defined using the Laplacian of Gaussian (LoG) as

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y), \tag{2.8}$$



(a) DoG Pyramid (b) DoG Extrema

Figure 2.4: Difference-of-Gaussian (DoG) Pyramid and Extrema Selection

where I denotes the original image, $*$ is the convolution operation, and σ is variance of the Gaussian function which is defined as

$$G(x,y,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x^2+y^2)/2\sigma^2}. \tag{2.9}$$

To efficiently detect stable features in scale space, the Difference-of-Gaussian (DoG) function is used [55, 56]. The DoG function is derived by subtracting two LoG functions with nearby scales

$$\begin{aligned} D(x,y,\sigma) =& L(x,y,k\sigma) - L(x,y,\sigma) \\ =& (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y), \end{aligned} \tag{2.10}$$

where k is a constant factor for separating two adjacent scales. With a series of scale pre-defined, DoG images are generated as illustrated in Figure 2.4 (a).

In order to detect the extrema, each pixel is compared to its 26 neighbors in 3×3 regions at the current and adjacent scales [56]. If the pixel is the maximum or minimum among the neighboring pixels, it is a keypoint candidate, as shown in Figure 2.4 (b). Each keypoint detected has its own location and scale.

2.3.3 A Contour-based Corner Detector (Fast-CPDA)

As discussed in Section 2.3.1, the Fast-CPDA corner detector [6] outperforms the other contour-based corner detectors. This corner detector is an improved version of the CPDA detector [4] which is based on the Chord-to-Point Distance Accumulation (CPDA) technique [27]. The corner detector [6] is called Fast-CPDA as efficiency is improved over [4].

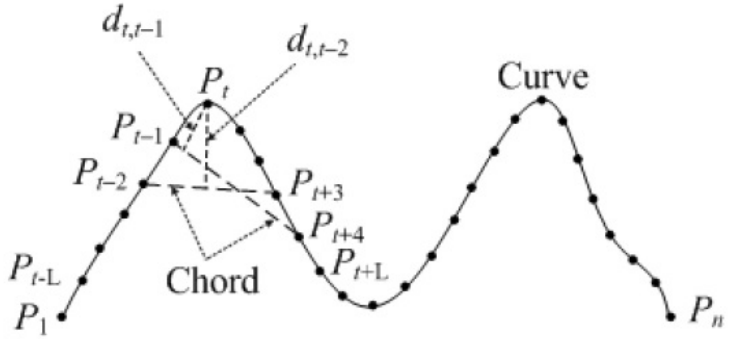


Figure 2.5: Illustrating How CPDA Works

As a contour-based corner detector, the Fast-CPDA corner detector firstly extracts contours from the edge image detected by the Canny edge detector [16]. Each contour is smoothed and the CPDA technique [27] is used to estimate curvatures of contour points. The contour points which correspond to the maxima of curvatures are treated as candidate corners. Herein, we summarize major steps in the Fast-CPDA corner detector [6] and the improvement over its original version [4] in terms of efficiency, as follows.

i. Extracting and Selecting Contours

Given a gray-scale image, the Canny edge detector [16] is used to detect edges. With the assumption that a very short contour might not contain strong corners, the length of a contour, n, should satisfy the condition

$$n > (w+h)/\alpha, \tag{2.11}$$

where w and h are the width and height of the image, and α is a parameter for controlling the length of contours.

ii. Smoothing Contours

To reduce the effects of noises on contours, all selected contours are smoothed using Gaussian convolution. A contour, $\Gamma(t) = (x(t), y(t))$, is smoothed by

$$\Gamma(t,\sigma) = \Gamma(t) * G(t,\sigma) = (x(t) * G(t,\sigma), y(t) * G(t,\sigma)), \tag{2.12}$$

where $\Gamma(t,\sigma)$ is the contour after being Gaussian-smoothed, $*$ is the convolution operation, and $G(t,\sigma)$ is the Gaussian function which is defined as

$$G(t,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-t^2}{2\sigma^2}}, \tag{2.13}$$

where σ is a scaling factor for smoothing.

iii. Estimating Curvatures

Figure 2.5 illustrates how the CPDA curvature estimation technique [27] works, which is as follows. As a chord moves along a contour, the perpendicular distances from point P_t to the chord are accumulated to represent the curvature at point P_t. With a chord of length L, the curvature for point P_t is estimated by

$$K_L(t) = \sum_{j=t-L+1}^{t-1} d_{t,j}, \tag{2.14}$$

where j is the index of the first intersected point between the moving chord and the contour, and $d_{t,j}$ denotes the distance between P_t and the moving chord.

In the CPDA detector [4], curvatures are estimated using chords of three different lengths and are then normalized as

$$K_j^{'}(t) = \frac{K_j(t)}{max(\left|K_j\right|)}, \tag{2.15}$$

where $1 \leq t \leq n$ and $1 \leq j \leq 3$. Finally, the three normalized curvatures are multiplied to determine a single curvature value by

$$K(t) = K_1^{'}(t) K_2^{'}(t) K_3^{'}(t), 1 \leq t \leq n. \tag{2.16}$$

iv. Refining Candidate Corners

By finding the local maxima of curvatures computed in Equation 2.16, candidate corners are decided. Candidate corners include strong, weak (also known as *round* [4, 30, 67]) and false corners. The weak and false corners are filtered out by applying curvature and angle thresholds [4].

v. Determining Final Corners

Apart from the corners that have been determined in Step iv, there may be a corner at the two ends of a closed contour. Such a corner is detected by estimating the angle at the end of a closed contour. Finally, all the corners are detected.

The computational cost of the CPDA detector [4] is high due to the following two reasons. First, the CPDA curvature estimation [27] is an expensive operation. Second, the CPDA detector [4] estimates a curvature value at each point of a given contour. Thus, the Fast-CPDA corner detector [6] aims to improve the efficiency of the original CPDA detector [4].

To reduce the time complexity of the original CPDA detector, a subset of all contour points are selected before the CPDA curvature estimation in [6]. The guideline in selecting contour points is that a contour segment with significant direction changes is more affected in the process of being Gaussian-smoothed as compared to a relatively straightforward contour segment. In other words, the distance from a point on the original contour to its location on the smoothed contour is relatively large, if this point is a corner or spatially close to a corner. In the Fast-CPDA corner detector [6], a distance function was established for the point-to-point distances between the original contour and its smoothed one. The maxima of the distance function is regarded as the candidate points before the CPDA curvature estimation in [6].

2.4 Mono-modal Image Registration Techniques

In this section, popular mono-modal image registration techniques are briefly reviewed. These techniques are summarized in two categories according to image features, i. e., gradient or binary features.

2.4.1 Techniques based on Gradient Features

2.4.1.1 SIFT and its Variants

SIFT (Scale Invariant Feature Transform) is one of the most popular techniques for detecting and describing local features in the past decade, which has been widely used in the community of computer vision. Here, SIFT and its variants are summarized.

i. SIFT [56]

The major stages of SIFT include:

a. Keypoint detection

With an initial image, a pyramid of Difference-of-Gaussian (DoG) images is generated. These DoG images represent images of various scales. In these DoG images, local maxima or minima are detected by comparing each point with its neighbors in the current DoG image and the two adjacent DoG images. An image point detected from DoG images is called a keypoint. For each keypoint, a main orientation is assigned. The main orientation is computed from a gradient histogram which is built in a local region centered at the keypoint. The frame of a keypoint includes its location, scale and orientation.

b. Assigning the main orientation for each keypoint

By assigning the main orientation for each keypoint, the keypoint descriptor can be represented relative to this orientation, thereby achieving invariance to image rotation. Firstly, the Gaussian smoothed image, $L(\sigma)$, is selected with the closest scale of the keypoint so that all computations are performed in a scale-invariant way. Then, a local region around the keypoint is determined by a Gaussian-weighted circular window that is derived based on the scale of the keypoint. The gradient magnitude, G_m, and orientation, G_θ, for each pixel, $L(x,y)$, within this region are calculated [56]:

$$\begin{aligned} G_m &= \sqrt{{d_x}^2 + {d_y}^2}, \\ G_\theta &= tan^{-1}(d_y/d_x), \end{aligned} \tag{2.17}$$

where

$$\begin{aligned} d_x &= L(x+1,y) - L(x-1,y), \\ d_y &= L(x,y+1) - L(x,y-1). \end{aligned} \tag{2.18}$$

Then, an orientation histogram consisting of 36 bins (covering 360° with an interval of 10°) is built based on the gradient orientations of all the pixels within the local region. The value in each orientation bin is incremented based on the gradient magnitude (weighted by the Gaussian window) of each pixel with a corresponding orientation in the neighboring region. The orientation bin with the highest value denotes the main orientation.

c. keypoint description

As illustrated in Figure 2.6, a SIFT descriptor is built as follows.

i. The gradient magnitude and orientation for each pixel located in this region are calculated.

ii. The gradient orientations are rotated relative to the main orientation of the keypoint to attain rotation invariance.

iii. All the gradients are weighted by a Gaussian window (indicated by a circle in Figure 2.6), giving less emphasis to gradients that are further from the center of the region.

iv. The local region is divided into 4×4 spatial bins and an orientation histogram is built in each spatial bin, with eight orientation bins evenly covering 360°(quantized with an interval of 45°).

v. The descriptor is normalized to reduce the effects of illumination changes.

d. keypoint matching

Firstly, a Euclidean distance is used to measure the similarity between two descriptors from the reference and target images. Relative to a descriptor from the reference image, all the descriptors from the target image are ranked by the descriptor distances. To ensure the distinctiveness of matches, a distance ratio is set between the closest neighbor and the second closest neighbor. A match that satisfies this constraint is decided as a SIFT match. As a result, a set of keypoint matches is generated.

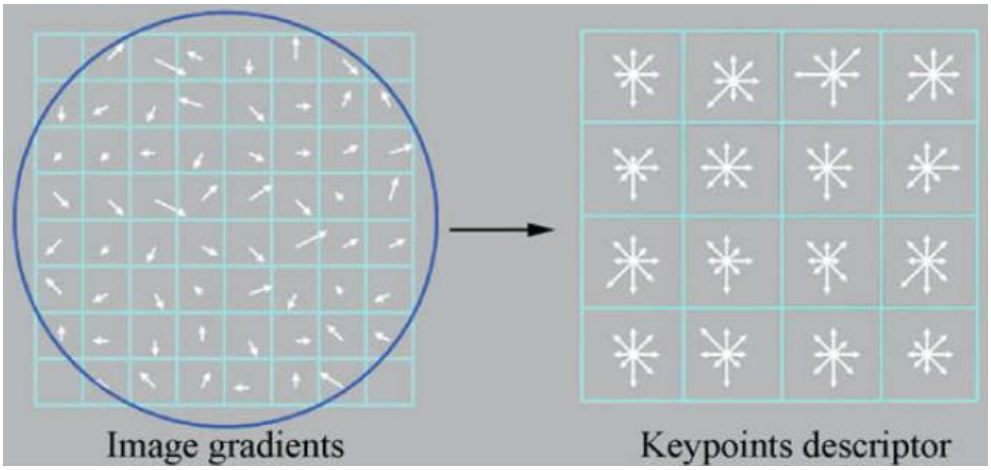


Figure 2.6: Building the SIFT Descriptor

e. Transformation estimation and alignment

Keypoint matches are refined by a technique for estimating and removing outliers such as RANSAC [23]. RANSAC will be described in Section 2.7. The pairs of keypoints in the refined keypoint matches are used to infer a transformation. Finally, the transformation is used to align the reference and target images.

ii. PCA-SIFT [43]

Different from SIFT, PCA-SIFT (PCA: Principal Component Analysis) encodes salient parts of image gradients within the local region centered at a keypoint. Firstly, the PCA-SIFT descriptor is a vector of image gradients in the horizontal and vertical (x and y axis) directions within this local region. Secondly, this vector is sampled at 39 $\times$ 39 locations. Thus, the vector consists of 3042(=39 $\times$ 39 $\times$ 2) elements. The dimension of this vector is then significantly reduced using PCA, leading to a much more compact feature representation. Based on the experiments in [43], the best registration performance is achieved when the dimension equates to 36. Moreover, PCA-SIFT is more discriminative as compared to the standard SIFT, which is primarily due to discarding the lower components in PCA. However, a major limitation of PCA-SIFT is that PCA is sensitive to noise. More specifically, principle subspaces in PCA may significantly change due to noise [117]. Thus, the discrimination of PCA-SIFT is likely to

decrease significantly in registering image pairs with a large amount of noise.

iii. SIFT+GC [70]

SIFT performs well in the scenarios where local regions surrounding keypoints are quite unique from the rest of local regions in images. However, SIFT cannot work well in scenarios where there exist multiple similar regions across an image. To address this issue, a SIFT descriptor with global context (called SIFT+GC) was proposed. SIFT+GC appropriately incorporates the SIFT descriptor with curvilinear shape information from a much larger local region, thereby reducing mismatches caused by multiple similar SIFT descriptors. Firstly, centered at a keypoint, a large circled region is defined by setting its diameter equivalent to the image diagonal. This region is then divided in a 5×12 log-polar coordinate system. Within this region a shape context descriptor is built by concatenating curvature values of all the pixels in each spatial bin. Note that, the curvature value of a given pixel is the absolute eigenvalue of the Hessian matrix. Finally, the shape context descriptor and its corresponding SIFT descriptor are weighted (the weighting factor is tentatively used and 0.5 is generally optimal) to form the SIFT+GC descriptor, which is $128+5 \times 12=188$ dimensional.

iv. GLOH [66]

GLOH (Gradient Location-Orientation Histogram), as an extension of SIFT, was designed to increase SIFT's robustness and distinctiveness. The GLOH descriptor is built in a log-polar grid with 17 location (spatial) bins. In each location bin, an orientation histogram consisting of 16 orientation bins is built. Then, a 272 ($=17\times16$) dimensional descriptor is formed. The dimensionality is reduced to 128 using PCA (Principal Component Analysis). Experiments in [66] show that GLOH outperforms SIFT in most cases while registering mono-modal image pairs from the commonly-used Affine Covariant Regions Datasets.

v. SURF [9, 10]

Speeded Up Robust Features (SURF) was proposed in the context of the high dimensionality of the SIFT descriptor, in order to reduce the time complexity for feature description and matching. In the stage of feature detection, various

sizes of Haar wavelet filters are convolved with the integral images instead of the original image convolved with a variable-scale Gaussian in SIFT. In the stage of feature description, a local region surrounding an interest point is divided into 4×4 sub-regions as in SIFT. In each sub-region, a descriptor is built, with four components: the sum of wavelet responses in the horizontal and vertical directions as well as the sum of absolute values of wavelet responses in the two directions. Hence, the SURF descriptor is 64 ($=4 \times 4 \times 4$) dimensional. Experiments in [9,10] show that SURF outperforms SIFT [56], PCA-SIFT [43] and GLOH [66] in the application of object recognition.

vi. ASIFT [68,112]

The assumption in ASIFT is that SIFT is fully invariant to changes in scale, rotation and translation, which covers four parameters out of six in affine transformations. Thus, ASIFT is focused on the other two transformation parameters that are associated with the angles defining the camera axis orientation. Firstly, all possible views are simulated with two variables: the horizontal angle and the vertical angle. Secondly, SIFT is used to detect and describe features from these simulations. As a result, the robustness to viewpoint changes is significantly improved as compared to SIFT. Thus, this method is called Affine-SIFT (ASIFT) as it claims to be fully affine invariant.

vii. SIFT Flow [52]

Inspired by optical flow [35, 57], SIFT Flow was proposed to align an image to its nearest neighbors in a large image database which contains a wide variety of scenes. In SIFT Flow, a SIFT descriptor is built at each pixel to capture local image structures and contextual information. In [52], the SIFT Flow algorithm was tested in applications such as motion field prediction from a single image, motion synthesis via object transfer, satellite image registration and face recognition. In registering satellite images, SIFT Flow outperforms the original SIFT.

viii. Edge-SIFT [113]

Edge-SIFT [113] aims to enhance the efficiency and discriminative power of the SIFT descriptor in mobile image search. The proposed Edge-SIFT descriptor

is formed as follows. First, keypoints are detected as in SIFT. A scale and main orientation are assigned to each keypoint. Second, this image patch is normalized to achieve rotation and scale invariance. The image patch is rotated to a fixed orientation from its main orientation, and is then resized into a fixed size (D×D). Next, edges are extracted in the normalized image patch and sub-edge maps are achieved according to the quantized edge orientations. Finally, each sub-edge map is regarded as a binary local descriptor and a D×D×O dimensional descriptor is constructed for this keypoint. Compared to SIFT, Edge-SIFT is more discriminative and efficient in the application of partial-duplicate mobile search.

2.4.1.2 GDB-ICP

In [111], an image registration framework called Generalized Dual-Bootstrap Iterative Closest Point (GDB-ICP) was proposed. The framework includes three primary components: the initialization algorithm, the estimation technique and the decision criteria. The three components are briefly summarized as follows.

The initialization algorithm is based on SIFT [56]. Keypoints are detected and matched as in SIFT. All the matches are sorted by the distance ratio between the closest and second closest matches, where top matches are selected. Each match is used to generate an initial bootstrap region which is centered at the keypoint from each of the two registered images. For each match, a similarity transformation is initialized.

The estimation technique is an iterative process for estimating a transformation between two images. The estimation works on the initial bootstrap regions and associated transformations. At each iteration, a new transformation is estimated and the bootstrap region is expanded till the region covers the entire overlap between two images.

The decision criteria determines whether an estimated transformation is accepted as correct or not. The decision is made based on three measurements including alignment accuracy, stability in the estimated transformation and consistency in geometric constraints.

One assumption in GDB-ICP is that there is at least one true match after matching SIFT descriptors. However, this assumption is not true in some difficult cases, as

pointed out in [17]. A second limitation is the poor applicability of GDB-ICP to registering multi-modal images, as pointed out in [37].

2.4.1.3 WLD: Weber Local Descriptor

WLD was proposed in [18] in accordance with the fact that human perception of a pattern depends not only on the change of a stimulus such as sound or lighting, but also on the original intensity of the stimulus. The stimulus refers to image intensities in building a WLD descriptor. Two components of a WLD descriptor are differential excitation and orientation. The differential excitation is a function of the ratio between two terms: one is the relative intensity differences of a current pixel against its neighbors, and the other is the intensity of the current pixel. The orientation is the gradient orientation of the current pixel. Experiments in [18] show that WLD outperforms SIFT and LBP [74] in face detection. LBP will be introduced in Section 2.4.2.1.

2.4.2 Techniques based on Binary Features

2.4.2.1 LBP and its Variants

In [74], a local image descriptor called Local Binary Pattern was proposed. LBP is one of the simplest texture descriptors. The LBP operator compares intensities of pixels in a circular neighborhood of radius R with the intensity of the central pixel. An array of binary codes is generated by following

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}, \tag{2.19}$$

where x is the difference between the intensity of a neighboring pixel and the intensity of the central pixel. The LBP value is then calculated by

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p, \tag{2.20}$$

where P and R are the number of neighboring pixels and radius from the central pixel respectively, and g_c and g_p are the intensity of the central pixel and the intensity of a neighboring pixel respectively. According to Equation 2.20, the LBP operator determines 2^P different patterns.

In [31], a modification was made on LBP and the proposed texture descriptor is called Center-Symmetric Local Binary Pattern (CS-LBP). Instead of comparing neighboring pixels with the central pixel in terms of intensity values, CS-LBP compares center-symmetric pairs of pixels in a neighborhood. CS-LBP halves the number of comparisons in LBP. Moreover, CS-LBP combines the strengths of SIFT and LBP by using a SIFT-like grid and replacing SIFT's gradient features with LBP-based features. Experiments in [31] show that CS-LBP outperforms SIFT in registering mono-modal images from the data sets [80].

A second variant of LBP is Local Relational String (LRS), which was proposed in [26] for image retrieval. As pointed out in [37], there are two differences between LRS and LBP. First, the number of neighboring pixels is kept to four regardless of the radius from the central pixel. Second, LRS differentiates three cases $>$, $=$ and $<$ in comparing a neighboring pixel and the central pixel in terms of intensity, which is different from the two cases in LBP as shown in Equation 2.19. The discrimination of LRS is insufficiently high as the LRS operator determines $3^4 (= 81)$ different patterns.

Local Triplet Pattern (LTP) was proposed in [29], which was also motivated by LBP. Like LRS, LTP also differentiates three cases $>$, $=$ and $<$ in comparing a neighboring pixel and the central pixel in terms of intensity. According to the three cases $>$, $=$ and $<$, triplet codes 2, 1, 0 are accordingly generated. But, a 3×3 neighborhood is used in LTP, which is different from the four neighboring pixels in LRS. Thus, the LTP operator determines $3^8 (= 6561)$ different patterns. Consequently, efficiency becomes a problem. To improve efficiency, a neighboring parameter was introduced to limit the neighborhood size base on the LTP value. Experiments in [29] show that LTP is promising for image classification and retrieval.

2.4.2.2 BRIEF and its Variants

In [15], a local feature descriptor called Binary Robust Independent Elementary Features (BRIEF) was proposed. The rationale behind the BRIEF descriptor is that image patches can be classified or differentiated on the basis of a relatively small number of pairwise intensity comparisons. With this rationale, intensity comparisons are carried out between sampled test points in a smoothed image patch, and binary codes are accordingly generated. It is concluded in [15] that BRIEF achieves similar or better recognition performance and is much faster as compared to SURF [9,10]. In [90], an improvement was made over BRIEF and a binary descriptor called ORB (Oriented FAST and Rotated BRIEF) was proposed. The ORB descriptor is built on the FAST [89] detector and BRIEF descriptor. The novelties of ORB lie in two aspects. First, ORB adds a fast and accurate orientation component to the FAST detector. Second, rotation invariance is achieved. Experiments in [90] show that ORB is much faster and achieves similar performance in image matching as compared to SIFT. However, ORB cannot achieve scale invariance.

2.4.2.3 BRISK

Binary Robust Invariant Scalable Keypoints (BRISK) was proposed in [47] for keypoint detection, description and matching. Compared to BRIEF [15] and ORB [90], there are two main differences as follows. First, to achieve scale invariance, BRISK detects FAST keypoints [89] in a scale-space pyramid. Second, sample points for intensity comparisons are equally located on concentric circles, which is different from a random sampling pattern used in BRIEF [15] and ORB [90]. Experiments in [47] show that BRISK achieves a comparable performance in registering mono-modal images from the data sets [80], as compared to SIFT and SURF.

2.4.2.4 FREAK

Inspired by the retina, Fast Retina Keypoint (FREAK) was proposed in [2] as a binary descriptor. The key difference from BRIEF [15] and ORB [90] and BRISK [47] lies in the sampling pattern for intensity comparisons. BRIEF [15] and ORB [90] use random

point pairs. BRISK [47] uses a circular pattern where sampled points are equally located on concentric circles. FREAK uses the retinal sampling pattern which is also circular but has more points near the center of a neighborhood. Experiments in [47] show that FREAK outperforms SIFT, SURF and BRISK [47] in registering mono-modal images from the data sets [80].

2.5 Multi-modal Image Registration Techniques

Images captured by different types of imaging modalities or capturing devices are known as multi-modal images. Registering multi-modal images is more challenging than registering mono-modal images due to the fact that the content differences between corresponding parts in two images can be substantial.

2.5.1 Gradient based Techniques

2.5.1.1 Multi-modal Variants of SIFT

i. SIFT-GM and SIFT-GMEP [44]

In [44], substantial and non-linear intensity variations across multi-modal images are investigated. Two characteristics of multi-modal images are taken into account. The first characteristic is *Gradient Mirroring*, that is, reversed image contrasts might appear at corresponding locations of multi-modal images. The second characteristic is called *Edge Precursors*, based on the assumption that the image information preserved across different modalities and strong illumination changes is primarily along the boundaries. With the two characteristics, modifications are made on the original SIFT. For the first characteristic, gradient orientations are restricted to $[0, \pi)$ instead of $[0, 2\pi)$ used in the original SIFT. For the second characteristic, only edge pixels around a keypoint are used to compute a local descriptor rather than using all the pixels as in the original SIFT. In modifying SIFT, SIFT-GM only considers the first characteristic, while SIFT-GMEP takes both characteristics into account. Experiments in [44] show that both SIFT-GM and SIFT-GMEP outperform SIFT in registering multi-modal images.

ii. SSIFT [19]

As pointed out in SSIFT (Symmetric SIFT) [19], gradient orientations of corresponding points across multi-modal images may point to opposite directions. This problem is also discussed in [37,38] and is called *gradient reversal*. For the referencing purpose, we use the same term in this thesis. To address *gradient reversal*, SSIFT is different from SIFT in two steps: assigning the main orientation for each keypoint and building descriptors. Note that a keypoint is associated with a local region and its size is determined by a scaling factor σ, which is carried out in the feature detection stage as in SIFT.

i. Assigning the main orientation for each keypoint

Different from SIFT, SSIFT introduced a new strategy of assigning main orientation for each keypoint, called *Gaussian-weighted average square gradients*. An assigned orientation is continuous, as compared to quantized or discrete orientations in SIFT.

Given a keypoint (x,y), its main orientation is assigned as follows. First, for each image pixel in a local region, the *squared gradient* is computed as

$$\begin{bmatrix} G_{sx}(x,y) \\ G_{sy}(x,y) \end{bmatrix} = \begin{bmatrix} G_x^2(x,y) - G_y^2(x,y) \\ 2G_x(x,y)G_y(x,y) \end{bmatrix}, \tag{2.21}$$

where $\begin{bmatrix} G_x(x,y) & G_y(x,y) \end{bmatrix}^T$ is the image gradients at x and y directions as

$$\begin{bmatrix} G_x(x,y) \\ G_y(x,y) \end{bmatrix} = sgn(\partial L(x,y)/\partial y) \begin{bmatrix} \partial L(x,y)/\partial x \\ \partial L(x,y)/\partial y \end{bmatrix}, \tag{2.22}$$

where $L(x,y)$ denotes image intensity at (x,y) in the Gaussian-smoothed or Laplacian-of-Gaussian (LoG) image which has been stated in Section 2.3.2. Next, the *Gaussian-weighted average square gradients* can be computed by

$$\begin{bmatrix} \overline{G_{sx}} \\ \overline{G_{sy}} \end{bmatrix} = \begin{bmatrix} G_{sx} * h_{\sigma'} \\ G_{sy} * h_{\sigma'} \end{bmatrix}, \tag{2.23}$$

where $h_{\sigma'}$ is the Gaussian-weighted kernel. Note that, σ' is 1.5 times the scale

of the keypoint σ, which determines the window size for Gaussian weighting. Lastly, the main orientation of the keypoint, $\varphi(x,y)$, is assigned by

$$\varphi(x,y) = \begin{cases} tan^{-1}(\overline{G_{sy}}/\overline{G_{sx}}) + \pi, & if \quad \overline{G_{sx}} \geq 0; \\ tan^{-1}(\overline{G_{sy}}/\overline{G_{sx}}) + 2\pi, & if \quad \overline{G_{sx}} < 0 \cap \overline{G_{sy}} > 0; \\ tan^{-1}(\overline{G_{sy}}/\overline{G_{sx}}), & if \quad \overline{G_{sx}} < 0 \cap \overline{G_{sy}} \leq 0. \end{cases} \tag{2.24}$$

ii. Building the SSIFT descriptor

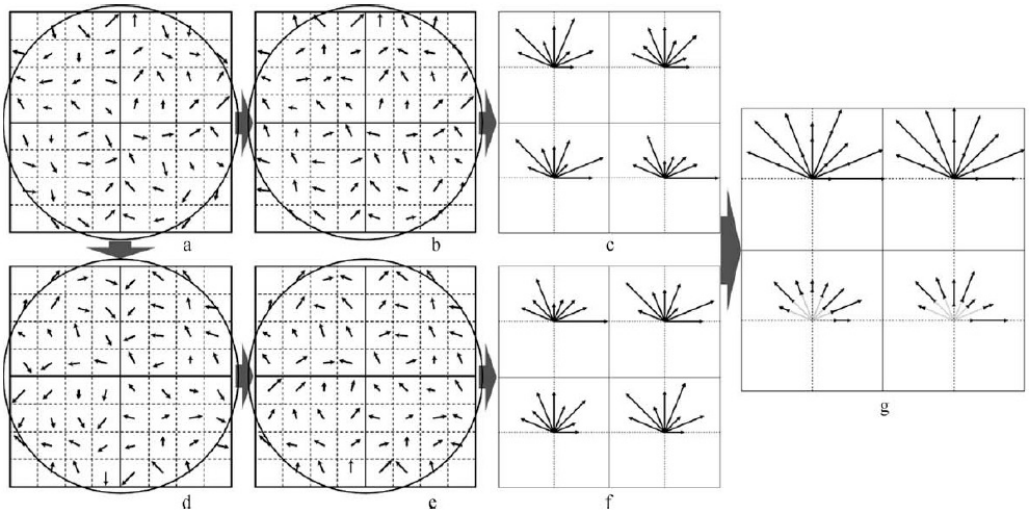


Figure 2.7: Building the SSIFT Descriptor. (a) The local region around a keypoint with gradient magnitudes and orientations; (b) All the gradient orientations in (a) are restricted in [0,π); (c) The orientation histogram corresponding to (b); (d-f) The corresponding operations with (a-c) by rotating 180° on the original region; (g) The final orientation histogram by combining the two histograms (c) and (f).

As illustrated in Figure 2.7, the process of building a SSIFT descriptor is summarized as follows. Note that, there are $4 \times 4 = 16$ spatial bins in a local region of a keypoint. Using $2 \times 2 = 4$ spatial bins in in Figure 2.7 is only for the purpose of illustration. First, the gradient magnitude and orientation are calculated for each pixel in a local region surrounding a given keypoint. Second, all the gradient orientations are restricted between 0 and π. Third, the operations in the first step are implemented in a local region that is rotated by 180° on the original local region. Next, orientation histograms are built for the original region and the rotated region separately. Finally, the two orientation

histograms are combined as

$$C(i,j,k) = \begin{cases} c_1 \left| A(i,j,k) + B(i,j,k) \right|, & i = 1,2; \\ c_2 \left| A(i,j,k) - B(i,j,k) \right|, & i = 3,4; \end{cases} \tag{2.25}$$

where i and j are the horizontal and vertical indexes of spatial bins of a descriptor respectively, and k is the index of orientation bins of a descriptor ($1 \leq i,j \leq 4$ and $1 \leq k \leq 8$), so $A(i,j,k)$ and $B(i,j,k)$ denote the gradient magnitudes at the k^{th} orientation bin of the (i,j) spatial bin for the original region and the rotated region respectively, c_1 and c_2 are constant factors to tune gradient magnitudes. With Equation 2.25, the descriptor is invariant to *gradient reversal*.

iii. MI-SIFT [58]

As a variant of SIFT, MI-SIFT (MI: Mirror and Inversion invariant) was proposed to achieve invariance to image mirroring and gray-scale inversion. Given a keypoint, SIFT descriptors built in the original, mirrored, grayscale-inverted and mirror&grayscale-inverted images are denoted as f, $f^{'}$, $f^{''}$ and $f^{'''}$, respectively. In order to achieve invariance of image mirroring and/or gray-scale inversion, a merging function is defined as

$$f_{mi} = \begin{bmatrix} \mathcal{A}_{tl} & \mathcal{B}_{tr} \\ \mathcal{C}_{bl} & \mathcal{D}_{br} \end{bmatrix}, \tag{2.26}$$

where the subscript tl, tr, bl and br denote the top-left, top-right, bottom-left and bottom-right quarters of an MI-SIFT descriptor respectively, and $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ are defined as

$$\begin{aligned} \mathcal{A} &= f + f^{'} + f^{''} + f^{'''} \\ \mathcal{B} &= \sqrt{f^2 + f^{'2} + f^{''2} + f^{'''2}} \\ \mathcal{C} &= \sqrt[3]{f^3 + f^{'3} + f^{''3} + f^{'''3}} \\ \mathcal{D} &= \sqrt[4]{f^4 + f^{'4} + f^{''4} + f^{'''4}}. \end{aligned} \tag{2.27}$$

iv. GO-SSIFT [39]

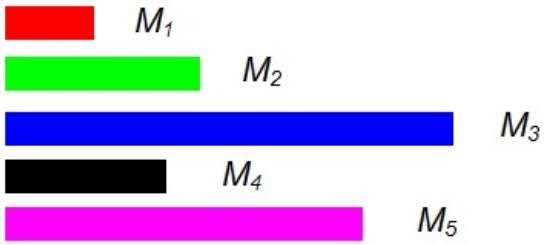


Figure 2.8: Illustrating the Difference between GM and GO for Building Descriptors

Different from SSIFT in building descriptors, Gradient Occurrences (GO) are used to increment the value at each orientation bin of gradient histograms instead of using Gradient Magnitudes (GM). GO is defined as the number of pixels where an image gradient occurs. The difference between GM and GO can be clearly seen through an example shown in Figure 2.8. Suppose the five horizontal bars in Figure 2.8 represent GM of five pixels relative to a particular orientation bin in building a SSIFT descriptor. GM and GO can be very different in calculating the value of the orientation bin. Using GM, the five GM values are summed up, whereas in GO the number of bars or image pixels is counted. As a result, the values of the orientation bin will be $\sum_{i=1}^{5} M_i$ and 5, respectively, using GM and GO. Experiments in [39] show that GO-SSIFT improves matching accuracy as compared to SSIFT [19]. However, with our analysis, we have found that both GM and GO are important gradient information in building SIFT-like descriptors. Regarding GM and GO, a thorough analysis will be made in Chapter 3, and subsequently a better way of utilizing GM and GO will be presented.

v. IS-SIFT

It is pointed out in IS-SIFT that the procedure of combining the descriptors of two reversed regions, stated in Equation 2.25, in SSIFT might cause ambiguities. It means that $A(i,j,k)$ and $B(i,j,k)$ in Equation 2.25 might represent two local regions which are unlikely to be a true match [37,38].

Based on the analysis above, the guideline for IS-SIFT is to avoid the descriptor combining procedure used in SSIFT. The major steps of IS-SIFT are summarized as follows.

a. A set of keypoint matches are determined using SSIFT.

b. A rotation difference between the reference and target images is estimated by averaging the rotation differences in all the keypoint matches. The rotation difference in a keypoint match is equivalent to the difference in main orientations between the two keypoints in this match. The estimated rotation difference is denoted as α.

c. Descriptors are built for the reference and target images. With the estimated rotation difference, the procedure of combining descriptors of two reversed regions is no longer necessary. Thus, only steps (a) to (c) in Figure 2.7 are relevant for building a descriptor.

It should be noted that GO proposed in GO-SSIFT can be incorporated with SIFT and IS-SIFT, due to the fact that GO is simply a way of weighting orientation bins in building SIFT-like descriptors. The two formed techniques are called GO-SIFT and GO-IS-SIFT respectively, and will be referred to in Chapter 3.

2.5.1.2 PIIFD

PIIFD (Partial Intensity Invariant Feature Descriptor) was proposed in [17] for registering multi-modal retinal images. We summarize how it works as follows.

i. Harris corners [28] are detected in the reference and target images.

ii. The main orientation is assigned for each corner.

The orientation histogram used in PIIFD is different from the one used in the original SIFT [56]. Average squared gradients are used in determining the main orientation for a corner. This is because main orientations determined in SIFT might point to unrelated directions in multi-modal images.

iii. A PIIFD descriptor is built using a fixed region surrounding each corner.

The local region is divided into 4×4 sub-regions. In each sub-region, an orientation histogram with eight orientation bins is built. The eight orientations are equally distributed between 0 and π. In each orientation bin, *normalized*

gradient magnitudes are accumulated. Specifically, all gradient magnitudes in the sub-region are ranked and categorized into different levels. Each level of gradient magnitudes is normalized to a particular value. Till now, a descriptor for the corner can be built and we call it an intermediate PIIFD descriptor. An unexpected problem is that the main orientations of a corner and the rotated version of the corner's local region by 180° may point to opposite directions. This problem is also discussed in [37,38] and called *region reversal*. For the referencing purpose, we use the same term in this thesis. To address *region reversal*, a linear combination is performed on two intermediate PIIFD descriptors that are built for a corner's local region and its rotated version by 180°. Finally, the PIIFD descriptor is built.

iv. The PIIFD descriptors in the reference and target images are matched using the bilateral best-bin-first (BBF) algorithm. The bilateral BBF improves the original BBF [11] by excluding cases where two or more descriptors in one image are matched to the same descriptor in another image.

v. False matches are removed using the main orientations of corners and the distance ratio between two matches.

vi. Locations of matches are refined.

Assume that a corner C_t in the target image is matched to C_r in the reference image. This match is denoted as $C_r \mapsto C_t$. An image pixel that is spatially close to C_t might better match C_r. Thus, in a small neighborhood surrounding C_t, PIIFD descriptors are built for each image pixel. If there is an image pixel, C_t', that is closer to C_r than C_t, then the match $C_r \mapsto C_t$ is updated to $C_r \mapsto C_t'$.

vii. A transformation is estimated from the matches determined above in order to align two images.

Based on our analysis, the limitations of PIIFD include:

a. The descriptor is not scale-invariant. The size of a local region for building a PIIFD descriptor is fixed at 40×40 pixels because the scale difference is usually up to 1.5 times in multi-modal retinal image registration PIIFD was designed for. However,

larger scale differences are common in many other multi-modal image registration applications, such as multi-modal microscopic image registration.

b. PIIFD keypoints are sensitive to intensity variations. In terms of robustness to content differences, it is by no means the best choice to use Harris corners as keypoints for building descriptors. This is because Harris corners are detected in a very small neighborhood according to intensity variations.

c. The descriptor is only partially invariant to intensity variations. However, it is common that intensity variations are very substantial in our tested multi-modal microscopic images.

2.5.2 Self-Similarity based Techniques

The self-similarity concept was firstly introduced in [14], although the term self-similarity was not used. In [14], self-similarity is used for image denoising. In specific, the estimated value for a pixel is computed by comparing a small patch centered at the pixel and all the other patches of the same size in the entire image. In the following, we will summarize popular multi-modal image registration techniques based on self-similarity.

2.5.2.1 Local Self-Similarity Descriptor

In [93], the Local Self-Similarity (LSS) descriptor was proposed by investigating internal layouts of local self-similarities. To the best of our knowledge, this is the first local descriptor based on self-similarity. To build an LSS descriptor at a pixel q, the surrounding image patch (the patch size is 5×5) is compared with a larger surrounding region center at pixel q (the region size is 41×41), using the sum of square differences (SSD) in terms of pixel intensities. The distance surface $SSD_q(x,y)$ is normalized and transformed into a 'correlation surface':

$$S_q(x,y) = exp(-\frac{SSD_q(x,y)}{max(var_{noise}, var_{auto}(q))}), \tag{2.28}$$

where var_{noise} is a constant which corresponds to acceptable photometric variations in color, illumination or noise, and $var_{auto}(q)$ takes into account the patch contrast and its

pattern structure, such that sharp edges are more tolerable to pattern variations than smooth patches. The correlation surface $S_q(x,y)$ is then transformed into log-polar coordinates centered at pixel q, and partitioned into 80 bins (4 bins at each of 20 angles). Within the 80 bins, maximal correlation values are selected to achieve insensitivity to small translations, and a vector of selected correlation values is formed. Finally, the vector is normalized to the final descriptor. As concluded in [93], the LSS descriptor outperforms SIFT and GLOH in object detection.

In [53], [93] was improved in two aspects. First, the LSS descriptor is represented in Cartesian coordinates and efficiency is also improved. Second, rather than selecting maximal correlation values to achieve small translation invariance in the LSS descriptor, a histogram representation is used as in the SIFT descriptor, leading to more robust translation invariance. As shown in [53], the improved LSS descriptor achieves favorably comparable performance in registering mono-modal images from the data sets [80].

2.5.2.2 NLSD

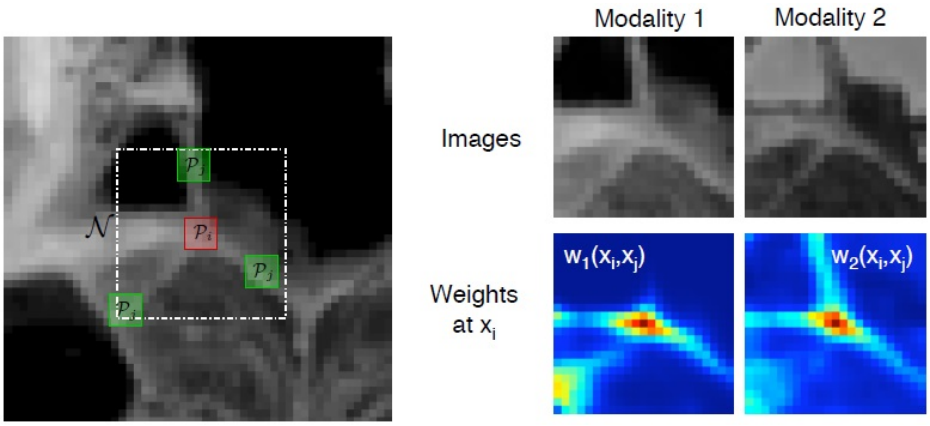


Figure 2.9: Illustration of Building an NLSD Descriptor

A descriptor called Non-Local Shape Descriptor (NLSD) was proposed in [32]. The process of building an NLSD descriptor is briefly summarized as follows. First, given a pixel x_i, a non-local search region $\mathcal{N}$ is defined and divided into a number of small patches with a pre-determined size of radius. As illustrated in Figure 2.9, $\mathcal{N}$ is delimited by dashed white lines, while a red square $\mathcal{P}_i$ and a green square $\mathcal{P}_j$ indicate

the central patch centered at x_i and an exemplary patch respectively. Second, each exemplary patch $\mathcal{P}_j$ is compared with the central patch $\mathcal{P}_i$. The similarity between $\mathcal{P}_j$ and $\mathcal{P}_i$ is calculated based on normalized cross correlation (NCC) [48]. A weight is assigned to each location around the point in the region through an exponentially decaying distance function based on the Euclidean distance, indicating the similarity between an exemplary patch and the central patch. Finally, the NLSD descriptor is built around pixel x_i.

The NLSD descriptor was modified in [33]. The descriptor proposed in [33] is called Modality Independent Neighborhood Descriptor (MIND). In MIND, the main modification over NLSD is that the distance between an exemplary patch $\mathcal{P}_j$ and the central patch $\mathcal{P}_i$ is Gaussian-weighted, so that a relatively higher response is obtained for similar patches. As claimed in [33], MIND is more robust to changes in local noise and contrast. However, a limitation of NLSD and MIND is that neither scale invariance nor rotation invariance is achieved.

2.5.2.3 Structural Representations of Images

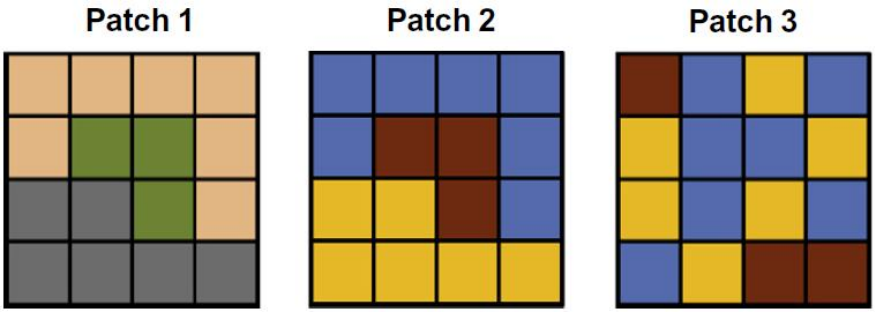


Figure 2.10: Three Patches with Two Different Structural Patterns. Different colors indicate different pixel intensities.

In [106], two approaches for representing local structural patterns were proposed, i.e., entropy and Laplacian images. The basic assumption is that there exist the same or similar structural patterns at corresponding locations in different modalities. Here, we summarize the process of converting a gray-scale image to an entropy image as follows. First, an image patch with a fixed size is defined around a pixel x_i in the gray-scale image. Second, the Probability Density Function (PDF) of pixel intensities

in the patch is computed. Note that a spatial weighting function is used for the PDF computation. Next, the entropy of the PDF of pixel intensities is computed for the patch and stored at pixel x_i. As illustrated in Figure 2.10, patches 1 and 2 have same structural pattern, while the structural pattern of patch 3 is different. As a result, the entropy values are derived for patches 1 and 2, while the entropy value for patch 3 is different. Finally, an entropy image of the gray-scale image is generated by computing entropy for each pixel. The entropy images of two original images which are being registered can be treated as input images of an image registration technique. As pointed out in [33], a limitation of the structural representation using entropy images is that a changing level of noise within and across images would influence the entropy computation.

2.5.3 Mappings of Keypoint Triplets

In [49], a registration framework was proposed to improve the initial pairwise matching of local descriptors by using spatial and geometrical relationships of triplets of descriptors. In [49], although the authors used SIFT [56] and PIIFD [17] to demonstrate the performance of their framework, the framework should work with any other types of local descriptors. For the purpose of describing the framework, we will use PIIFD. First of all, let $P_r^i, i = 1,2,\ldots,N_r$ denote keypoints in the reference image, and $P_t^j, j = 1,2,\ldots,N_t$ denote keypoints in the target image. Likewise, let $D_r^i, i = 1,2,\ldots,N_r$ denote PIIFD descriptors in the reference image, and $D_t^j, j = 1,2,\ldots,N_t$ denote PIIFD descriptors in the target image.

i. Relative to a keypoint, P_r^i, in the reference image, all the PIIFD descriptors in the target image are ranked in terms of the distance to P_r^i. As a result, an initial mapping for each reference keypoint is obtained:

$$P_r^i \mapsto \{P_t^1, P_t^2, \ldots P_t^{N_c}\}, \tag{2.29}$$

where N_c denotes the number of candidate matches.

ii. Keypoint triplets are generated in the reference and target images.

The rationale behind using keypoint triplets is that at least three keypoint mappings are required to determine an affine transformation. In the reference image, $\binom{N_r}{3}$ keypoint triplets are generated. Accordingly, there are N_c^3 keypoint triplets generated in the target image relative to a keypoint triplet in the reference image.

iii. For each reference keypoint, P_r^i, the best match is determined as follows.

- In the reference image, all the keypoint triplets associated with P_r^i are selected.
- Each associated keypoint triplet in the reference image is compared with its candidate keypoint triplets in the target image. For the purpose of reducing time complexity, certain geometric constraints are imposed. If two keypoint triplets that are compared satisfy these constraints, an affine transformation is computed from the three pairs of keypoints.
- With an affine transformation, the reference image can be transformed onto the target image. Two edge images are derived from the transformed reference image and the target image. The two edge images are then overlapped and the Number of Overlapped Pixels (NOP) is computed.
- Assuming that the transformation estimated from $P_r^i, P_r^j, P_r^k \mapsto P_t^i, P_t^j, P_t^k$ achieves the maximum NOP, then keypoint P_t^i is the best match to P_r^i. This NOP value is attached to the match from P_r^i to P_t^i.

iv. All the keypoint matches are ranked by their NOP values. A threshold is set to select keypoint matches that hold highest NOP values.

v. RANSAC [23] is used to refine keypoint matches. We will introduce RANSAC in Section 2.7.

vi. A transformation is estimated from the refined keypoint matches and is used for aligning the reference and target images.

In [49], the registration framework of using spatial and geometrical relationships of keypoint triplets is called Global Information (GI). For the referencing purpose, the

aforementioned method is called GI-PIIFD in this thesis. Likewise, GI-SIFT can be derived if SIFT as the local descriptor is used for the registration framework in [49]. Experiments in [49] have shown that GI-PIIFD outperforms GI-SIFT in registering multi-modal images. Also, GI-PIIFD will be used as a benchmark technique for performance comparisons in Chapter 5. The main reason for doing this is that GI-PIIFD takes into account both local representations and spatial relationships between keypoints.

2.6 Techniques for Feature Matching

In Sections 2.4 and 2.5, we have reviewed various techniques for describing feature points. With a particular feature description technique, descriptors are built for all feature points. The next step is to match descriptors in the reference and target images so that a set of matches can be determined. Here, we summarize three strategies for matching descriptors: threshold-based matching, nearest neighbor (NN) matching and matching based on the nearest neighbor distance ratio (NNDR) [66].

First of all, we use D_A to denote a descriptor in the reference image, D_1 and D_2 as the nearest neighbor and the second nearest neighbor to D_A, respectively, in the target image. And D_i represents an arbitrary descriptor in the target image. The three strategies for matching descriptors work as follows.

i. Threshold-based Matching

   D_A and D_i are matched if the distance between the two descriptors is below a threshold d_t. An explicit disadvantage of this matching strategy is that multiple matches are potentially generated relative to one descriptor in the reference image.

ii. NN Matching

   D_A and D_1 are matched if the distance between the two descriptors is below a threshold d_t. In the case of NN matching, the nearest neighbor might be very close to the second nearest neighbor in terms of descriptor distance, so that the discrimination of a match cannot be gauranteed.

iii. NNDR Matching

If the distance ratio between the nearest neighbor and the second nearest neighbor is below a threshold t, i.e.,

$$\frac{||D_A - D_1||}{||D_A - D_2||} < t, \tag{2.30}$$

where $||D_A - D_1||$ and $||D_A - D_2||$ are Euclidean distances of D_1 to D_A and D_2 to D_A respectively.

Comparing the three strategies for matching descriptors stated above, the NNDR matching is the most favorable choice for the reason that it decides a match by considering both the actual descriptor distance and the discrimination from all the other descriptors. The NNDR matching is used for keypoint matching in SIFT, where the threshold of distance ratio $t = 0.80$. By rejecting those matches in which the distance ratio is over t, 90% of false matches are eliminated while less than 5% of true matches are discarded, according to the experiments in SIFT [56]. Thus, we use NNDR matching in our experiments.

2.7 Techniques for Refining Matches and Estimating Transformation

In a feature-based image registration technique, a set of matches are determined after feature matching. The next step is to refine these matches and estimate a transformation. In this section, we will review three relevant techniques.

2.7.1 Least Squares

Least Squares is an old, commonly-used data fitting technique which was proposed in [12]. Below is the basic idea of Least Squares. Suppose there is a set of data points $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$, where x is the independent variable and y is the dependent variable. The fitting function $f(x)$ has a deviation from each data point, i.e., $d_i = y_i - f(x_i)$, where $1 \leq i \leq n$. Such a deviation d_i is called a residual. The best fitting function

minimizes the sum of squared residuals, i.e.,

$$\sum_{i=1}^{n} d_i^2 = minimum. \tag{2.31}$$

Least Squares is used to refine matches as follows. A set of keypoint matches is given, and suppose that (x_i, y_i) is matched to (x_i', y_i'). First, a transformation H is estimated by a random subset of keypoint matches, so the transformed coordinate of (x_i, y_i) is $H(x_i, y_i)$. Second, the Euclidean distance between (x_i', y_i') and $H(x_i, y_i)$ is computed as

$$d_i = ||(x_i', y_i') - H(x_i, y_i)||. \tag{2.32}$$

Next, $\sum_{i=1}^{n} d_i^2$ is minimized, where n is the number of keypoint matches. The final transformation which achieves the minimum of $\sum_{i=1}^{n} d_i^2$ is denoted as H_f. The keypoint matches which drive H_f are preserved, and accordingly the other matches are removed. For the purpose of refining matches, Least Squares is simple and efficient. However, the major limitation of this technique is its low tolerance to outliers.

2.7.2 Hough Transform

Hough transform was first introduced in [40], which was later generalized in [22]. In image processing, the purpose of Hough transform is to find imperfect instances within a number of shapes by a voting scheme. The shapes include straight lines, circles and ellipse, etc. Let us take detecting straight lines as an example to illustrate how Hough transform works as follows.

i. A straight line in image space is described as $y = mx + b$, where m is the line slope and b is the intercept at y axis.

ii. The straight line $y = mx + b$ is associated with a point (r, θ) of a polar coordinate in Hough parameter space. r represents the algebraic distance between the line and the pre-defined origin, while θ is the angle of the vector which is orthogonal to the line and points towards the upper half plane.

iii. The Hough parameter space is discretized into a number of bins. For each point in the format of (r, θ), a vote is put into the corresponding bin.

iv. The bin or (bins) holding the most votes is selected. Accordingly, straight lines corresponding to selected bins are detected.

More broadly, Hough transform has been successfully used in various applications such as detecting arbitrary shapes [7], object detection [61] and object recognition [56]. To refine keypoint matches, Hough transform is used to identify clusters of keypoint matches which share similar transformations [56]. Scale, rotation and translations at x and y axis make up a four-dimensional Hough space. Each match contributes a vote to the Hough space. Ideally, all true matches contribute to the same bin in the Hough space. The bin with the most votes accumulates those matches which are true with a high confidence. The matches which do not fall into the bin are removed from the original keypoint matches, so that the original keypoint matches are refined. However, Hough transform has two limitations as follows. First, it is difficult to decide the dimension of Hough parameter space. Second, the complexity is too high if the required dimension of Hough parameter space is high [37].

2.7.3 RANSAC

RANdom SAmple Consensus (RANSAC) was proposed in [23] as a parameter estimation technique. How RANSAC works is summarized in the following steps.

a. Given a set of data points, the minimum number of points required to determine a model is randomly selected.

b. With randomly selected points, an initial model is determined.

c. The initial set of data points is enlarged by searching those points which fit the initial model with a pre-defined tolerance ε. The enlarged set is called a consensus set, where the included data points are treated as inliers.

d. If the fraction of inliers within the whole set of data points, F_i, is above a pre-defined threshold τ, a new model is estimated using the consensus set and the algorithm is completed.

e. If F_i is below τ, steps a to d are repeated for a pre-defined number of times.

RANSAC is used to refine keypoint matches as follows. A transformation is calculated using four keypoint matches randomly selected. The transformation is iteratively calculated many times (normally between 500 and 1000). In testing an estimated transformation, if a match is true with a pre-defined, acceptable error, the match is regarded as an inlier. The iteration in which the number of inliers is largest is recorded, and accordingly the outliers are removed from the original keypoint matches.

RANSAC has been widely used for robust estimation problems in computer vision, primarily due to its high accuracy of estimation even when there are a significant number of outliers in the input data [84]. Admittedly, if the percentage of inliers is too low, RANSAC cannot estimate an optimal model. One limitation is that RANSAC only estimates one model for a particular set of data points. However, only one model (a transformation) is to be estimated in registering our tested image pairs as the transformation is uniform across the entire image. Note that Least Squares is used to estimate an image transformation after keypoint matches are refined by RANSAC in our experiments.

2.8 Summary

In this chapter, we have given a systematic and thorough review on exiting image registration techniques. We have identified the most promising registration techniques which will be the basis of our work in Chapters 3, 4 and 5, as follows.

i. SIFT-like descriptors have shown their effectiveness in the domain of image registration. In reviewing SIFT-like descriptors, we have mentioned two types of gradient information, i.e., GM (Gradient Magnitudes) and GO (Gradient Occurrences). However, both GM and GO have limitations in building and matching descriptors, which will be detailed in Chapter 3. A better way of gradient utilization must exist. Improvements will be made in both mono-modal and multi-modal cases. In mono-modal cases, improvements will be made on the basis of SIFT and GO-SIFT, while IS-SIFT and GO-IS-SIFT are the foundation for making improvements in multi-modal cases.

ii. The PIIFD descriptor [17] was designed for registering multi-modal images. PIIFD achieves invariance to image rotation, *gradient reversal*, *region reversal* and partial invariance to intensities. However, there are two problems in PIIFD. First, it is not scale-invariant. Second, using Harris corners as keypoints for building descriptors is by no means the best choice, due to the fact that these corners are sensitive to intensity variations in a small neighborhood. Third, the robustness to content differences is not guaranteed because the PIIFD descriptor is only partially invariant to intensity changes. PIIFD will be used in Chapter 4 for performance comparisons and the aforementioned two problems will be addressed in Chapter 5.

iii. In [49], a multi-modal image registration framework of using spatial and geometrical relationships of keypoint triplets was proposed, as described in Section 2.5.3. By incorporating the PIIFD descriptor into the registration framework, GI-PIIFD is formed as a multi-modal image registration technique. The multi-modal image registration framework will be used in our proposed registration technique and GI-PIIFD will be used as a benchmark technique for performance comparisons in Chapter 5.

Chapter 3

Improving SIFT by Better Utilization of Image Gradients

3.1 Overview of Gradient Utilization in SIFT-based Registration Techniques

SIFT [56] is a very popular technique for detecting and describing local features in images and it has been widely used in the field of image registration such as [19, 38, 39, 66, 68, 70]. In a SIFT-based image registration technique, describing a keypoint is equivalent to describing image information in a local region around this keypoint, as elaborated in Section 2.4.1.1 of Chapter 2. Thus, how image information is described in a local region directly affects the discrimination power of the formed local descriptor.

A SIFT-like descriptor is built based on a local region around a given keypoint. This local region is divided into a number of sub-regions, e.g. 4×4 sub-regions in the original SIFT. In each sub-region, a histogram of gradient orientations is built. At each orientation bin of the histogram, Gradient Magnitude (GM) of each pixel, whose quantized gradient orientation corresponds to the orientation bin, is accumulated. Instead of utilizing GM, Gradient Occurrence (GO) was proposed in [39] for building histograms of gradient orientations. GO is defined as the number of occurrences of image gradients whose quantized orientations correspond to a particular orientation bin. GM and GO are two categories of gradient information which are used for building SIFT-like local descriptors.

The purpose of this chapter is to explore a better way of utilizing gradient information, thereby improving the discrimination power of SIFT-like descriptors in

image registration. The rest of the chapter is structured as follows. Section 3.2 gives a detailed analysis of the utilization of either GM or GO in building and matching SIFT-like descriptors. In Section 3.3, we will propose a technique to better utilize GM and GO. Our experimental results are shown and discussed in Section 3.4. Finally, the chapter is summarized in Section 3.5.

3.2 Analysis of the Utilization of either GM or GO

In this section, we will make a theoretical analysis of the utilization of either GM or GO (Sections 3.2.1 and 3.2.2). Also, the limitations of utilizing only GM or GO will be clearly illustrated through examples (Sections 3.2.3 and 3.2.4).

3.2.1 Utilizing Only GM

As stated in [19, 56], in SIFT-like registration techniques which use GM for building descriptors, GM values are incremented for each orientation bin of a gradient histogram. Assume that there are n pixels whose quantized gradient orientations correspond to the o^{th} orientation bin of the (x,y) spatial bin in a histogram of gradient orientations. For brevity, we can call the bin the (x,y,o) orientation bin. The GM value for the (x,y,o) orientation bin is calculated by

$$D_{GM}(x,y,o) = \sum_{i=1}^{n} M_i, \tag{3.1}$$

where M_i is the GM of the i^{th} pixel.

Now let us examine three scenarios for the (x,y,o) orientation bin. The three scenarios have the same sum of GM values, but there are n, two and three pixels whose quantized gradient orientations correspond to this orientation bin. Without loss of generality, there can be many scenarios which have the same sum of GM values but are different in the number of pixels. These scenarios are likely to represent different image contents, but cannot be distinguished by the utilization of GM.

3.2.2 Utilizing Only GO

According to the definition of GO stated in Section 3.1, we define a function f_{GO} as

$$f_{GO}(\{M_1, M_2, \dots, M_n\}) = n, \tag{3.2}$$

where $\{M_1, M_2, \dots, M_n\}$ represent GM values of n pixels corresponding to a particular orientation bin. Regardless of the GM value of each pixel, when utilizing GO for building descriptors, the value for this orientation bin is n. However, GM value of each of the n pixels can be arbitrary, indicating many different scenarios in image contents. These different scenarios cannot be distinguished by the utilization of GO.

3.2.3 An Artificial Example

In [39], utilizing GO was proposed for building descriptors, rather than utilizing GM in the original SIFT. In the following we will point out the limitations of only utilizing either GM or GO through an example.

In [39], GO was proposed on Symmetric SIFT (called SSIFT for the referencing purpose) [19], so we call [39] GO-SSIFT. In SSIFT, orientation histograms are used for building a SSIFT descriptor. The value in each orientation bin is incremented by the GM of each pixel with the corresponding orientation. However, utilizing GM for incrementing the values in the orientation bins will result in descriptors which would potentially cause an ambiguity. Figure 3.1 gives one example illustrating the limitation of utilizing GM for building descriptors.

In Figure 3.1, each sub-figure denotes a spatial bin in a local region, as stated in Section 2.4.1.1, for building a descriptor, and each cell corresponds to a pixel. Each arrow indicates an occurrence of image gradient. The length and orientation of each arrow denote the gradient magnitude and orientation of a particular pixel respectively. A blank cell means that the gradient magnitude of this pixel is zero. Without loss of generality, assume that the orientations of all the arrows in the four spatial bins are 45°. Based on the process of building a SSIFT descriptor in Section 2.5.1.1 of Chapter 2, the sum of gradient magnitudes at the 45° orientation bin for the four spatial bins is 4 and the values in all the other seven orientation bins are zero.

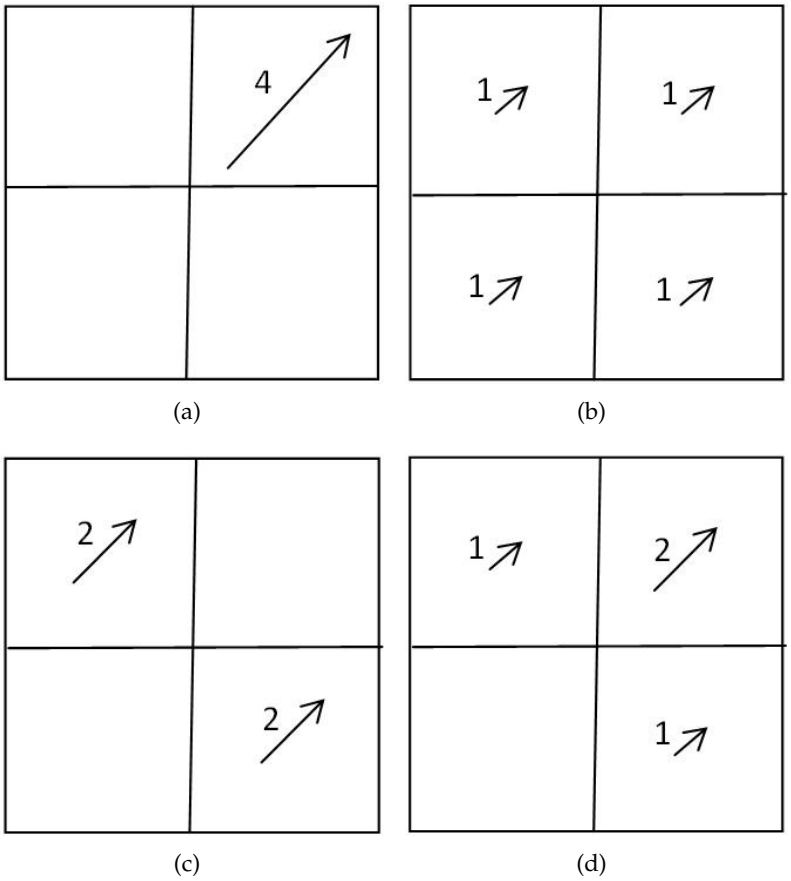


Figure 3.1: Ambiguity of incrementing the values in the orientation bins based on GM: the four visually different spatial bins have the same orientation histogram. Note that, there are many other combinations with different pixel locations, which also applies to Figure 3.2.

Consequently, the same orientation histogram will be built for the four spatial bins. But the contents represented by the four spatial bins are completely different. Thus, an ambiguity has arisen. However, utilizing GO [39], the value in the 45° orientation bin for the four spatial bins will be 1, 4, 2 and 3 respectively in Figure 3.1.

Utilizing GO for incrementing the values in the orientation bins can successfully distinguish those regions similar to the ones depicted in Figure 3.1. However, we have found that utilizing GO for building descriptors might cause a similar ambiguity to utilizing GM. A typical example is shown in Figure 3.2. All the assumptions in Figure 3.2 are consistent with those in Figure 3.1. Let us build the orientation histograms for the four regions in Figure 3.2 utilizing GO to increment the value in each orientation bin. The value in the 45° orientation bin would be consistently

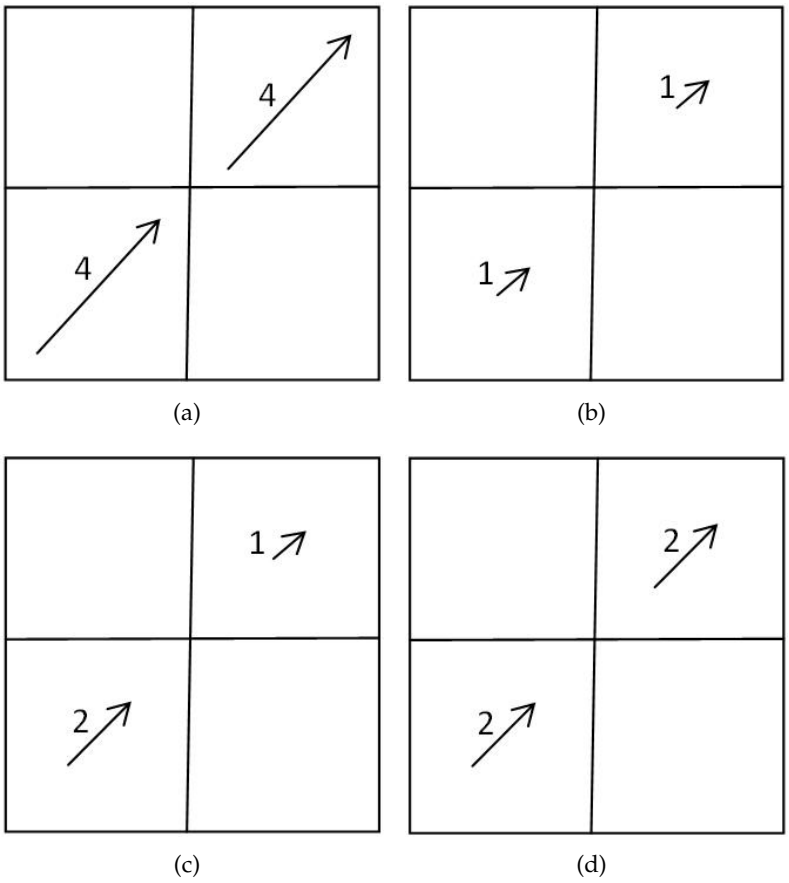
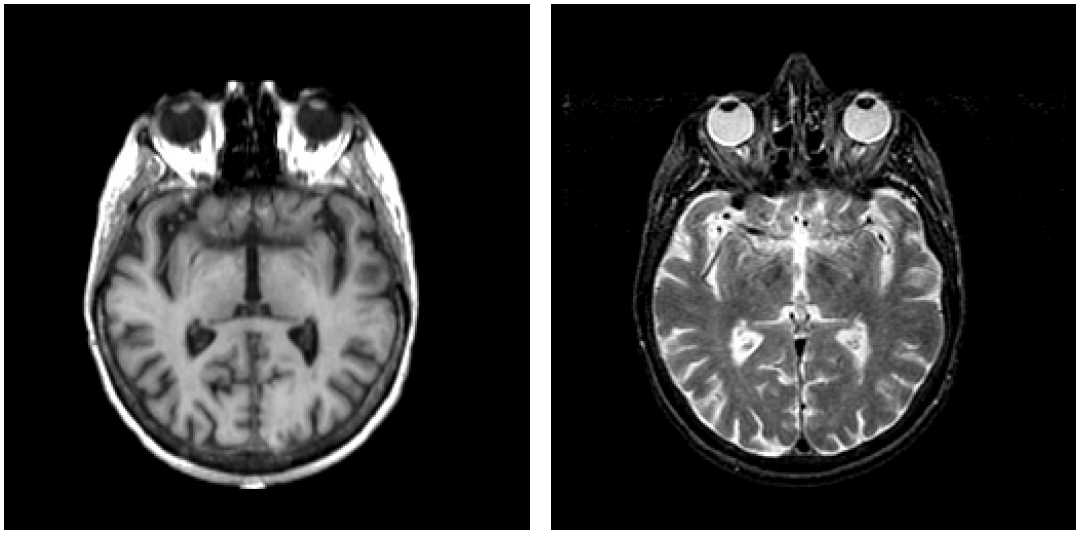


Figure 3.2: Ambiguity of incrementing the values in the orientation bins based on GO: the four visually different regions have the same orientation histogram.

equivalent to 2 for the four spatial bins. Thus, the same orientation histogram will be built for them. However, the four spatial bins are likely to represent different image contents. If orientation bins are incremented utilizing GM, the value in the 45° orientation bin would be 8, 2, 3 and 4 respectively, leading to four different orientation histograms.

In conclusion, neither GM nor GO is able to distinguish local image regions with different visual contents in all circumstances, such as the two examples illustrated in Figure 3.1 and Figure 3.2. Since GM and GO are both important visual properties in images, utilizing both types of gradient information in the feature description and matching will potentially improve the registration performance.

3.2.4 A Real Example



(a) DoG Pyramid (b) DoG Extrema

Figure 3.3: A Pair of Multi-modal MRI Images

In Section 3.2.3, the limitations of only utilizing either GM or GO are pointed out using an artificial example. Now we give a real example of registering two images using IS-SIFT and GO-IS-SIFT which uses GM and GO for building descriptors respectively. Details for IS-SIFT and GO-IS-SIFT can be found in Section 2.5.1.1. The two images are shown in Figure 3.3, which are a pair of multi-modal MRI (Magnetic Resonance Imaging) [36] images. In the example, we will analyze false matches determined by IS-SIFT and GO-IS-SIFT respectively. More specifically, we are interested in seeing how GO-IS-SIFT deals with the false matches which are determined by IS-SIFT, and vice versa.

In registering the two images shown in Figure 3.3, 13 false matches are determined by IS-SIFT. We analyze the matching status of these 13 false matches when IS-SIFT and GO-IS-SIFT are respectively used. First of all, the two images in Figure 3.3 are denoted as I_r and I_t, so we use a keypoint mapping $P_r^i \mapsto P_t^j$ to refer to a false match from I_r to I_t. For a false match $P_r^i \mapsto P_t^j$ determined by IS-SIFT, the matching status using GO-IS-SIFT can be divided into three cases as follows.

A. Keypoint P_r^i is not the closest neighbor to keypoint P_t^j.

Table 3.1: Matching Status of False Matches Determined by IS-SIFT

ID	Distance Ratio[a] by IS-SIFT	If Closest Neighbor by GO-IS-SIFT?	Distance Ratio by GO-IS-SIFT	Case
1	0.729	Yes	0.823	B
2	0.779	Yes	0.660	C
3	0.798	No	N/A	A
4	0.780	Yes	0.909	B
5	0.793	Yes	0.961	B
6	0.661	Yes	0.994	B
7	0.726	Yes	0.826	B
8	0.791	No	N/A	A
9	0.728	Yes	0.918	B
10	0.729	Yes	0.822	B
11	0.779	Yes	0.899	B
12	0.616	Yes	0.572	C
13	0.766	Yes	0.876	B

[a] The threshold of distance ratio is set to 0.800 as in the original SIFT.

Table 3.2: Matching Status of False Matches Determined by GO-IS-SIFT

ID	Distance Ratio by GO-IS-SIFT	If Closest Neighbor by IS-SIFT?	Distance Ratio by IS-SIFT	Case
1	0.660	Yes	0.779	C
2	0.783	Yes	0.995	B
3	0.758	No	N/A	A
4	0.797	Yes	0.809	B
5	0.768	Yes	0.852	B
6	0.572	Yes	0.616	C
7	0.749	Yes	0.943	B

B. Keypoint P_r^i is the closest neighbor to keypoint P_t^j, but the distance ratio between the closest neighbor and the second closest neighbor is above the pre-defined threshold. Note that, a keypoint mapping is determined as a match if this distance ratio is below the threshold, indicating the closest neighbor is sufficiently distinctive from all the rest keypoints.

C. Keypoint P_r^i is the closest neighbor to keypoint P_t^j, and the distance ratio between the closest neighbor and the second closest neighbor is below the threshold.

In Cases A and B, a false match $P_r^i \mapsto P_t^j$ determined by IS-SIFT is not regarded as a match by GO-IS-SIFT. In Case C, GO-IS-SIFT also determines $P_r^i \mapsto P_t^j$ as a false match,

which means neither IS-SIFT nor GO-IS-SIFT can successfully distinguish Keypoint P_r^i from P_t^j. Table 3.1 lists which case each false match belongs to. It can be seen that 11 out of 13 false matches fall into either Case A or Case B, which is 84.62%. In other words, the majority of false matches determined by IS-SIFT are not matches at all when using GO-IS-SIFT. Therefore, GO-IS-SIFT has advantages over IS-SIFT in dealing with these matches. Likewise, Table 3.2 shows the matching status of false matches that are determined by GO-IS-SIFT. In the seven false matches determined by GO-IS-SIFT, five matches belong to Case A or B when using IS-SIFT. Thus, IS-SIFT has advantages over GO-IS-SIFT in dealing with these matches.

This aforementioned real example suggests that GM suits some circumstances better than GO, and vice versa. Thus, only utilizing either GM or GO is by no means the optimal choice for building and matching SIFT-like descriptors.

3.3 A New Way of Utilizing Gradients

In this section, we will introduce our proposed way of utilizing gradient information in building and matching SIFT-like descriptors. The proposed technique is called MOG (Magnitudes and Occurrences of Gradients) for the referencing purpose. First, we will describe the rationale and steps of MOG. Second, the characteristics of MOG matches will be analyzed.

3.3.1 Rationale and Steps of MOG

The analysis in Section 3.2 has inspired us to find a new way of utilizing gradients for feature description and matching. Both GM and GO have limitations in building and matching SIFT-like descriptors, as analyzed in Section 3.2. The examples shown in Sections 3.2.3 and 3.2.4 convey the message that GM and GO are complementary gradient information. Therefore, a keypoint match that satisfies the matching criteria of both GM-based and GO-based descriptors is more likely to be a true match as compared to a match only satisfying either of the two. This assumption will be clearly illustrated in Section 3.3.2.

It is noted that the proposed MOG can be incorporated with SIFT and IS-SIFT for registering mono-modal and multi-modal images respectively. Accordingly, the incorporated techniques are called MOG-SIFT and MOG-IS-SIFT. Here, we illustrate the steps of MOG-IS-SIFT as follows.

i. IS-SIFT is performed to determine a set of keypoint matches

$$M_{GM} = \{M_{GM}^1, M_{GM}^2, \dots M_{GM}^{N_1}\}. \tag{3.3}$$

ii. GO-IS-SIFT is performed to determine a set of keypoint matches

$$M_{GO} = \{M_{GO}^1, M_{GO}^2, \dots M_{GO}^{N_2}\}. \tag{3.4}$$

iii. The common matches of the two sets of keypoint matches, M_{GM} and M_{GO}, constitute a new set of keypoint matches

$$M_{MOG} = M_{GM} \bigcap M_{GO}. \tag{3.5}$$

The relationships between M_{GM}, M_{GO} and M_{MOG} are illustrated in Figure 3.4.

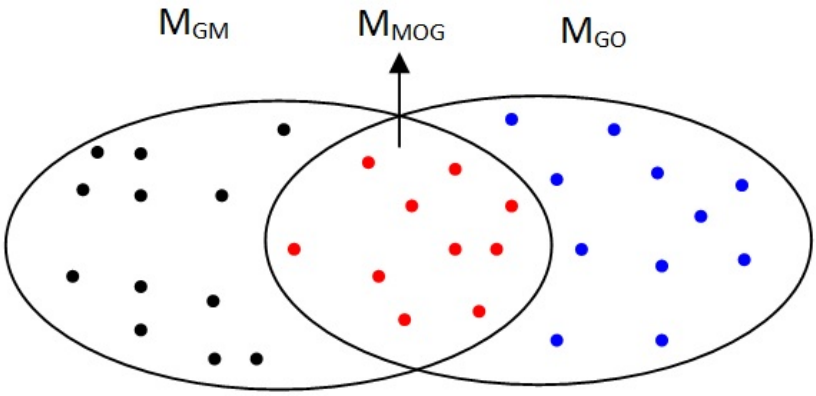


Figure 3.4: Illustrating MOG Matches. Each dot in the figure indicates a keypoint match.

3.3.2 Characteristics of MOG Matches

The matches that are included in M_{MOG} satisfy the matching criteria by both IS-SIFT descriptors and GO-IS-SIFT descriptors. By imposing the matching criteria of both

GM-based and GO-based descriptors, a MOG match is expected to have stronger discrimination power, as compared to those matches that fall into either M_{GM} or M_{GO}, but not included in M_{MOG}. Thus, a MOG match is more likely to be a true match as compared to a match determined by IS-SIFT or GO-IS-SIFT.

3.4 Performance Study

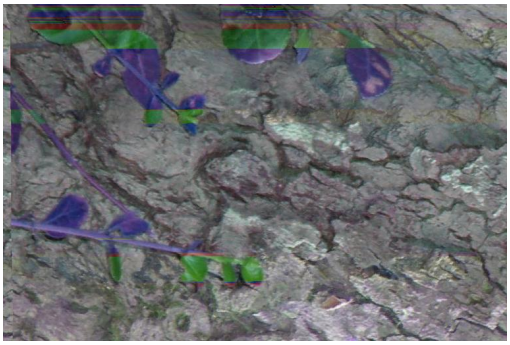
We evaluate the proposed MOG on both mono-modal and multi-modal data sets. In mono-modal cases, SIFT, GO-SIFT and MOG-SIFT are compared, whereas in multi-modal cases, IS-SIFT, GO-IS-SIFT and MOG-IS-SIFT are compared.

3.4.1 Test Data

3.4.1.1 Mono-modal Data Sets

In the test data, there are two mono-modal data sets: Affine Covariant Regions Data Set [80] and Mono-modal Microscopic Data Set. In the Affine Covariant Regions Data Set, there are five different changes in imaging conditions: scale and rotation, viewpoint, blur, illumination and JPEG compression. The scale and blur changes are obtained by varying the camera zoom and focus, respectively. The scale and rotation changes are up to four times and approximately 180° respectively. With regard to viewpoint changes, the camera varies from a fronto-parallel view to one with significant foreshortening at approximately 50° to 60° [66]. The illumination changes are introduced by varying the camera aperture [66]. The JPEG compression changes are generated using a standard xv image browser with the image quality parameter varying from 40% to 2%. There are eight original images and five image pairs are generated for each original image. The eight images are shown in Figure 3.5 and the corresponding imaging transformations are listed in Table 3.4 (See Section 3.4.3). Thus, the Affine Covariant Regions Data Set includes 40 image pairs.

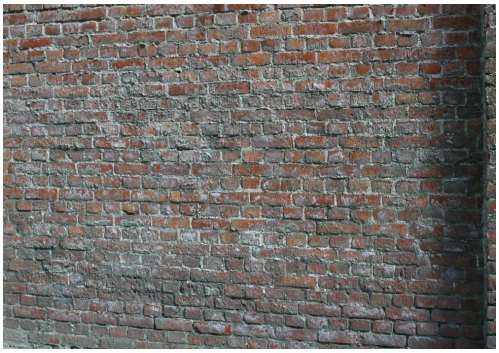
In the Mono-modal Microscopic Data Set, changes in imaging condition are in scale and rotation. The scale and rotation changes are up to four times and 90° respec-



(a) bark

(b) boat

(c) graffiti



(d) wall

(e) bikes

(f) trees

(g) leuven

(h) ubc

Figure 3.5: Eight Original Images from Affine Covariant Regions Data Set

tively. There are eight original image pairs. Two out of the eight pairs are shown in Figure 3.6. In each image pair, the reference image remains unchanged and the target image is rotated up to 90° with an increment of 15°. With rotation changes, the Mono-modal Microscopic Data Set includes 56 image pairs.

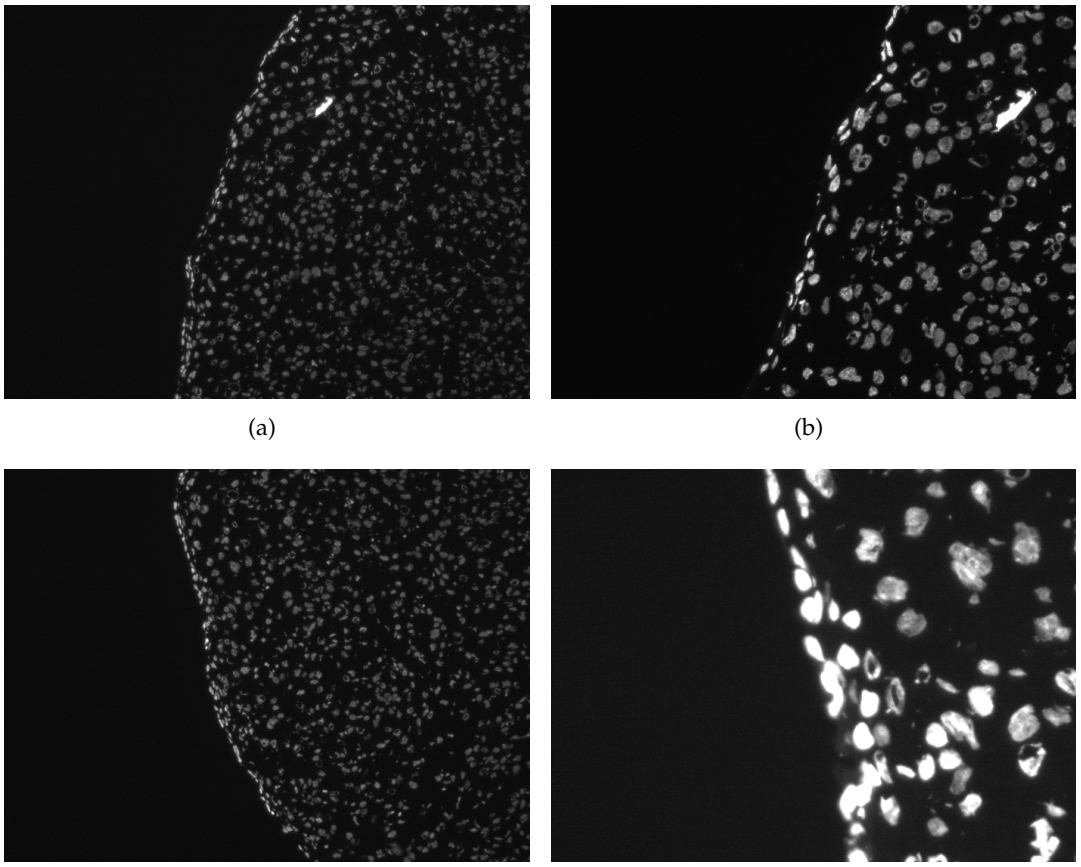
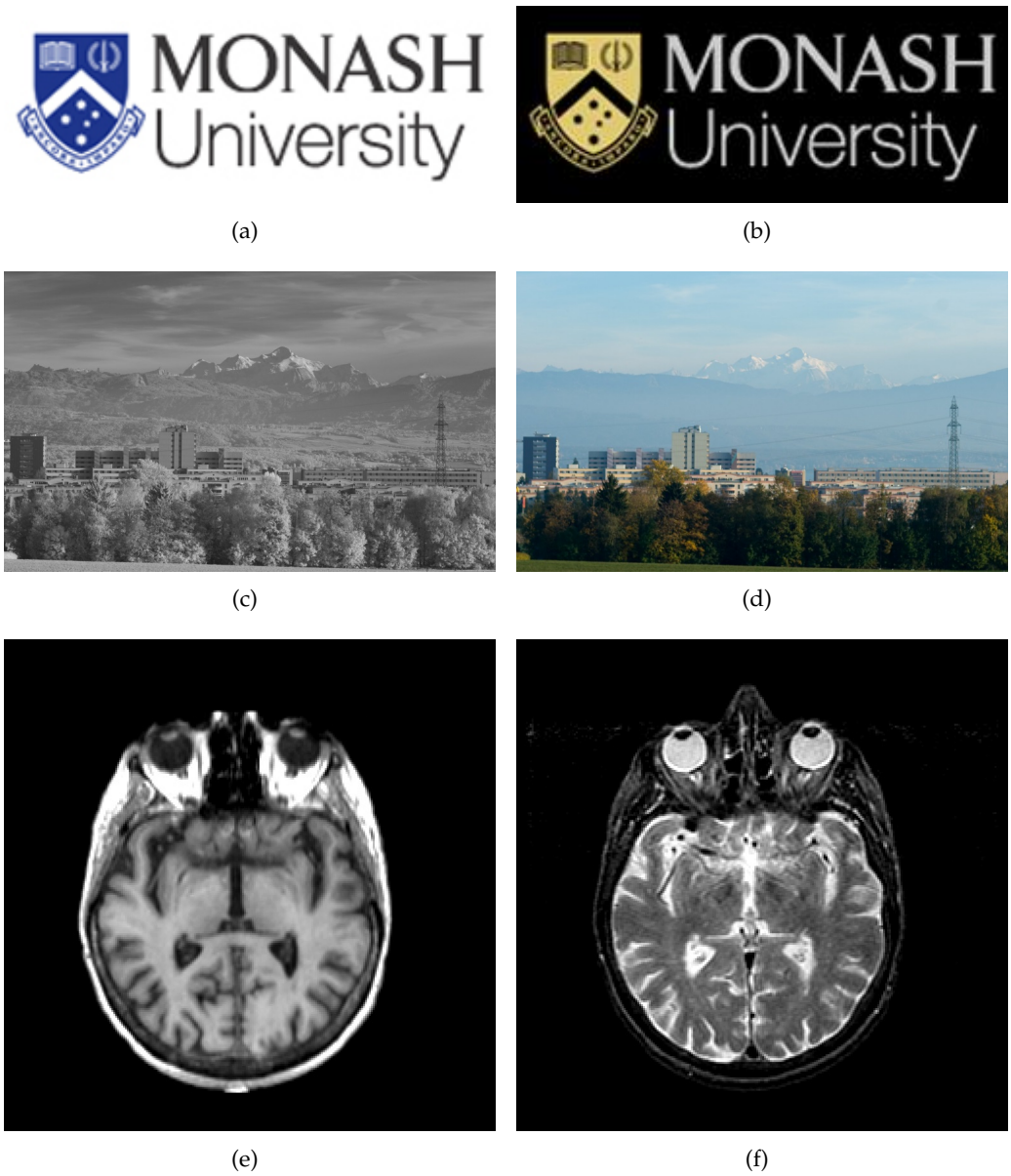


Figure 3.6: Two Pairs of Mono-modal Microscopic Images. The scale difference between (a) and (b) is 2X. The scale difference between (c) and (d) is 4X.

3.4.1.2 Multi-modal Data Sets

There are four multi-modal data sets in our test data. Data Set 1 includes two artificial pairs in which image contrast is reversed between the reference and target images. Data Set 2 includes 18 NIR (Near Infra-Red) vs EO (Electro-Optical) image pairs. Data Set 3 includes four image pairs used in [111]. The four image pairs include three MRI



(a) (b)

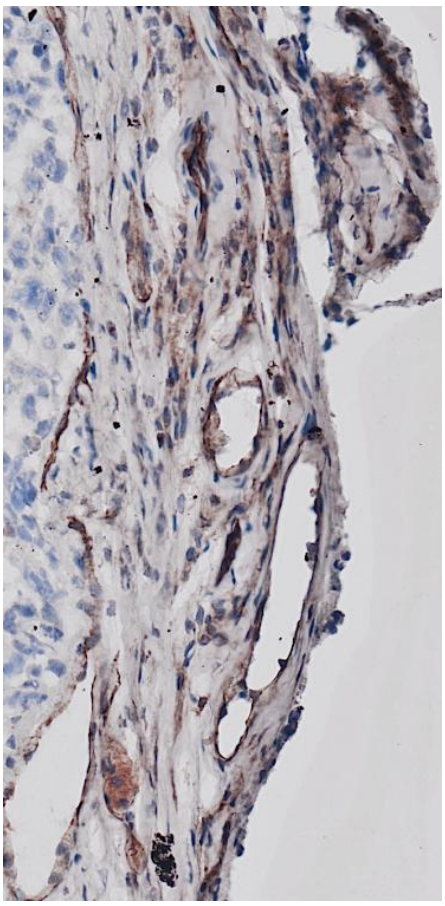
(c) (d)

(e) (f)

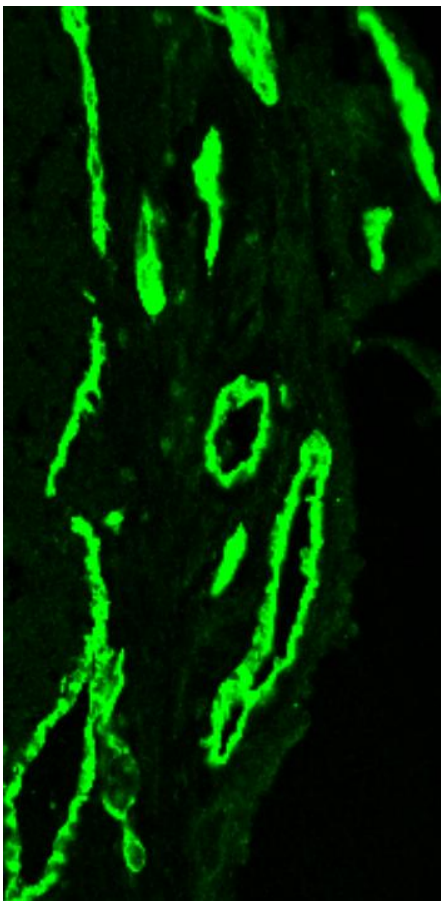
Figure 3.7: Examples of Image Pairs from Multi-modal Data Sets. (a) and (b): Artificial; (c) and (d): NIR vs EO; (e) and (f): MRI (T1 vs T2).

Table 3.3: Pair IDs and Imaging Category

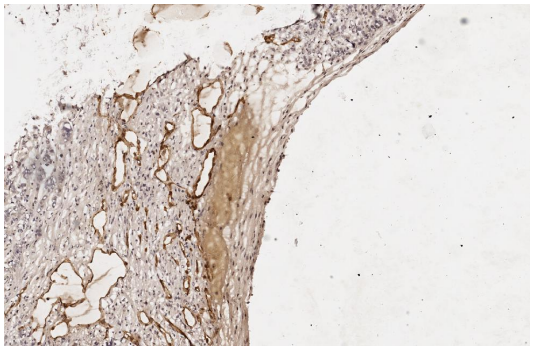
Data Set	ID	Category
1	1-2	Artificial
2	3-20	NIR vs EO
3	21-24	MRI, EO vs IR
4	25-40	Multi-modal Microscopic



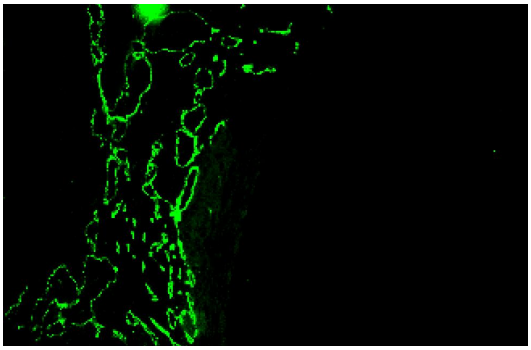
(a) Color Image 2



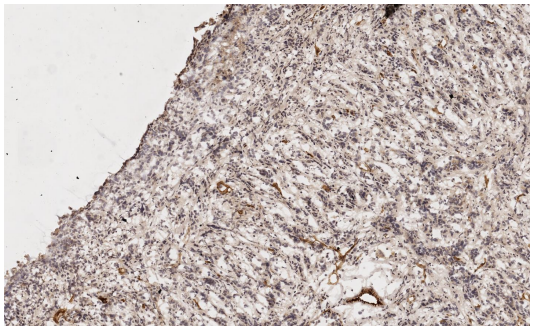
(b) Confocal Image 2



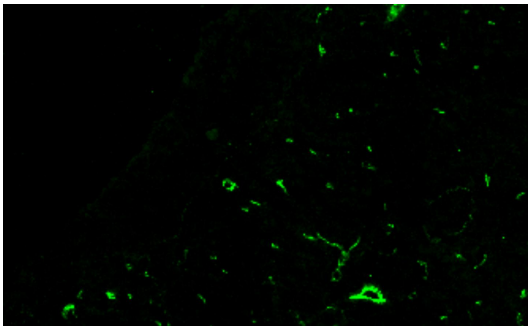
(c) Color Image 6



(d) Confocal Image 6



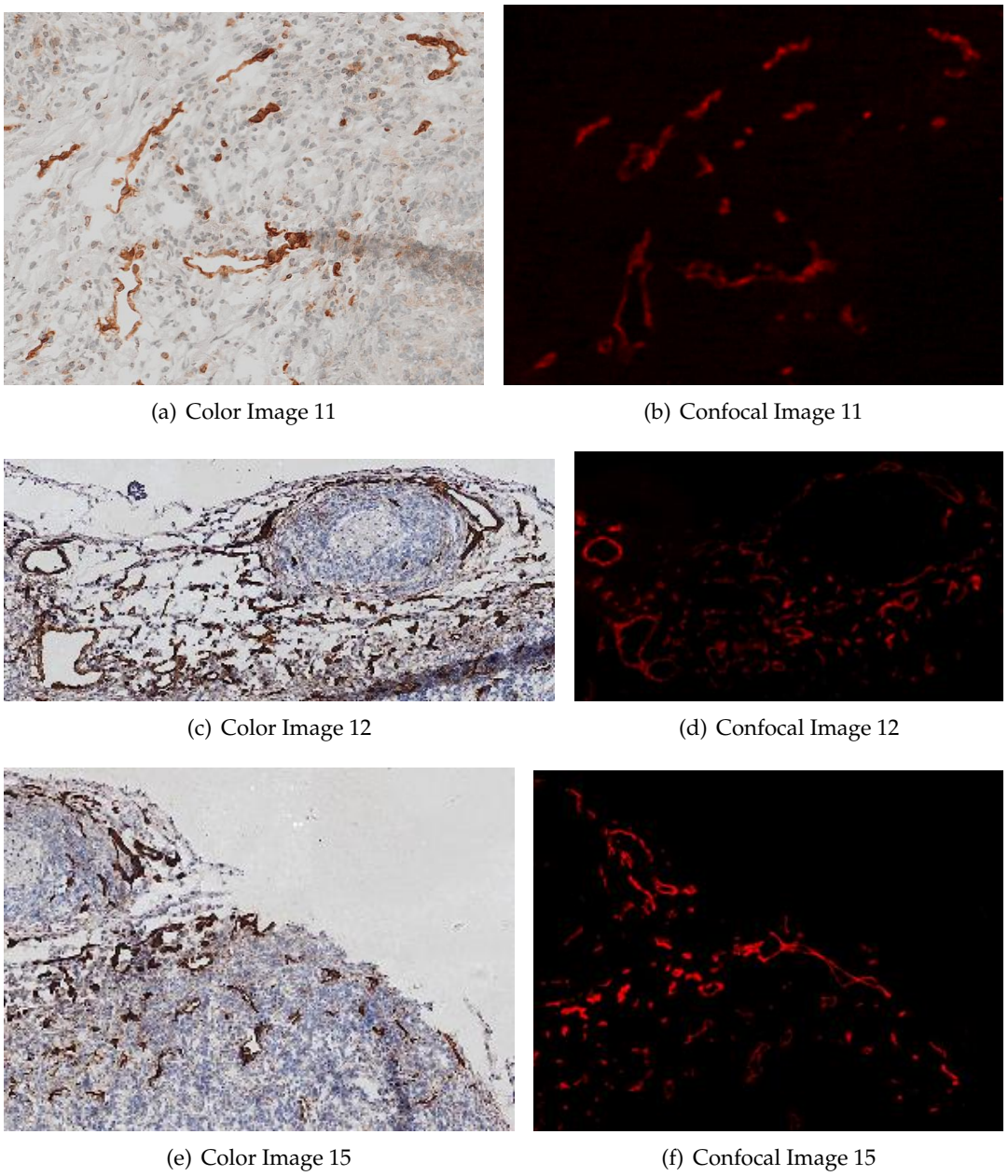
(e) Color Image 7



(f) Confocal Image 7

Figure 3.8: Sample Multi-modal Microscopic Image Pairs (Part 1)

pairs and one EO vs IR (Infra-Red) pair. The three MRI pairs are of different weighting patterns [36]: T1 vs T2, T1 vs PD (Proton Density), and T2 vs PD, for each. Data Set 4 includes 16 multi-modal microscopic image pairs. For each data set, the IDs of image pairs and their corresponding imaging category are listed in Table 3.3. Figure 3.7 shows three image pairs for the categories Artificial, NIR vs EO and MRI, respectively.



(a) Color Image 11 (b) Confocal Image 11

(c) Color Image 12 (d) Confocal Image 12

(e) Color Image 15 (f) Confocal Image 15

Figure 3.9: Sample Multi-modal Microscopic Image Pairs (Part 2)

Samples of multi-modal microscopic image pairs are shown in Figure 3.8 and Figure 3.9, where the six image pairs represent six different specimens.

3.4.2 Evaluation Criterion

The accuracy of an image registration technique, to a high degree, depends on the matching accuracy. The higher the percentage of true matches, the more accurate the final registration will be. Therefore, we evaluate our proposed technique using the matching accuracy where

$$accuracy = \frac{Number\ of\ true\ matches}{Number\ of\ total\ matches} \times 100\%. \tag{3.6}$$

In the tested data sets, the ground truth for non-microscopic image pairs is known or provided [80, 111]. A maximum error of four pixels, which is consistent with the setting in [111], is used when determining a true match. However, microscopic image pairs, in both mono-modal and multi-modal cases, show a higher complexity from the perspective of image registration, as compared to other tested image pairs. Thus, the maximum error for determining a true match is set to five in registering microscopic image pairs, which is reasonably acceptable.

3.4.3 Experiments on Mono-modal Images

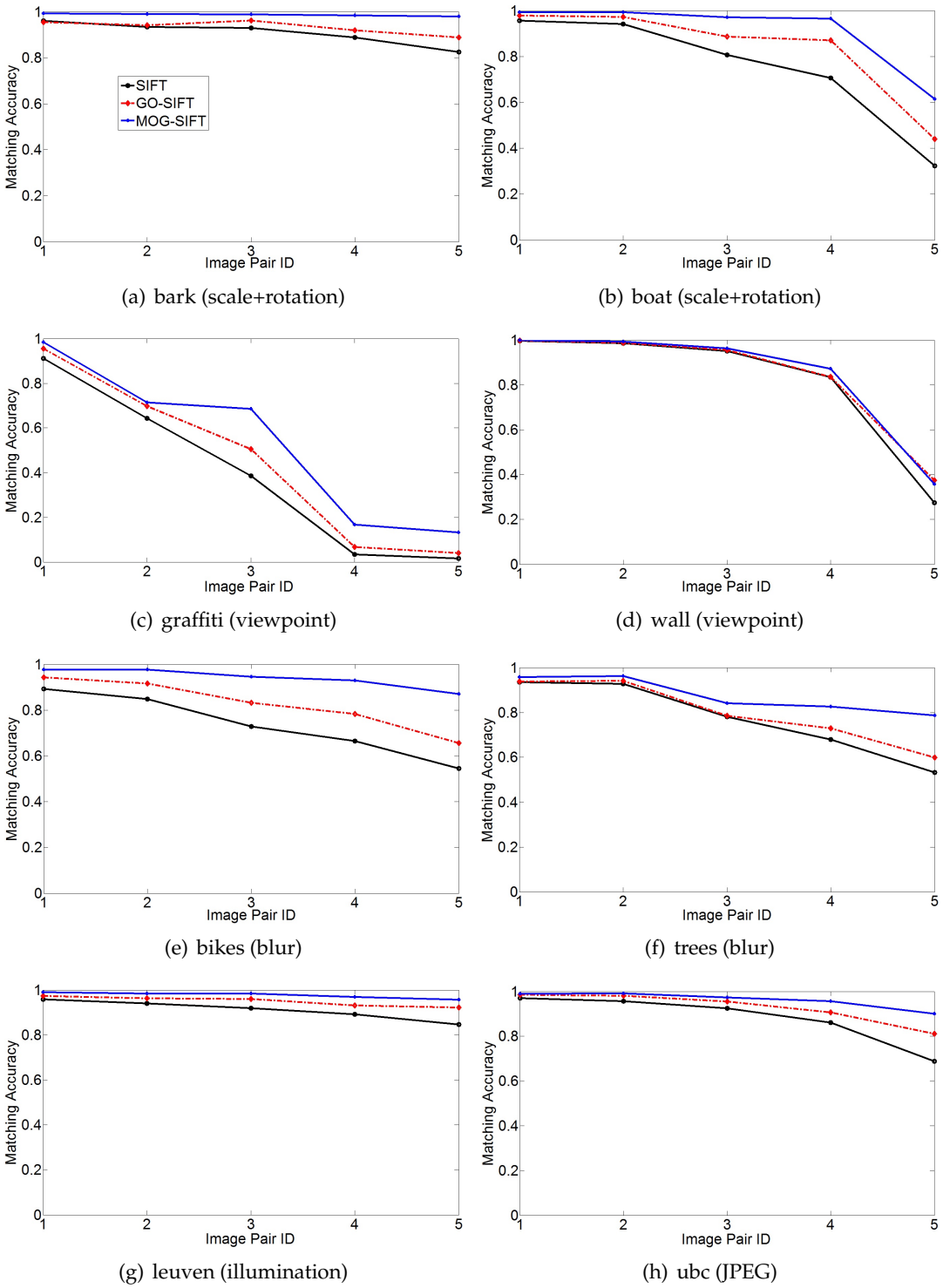
For the Affine Covariant Regions Data Set, Table 3.4 lists the number of true matches, the number of false matches and matching accuracy. Note that the values in the *Accuracy* column are averaged matching accuracy for registering five image pairs relative to each original image. As shown in the *Accuracy* column in the italicized bold font, MOG-SIFT consistently achieves the highest matching accuracy. With regard to the number of true matches, it is expected that the value for MOG-SIFT is smaller than that for SIFT and that for GO-SIFT. For all the image pairs in this data set, the average accuracies for SIFT, GO-SIFT and MOG-SIFT are 77.00%, 81.88% and 87.79%, respectively. There is an improvement of 5.91% in matching accuracy from GO-SIFT to MOG-SIFT. In addition, Figure 3.10 shows the changes of matching accuracy within five pairs for each original image. We can conclude two trends from

Table 3.4: Matching Accuracies for Affine Covariant Regions Data Set [80]

Image	T[a]	Technique	#True[b]	#False[b]	Accuracy(%)
bark	scale + rotation	SIFT	3774	307	92.48
		GO-SIFT	3678	224	94.26
		MOG-SIFT	3555	39	***98.91***
boat	scale + rotation	SIFT	7003	996	87.55
		GO-SIFT	6746	460	93.62
		MOG-SIFT	6421	140	***97.87***
graffiti	viewpoint	SIFT	2268	1057	68.21
		GO-SIFT	2211	556	79.91
		MOG-SIFT	2085	288	***87.86***
wall	viewpoint	SIFT	16817	466	97.30
		GO-SIFT	16211	341	97.94
		MOG-SIFT	15671	211	***98.67***
bikes	blur	SIFT	4410	1197	78.65
		GO-SIFT	4264	638	86.98
		MOG-SIFT	4124	181	***95.80***
trees	blur	SIFT	5808	893	86.67
		GO-SIFT	5399	704	88.46
		MOG-SIFT	4978	377	***92.79***
leuven	illumination	SIFT	6634	563	92.18
		GO-SIFT	6810	317	95.55
		MOG-SIFT	6472	131	***98.02***
ubc	JPEG	SIFT	9865	695	93.42
		GO-SIFT	9216	317	96.67
		MOG-SIFT	8954	161	***98.23***

[a] Column T denotes transformations between image pairs.
[b] Column #True and column #False indicate the number of true matches and false matches separately.
[c] The highest accuracy in the *Accuracy* column is shown in the italicized bold font.

Figure 3.10. First, among the three techniques compared (SIFT, GO-SIFT, MOG-SIFT), MOG-SIFT achieves the highest matching accuracy in the overwhelming majority of cases. Second, in most cases for each original image the matching accuracy decreases from image pair 1 to 5. To further illustrate the improvement MOG-SIFT has achieved, matching results for GO-SIFT and MOG-SIFT are compared in Figure 3.11 and Figure 3.12. Due to the original number of matches is quite large as shown in Figure 3.11, keypoint matches are down-sampled by 10 times in Figure 3.12 so that it is easier for readers to visually evaluate the correctness of the correspondences identified.



(a) bark (scale+rotation)

(b) boat (scale+rotation)

(c) graffiti (viewpoint)

(d) wall (viewpoint)

(e) bikes (blur)

(f) trees (blur)

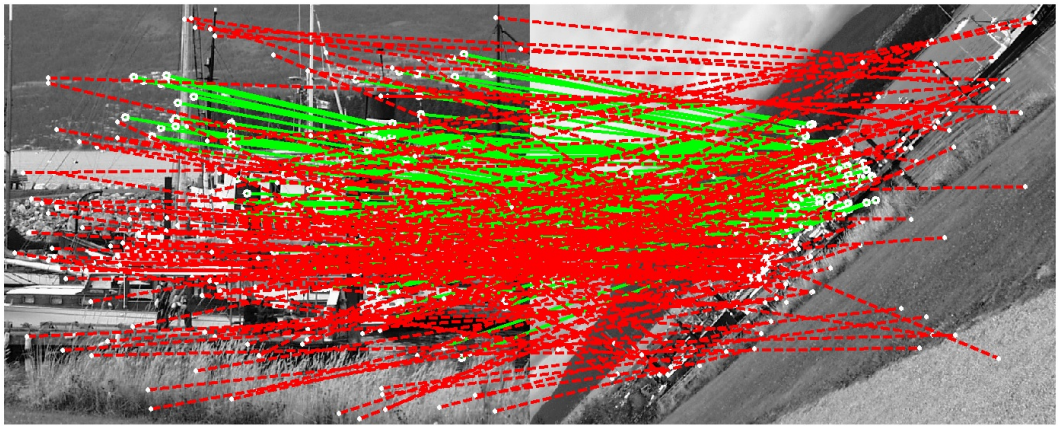
(g) leuven (illumination)

(h) ubc (JPEG)

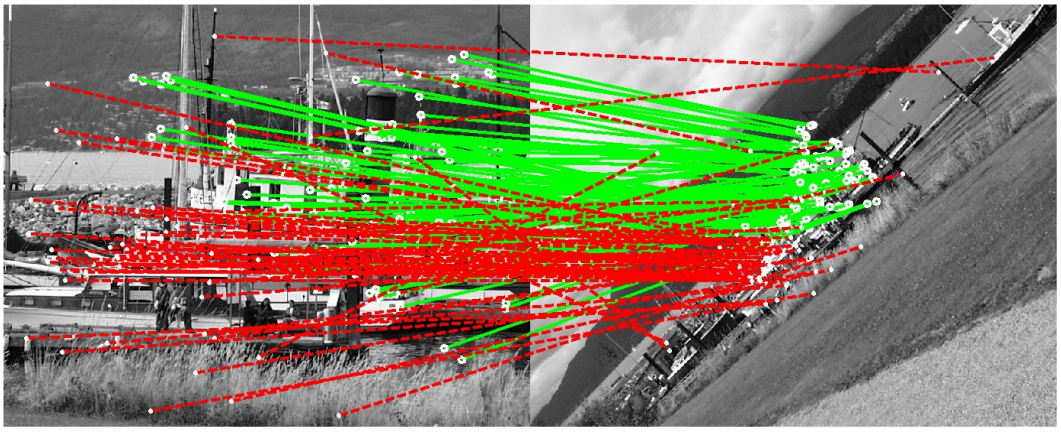
Figure 3.10: Matching Accuracy for Each Base Pair of Affine Covariant Regions Data Set

(a) Reference Image

(b) Target Image

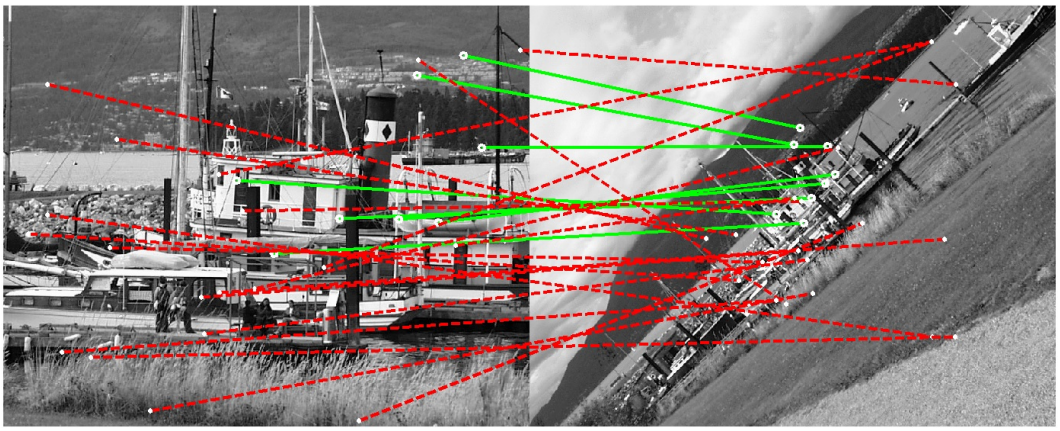


(c) GO-SIFT (matching accuracy: 118/268=44.03%)

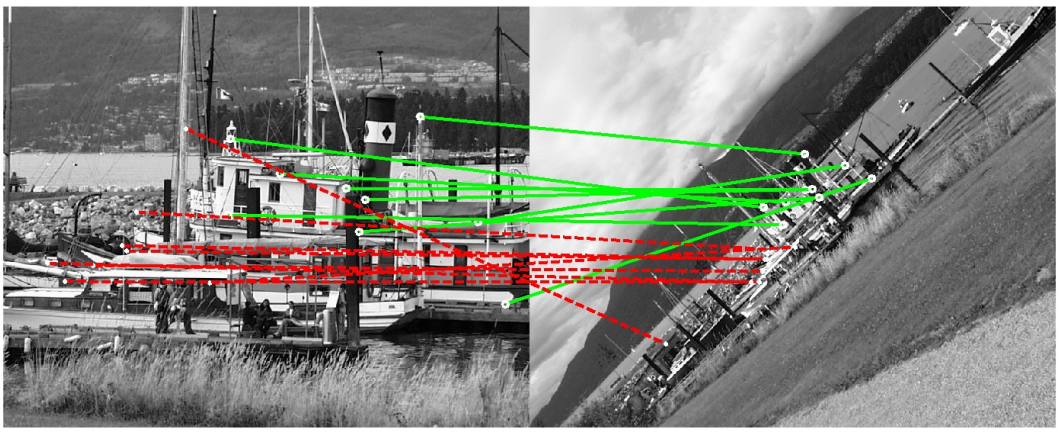


(d) MOG-SIFT (matching accuracy: 102/166=61.45%)

Figure 3.11: A Matching Example for comparing GO-SIFT with MOG-SIFT. Green and red lines indicate true and false matches respectively.



(a) GO-SIFT (matching accuracy: 8/27=29.63%)



(b) MOG-SIFT (matching accuracy: 9/17=52.94%)

Figure 3.12: Down-sampled Matches for boat 1 to 6. The matches of both GO-SIFT and MOG-SIFT are down-sampled by 10:1.

Table 3.5 presents the number of true matches, the number of false matches and matching accuracy for the Mono-modal Microscopic Data Set. The results of seven image pairs for each original image are averaged in Table 3.5. On average, the matching accuracy for all the pairs in this data set has improved from 62.94% (SIFT) to 75.71% using GO-SIFT, then to 90.81% using MOG-SIFT. Figure 3.13 shows the changes to matching accuracy as the rotation difference increases for each original image. It can be seen from Figure 3.13 that rotation changes hardly affect the matching performance.

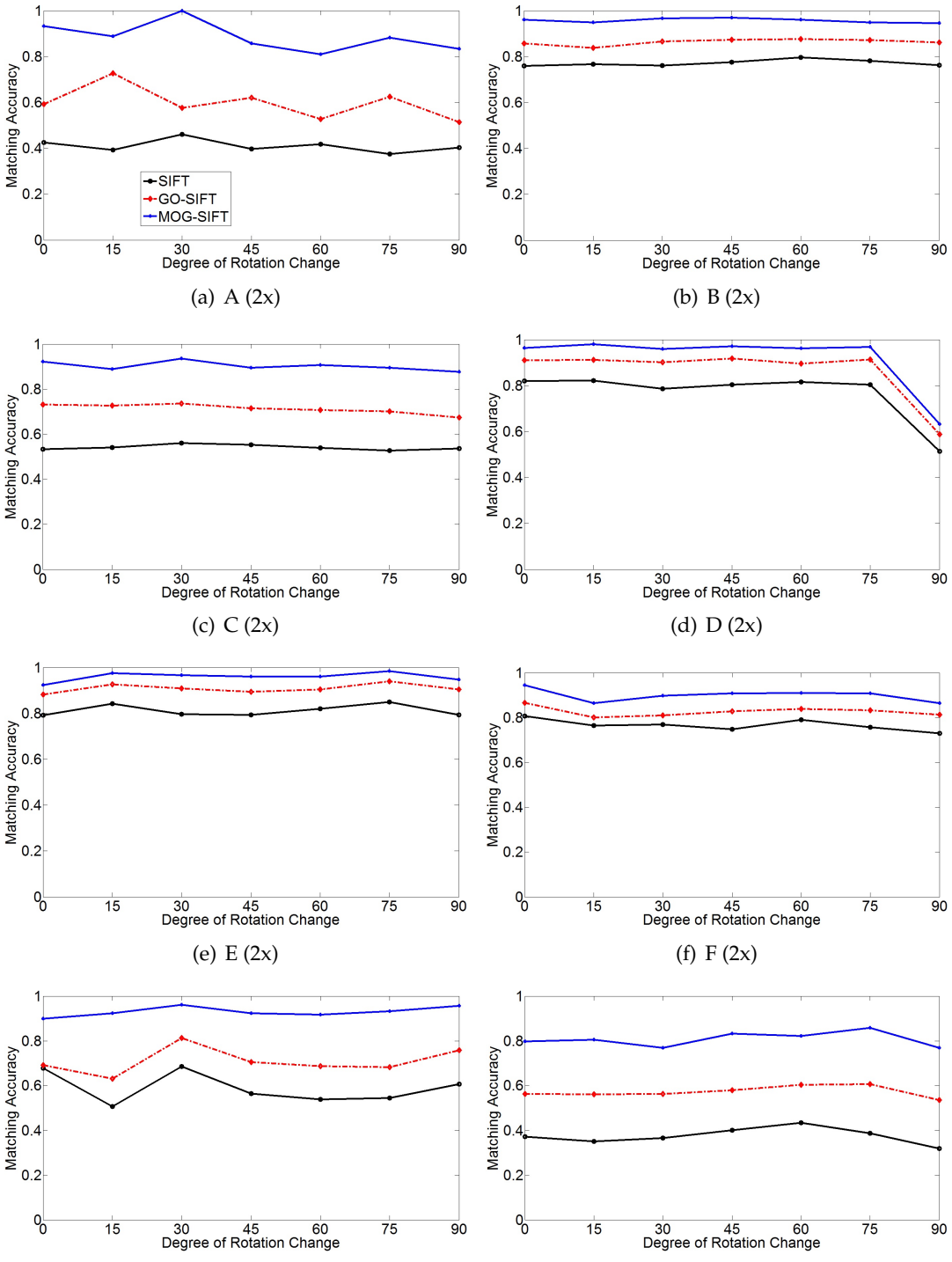
Table 3.5: Matching Accuracies for Mono-modal Microscopic Images

Specimen[a]	Δ_σ^b	**Technique**	**#True**	**#False**	**Accuracy(%)**
A	2*x*	SIFT	160	230	41.03
		GO-SIFT	117	82	58.79
		MOG-SIFT	110	15	***88.00***
B[c]	2*x*	SIFT	3442	1015	77.23
		GO-SIFT	3351	529	86.37
		MOG-SIFT	3245	144	***95.75***
C	2*x*	SIFT	1324	1121	54.15
		GO-SIFT	1277	515	71.26
		MOG-SIFT	1230	133	***90.24***
D	2*x*	SIFT	4996	1518	76.70
		GO-SIFT	4798	764	86.26
		MOG-SIFT	4713	411	***91.98***
E	2*x*	SIFT	1034	239	81.23
		GO-SIFT	929	93	90.90
		MOG-SIFT	860	36	***95.98***
F	2*x*	SIFT	898	274	76.62
		GO-SIFT	777	162	82.75
		MOG-SIFT	742	82	***90.05***
A	4*x*	SIFT	258	184	58.37
		GO-SIFT	173	72	70.61
		MOG-SIFT	172	13	***92.97***
D[d]	4*x*	SIFT	877	1467	37.41
		GO-SIFT	841	626	57.33
		MOG-SIFT	804	192	***80.72***

[a] The first column denotes labels of the original images.
[b] The second column gives the scale difference between image pairs.
[c] The specimen corresponds to the two images in Figure 3.6 (a) and (b).
[d] The specimen corresponds to the two images in Figure 3.6 (c) and (d).

3.4.4 Experiments on Multi-modal Images

There are four data sets for multi-modal images, as listed in Table 3.3. Three registration techniques IS-SIFT, GO-IS-SIFT and MOG-IS-SIFT have been compared. We firstly look into the comparisons in terms of matching accuracy between the three techniques on Data Sets 1-3, as shown in Figure 3.14. In registering two artificial image pairs from Data Set 1, on average, the matching accuracies achieved by IS-SIFT, GO-IS-SIFT and MOG-IS-SIFT are 93.79%, 94.76% and 96.04% respectively. It is not difficult to effectively register the two image pairs as the content difference at corresponding parts only lies in contrast reversal. For Data Set 2, the average



(a) A (2x) (b) B (2x)

(c) C (2x) (d) D (2x)

(e) E (2x) (f) F (2x)

(g) A (4x) (h) D (4x)

Figure 3.13: Rotation Changes vs Matching Accuracy for Mono-modal Microscopic Images

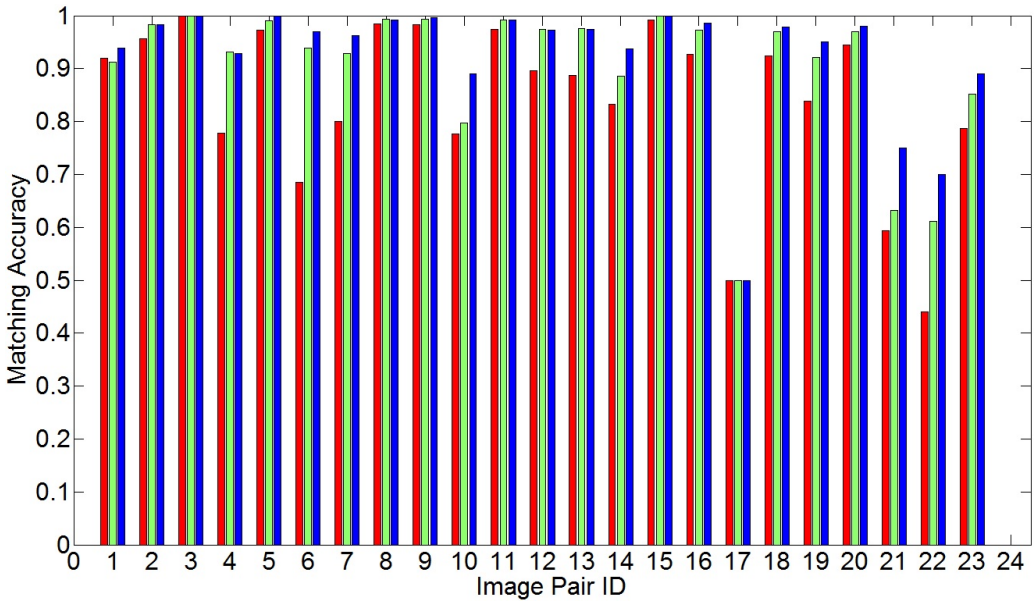
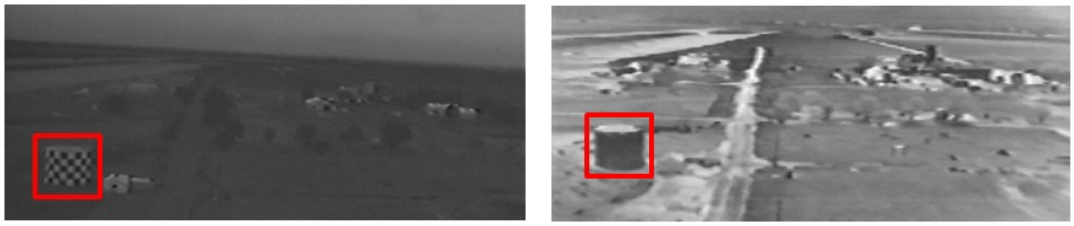


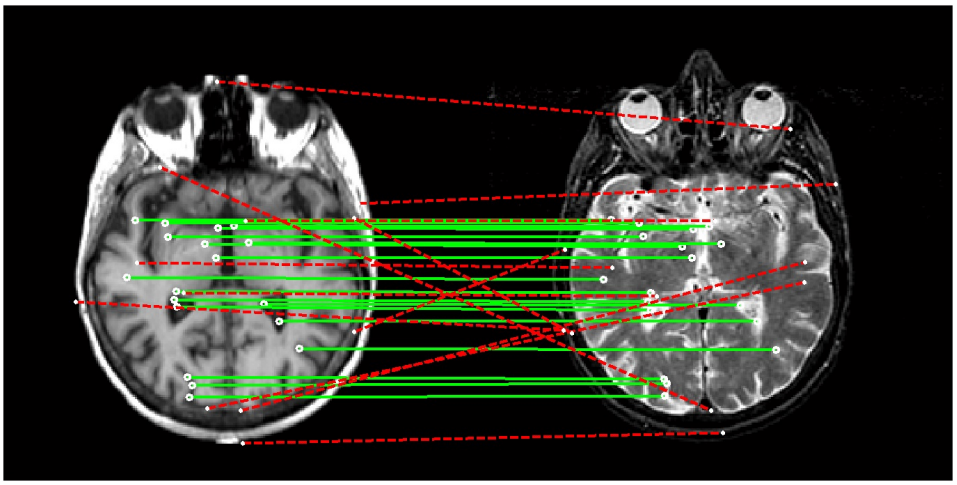
Figure 3.14: Matching Accuracy for Multi-modal Image Pairs (Non-microscopic)

matching accuracy improves from 87.20% (IS-SIFT) to 92.93% using GO-IS-SIFT, then to 94.50% using MOG-IS-SIFT. Considering Data Sets 1 and 2 collectively, the improvements of MOG-IS-SIFT over IS-SIFT and GO-IS-SIFT are small. However, for 17 out of 20 image pairs in Data Sets 1 and 2, the matching accuracies are over 90.00% when using GO-IS-SIFT, which can be further improved using a technique for refining keypoint matches such as RANSAC [23] discussed in Section 2.7 of Chapter 2. As a result, an estimated transformation in each image pair should be sufficiently accurate in these cases. As shown in Figure 3.14, matching accuracies for pairs 10, 14 and 17 are below 90.00%. For pairs 10 and 14, matching accuracy is improved by 9.33% and 5.19% respectively, when using MOG-IS-SIFT. Only for pair 17, MOG-IS-SIFT does not make any improvement over IS-SIFT and GO-IS-SIFT.

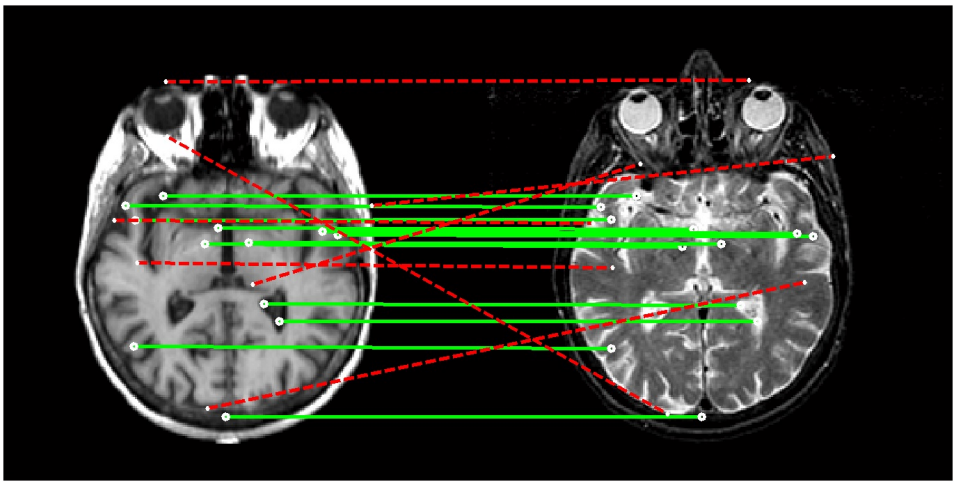


(a) EO (b) IR

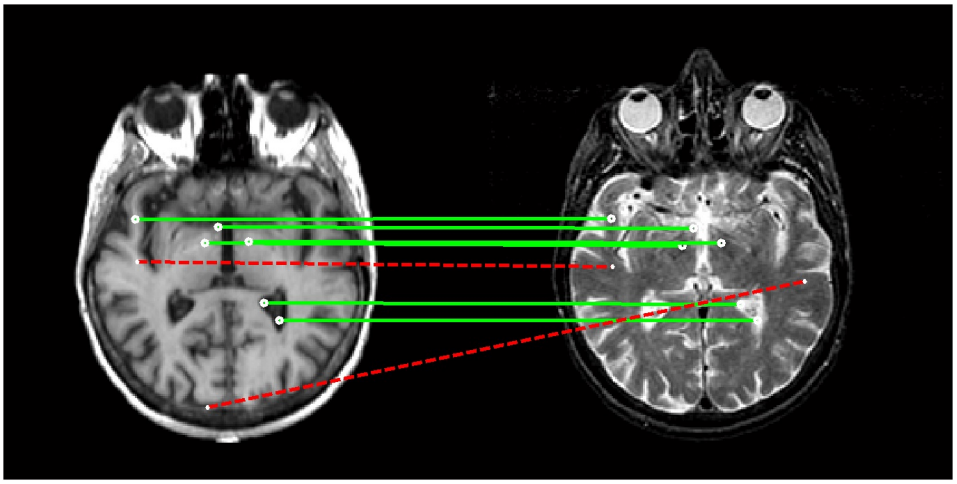
Figure 3.15: Image Pair 24. The highlighted regions are corresponding.



(a) IS-SIFT (19/32=59.38%)



(b) GO-IS-SIFT (12/19=63.16%)



(c) MOG-IS-SIFT (6/8=75.00%)

Figure 3.16: Keypoint Matches Achieved by IS-SIFT, GO-IS-SIFT and MOG-IS-SIFT on Pair 21

For Data Set 3, i.e. pairs 21 to 24 as shown in Figure 3.14, we firstly discuss pair 24 in which the matching accuracy achieved by each of the three registration techniques is all 0.00%. It can be seen in Figure 3.15 that the objects in the two images are very unclear and that the content differences are very large even at corresponding regions as highlighted. In registering image pair 24, IS-SIFT determines only one match and it is a false match, whereas there is no match when using GO-IS-SIFT. In this case, MOG-IS-SIFT can do nothing to improve the situation as MOG-IS-SIFT is designed to find common matches which are determined by both IS-SIFT and GO-IS-SIFT. With regard to the three MRI image pairs in Data Set 3, the average matching accuracies obtained by IS-SIFT, GO-IS-SIFT and MOG-IS-SIFT are 60.67%, 69.82% and 78.01% respectively. The keypoint matches achieved by the three registration techniques for pair 21 are shown in Figure 3.16.

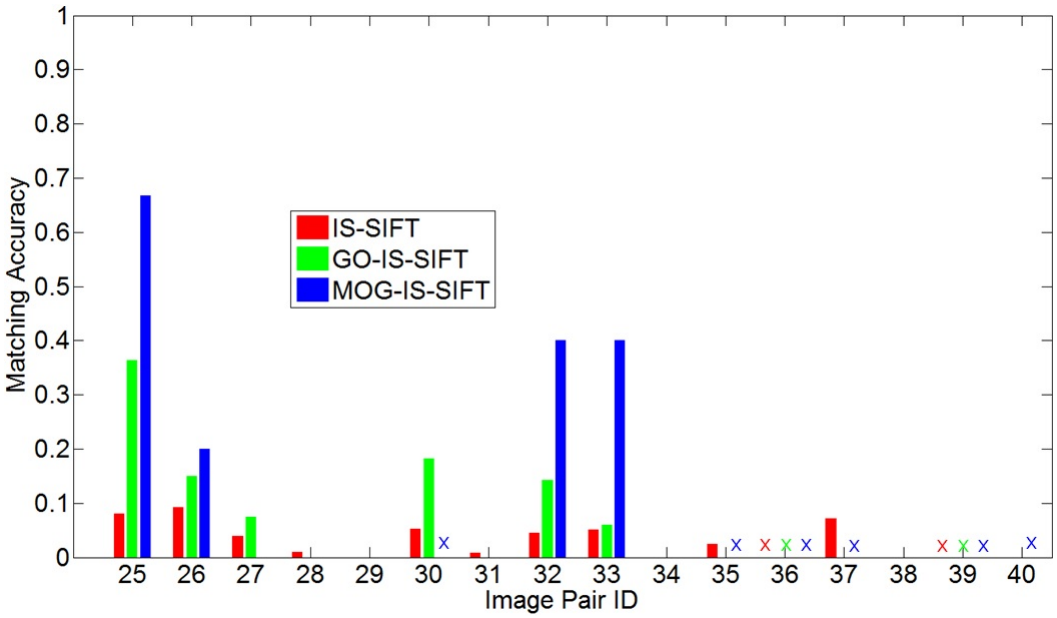
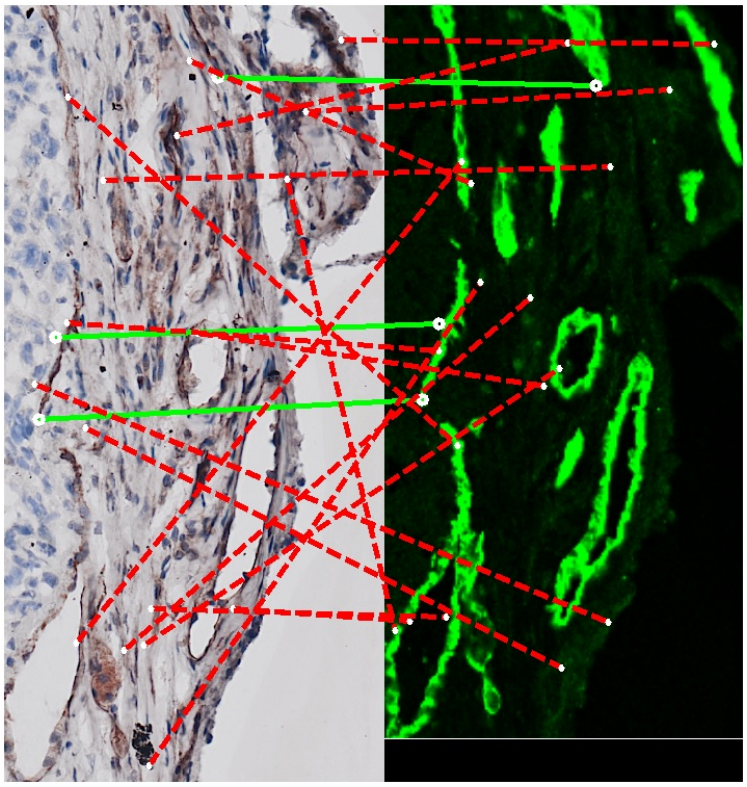


Figure 3.17: Matching Accuracy for Multi-modal Image Pairs (Microscopic)

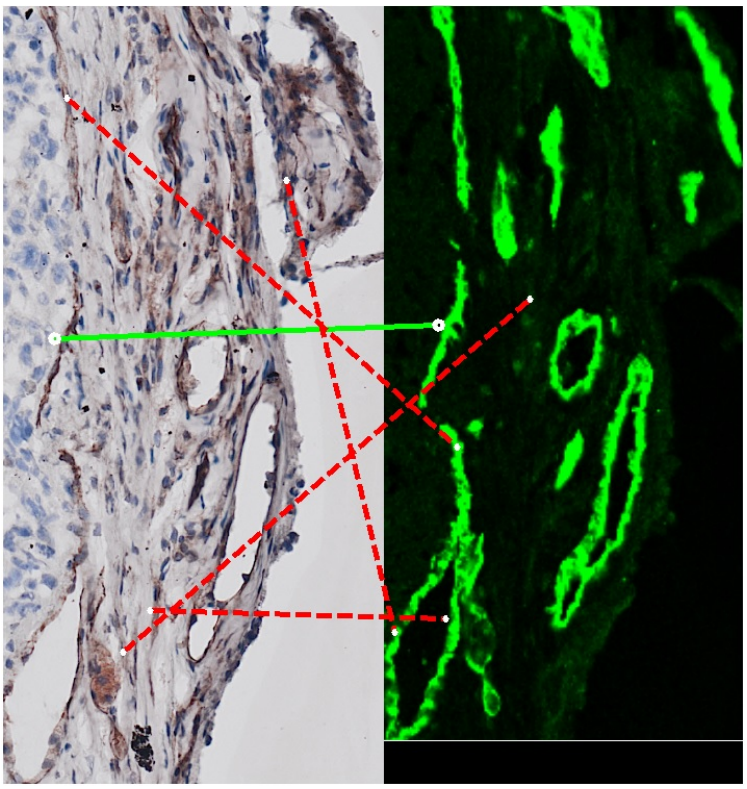
For Data Set 4, the matching results for the 16 multi-modal microscopic image pairs are presented in Table 3.6. The *Specimen* column specifies which specimen each image pair belongs to. Overall, all the three registration techniques, IS-SIFT, GO-IS-SIFT and MOG-IS-SIFT, perform a lot worse in registering the 16 image pairs as compared to registering image pairs in Data Sets 1-3. The poor performance simply verifies the fact that the image characteristics in multi-modal microscopic images are

Table 3.6: Matching Accuracies for Multi-modal Microscopic Images

Pair ID	Specimen	Technique	#True	#False	Accuracy (%)
25	A	IS-SIFT	5	57	8.06
		GO-IS-SIFT	4	7	36.36
		MOG-IS-SIFT	2	1	***66.67***
26	A	IS-SIFT	5	49	9.26
		GO-IS-SIFT	3	17	15.00
		MOG-IS-SIFT	1	4	***20.00***
27	A	IS-SIFT	4	98	3.92
		GO-IS-SIFT	2	25	***7.41***
		MOG-IS-SIFT	0	7	0.00
28	A	IS-SIFT	1	98	***1.01***
		GO-IS-SIFT	0	29	0.00
		MOG-IS-SIFT	0	5	0.00
29	A	IS-SIFT	0	77	0.00
		GO-IS-SIFT	0	29	0.00
		MOG-IS-SIFT	0	5	0.00
30	A	IS-SIFT	2	36	5.26
		GO-IS-SIFT	2	9	18.18
		MOG-IS-SIFT	2	0	***100.00***
31	A	IS-SIFT	1	118	***0.84***
		GO-IS-SIFT	0	23	0.00
		MOG-IS-SIFT	0	6	0.00
32	B	IS-SIFT	2	43	4.44
		GO-IS-SIFT	2	12	14.29
		MOG-IS-SIFT	2	3	***40.00***
33	C	IS-SIFT	6	111	5.13
		GO-IS-SIFT	4	63	5.97
		MOG-IS-SIFT	4	6	***40.00***
34	C	IS-SIFT	0	82	0.00
		GO-IS-SIFT	0	50	0.00
		MOG-IS-SIFT	0	10	0.00
35	D	IS-SIFT	1	40	***2.44***
		GO-IS-SIFT	0	11	0.00
		MOG-IS-SIFT	0	1	0.00
36	E	IS-SIFT	0	1	0.00
		GO-IS-SIFT	1	1	***50.00***
		MOG-IS-SIFT	0	1	0.00
37	E	IS-SIFT	1	13	***7.14***
		GO-IS-SIFT	0	10	0.00
		MOG-IS-SIFT	0	0	0.00
38	E	IS-SIFT	0	80	0.00
		GO-IS-SIFT	0	38	0.00
		MOG-IS-SIFT	0	6	0.00
39	F	IS-SIFT	0	0	0.00
		GO-IS-SIFT	0	0	0.00
		MOG-IS-SIFT	0	0	0.00
40	F	IS-SIFT	0	15	0.00
		GO-IS-SIFT	0	9	0.00
		MOG-IS-SIFT	0	1	0.00



(a) GO-IS-SIFT (3/20=15.00%)



(b) MOG-IS-SIFT (1/5=20.00%)

Figure 3.18: Keypoint Matches Achieved by GO-IS-SIFT and MOG-IS-SIFT on Pair 26

much more complex than pairs in Data Sets 1-3 for the registration purpose. First of all, it should be emphasized that at least three matches are needed to estimate a transformation between two images to do the final alignment. In the 16 image pairs, the number of matches is no smaller than three in registering 10 pairs. The matching accuracies achieved by the three registration techniques for these 10 pairs are compared in Figure 3.17. Note that in Figure 3.17, if the number of matches is smaller than three, an image registration cannot be achieved and $\times$ marks the six image pairs where this is the case. The following discusses the 10 pairs where the number of matches is no smaller than three: pairs 25, 26, 27, 28, 29, 31, 32, 33, 34 and 38. As Table 3.6 suggests, on average, the matching accuracy for IS-SIFT, GO-IS-SIFT and MOG-IS-SIFT increases from 3.27% to 7.90%, and to 16.67%. More specifically, the 10 pairs are divided into two categories according to whether there is any true match for MOG-IS-SIFT. The first category includes pairs 25, 26, 32 and 33, where there is at least one true match, whereas pairs 27, 28, 29, 31, 34 and 38 without any true match are included in the second category. For the four pairs in the first category, the average matching accuracies for IS-SIFT, GO-IS-SIFT and MOG-IS-SIFT are 6.72%, 17.90% and 41.67%. It is clear that MOG-IS-SIFT has made large improvements over both IS-SIFT and GO-IS-SIFT in matching accuracy, by 34.95% and 23.77% respectively. Since MOG-IS-SIFT is proposed for seeking the common matches which are determined by both IS-SIFT and GO-IS-SIFT, it is possible that the number of true matches is zero for MOG-IS-SIFT in the cases where the number of true matches for IS-SIFT and GO-IS-SIFT are very small. With this analysis in mind, the matching results for the six pairs in the second category make sense, as shown in Table 3.6. In fact, 7.41% is the highest matching accuracy for the six pairs in the second category using IS-SIFT or GO-IS-SIFT, which is too low to achieve an effective registration. Also, we have shown keypoint matches achieved by GO-IS-SIFT and MOG-IS-SIFT in Figure 3.18 for registering image pair 26. To summarize, MOG-IS-SIFT has shown improvements over IS-SIFT and GO-IS-SIFT as long as three matches are determined by the three registration techniques.

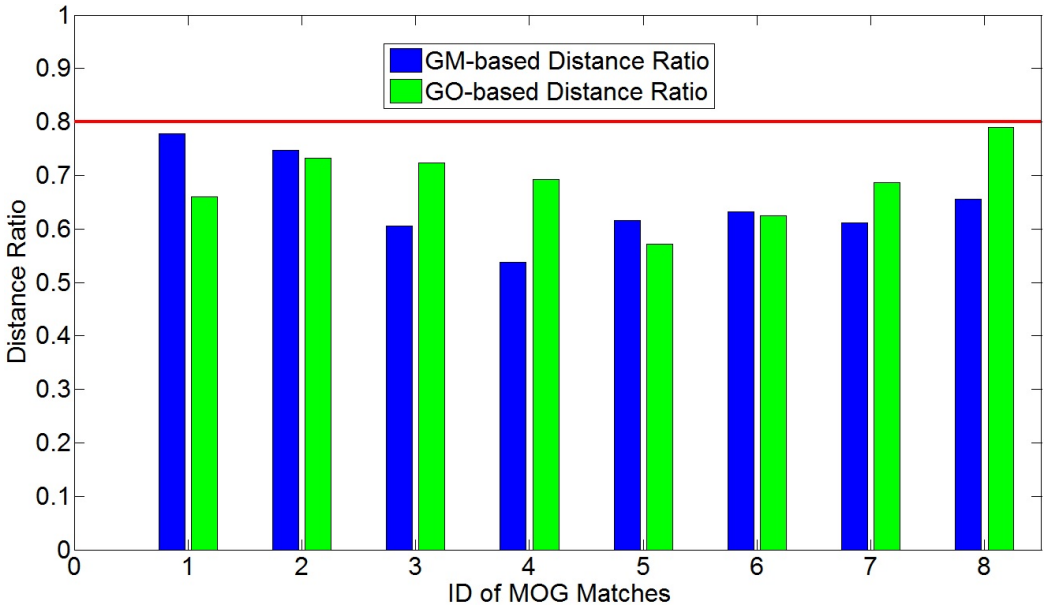


Figure 3.19: Illustrating Characteristics of MOG Matches. The *Distance Ratio* at y axis refers to the distance ratio between the closest neighbor and the second closest neighbor. The horizontal line denotes the threshold of distance ratio pre-defined. *GM-based* and *GO-based* correspond to IS-SIFT and GO-IS-SIFT respectively.

3.4.5 An Example Illustrating Distance Ratios of MOG Matches

As stated in Section 3.3.2, MOG matches satisfy the matching criteria by both IS-SIFT descriptors and GO-IS-SIFT descriptors. Here, we illustrate characteristics of MOG matches using an example of registering a particular multi-modal image pair. The keypoint matches determined by IS-SIFT, GO-IS-SIFT and MOG-IS-SIFT can be found in Figure 3.16. Figure 3.19 clearly shows that, the distance ratio between the closest neighbor and the second closest neighbor is below the threshold pre-defined for each MOG match. Thus, a MOG match is a true one with a higher possibility, as compared to those matches which are determined by IS-SIFT or GO-IS-SIFT, but not determined by MOG-IS-SIFT. As Figure 3.16 shows, MOG-IS-SIFT improves IS-SIFT and GO-IS-SIFT by 15.62% and 11.84%, respectively, in matching accuracy.

3.5 Summary

We have proposed a technique called MOG to utilize both Gradient Magnitudes (GM) and Gradient Occurrences (GO) for feature description and matching in SIFT-based

registration techniques. The proposed MOG can be incorporated with both SIFT and IS-SIFT for registering mono-modal and multi-modal images respectively. We believe that the idea of utilizing both GM and GO in feature description and matching can be broadly applied to SIFT-like descriptors.

It is noted that MOG-SIFT has been proposed on the basis of SIFT for mono-modal image registration. We believe that MOG can be applied to some variants of SIFT, such as PCA-SIFT [43] and GLOH [66]. It is likely that the registration performance can be improved accordingly.

In general, our experiments have shown that MOG improves SIFT-like descriptors in both mono-modal and multi-modal cases. However, SIFT-like descriptors may not be suitable for registering multi-modal microscopic images as contents in these images vary greatly. Based on our analysis, this is caused by a low structural similarity and substantial content differences in these images. Thus, we will explore other techniques in Chapters 4 and 5, in order that multi-modal microscopic images can be effectively registered.

Chapter 4

Detection of Structural Similarity for Multi-modal Microscopic Image Registration

4.1 Introduction

From the experimental study on multi-modal data sets in Section 3.4.4 of Chapter 3, it is clear that achieving an effective registration for multi-modal microscopic image pairs is more challenging as compared to the other pairs in the test data. By looking into visual characteristics, we have found that there exists much lower structural similarity in multi-modal microscopic image pairs than other tested image pairs. Based on our analysis, we will conclude that a low structural similarity has a huge impact on the registration process and significantly decreases the registration performance. Unfortunately, many existing multi-modal image registration techniques such as [17, 19, 32, 33, 38, 39, 56, 66, 68, 70, 106] assume that the same or similar structures exist at corresponding parts between two images. With this assumption, these existing multi-modal image registration techniques cannot achieve a satisfactory performance in registering multi-modal microscopic images.

Due to the great significance of structural similarity to the registration process, we will propose a technique to detect intrinsic structural similarity between color and confocal microscopic images. The proposed technique is called Detector of Structural Similarity (DSS in short). DSS increases the structural similarity between the color and confocal images by exploiting the characteristics in intensity relationships between the RGB color channels and detecting structures of interest. Compared to the original

multi-modal microscopic images, the structural similarity is a lot higher in the microscopic images after being processed by the proposed DSS technique.

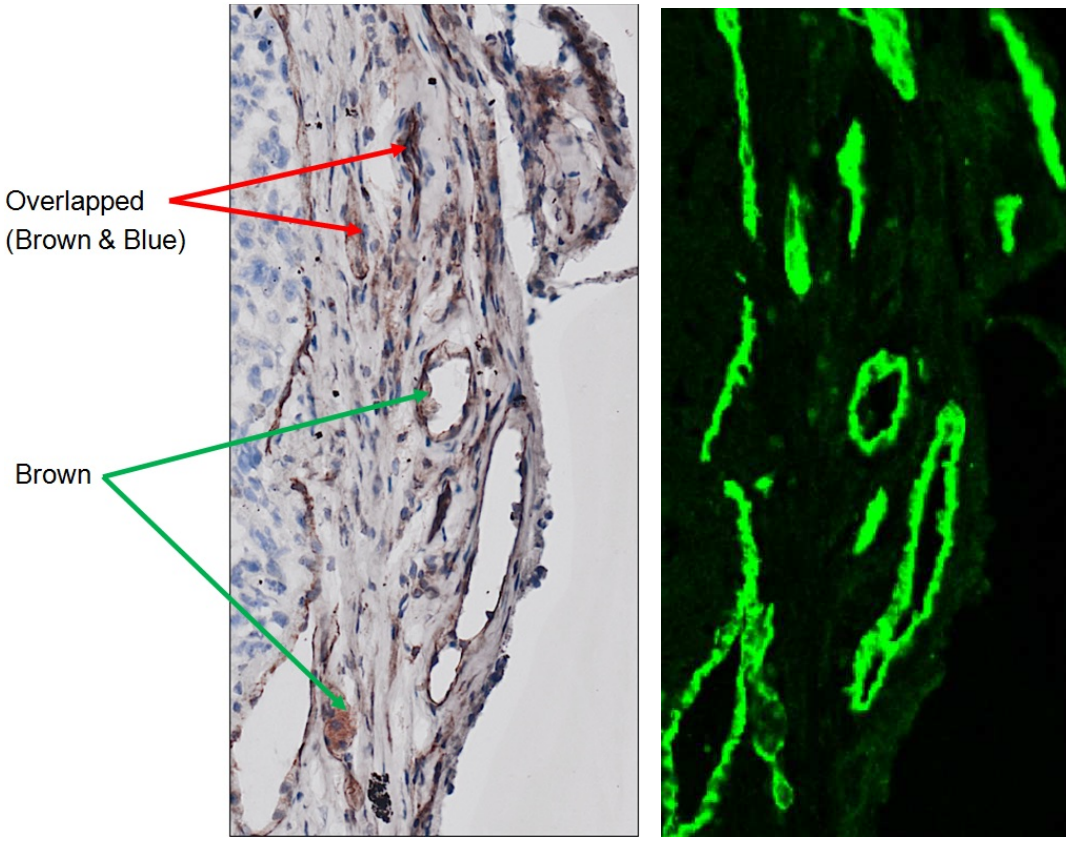
The rest of the chapter is outlined as follows. In Section 4.2, structures of interest in color and confocal images are identified. Section 4.3 discusses the low structural similarity between color and confocal images; it also points out the reasons for this and analyzes its significance to image registration. In Section 4.4, we will describe our proposed DSS. Section 4.5 illustrates the significance of a critical parameter in DSS, while Section 4.6 proposes a way of adaptively selecting the parameter. In Section 4.7, we will demonstrate and discuss our experimental results. Finally, the chapter is summarized in Section 4.8.

4.2 Structures of Interest in Multi-modal Microscopic Images

In this section, we will identify structures of interest in multi-modal microscopic images. Due to different characteristics, structures of interest in color and confocal images are analyzed in different ways. Given a color or confocal image, there are structures of interest and structures of no interest. For the referencing purpose, structures of interest and structures of no interest are called SOI and non-SOI respectively.

4.2.1 Structures of Interest in Color Images

Figure 4.1 shows a pair of color and confocal images. Each of the two images is part of a tissue section. Brown and blue structures in the color image as well as green structures in the confocal image represent cells in the tissue [81]. In the color image shown in Figure 4.1(a), *Brown Structures* should be appropriately detected as SOIs, whereas the blue structures and background pixels should be eliminated. By doing so, the detected structures in the color image will correspond to SOIs in the confocal image shown in Figure 4.1(b). Observing the color image shown in Figure 4.1(a) closely, another group of image structures exist to correspond to structures in the confocal image. We call these structures *Brown/Blue Overlapped Structures*, as illustrated in Figure 4.1(a). Specifically, *Brown/Blue Overlapped Structures*



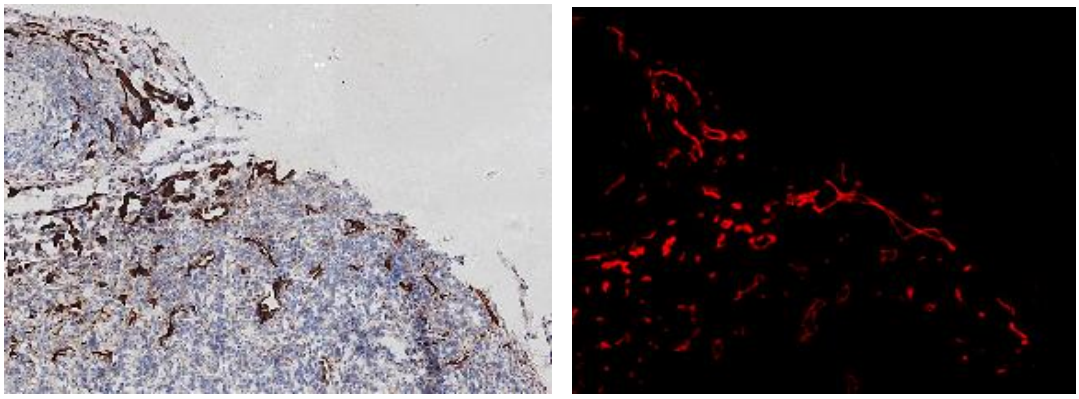
(a) SOIs in Color Image (b) Confocal Image

Figure 4.1: Structures of Interest in a Color Image

refer to those pixels where their intensities at the blue channel are slightly higher than their intensities at the red channel. These structures are generally enclosed by brown structures and their counterparts in the confocal image are SOIs. Thus, our aim is to detect both *Brown Structures* and *Brown/Blue Overlapped Structures* in the color image.

4.2.2 Structures of Interest in Confocal Images

In the confocal image shown in Figure 4.1(b), image contents are a lot simpler as compared to the corresponding color image. The only issue is to eliminate background pixels as we believe these pixels are not useful in feature description and matching for registration purposes. In addition to confocal images such as Figure 4.1(b), the test data includes confocal images with different characteristics such as the one shown in Figure 4.2(b). Figure 4.1(b) and Figure 4.2(b) represent two



(a) Color (b) Confocal

Figure 4.2: An Example of Multi-modal Microscopic Images

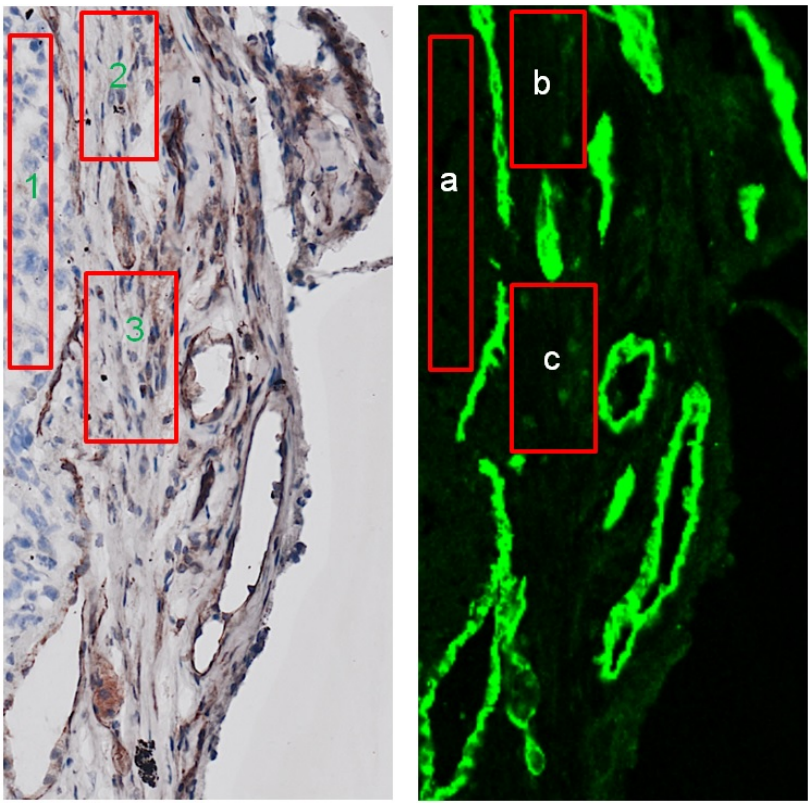
categories of confocal images, where image information only lies at the green channel and red channel respectively.

4.3 Low Structural Similarity in Multi-modal Microscopic Images

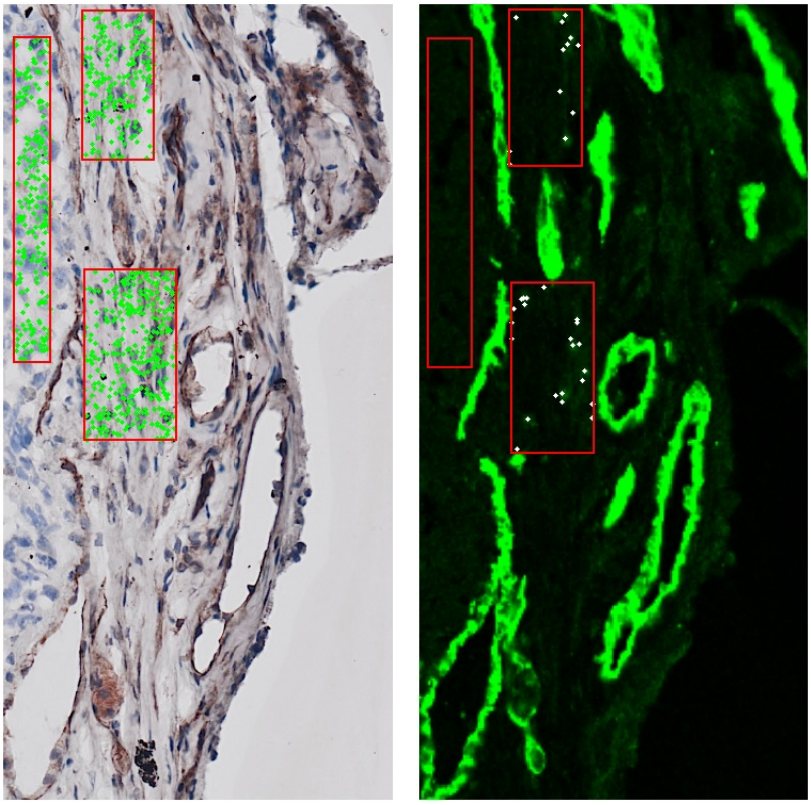
This section describes the low structural similarity in multi-modal microscopic images. First, low structural similarity is illustrated through an example of multi-modal microscopic images. Next, the root of low structural similarities is pointed out. Thirdly, we will analyze how a low structural similarity affects the registration process.

4.3.1 Low Structural Similarity

A pair of multi-modal microscopic images is shown in Figure 4.3(a) and (b). Visually, the structural similarity is very low between the two images. Due to different staining and capturing techniques in the two images [42, 85], it is known that brown structures in the color image should correspond to green structures in the confocal image. Moreover, the color image contains some blue structures, but these structures do not appear in the confocal image.



(a) Color (b) Confocal



(c) Color (d) Confocal

Figure 4.3: Illustrating Low Structural Similarity in Multi-modal Microscopic Images

In Figure 4.3(a) and (b), the low structural similarity is illustrated by highlighting a few corresponding regions in the color and confocal images. The three regions 1, 2 and 3 in the color image correspond to regions a, b and c in the confocal image, respectively. It can be clearly seen that each of the three regions in the color image contains a large amount of image information. In contrast, there is much less information in each of corresponding regions in the confocal image. Figure 4.3(c) and (d) show the keypoints within the highlighted regions, where the keypoints are detected using [19]. In the color image, the number of keypoints within regions a, b and c are 200, 171 and 300, respectively. In striking contrast, the number of keypoints within regions 1, 2 and 3 in the confocal image are 0, 12 and 24. The comparison between the number of keypoints at corresponding regions also illustrates a low structural similarity between the two images.

4.3.2 What Causes Low Structural Similarity?

The two images shown in Figure 4.3(a) and (b) are captured by a light microscope and a confocal microscope respectively. For the referencing purpose, images captured using a light microscope and a confocal microscope are called color and confocal images respectively. The terms, *color* and *confocal*, are used for microscopic image pairs in the entire thesis. The low structural similarity appearing in the images such as Figure 4.3(a) and (b) is caused by different staining techniques which are used in two types of microscopes.

The low structural similarity in a pair of color and confocal images is caused by differences in light filtering and staining which are used for the two types of microscopes. In a light microscope, an entire specimen is exposed to visible light, while in a confocal microscope only a single point is illuminated at a time to exclude unwanted scattering of light [81, 82]. Consequently, different types of signals are mixed in a color image, while these signals are discriminated in a confocal image.

Staining is a technique used to enable better visualization of cells in the acquisition of microscopic images [107]. Specifically, a staining pattern in our application is generated by binding antibodies to molecules of different colors that are present on certain cells. To generate two images like the ones shown in Figure 4.3(a) and

(b), one tissue is used and stained with two different antibodies. The antibodies in Figure 4.3 (a) are visible under the range within the visible color spectrum. By comparison, in Figure 4.3 (b) the antibodies are only visible under the wavelength range within the laser spectrum. Moreover, another reason leading to partially similar structures is that different types of antibodies present different sensitivities of the binding to molecules. Therefore, blue portions in Figure 4.3 (a) are clear but their corresponding portions in Figure 4.3 (b) are not, leading to partially similar structures between the two images.

4.3.3 Significance of Low Structural Similarity to Image Registration

Intuitively, the lower the structural similarity between two images is, the more challenging the registration process will be. We will analyze the significance of the low structural similarity to image registration from the following two aspects: the negative impact of non-SOIs on feature matching and descriptor distances between corresponding keypoints.

First, we look into how non-SOIs in a color image have a negative impact on the stage of feature matching. As shown in Figure 4.3(a) and (b), the color image contains a great deal of non-SOIs which do not appear in the corresponding confocal image. Let us first denote the keypoints in the confocal image as

$$P_r = \{P_r^1, P_r^2, \dots P_r^{N_r}\}. \tag{4.1}$$

In the color image, structures are divided into two categories: SOIs and non-SOIs, according to whether or not there are corresponding structures in the confocal image. In the color image, the keypoints which fall into SOIs and non-SOIs are denoted as

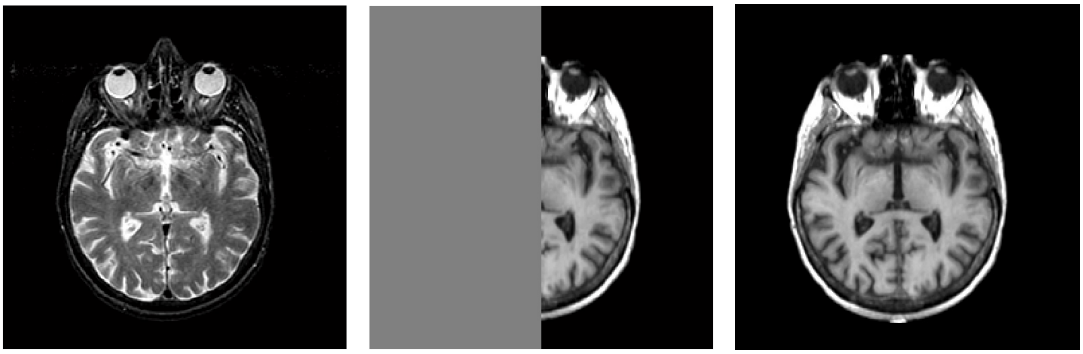
$$P_{t1} = \{P_{t1}^1, P_{t1}^2, \dots P_{t1}^{N_{t1}}\} \tag{4.2}$$

and

$$P_{t2} = \{P_{t2}^1, P_{t2}^2, \dots P_{t2}^{N_{t2}}\} \tag{4.3}$$

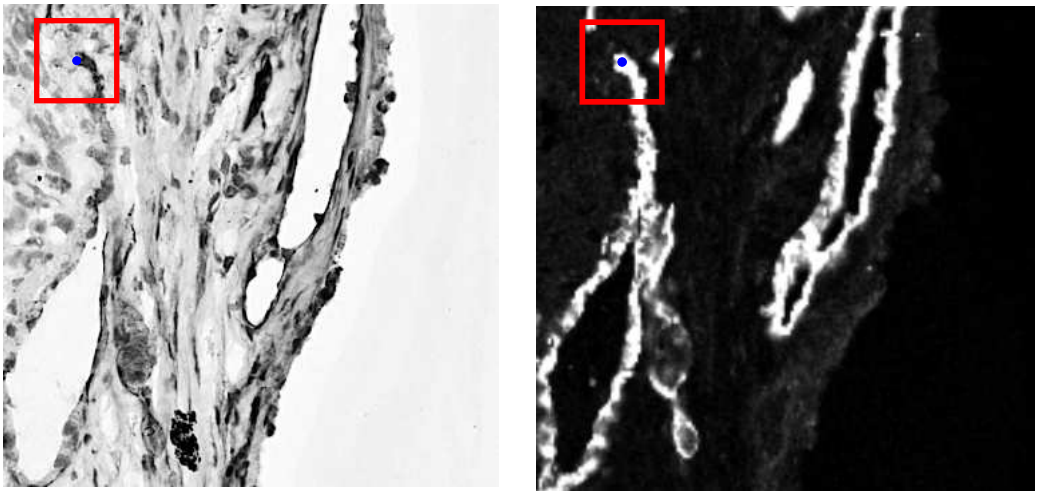
respectively. For a keypoint P_r^i in the confocal image, its true match in the color image should come from P_{t1} as P_{t1} represents the set of keypoints which are detected in SOIs.

The true match of P_r^i is denoted as P_{t1}^j. However, the true match $P_r^i \mapsto P_{t1}^j$ might be hindered due to keypoints from P_{t2} in two different scenarios as follows. First, there may exist a keypoint, P_{t2}^j, which is the closest neighbor to P_r^i, but P_{t2}^j is detected from non-SOIs. Second, P_{t1}^j is the closest neighbor to P_r^i, but a keypoint, P_{t2}^k, from non-SOIs is visually similar to P_{t1}^j. Consequently, P_{t1}^j is not sufficiently distinctive so that P_{t1}^j is not matched to P_r^i.



(a) (b) (c)

Figure 4.4: An Example of Non-overlapping



(a) Color (b) Confocal

Figure 4.5: An Example of Low Structural Similarity at Corresponding Keypoints

It is noticeable that the aforementioned non-SOIs should be differentiated from the non-overlapping issue in image registration. In registering a pair of color and confocal images, there exist non-SOIs even if two entire images are overlapping, such

as the two images shown in Figure 4.1. Figure 4.4 illustrates a different scenario from non-SOIs existing in color and confocal images. In Figure 4.4, image (b) is the right half of image (c). The image pair (a) and (c) in Figure 4.4 are overlapping, however there is no non-SOI between the two images.

Second, descriptor distances of corresponding keypoints can be affected by a low structural similarity between the color and confocal images. Figure 4.5 illustrates an example of low structural similarity between two corresponding keypoints. Let D_A and D_B denote the descriptors built in the two regions which are marked in Figure 4.5. Note that the sizes of the two regions marked in Figure 4.5 are proportional to the actual scale difference between the two images in order to eliminate the impact of the accuracy of scale estimation. Given corresponding keypoints with a low structural similarity, different amounts of image contents exist, as clearly shown in Figure 4.5. From the perspective of image registration, the image contents in the two regions that are not visually corresponding are regarded as noises. Hence, the corresponding descriptors built within the two regions will not be close no matter how discriminative the local descriptor itself is. A large descriptor distance between D_A and D_B increases the likelihood of rejecting a true match in the following two possibilities: 1) D_B is not the closest neighbor to D_A; 2) D_B is the closest neighbor to D_A, but D_B is insufficiently discriminative from all the other descriptors. Consequently, the accuracy of keypoint matches is unlikely to be high, leading to a poor registration performance.

4.4 Detector of Structural Similarity (DSS)

Following the discussion on SOIs in Section 4.2, we will describe the proposed DSS technique for increasing the structural similarity between color and confocal microscopic images. In color images, characteristics in intensity relationships between RGB color channels are utilized to detect *Brown Structures* and *Brown/Blue Overlapped Structures* which have been identified in Section 4.2.1. In confocal images, a particular color channel is extracted as SOIs which correspond to the ones in color images. An additional operation for confocal images is to eliminate the pixels with very weak intensities. Finally, background noise is eliminated in both color images and confocal images.

4.4.1 DSS in Color Images

To detect the two categories of structures in color images, i. e. *Brown Structures* and *Brown/Blue Overlapped Structures* as illustrated in Figure 4.1(a), we exploit the intensity relationships between the RGB channels. By analyzing the image characteristics of the *Brown Structures*, we found that the intensity at the red channel is highest of the three channels and we call this characteristic *Intensity Relationship*. To accurately detect the structures as required, we have also taken into account the *Intensity Separation* between the three channels. The *Intensity Separation* refers to how separate two color channels are in intensity values. The image pixels which have very similar intensity values in all three color channels should not be extracted as such pixels appear to be gray in the color images. With the two characteristics identified, the *Brown Structures* are formulated as

$$(I_G \leq k \times I_R \bigcap I_B < I_R) \bigcup (I_B \leq k \times I_R \bigcap I_G < I_R), \tag{4.4}$$

where I_R, I_G and I_B denote the intensity values at the red, green and blue channels respectively, k is a parameter for imposing constraints on the *Intensity Separation* between the RGB channels.

As for the *Brown/Blue Overlapped Structures*, the characteristic *Intensity Relationship* is that intensities at blue channel are slightly higher than intensities at red channel. Furthermore, the characteristic *Intensity Separation* used for the *Brown Structures* is also applicable to the *Brown/Blue Overlapped Structures*. Likewise, this group of structures is formulated as

$$I_G \leq k \times I_R \bigcap I_R \leq I_B. \tag{4.5}$$

By observing Equations 4.4 and 4.5, the two equations can be merged into

$$I_G \leq k \times I_R \bigcup (I_B \leq k \times I_R \bigcap I_G < I_R). \tag{4.6}$$

Thus, Equation 4.6 is the criterion which is used to judge whether a pixel belongs to the *Brown Structures* or the *Brown/Blue Overlapped Structures* in color images. Note that k in Equation 4.6 is of great importance in detecting SOIs of color images. The

significance of k and how to select an appropriate k value will be elaborated in Sections 4.5 and 4.6.

4.4.2 DSS in Confocal Images

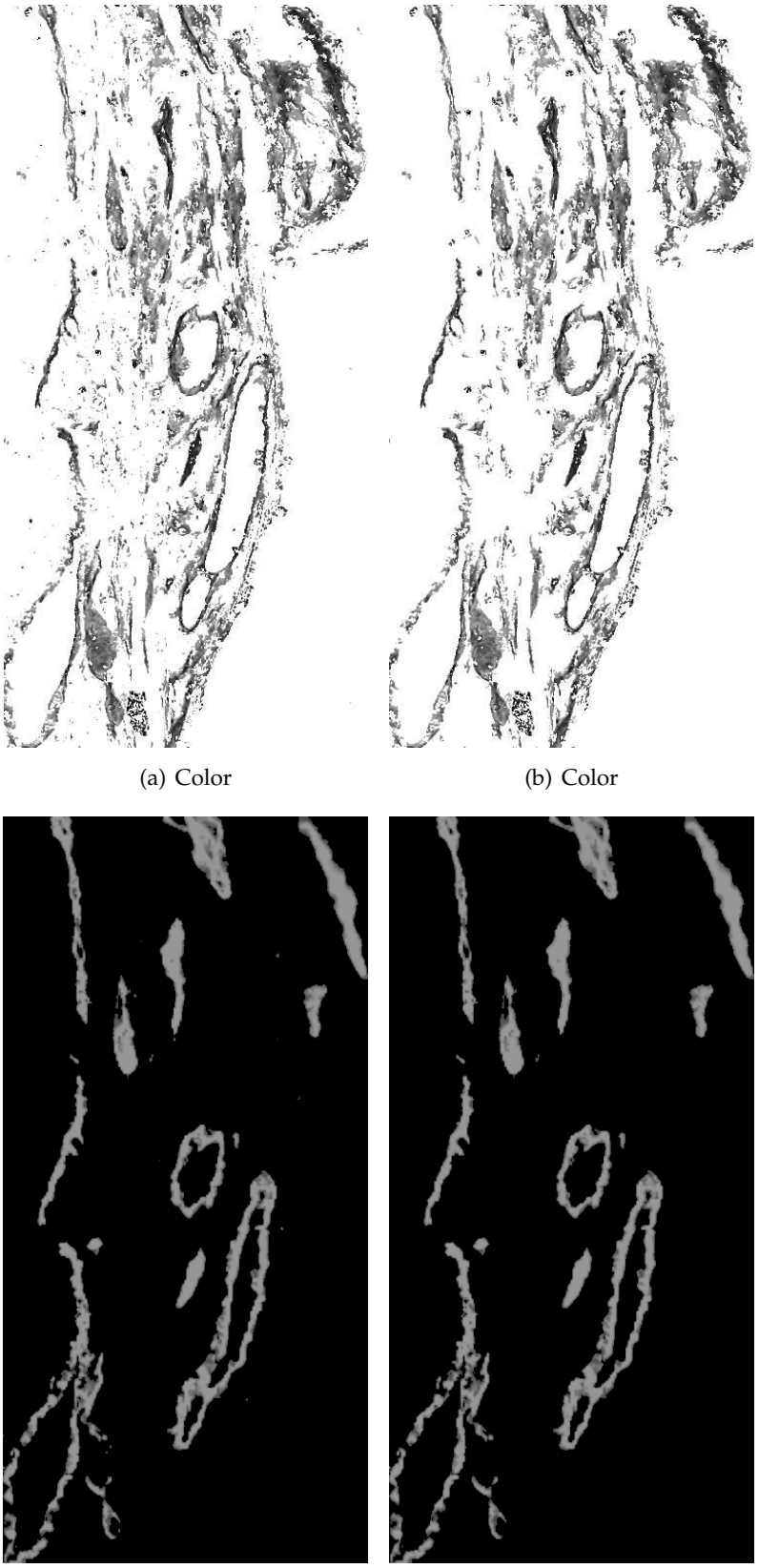
Unlike the complex visual characteristics in color images, SOIs in a confocal image are only at one color channel, green or red, as stated in Section 4.2.2. Given a confocal image such as the ones shown in Figure 4.1 (b) and Figure 4.2 (b), there exists a number of pixels whose intensities are very weak. For the purpose of image registration, those pixels with relatively strong intensities will correspond to SOIs in the corresponding color image. The pixels with strong intensities are regarded as the foreground and those with weak intensities as the background. The objective is to segment foreground and background by automatically determining an intensity threshold. To achieve this, we use Otsu's thresholding [79].

4.4.3 Eliminating Background Noise

After performing the operations stated in Sections 4.4.1 and 4.4.2, we found in the transformed images that there exists some background noise, as shown in Figure 4.6 (a) and (c), primarily in the color image. For the purpose of image registration, it is unlikely that the background noise contains sufficient and useful information for building a local descriptor. Thus, we believe that eliminating background noise can further increase the structural similarity between color and confocal images. This is done by morphologically opening the corresponding binary images [1]. In our implementation image regions containing fewer than 16 pixels are eliminated. The images transformed after eliminating background noise are shown in Figure 4.6 (b) and (d).

4.5 Significance of k in DSS

We will analyze the significance of k in DSS from the following two aspects. First, different k values can make a big difference in the final registration performance. Second, color images with different characteristics can have different optimal k values.



(a) Color (b) Color

(c) Confocal (d) Confocal

Figure 4.6: Color and Confocal Images Transformed by DSS. (a) and (c): before eliminating background noise, (b) and (d): after eliminating background noise.

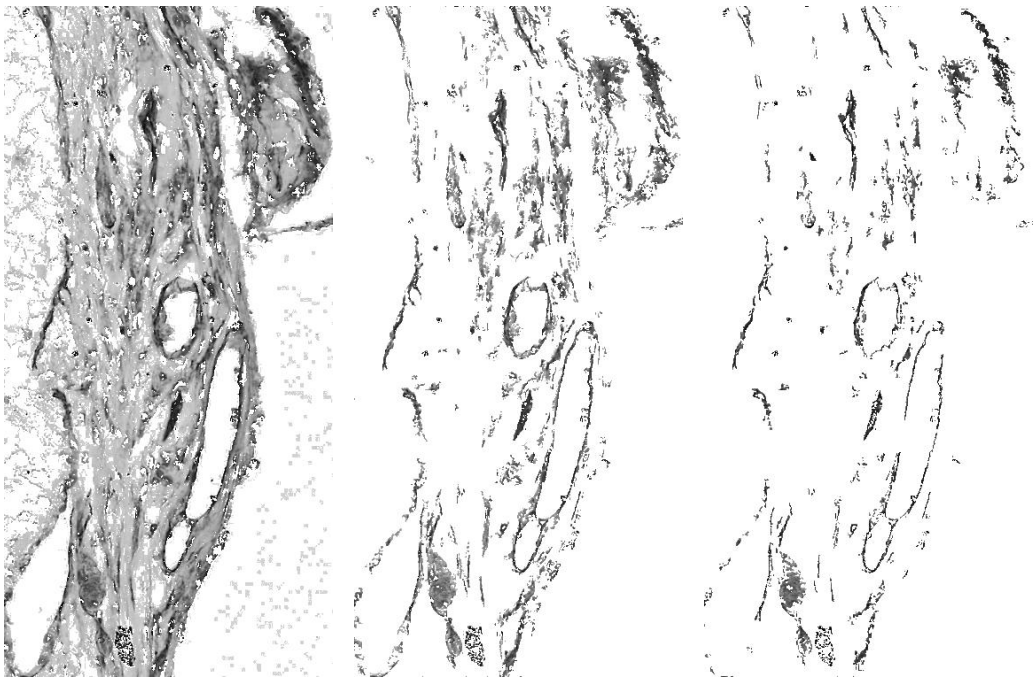
4.5.1 Impact of k on Registration Performance

In Section 4.4.1, k is used as a parameter for imposing constraints on the *Intensity Separation* between the RGB channels. The parameter k is critical in Equation 4.6 for detecting SOIs in color images, thereby determining the structural similarity in a pair of color and confocal images. Let us use k_{opt} to denote the optimal k value which can detect the most appropriate amount of SOIs in a color image. If a tuned k is far from k_{opt}, much higher or lower, the structural similarity between the color and confocal images is not optimally detected. On the one hand, a k value that is too high preserves many gray pixels in the background which belong to non-SOIs (Scenario 1). Compared with the confocal image, the color image contains redundant image structures. On the other hand, a k value that is too low eliminates image structures which belong to SOIs (Scenario 2). Unexpectedly, the confocal image contains more structures than the color image. In both of the two scenarios, the structural similarity is low. As analyzed in Section 4.3.3, the low structural similarity undermines the registration performance.

Table 4.1: Matching Accuracies Using MOG-IS-SIFT with Different k Values

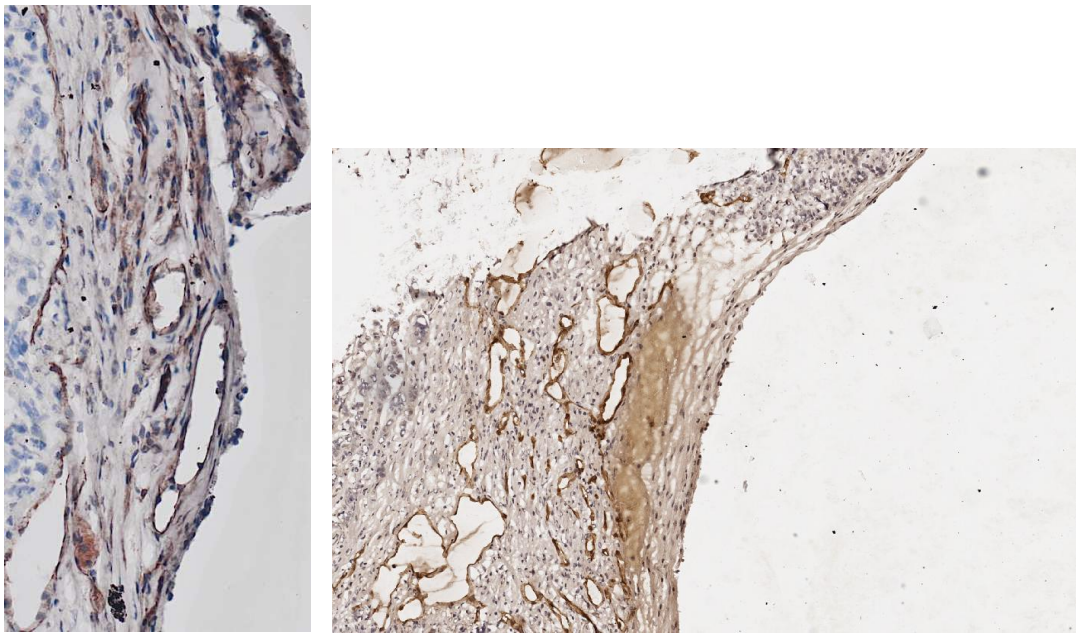
k	**Total**	**True**	**Accuracy (%)**
0.89	7	4	57.14
0.99	3	0	0.00
0.80	3	1	33.33

We now give an example to illustrate the impact of different k values on the registration performance. Figure 4.7 shows color images in which SOIs have been detected using three different k values: 0.99, 0.89 and 0.80. By comparison, Figure 4.7 (b) shows much higher structural similarity with the corresponding confocal image in Figure 4.6 (d) as compared to Figure 4.7 (a)and (c). Table 4.1 compares the matching accuracies of MOG-IS-SIFT when using the three different k values. This example clearly illustrates to what extent different k values can impact the registration performance.



(a) k=0.99 (b) k=0.89 (c) k=0.80

Figure 4.7: Color Images Processed by Different k Values



(a) (b)

Figure 4.8: Two Color Microscopic Images with Different Characteristics

4.5.2 Color Images with Different Characteristics can have Different Optimal *k*

The parameter *k* is used in Equation 4.6 for imposing constraints on *Intensity Separation* between the RGB channels, which is clearly associated with image characteristics. This can be illustrated through an example shown in Figure 4.8. The two color images in Figure 4.8 present different image characteristics as the color intensities are different. Consequently, the *Intensity Separation* between the RGB channels defined in Equation 4.6 between Figure 4.8(a) and Figure 4.8(b) will be different, leading to different optimal *k* values. The optimal *k* values for the two images in Figure 4.8 are 0.89 and 0.74 respectively, when using MOG-IS-SIFT as the registration technique. Details about optimal *k* values for different color images can be found in Section 4.7. Thus, the selection of *k* must be dependent on image characteristics which might vary across different color images in our test data.

4.6 Adaptively Selecting *k* in DSS

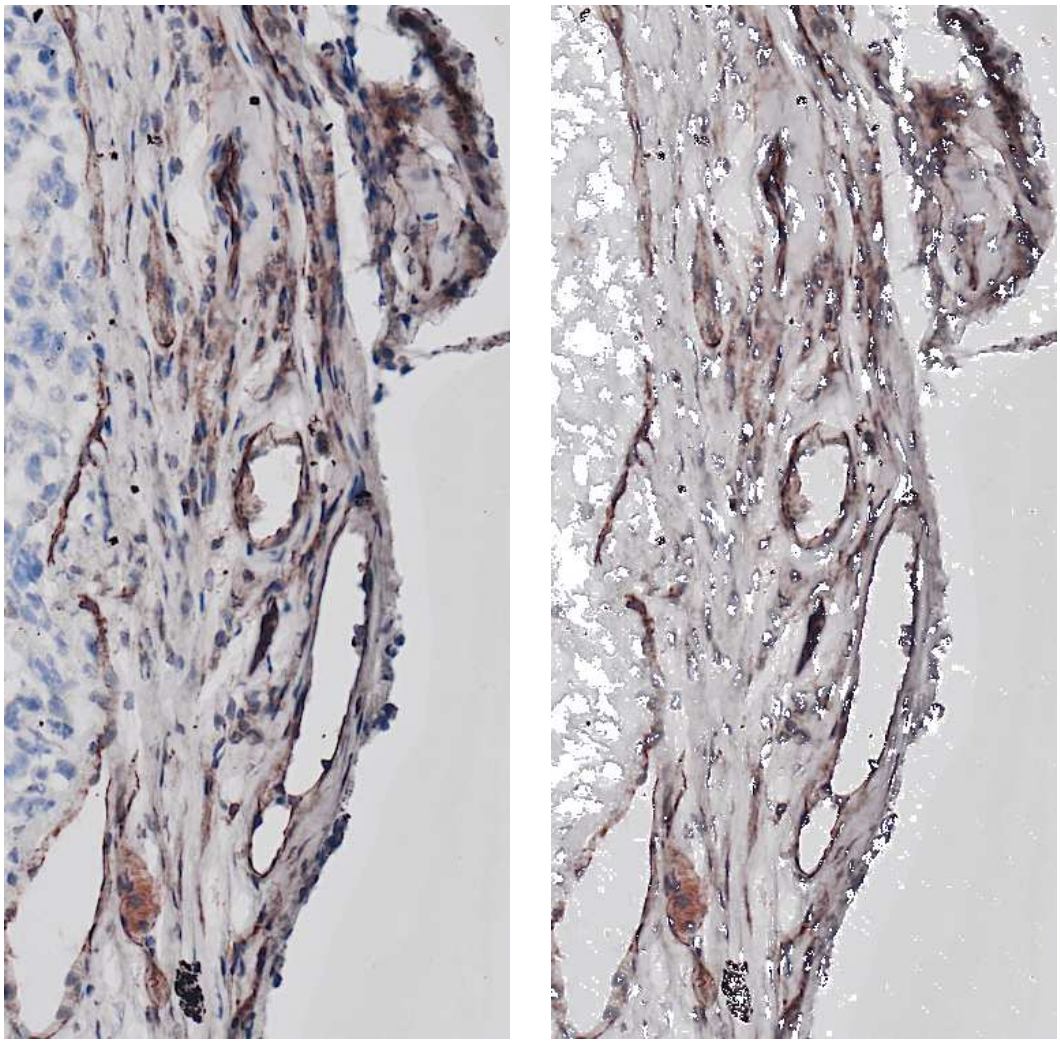
As stated in Section 4.5, the parameter *k* is of great importance in DSS for detecting SOIs in color images. This section aims to select an appropriate *k* for detecting SOIs in a color image, leading to a high structural similarity between the color and confocal images. First, we will illustrate the transformations of a color image when *k* is equivalent to 1. Second, we will analyze how SOIs and non-SOIs are affected by tuning *k*, and therefore derive the criteria for adaptively selecting *k*.

4.6.1 Transformations of Color Images When *k*=1 in DSS

We now review Equation 4.6 that is for detecting SOIs in color images. If *k* is equivalent to 1, Equation 4.6 can be expressed as

$$I_G \leq I_R \bigcup (I_B \leq I_R \bigcap I_G < I_R). \tag{4.7}$$

By doing this, only the first image characteristic of the two described in Section 4.4.1, *Intensity Relationship*, is applied. Consequently, Equation 4.7 removes the blue



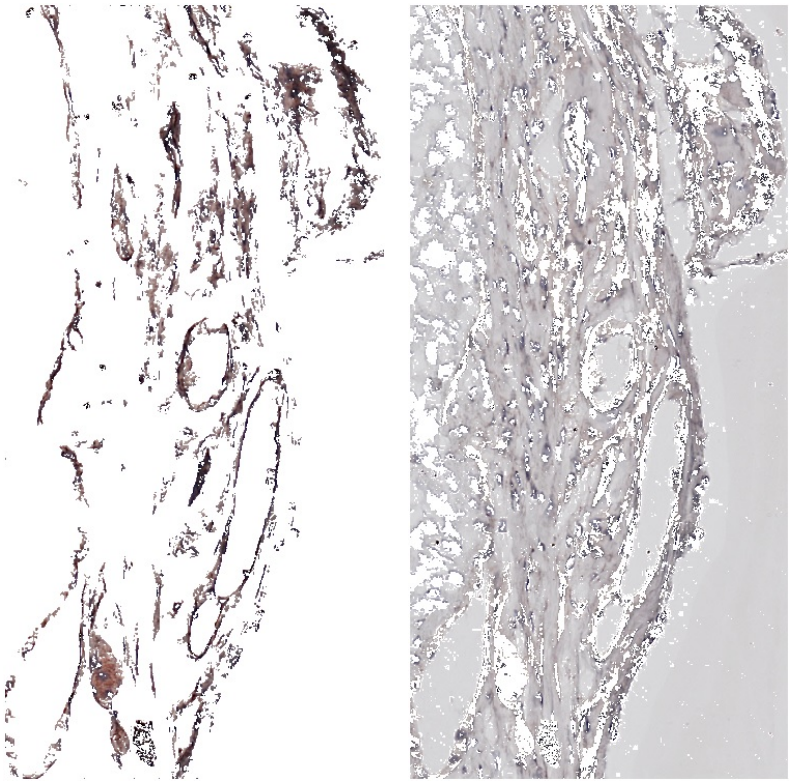
(a) Original Image (b) Intermediate Image

Figure 4.9: Original and Intermediate Images

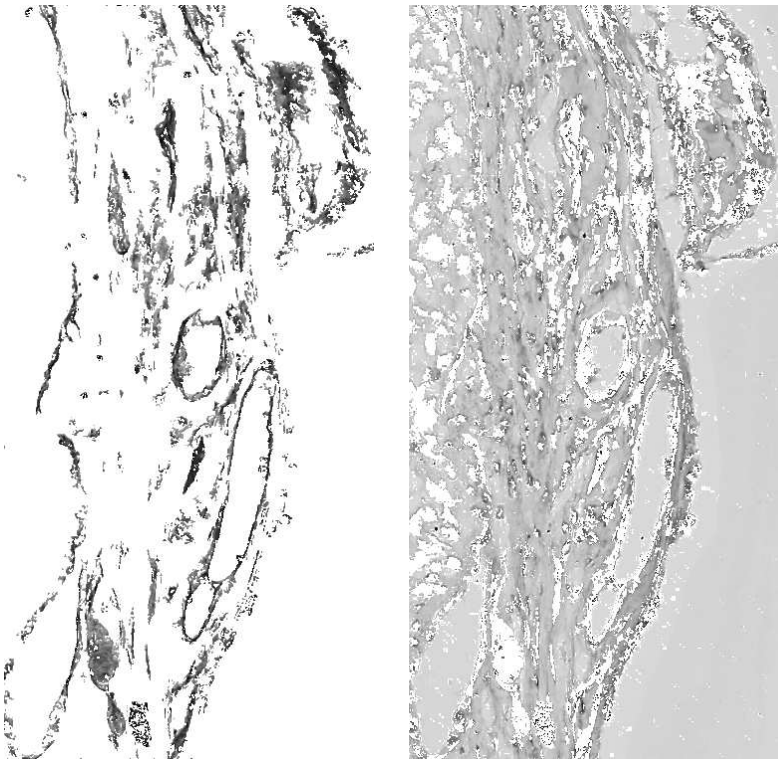
structures and preserves the brown structures as well as gray pixels. As shown in Figure 4.9 (b), this image is called *Intermediate Image*.

4.6.2 Tuning *k* and Deriving Criteria for Selecting *k*

Once an *Intermediate Image* is identified as described in Section 4.6.1, we can start to look into image transformations while tuning k using Equation 4.6. As k is decreased from the value 1 with a fixed interval (0.01 is used in our implementation), a certain amount of background pixels will be eliminated from the *Intermediate Image* and these



(a) Preservation Image (RGB) (b) Elimination Image (RGB)



(c) Preservation Image (grayscale) (d) Elimination Image (grayscale)

Figure 4.10: Preservation and Elimination Images

pixels make up a new image, called *Elimination Image* as shown in Figure 4.10 (b). Accordingly, all the pixels preserved comprise Figure 4.10 (a), called *Preservation Image*. Apparently, the *Preservation Image* and the *Elimination Image* constitute an *Intermediate Image* which is shown in Figure 4.9 (b). Ideally, our objective is to find the optimal k, k_{opt}, so that the *Preservation Image* contains the biggest amount of SOIs and the *Elimination Image* contains the biggest amount of non-SOIs.

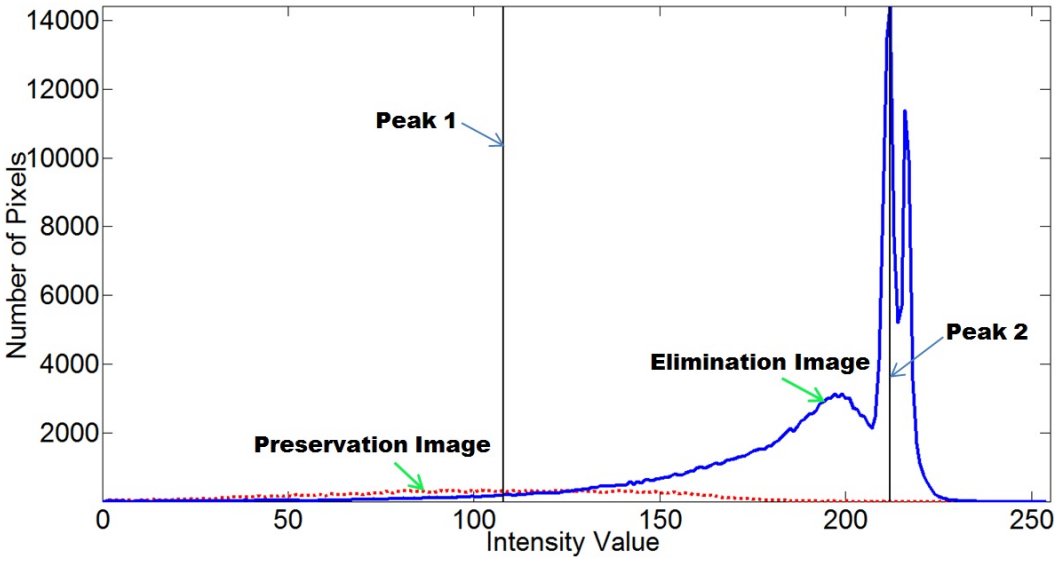


Figure 4.11: Histograms of Preservation Image and Elimination Image

In order to find an appropriate k, we investigate the relationship between the *Preservation Image* and *Elimination Image*. Based on Figure 4.10 (a) and (b), their grayscale images are obtained, as shown in Figure 4.10 (c) and (d). While tuning k, we track the changes that occur in the histograms derived from the grayscale *Preservation Image* and *Elimination Image*. Figure 4.11 shows an example of histograms of the grayscale *Preservation Image* and *Elimination Image*, where a tested k is used. Peaks 1 and 2 have been highlighted in Figure 4.11 for the *Preservation Image* and *Elimination Image* respectively.

Based on our analysis, if a tuned k is around k_{opt}, the *Preservation Image* contains dominant SOIs while the *Elimination Image* contains dominant non-SOIs. We are interested in seeing how SOIs and non-SOIs are affected when k is tuned. In particular, if k is tuned slightly around k_{opt}, this will change the pixels around the edges of SOIs

to non-SOIs, or non-SOIs to SOIs. One important consideration is that dominant SOIs and dominant non-SOIs should remain stable. Accordingly, the histogram peaks of *Preservation Image* and *Elimination Image*, Peaks 1 and 2, are stable. Based on this consideration, we derive the following two criteria for adaptively selecting k.

i. **Criterion 1: Stability of Histogram Peaks in Intensity**

Around k_{opt}, there must exist a stage range between Peaks 1 and 2 where their intensity separation remains unchanged. Specifically, if the absolute difference between the intensities of the two peaks remains constant for at least two adjacent k values, i.e.

$$\exists |I_{p1}(k) - I_{p2}(k)| = |I_{p1}(k-\delta) - I_{p2}(k-\delta)| \tag{4.8}$$

where $I_{p1}(k)$ and $I_{p2}(k)$ denote the intensity of Peaks 1 and 2 respectively for a particular tuned k, and δ is the fixed interval while tuning k.

ii. **Criterion 2: Stability of Histogram Peaks in Number of Pixels**

The stability between Peaks 1 and 2 can also be reflected by their corresponding number of pixels. For two adjacent k values, from k to $k-\delta$, the ratio between the numbers of pixels at the two peaks should remain stable. Unlike the intensity of the two peaks discussed in **Criterion 1**, a slight change in the number of pixels for Peaks 1 and 2 can be allowed. Thus, from k to $k-\delta$, the ratio difference between the numbers of pixels at the two peaks is approaching to zero, i.e.

$$RD(k, k-\delta) = \frac{P_{p1}(k)}{P_{p2}(k)} - \frac{P_{p1}(k-\delta)}{P_{p2}(k-\delta)} \approx 0 \tag{4.9}$$

where $P_{p1}(k)$ and $P_{p2}(k)$ indicate the number of pixels at the two peaks for a particular tuned k. The closer to zero the absolute value derived from the left of $\approx$ in Equation 4.9 is, the better the tuned k is.

Another consideration is that our focus is on SOIs. However, we have found that a large amount of non-SOIs may be preserved in *Preservation Image* when k is above and far away from k_{opt}. To avoid this kind of scenarios, the intensity range for the color brown, between rosy brown and dark brown, is used to narrow down the

range of tuned k values. This criterion is called **Appropriate Intensity Range for SOIs (Criterion 3)**.

In **Criterion 1**, a stable range of intensity separation between the two peaks is discussed. But we have found that there may exist multiple such stable ranges. Based on our analysis, there are two possible reasons which account for this phenomenon. One reason is that Peak 1 represents non-SOI pixels when many non-SOIs are preserved in the *Preservation Image* in a certain range of tuned k values. The second reason is the potential existence of two categories of SOIs, i. e. *Brown Structures* and *Brown/Blue Overlapped Structures* in color images, as described in Section 4.2.1.

After deriving the three criteria, various combinations can be applied in DSS for adaptively selecting the parameter k. In our experiments, we have tested the following four combinations:

- DSS^1

  Only **Criterion 2** is used. As formulated in Equation 4.9, a set of $RD(k, k-\delta)$ values are derived as k is tuned. As k is decreased from the value 1, we select the first k value which causes a dramatic change in $RD(k, k-\delta)$ values. More specifically, *k-means* [59] is used to automatically determine where such a dramatic change is located.

- DSS^2

  Criteria 1 and 2 are used. In this case, the stability between the two histogram peaks is assured by both the intensity and number of pixels. Specifically, after identifying a k value determined by DSS^1, k is further tuned until the first stable range of intensity separation occurs between the two peaks. The first tuned k value in this stable range is selected.

- DSS^3

  All three criteria are used. First, **Criterion 3** is used to narrow down the range of tuned k values Second, k is further tuned until the first stable range of intensity separation occurs between the two peaks (**Criterion 1**). Third, we select the k value which leads to the smallest absolute value of ratio difference between the number of pixels at the two peaks.

- DSS^4

  Similar to DSS^3, all the three criteria are used. The difference from DSS^3 is that the potential multiple ranges of intensity separation between the two peaks are taken into consideration.

4.7 Performance Study

We evaluate the proposed DSS using two multi-modal image registration techniques: MOG-IS-SIFT which we have proposed in Chapter 3 and PIIFD [17]. The two registration techniques, MOG-IS-SIFT and PIIFD, are applied to register both original multi-modal microscopic images and these images after DSS is performed. Each of the four combinations of k-selection criteria stated in Section 4.6.2 can be used together with MOG-IS-SIFT and PIIFD. Thus, performance comparisons will be carried out between MOG-IS-SIFT and DSS^i-MOG-IS-SIFT, and between PIIFD and DSS^i-PIIFD, where $1 \leq i \leq 4$. With regard to evaluation criterion, we use matching accuracy which has been defined in Section 3.4.

4.7.1 DSS with MOG-IS-SIFT

Table 4.2 presents matching accuracies and corresponding k values when five registration techniques, MOG-IS-SIFT, DSS^1-MOG-IS-SIFT, DSS^2-MOG-IS-SIFT, DSS^3-MOG-IS-SIFT and DSS^4-MOG-IS-SIFT, are used to register the 16 multi-modal microscopic image pairs. In some cases, very few matches have been determined, such as two, one or even zero. As pointed out in Section 3.4.4 of Chapter 3, at least three matches are needed for estimating a transformation between two images. Thus, a matching accuracy in cases where the number of matches is smaller than three is meaningless for the registration purpose. An example of this is registering pair 6 using MOG-IS-SIFT, where all of the two matches are true, leading to a 100% matching accuracy.

To better illustrate the matching results shown in Table 4.2, we use $\times$ in Figure 4.12 to represent the matching accuracies for cases in which the number of matches

is smaller than three. Accordingly, matching accuracies for the 16 multi-modal microscopic image pairs are presented in Figure 4.12. For pairs 2-5, the proposed techniques DSS^i-MOG-IS-SIFT ($1 \leq i \leq 4$) consistently outperform MOG-IS-SIFT. For pairs 8-10, DSS^3-MOG-IS-SIFT

Table 4.2: MOG-IS-SIFT vs DSS^i-MOG-IS-SIFT in Matching Accuracy

ID	MOG-IS-SIFT	DSS^1-MOG-IS-SIFT	DSS^2-MOG-IS-SIFT	DSS^3-MOG-IS-SIFT	DSS^4-MOG-IS-SIFT
1	2/3=66.67%	4/17=23.53% (0.97)	3/10=30.00% (0.96)	3/10=30.00% (0.96)	3/10=30.00% (0.96,0.93,0.88)
2	1/5=20.00%	1/4=25.00% (0.96)	4/7=57.14% (0.89)	4/7=57.14% (0.89)	4/7=57.14% (0.89)
3	0/7=0.00%	35/57=61.40% (0.97)	6/13=46.15% (0.92)	3/9=33.33% (0.91)	3/9=33.33% (0.91,0.85,0.79,0.71)
4	0/5=0.00%	1/12=8.33% (0.97)	4/17=23.53% (0.93)	2/10=20.00% (0.89)	2/10=20.00% (0.89)
5	0/5=0.00%	19/78=24.36% (0.97)	26/75=34.67% (0.93)	27/65=41.54% (0.92)	27/65=41.54% (0.92,0.87,0.78)
6	2/2=100.00%	0/0=0.00% (0.96)	0/0=0.00% (0.92)	0/0=0.00% (0.92)	0/0=0.00% (0.92,0.89,0.88,0.80)
7	0/6=0.00%	0/0=0.00% (0.96)	1/1=100.00% (0.93)	0/1=0.00% (0.92)	0/1=0.00% (0.92,0.88)
8	2/5=40.00%	7/23=30.43% (0.93)	7/23=30.43% (0.93)	23/42=54.76% (0.88)	14/24=58.33% (0.88,0.87,0.74)
9	4/10=40.00%	7/21=33.33% (0.93)	11/20=55.00% (0.91)	2/4=50.00% (0.75)	2/4=50.00% (0.75,0.72)
10	0/10=0.00%	0/7=0.00% (0.92)	0/13=0.00% (0.88)	0/16=0.00% (0.87)	2/16=12.50% (0.87,0.86,0.78,0.73)
11	0/1=0.00%	0/1=0.00% (0.89)	0/3=0.00% (0.83)	0/3=0.00% (0.83)	0/3=0.00% (0.83,0.76)
12	0/1=0.00%	0/1=0.00% (0.96)	0/1=0.00% (0.96)	0/1=0.00% (0.88)	0/1=0.00% (0.88)
13	0/0=0.00%	0/4=0.00% (0.96)	0/4=0.00% (0.96)	0/1=0.00% (0.89)	0/1=0.00% (0.89)
14	0/6=0.00%	0/8=0.00% (0.97)	0/8=0.00% (0.97)	0/14=0.00% (0.91)	0/14=0.00% (0.91)
15	0/0=0.00%	1/1=100.00% (0.91)	1/1=100.00% (0.91)	1/1=100.00% (0.91)	1/1=100.00% (0.91)
16	0/1=0.00%	0/3=0.00% (0.97)	0/6=0.00% (0.91)	0/4=0.00% (0.90)	0/5=0.00% (0.85)

[a] For i in DSS^i-MOG-IS-SIFT, $1 \leq i \leq 4$.

[b] A matching accuracy in the DSS^4-MOG-IS-SIFT column is obtained using the underlined k value.

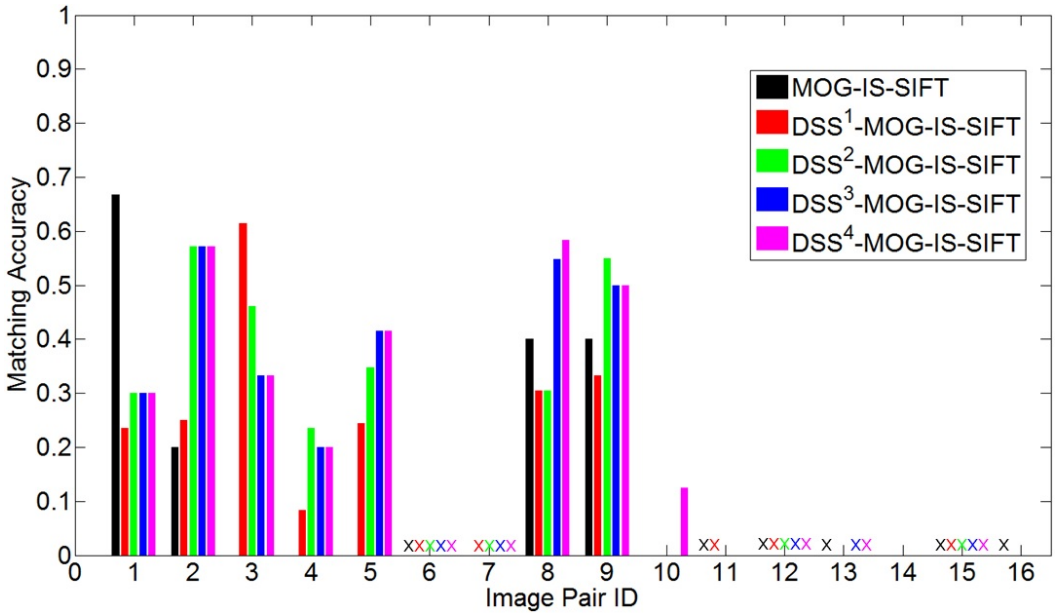
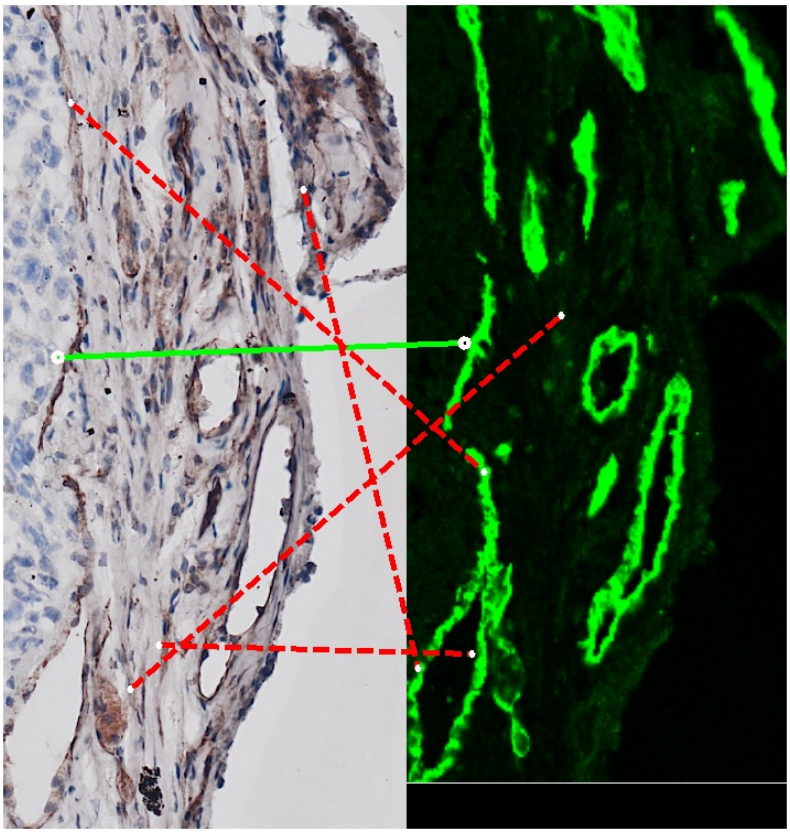


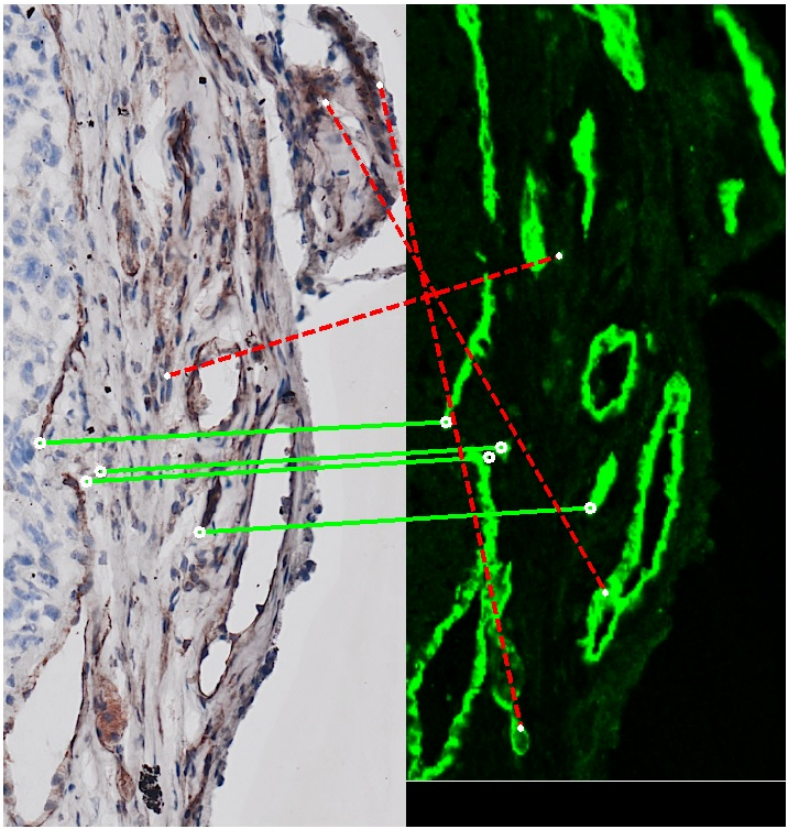
Figure 4.12: Comparisons in Matching Accuracy between MOG-IS-SIFT and DSS^i-MOG-IS-SIFT. In DSS^i-MOG-IS-SIFT, $1 \leq i \leq 4$. A case where there is no bar emerging indicates that the matching accuracy is 0.00%.

or DSS^4-MOG-IS-SIFT achieves higher matching accuracies than MOG-IS-SIFT. In registering pairs 6, 7 and 11-16, the number of matches is smaller than three or the matching accuracy is 0.00% for all the five registration techniques. It is only in registering pair 1 that MOG-IS-SIFT is able to achieve a higher matching accuracy than the other four proposed techniques. On average, the matching accuracies of MOG-IS-SIFT and DSS^i-MOG-IS-SIFT for all the 16 pairs are 10.42%, 12.90%, 17.31%, 17.92% and 18.93%, respectively. Note that, in averaging matching accuracies, 0.00% is used for the matching accuracy of a case where the number of matches is smaller than three due to its meaninglessness for registration.

Also, a matching example is given in Figure 4.13 for registering microscopic pair 2. The matching shown in Figure 4.13 applies to DSS^2-MOG-IS-SIFT, DSS^3-MOG-IS-SIFT and DSS^4-MOG-IS-SIFT as the selected k values for the three registration techniques are equivalent. In registering this image pair, an improvement of 37.14% in matching accuracy is achieved from MOG-IS-SIFT to DSS^j-MOG-IS-SIFT, where $2 \leq j \leq 4$.



(a) MOG-IS-SIFT (1/5=20.00%)



(b) DSS^j-MOG-IS-SIFT (4/7=57.14%, $2 \leq j \leq 4$)

Figure 4.13: A Matching Example of Evaluating DSS by MOG-IS-SIFT

4.7.2 DSS with PIIFD

Table 4.3: PIIFD vs DSS^i-PIIFD in Matching Accuracy

ID	PIIFD	DSS^1-PIIFD	DSS^2-PIIFD	DSS^3-PIIFD	DSS^4-PIIFD
1	1/1=100.00%	2/9=22.22% (0.97)	2/11=18.18% (0.96)	2/11=18.18% (0.96)	7/16=43.75% (0.96,0.93,0.88)
2	0/1=0.00%	7/21=33.33% (0.96)	10/24=41.67% (0.89)	10/24=41.67% (0.89)	10/24=41.67% (0.89)
3	0/0=0.00%	17/44=38.64% (0.97)	11/33=33.33% (0.92)	7/32=21.88% (0.91)	3/9=33.33% (0.91,0.85,0.79,0.71)
4	0/0=0.00%	2/19=10.53% (0.97)	2/19=10.53% (0.93)	5/19=26.32% (0.89)	5/19=26.32% (0.89)
5	0/0=0.00%	38/61=62.30% (0.97)	41/68=60.29% (0.93)	42/68=61.76% (0.92)	42/68=61.76% (0.92,0.87,0.78)
6	0/7=0.00%	5/14=35.71% (0.96)	6/14=42.86% (0.92)	6/14=42.86% (0.92)	6/14=42.86% (0.92,0.89,0.88,0.80)
7	0/8=0.00%	0/11=0.00% (0.96)	0/7=0.00% (0.93)	0/9=0.00% (0.92)	0/9=0.00% (0.92,0.88)
8	0/0=0.00%	13/26=50.00% (0.93)	13/26=50.00% (0.93)	20/30=66.67% (0.88)	16/22=72.73% (0.88,0.87,0.74)
9	0/0=0.00%	0/9=0.00% (0.93)	1/11=9.09% (0.91)	3/14=21.43% (0.75)	3/14=21.43% (0.75,0.72)
10	0/5=0.00%	6/13=46.15% (0.92)	10/21=47.62% (0.88)	12/21=57.14% (0.87)	14/22=63.64% (0.87,0.86,0.78,0.73)
11	0/12=0.00%	1/16=6.50% (0.89)	2/13=15.38% (0.83)	2/13=15.38% (0.83)	3/10=30.00% (0.83,0.76)
12	0/0=0.00%	0/7=0.00% (0.96)	0/7=0.00% (0.96)	0/10=0.00% (0.88)	0/10=0.00% (0.88)
13	0/1=0.00%	0/8=0.00% (0.96)	0/8=0.00% (0.96)	0/6=0.00% (0.89)	0/6=0.00% (0.89)
14	0/6=0.00%	0/16=0.00% (0.97)	0/16=0.00% (0.97)	0/18=0.00% (0.91)	0/18=0.00% (0.91)
15	0/0=0.00%	0/4=0.00% (0.91)	0/4=0.00% (0.91)	0/4=0.00% (0.91)	0/4=0.00% (0.91)
16	0/0=0.00%	1/10=10.00% (0.97)	0/14=0.00% (0.91)	0/13=0.00% (0.90)	0/13=0.00% (0.85)

[a] For i in DSS^i-PIIFD, $1 \leq i \leq 4$.
[b] A matching accuracy in the DSS^4-PIIFD column is obtained using the underlined k value.

Similar to showing results based on MOG-IS-SIFT in Section 4.7.1, Table 4.3 and Figure 4.14 present the matching results when using PIIFD, DSS^1-PIIFD, DSS^2-PIIFD, DSS^3-PIIFD and DSS^4-PIIFD to register the 16 multi-modal microscopic image pairs.

It can be seen in Table 4.3 that PIIFD performs very poorly in registering these image pairs. With the analysis of the minimum number of matches in Section 4.7.1, 1/1=100.00% for pair 1 is meaningless for the registration purpose. To be specific, all the four registration techniques, DSS^1-PIIFD, DSS^2-PIIFD, DSS^3-PIIFD and DSS^4-PIIFD, substantially improve PIIFD in matching accuracy in registering pairs 1-6, 8, 10 and 11. For pair 9, there is also a big improvement from PIIFD to DSS^2-PIIFD, DSS^3-PIIFD and DSS^4-PIIFD, whereas DSS^1-PIIFD has achieved a 10.00% improvement over PIIFD for pair 16. In registering pairs 7 and 12-15, none of the five registration techniques can determine a true match. On average, DSS^1-PIIFD, DSS^2-PIIFD, DSS^3-PIIFD and DSS^4-PIIFD improve PIIFD in matching accuracy by 19.70%, 20.56%, 23.33% and 27.34%, respectively. Figure 4.15 shows the keypoint matches using PIIFD and DSS^j-PIIFD ($2 \leq j \leq 4$) to register pair 2. It is clear that the proposed DSS has made a significant improvement based on PIIFD.

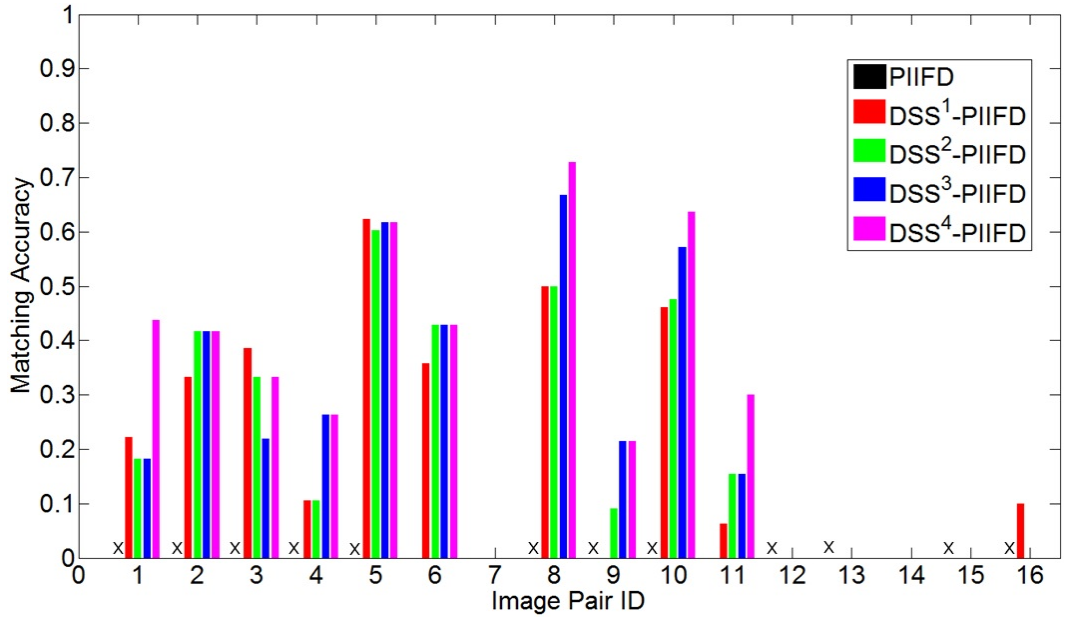
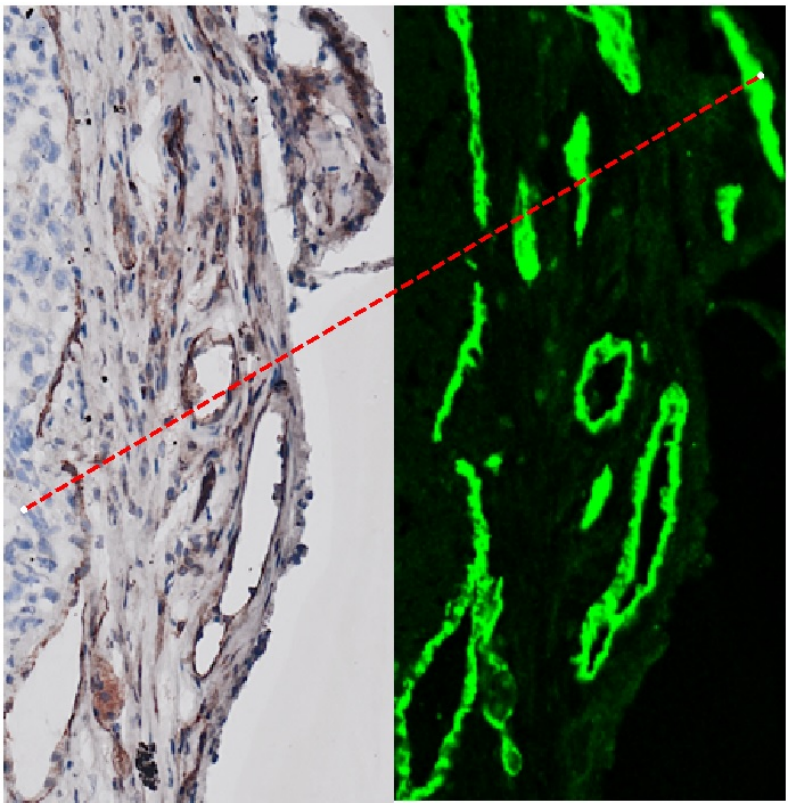
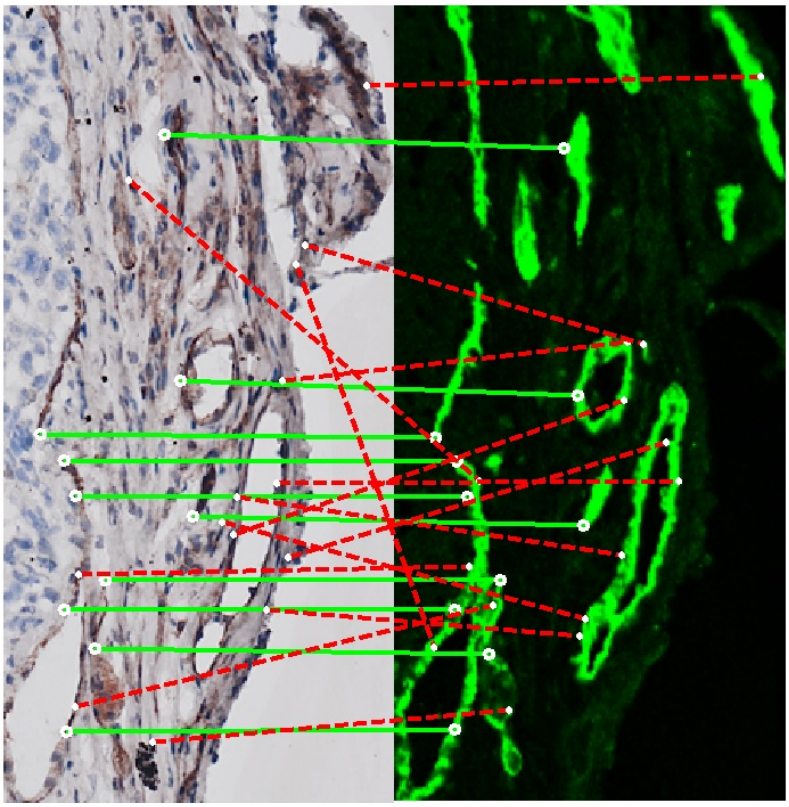


Figure 4.14: Comparisons in Matching Accuracy between PIIFD and DSS^i-PIIFD. In DSS^i-PIIFD, $1 \leq i \leq 4$.



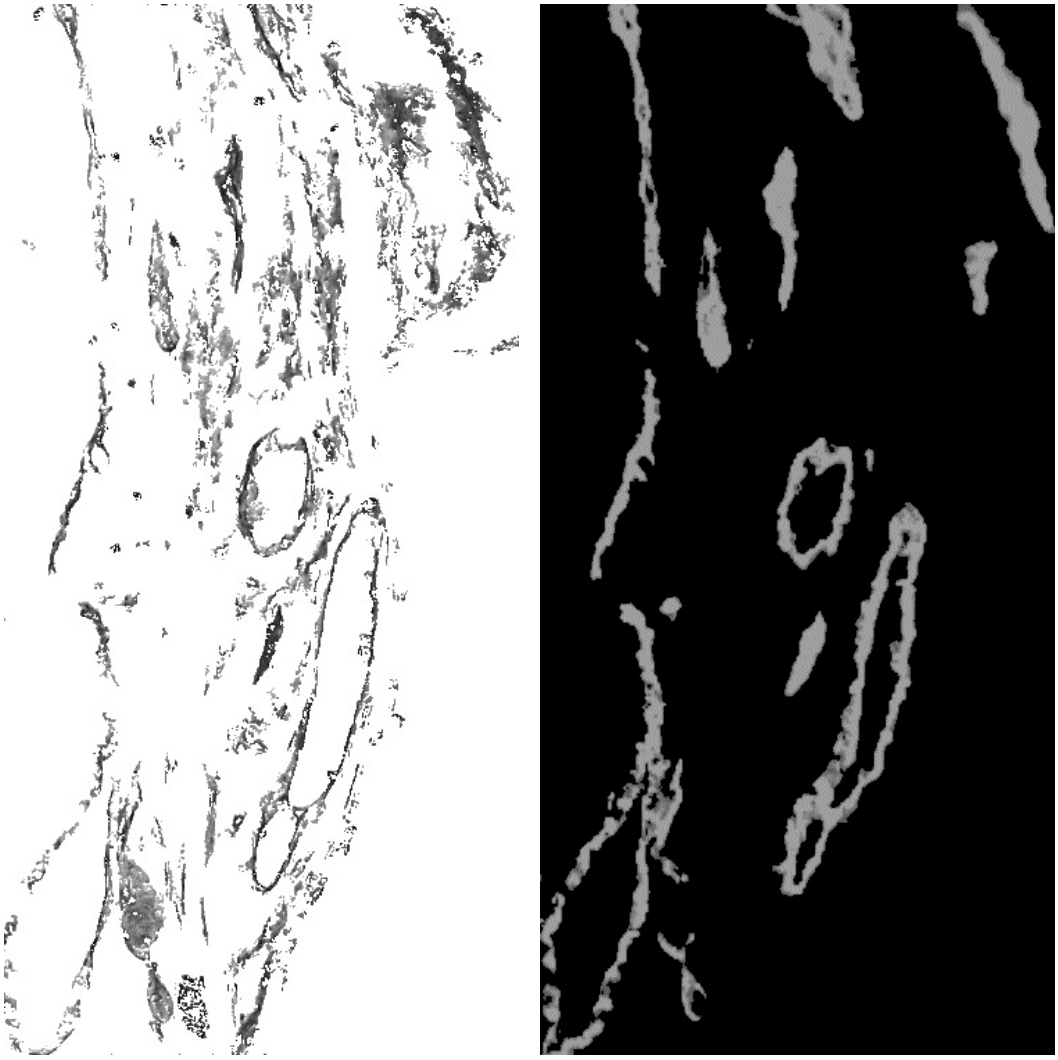
(a) PIIFD (0/1=0.00%)



(b) DSS^j-PIIFD (10/24=41.67%, $2 \leq j \leq 4$)

Figure 4.15: A Matching Example of Evaluating DSS by PIIFD

4.7.3 Discussions



(a) Color (b) Confocal

Figure 4.16: Color and Confocal Images after DSS is Performed. $k = 0.89$, which is identical for DSS^2, DSS^3 and DSS^4.

In Sections 4.7.1 and 4.7.2, we have evaluated the performance of DSS using MOG-IS-SIFT and PIIFD respectively. Overall, the matching accuracy has been significantly improved after applying DSS on the original multi-modal microscopic images. However, we need to highlight two issues from the results shown in Sections 4.7.1 and 4.7.2, as follows. First, even after applying DSS, there are eight pairs for MOG-IS-SIFT and five pairs for PIIFD where no true match has been

obtained. Second, the overall matching accuracy is still low. On average, the matching accuracies achieved by DSS^4-MOG-IS-SIFT and DSS^4-PIIFD are 18.93% and 27.34% respectively, which are too low to effectively align the color and confocal images. However, the structural similarity for each pair of color and confocal images has been significantly increased as compared to the original image pairs. For instance, Figure 4.16 shows the color and confocal images after DSS is performed, which can be compared with the original image pair shown in Figure 4.3. With the detected structural similarity, we will further explore ways in Chapter 5 to improve the registration performance on these images.

4.8 Summary

In registering multi-modal microscopic images, a big issue is that the structural similarity is low between a color image and the corresponding confocal image. In this chapter we have analyzed the significance of structural similarity in registering these images. In order to improve the registration performance, we have proposed the DSS technique to detect the structural similarity between color and confocal microscopic images. After performing DSS in the original color and confocal microscopic images, the structural similarities in these images have been significantly increased, thereby improving the registration performance. Although we have focused on increasing the structural similarities between color and confocal images, the proposed methodology in this chapter can be used to increase structural similarity in other types of medical images. But we have found that there still exist large content differences between color and confocal microscopic images which have been processed by DSS and that the overall registration performance is far from satisfactory. In Chapter 5, we will continue working on these color and confocal microscopic images to further improve registration accuracy.

Chapter 5

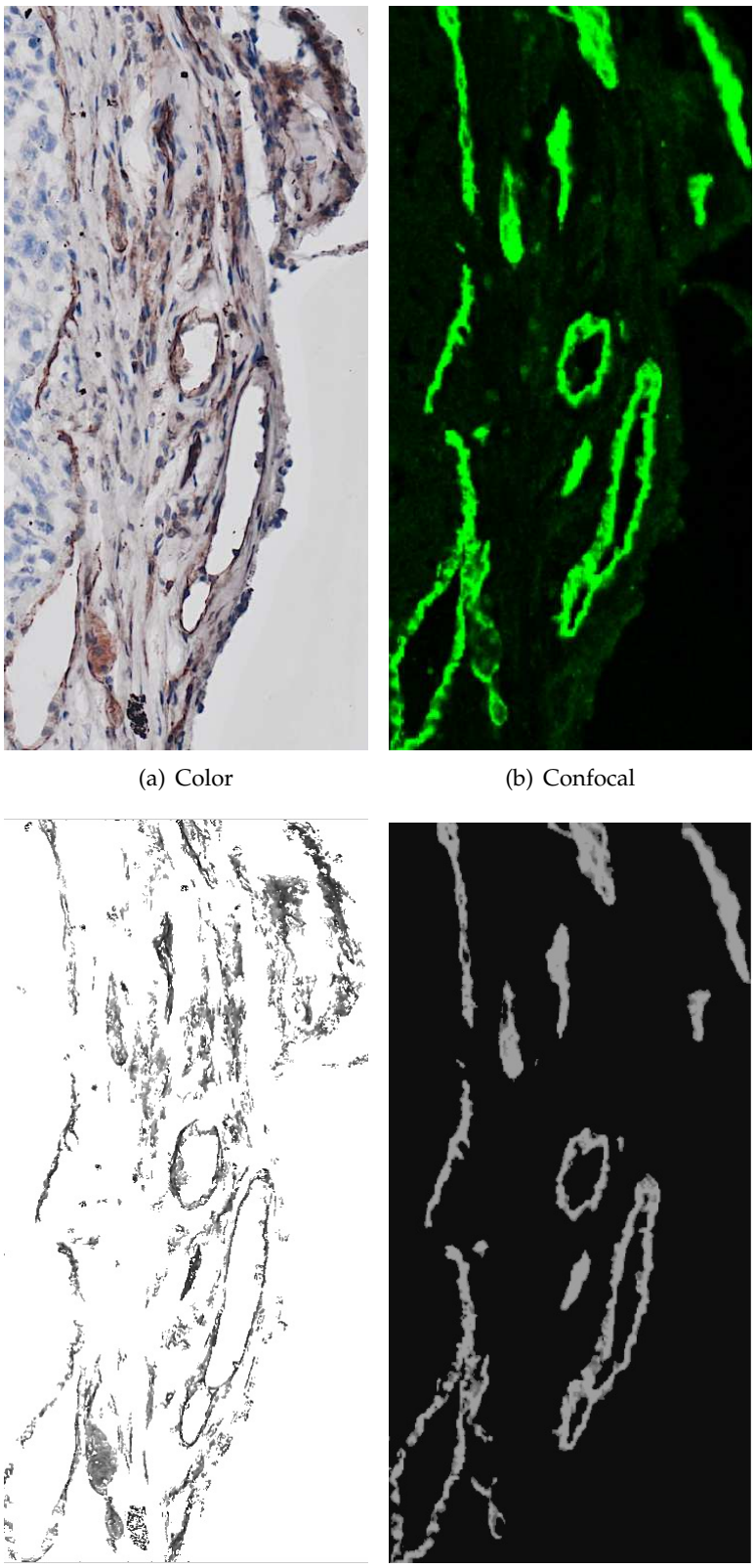
A Novel Multi-modal Image Registration Technique based on Corners

5.1 Introduction

In Chapter 4, we have proposed a Detector of Structural Similarity (DSS) technique to increase the structural similarity between color and confocal images. Figure 5.1 shows an example of the images before and after applying DSS. It is clear that, Figure 5.1 (c) and (d) display a much higher structural similarity as compared to Figure 5.1 (a) and (b). However in many cases, even after processing the color and confocal images with DSS, the processed images might remain too challenging for existing image registration techniques or our technique proposed in Chapter 3 to achieve a satisfactory performance. This has been shown in the experimental results in Section 4.7. Thus in this chapter we continue developing a more robust multi-modal image registration technique which can more accurately register such images with such difference in their contents.

By closely looking into color and confocal images such as Figure 5.1 (c) and (d), we have found that content differences between corresponding parts are still large. Due to the large content differences, it would be very challenging to effectively register these images using registration techniques which are sensitive to intensity or gradient changes. Moreover, if the scale difference between the color and confocal images is large, registering these images would be more difficult.

In this chapter, we will propose a multi-modal image registration technique based on corners, in order to achieve greater robustness to content and scale differences than



(a) Color (b) Confocal

(c) Color (d) Confocal

Figure 5.1: An Example of Original and DSS Color and Confocal Images. (a) and (b): original images; (c) and (d): images after applying DSS.

existing techniques such as [17, 45, 49]. In the proposed technique, we will use the Fast-CPDA corner detector [6] which is based on contours. Fast-CPDA corners are independent of intensity or gradient changes, leading to greater robustness to content differences. In addition, it is a big challenge to deal with large scale differences. We will propose estimating scale difference by making use of geometric relationships between corner triplets from the reference and target images. It is noted that a corner triplet is formed by three non-collinear corners. With the estimated scale difference, the original reference and target images are resized to have similar scales. Thus, the proposed technique is more robust to scale differences. Our main contributions include the following. First, the curvature similarity between corners is for the first time explored for the purpose of multi-modal image registration. Second, a new way of estimating scale difference in an image pair is proposed. Moreover, a novel corner descriptor is proposed to represent edges in the neighborhood of corners.

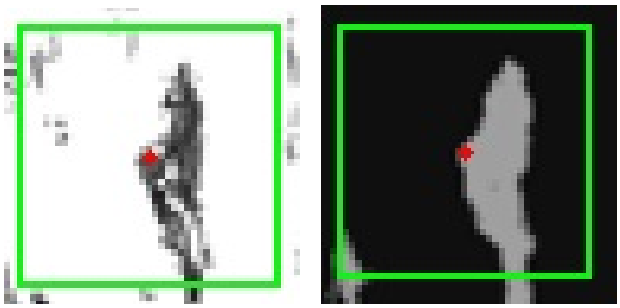
The rest of this chapter is structured as follows. In Section 5.2, we will illustrate content differences in image pairs. Section 5.3 discusses the significance of scale invariance to image registration and how the PIIFD descriptor is scale invariant. The robustness of the Fast-CPDA corner detector to content differences is demonstrated in Section 5.4. In Section 5.5, the proposed registration technique is elaborated, followed by a performance study in Section 5.6. Finally this chapter is summarized in Section 5.7.

5.2 Content Differences between Images

In the color and confocal images which have been processed by DSS such as Figure 5.1 (c) and (d), there are still large content differences between corresponding parts. In this section, we will first analyze an example of corresponding parts in the color and confocal images. Next, the content differences will be evaluated by descriptor distances between corresponding keypoints which are detected using PIIFD [17]. Note that PIIFD uses Harris corners [28] as keypoints for feature description and matching. For the referencing purpose, the feature points in PIIFD are called keypoints in this thesis.

5.2.1 An Example Illustrating Content Differences

Large content differences are illustrated in Figure 5.2, where Figure 5.2 (a) and (b) are corresponding parts manually extracted from Figure 5.1 (c) and (d).



(a) Color (b) Confocal

Figure 5.2: Illustrating Large Content Differences. A red dot represents a keypoint detected by PIIFD [17]. A PIIFD descriptor is built in a local region as enclosed by a green square.

Comparing the two images in Figure 5.2 (a) and (b), content differences are displayed in two aspects. First, the pixels in the confocal image are all spatially close each other, whereas many pixels in the color image are unconnected. Second, the color image in Figure 5.2 (a) presents more intensity variations as compared to the confocal image in Figure 5.2 (b).

5.2.2 A Measure for Evaluating Content Differences

Like the two regions shown in Figure 5.2 (a) and (b), color and confocal images are visually different. In order to quantitatively observe content differences between corresponding regions, we tentatively use the distance between descriptors as a measure and the PIIFD descriptor [17] is used. In describing the local region around a PIIFD keypoint, $4 \times 4 = 16$ orientation histograms are built. In an orientation histogram, normalized gradient magnitudes are incremented to be the value of each orientation bin. Details about building PIIFD descriptors can be found in Section 2.5.1.2 of Chapter 2. Thus, if image contents between two regions are very similar, the descriptor distance is very small; otherwise, the distance is relatively large. Firstly, a

PIIFD descriptor in the reference image is denoted as

$$D_r^i = \{D_r^{i1}, D_r^{i2}, \ldots D_r^{in}\}, \tag{5.1}$$

where n is the dimensionality of the descriptor. Likewise, a PIIFD descriptor in the target image is represented as

$$D_t^i = \{D_t^{i1}, D_t^{i2}, \ldots D_t^{in}\}. \tag{5.2}$$

Then, the Euclidean distance between two PIIFD descriptors, D_r^i and D_t^i, is calculated by

$$d(D_r^i, D_t^i) = \sqrt{\sum_{k=1}^{n} (D_r^{ik} - D_t^{ik})^2}. \tag{5.3}$$

The smaller $d(D_r^i, D_t^i)$ is, the closer the two descriptors are. In other words, a relatively large $d(D_r^i, D_t^i)$ indicates the content differences between corresponding regions are relatively large. Thus, Equation 5.3 is used to measure the content differences between the regions of corresponding descriptors. With the distance between two descriptors defined in Equation 5.3, the average distance between corresponding descriptors in the reference and target images can be calculated by

$$\overline{d} = \frac{1}{N} \sum_{i=1}^{N} d(D_r^i, D_t^i), \tag{5.4}$$

where N is the number of corresponding descriptors in two images. Thus, the average distance between corresponding descriptors, $\overline{d}$, is used to measure the content differences between two images.

5.2.3 Statistics on Content Differences

With Equation 5.3, the distance between each pair of corresponding descriptors can be calculated for an image pair. Figure 5.3 plots the distances of all corresponding descriptors for the color and confocal images shown in Figure 5.1 (c) and (d). For 33 corresponding descriptors, the average distance between corresponding descriptors, $\overline{d}$, is equal to 0.62. According to the Euclidean distance between two PIIFD descriptors in Equation 5.3, the average distance is relatively large.

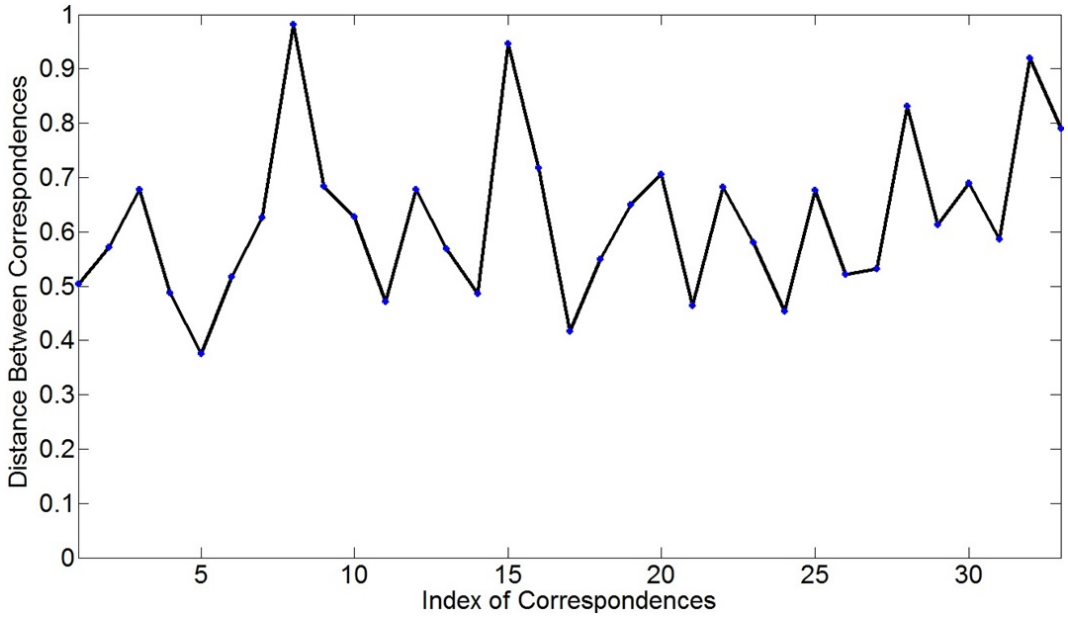


Figure 5.3: Descriptor Distances between Correspondences for the Color and Confocal Images Shown in Figure 5.1 (c) and (d)

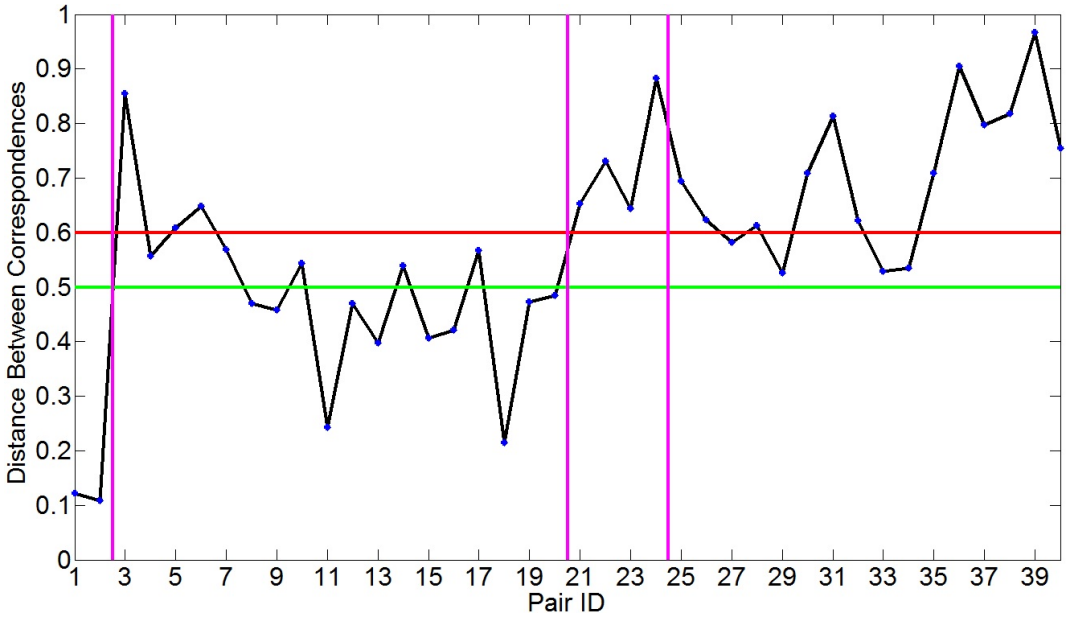


Figure 5.4: Average Descriptor Distance of Correspondences. The three vertical lines separate image pairs from the tested four data sets.

In order to compare the content differences between different image pairs, the average distance between corresponding descriptors, $\overline{d}$, is calculated for each of the 40 tested image pairs using Equation 5.4, as shown in Figure 5.4. Regarding which data

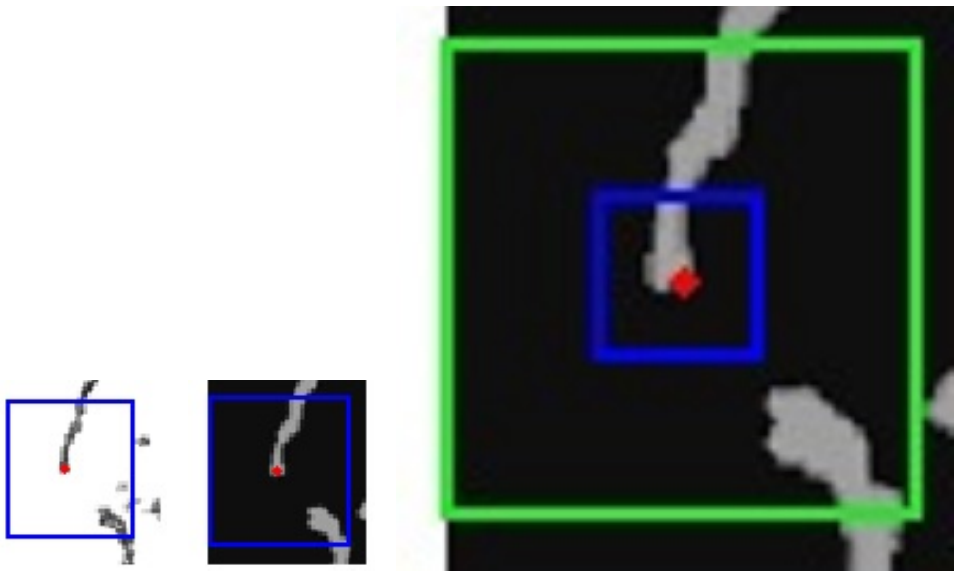
set and imaging category an image pair belongs to, details can be found in Section 3.4.1.2 of Chapter 3. For Data Set 1 (pairs 1-2), $\overline{d}$ is very small, indicating that the content differences in the two image pairs are quite small. For Data Set 2 (pairs 3-20), the $\overline{d}$ values for 15 pairs out of 18 are below 0.60 and for 10 pairs the values are smaller than 0.50. For Data Set 3 (pairs 21-24), none of the four image pairs hold a $\overline{d}$ value which is below 0.60. For Data Set 4 (pairs 25-40), the $\overline{d}$ values for all 16 pairs are larger than 0.50 and only four pairs have $\overline{d}$ values below 0.60. Overall, the $\overline{d}$ values are generally large for image pairs in Data Sets 3 and 4, which indicates the content difference are relatively large.

5.3 Discussing Scale Invariance

Scale invariance will be discussed in this section. First, we will analyze the significance of scale invariance to image registration. Next, we will illustrate how invariant the PIIFD descriptor is to scale differences and its impact on GI-PIIFD.

5.3.1 Significance of Scale Invariance to Image Registration

It is important to achieve scale invariance in registering images as the reference and target images may contain structures at different scales [54]. For a feature-based image registration technique such as [19, 56], a scale is estimated and assigned to each keypoint in a scale-space representation [50]. The scale of a keypoint determines the size of the local region in which a descriptor is built. Thus, whether the scale estimation is accurate directly affects the feature description and matching stages. If the estimated scale is inaccurate, the distance between a pair of corresponding keypoints is likely to be larger than it should be. Consequently, there will be a high possibility that this potentially true match is rejected in the matching stage. Due to an inaccurate scale estimation, the final registration performance is likely to be undermined.



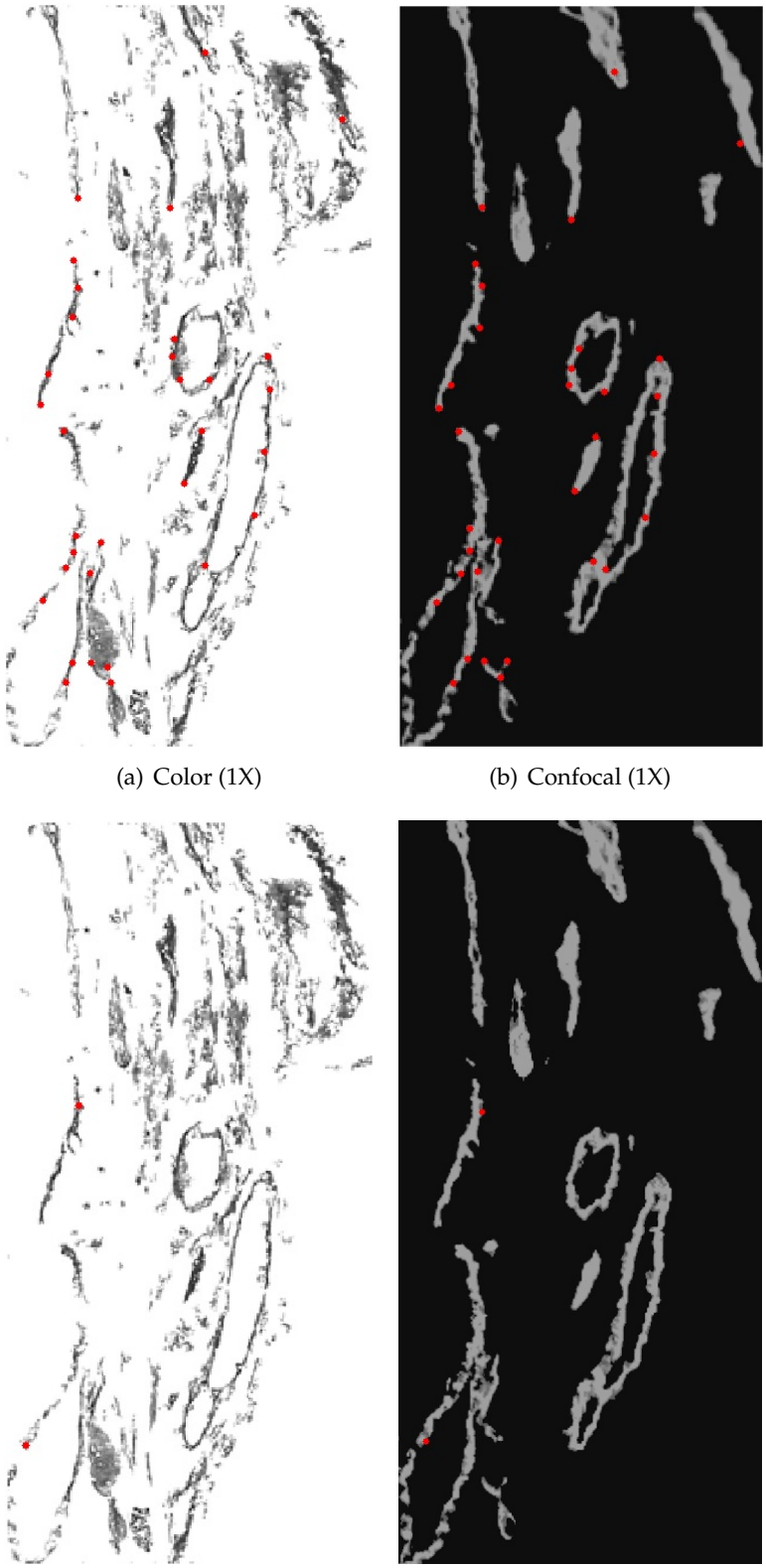
(a) (b) (c)

Figure 5.5: Regions for Building PIIFD Descriptors at Different Scales. A red dot in each sub-figure represents a PIIFD keypoint. Images in (a) and (b) are at similar scales. The scale difference between (c) and (b) is three times. In (c), the local region in the blue square is used in building the PIIFD descriptor, and the region within the green square corresponds to the regions in (a) and (b).

5.3.2 Scale Variance of PIIFD Descriptor

The PIIFD descriptor was proposed in [17] for registering multi-modal retinal images. The size of a local region for building PIIFD descriptor is fixed at 40×40 pixels because there is a minor scale difference between retinal images tested in [17]. Using the same setting as [17] for the size of local regions, we have illustrated corresponding keypoints which are manually extracted from color and confocal images, as shown in Figure 5.5. Figure 5.5 (c) is three times Figure 5.5 (a) and (b) with respect to scale difference. The local regions in Figure 5.5 (a) and (c) only partially correspond. Accordingly, the image structures which are represented in building PIIFD descriptors are not equivalent.

We now explain how GI-PIIFD [49] is affected by the scale variance of the PIIFD descriptor as GI-PIIFD is the benchmark multi-modal image registration technique in our work. As elaborated in Section 2.5.3 of Chapter 2, GI-PIIFD determines initial mappings of keypoints by selecting a set of closest descriptors, followed by matching



(a) Color (1X) (b) Confocal (1X)

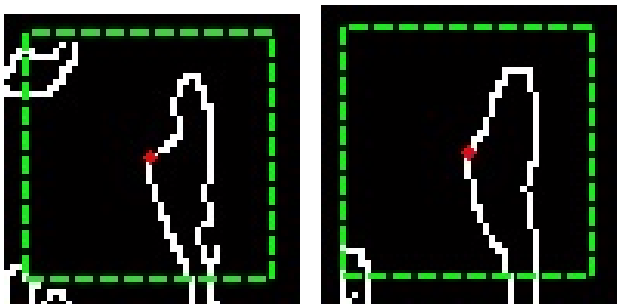
(c) Color (1X) (d) Confocal (3X)

Figure 5.6: An Example of Correspondences in Initial Mappings of Keypoints Using GI-PIIFD. (a) and (b) are at similar scales; the scale of (d) is three times that of (c).

triplets of keypoints. Due to the scale variance of the PIIFD descriptor, the number of correspondences appearing in initial mappings is likely to decrease as the scale difference between the reference and target images increases. Figure 5.6 gives two examples of correspondences in initial mappings of GI-PIIFD in registering images with similar scales and with a scale difference of three times respectively. There are 33 correspondences of 58 in registering Figure 5.6 (a) and (b), whereas there are only two correspondences of 21 in registering Figure 5.6 (c) and (d). Obviously, there is no chance of matching a triplet pair where all the three pairs of keypoints are corresponding, in registering Figure 5.6 (c) and (d). Consequently, it is impossible to effectively register the two images.

5.4 Robustness of Curvatures of Fast-CPDA Corners to Content Differences

As illustrated in Sections 5.2 and 5.3, PIIFD descriptors have limitations with regard to robustness to content differences. In this section, we will illustrate how curvatures of Fast-CPDA corners [6] are robust to content differences. Details for the Fast-CPDA corner detector can be found in Section 2.3.3 of Chapter 2.



(a) Color (b) Confocal

Figure 5.7: Illustrating Curvature Similarity between Corresponding Corners. A red dot represents a corner detected by the Fast-CPDA corner detector [6]. The dashed square is used to highlight corresponding regions shown in Figure 5.2.

In PIIFD, keypoints are detected using the Harris corner detector which relies on intensity variations in a small neighborhood [28]. The PIIFD descriptor is built based

on a local region around each keypoint, where normalized gradient magnitudes are used to build orientation histograms. Due to the use of gradient information, the PIIFD descriptor is sensitive to content differences within the local region.

Let us now look into how curvatures of Fast-CPDA corners are robust to content differences. As elaborated in Section 2.3.3 of Chapter 2, the Fast-CPDA corner detector [4, 6] estimates curvatures of contour points using the chord-to-point distance accumulation technique [27] and treats maxima contour points with regard to curvature values as candidate corners. Thus, the curvature of a Fast-CPDA corner is independent of intensity or gradient changes in the neighborhood of the corner.

Figure 5.7 shows a pair of corresponding corners which are detected by the Fast-CPDA corner detector. Note that, the local regions highlighted in Figures 5.7 and 5.2 are equivalent. Based on the curvature estimation in the Fast-CPDA corner detector, the curvatures for the two corners in Figure 5.7 (a) and (b) are very similar. As stated in Section 5.2.1, there are large content differences between the two regions shown in Figure 5.7. Hence, curvatures of Fast-CPDA corners are more robust to content differences as compared to PIIFD descriptors.

5.5 COREG: A Multi-modal Image Registration Technique based on Corners

In this section, we will propose a COrner based REGistration technique which is called COREG for the referencing purpose. An overview of COREG is first given, followed by a few key issues in detail.

5.5.1 Overview of COREG

The proposed COREG is designed based on the registration framework in [49]. GI-PIIFD [49] has two mains limitations. First, GI-PIIFD can only deal with small changes in affine transformation, and is therefore not scale-invariant. Second, the PIIFD descriptor is only partially invariant to intensity variations, so that the robustness to content differences cannot be ensured in registering multi-modal

images. Overall, our aim is to achieve greater robustness to large differences in image contents and scale as compared to GI-PIIFD [49]. To achieve greater robustness to large content differences, we will explore curvature similarity between corners and propose a novel corner descriptor, which will be elaborated in Sections 5.5.2 and 5.5.4. To deal with scale differences, geometric relationships between corner triplets are taken into account, as stated in Section 5.5.3.

The steps in COREG are as follows.

i. Detecting corners

   Corners are detected in the reference and target images using the Fast-CPDA corner detector [6].

ii. Determining initial mappings of corners using curvature similarities

   Relative to each reference corner, curvature similarities of all the corners in the target image are ranked. By selecting highly-ranked corners, candidate matches of each reference corner are determined. Curvature similarity will be described in Section 5.5.2.

iii. The first round of matching of corner triplets

   With initial mappings of corners determined in Step ii, all the possible mappings of corner triplets are generated. Each pair of corner triplets in the reference and target images are compared and accordingly a transformation is computed. The transformation is used to transform the target image onto the reference image. The corresponding edge images are overlapped and therefore the Number of Overlapped Pixels (NOP) is computed. By comparing NOP values, the pair of corner triplets with the maximum NOP is selected. The triplet pair selected is denoted as TP_1. Note that details about NOP can be found in Section 2.5.3 of Chapter 2.

iv. Estimating a scale difference between the reference and target images

   The scale difference between the reference and target images is estimated from the pair of corner triplet TP_1. The estimated scale difference is obtained by averaging the length ratios between corresponding line segments in the two corner triplets. This will be illustrated in Section 5.5.3.

v. The second round of matching of corner triplets

First, the reference and target images are resized using the scale difference estimated in Step iv. Second, a novel local descriptor called Distribution of Edge Pixels Along Contour (DEPAC) is built for each corner. The proposed DEPAC descriptor will be stated in Section 5.5.4. Similar to Step ii, the initial mappings of corners can be determined by ranking the DEPAC descriptor distances. Next, matching of corner triplets is carried out based on curvature similarity and the DEPAC descriptor respectively. Accordingly, two pairs of corner triplets are obtained. The pair of corner triplets which correspond to a higher NOP is denoted as TP_2.

Table 5.1: Comparing Steps in COREG and GI-PIIFD

No.	**COREG**	**GI-PIIFD**
i	Detecting corners	Detecting PIIFD keypoints
ii	Determining initial mappings of corners using curvature similarities	Determining initial mappings of keypoints using PIIFD descriptors
iii	The first round of matching of corner triplets	Matching of keypoint triplets
iv	Estimating a scale difference	N/A
v	The second round matching of corner triplets	N/A
vi	Determining a triplet pair	Determining a triplet pair
vii	Refining localization of the selected pair of corner triplet	N/A
viii	Estimating a transformation and aligning images	Estimating a transformation and aligning images

vi. Determining a triplet pair

The two triplet pairs, TP_1 and TP_2, are compared in terms of NOP. A decision is made to select the triplet pair with the higher NOP. The selected triplet pair is denoted as TP_s.

vii. Refining localizations of the selected pair of corner triplets TP_s

With the triplet pair determined, the localizations of corner pairs in the triplet pair are refined in a small neighborhood. If a higher NOP can be achieved, then the triplet pair is updated with the refined corner localizations.

viii. Estimating a transformation and aligning images

A transformation is estimated from the selected pair of corner triplet TP_s. The estimated transformation is finally used for aligning the reference and target images.

Table 5.1 compares the steps in COREG and GI-PIIFD [49], which clearly indicates the differences between the two techniques. Compared with GI-PIIFD, the novelties of COREG lie in Steps ii, iv, v and vii. For Steps ii and v, we will describe curvature similarity between corners in Section 5.5.2 and the DEPAC descriptor in Section 5.5.4. Steps iv and vii will be elaborated in Sections 5.5.3 and 5.5.5 respectively.

5.5.2 Curvature Similarity between Corners

Let us firstly define corners in the reference and target images as

$$C_r = \{C_r^1, C_r^2, \dots C_r^{N_r}\} \tag{5.5}$$

and

$$C_t = \{C_t^1, C_t^2, \dots C_t^{N_t}\}, \tag{5.6}$$

where N_r and N_t denote the number of corners in the reference and target images. Likewise, the curvatures of corners are defined as

$$K_r = \{K_r^1, K_r^2, \dots K_r^{N_r}\} \tag{5.7}$$

and

$$K_t = \{K_t^1, K_t^2, \dots K_t^{N_t}\}. \tag{5.8}$$

Given two corners from the reference and target images, their curvature similarity is defined as

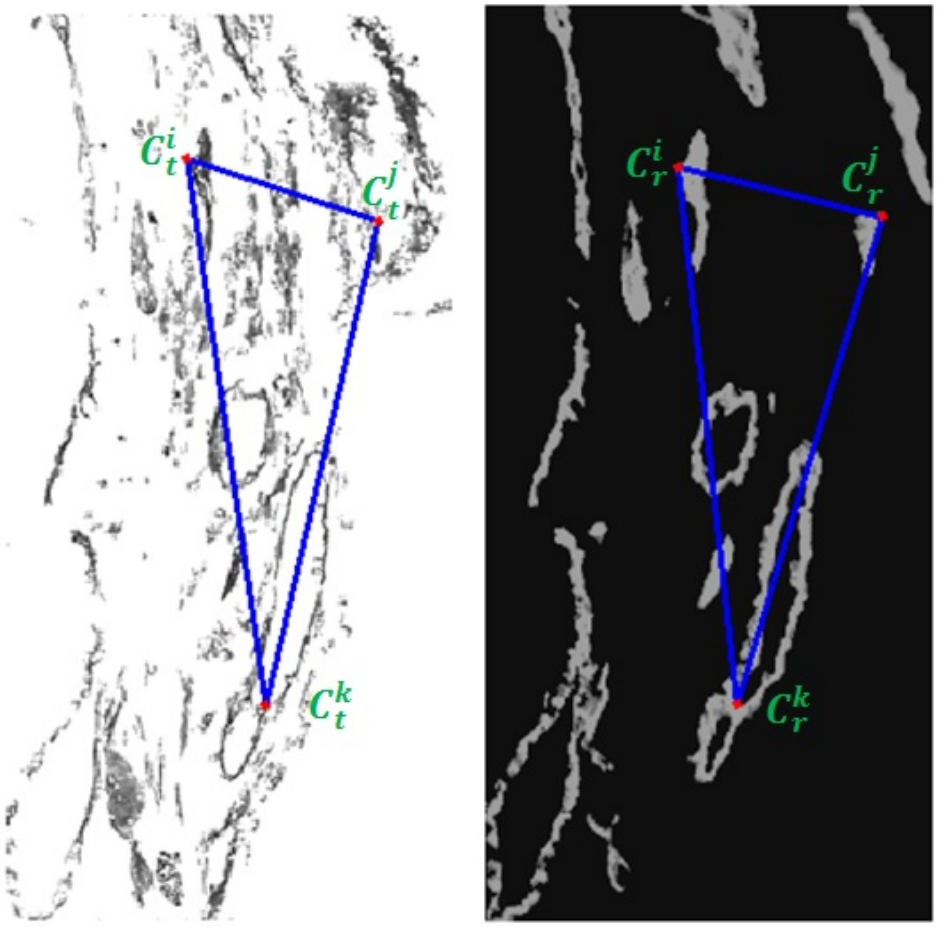
$$s^{ij} = \frac{|K_r^i - K_t^j|}{K_r^i}, \tag{5.9}$$

where $1 \leq i \leq N_r$ and $1 \leq j \leq N_t$. Explicitly, the smaller a s^{ij} value is, the higher the curvature similarity between two corners is.

With the curvature similarity defined in Equation 5.9, all the corners in the target image are ranked by their curvature similarities relative to each reference corner. The highly-ranked corners comprise candidate matches. Thus, a reference corner is mapped to these candidate matches as

$$C_r^i \mapsto \{C_t^1, C_t^2, \dots C_t^{N_c}\}, \tag{5.10}$$

where N_c represents the number of candidate matches. Given three corners C_r^i, C_r^j and C_r^k in the reference image, a corner triplet is generated. With candidate matches relative to each reference corner as Equation 5.10 describes for C_r^i, all the possible corner triplets are generated in the target image.



(a) Color (b) Confocal

Figure 5.8: An Example of a Triplet Pair for Estimating a Scale Difference. Note that the image size in the figure does not reflect the actual scale difference between the two images.

5.5.3 Scale Estimation

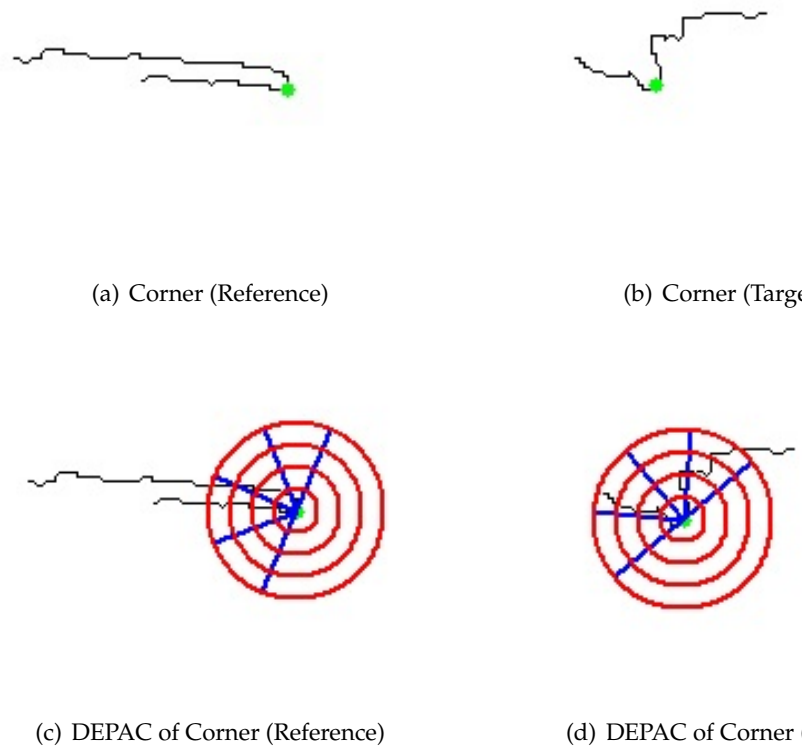
As stated in Step iii of COREG in Section 5.5.1, a pair of corner triplets, TP_1, is selected after the first round matching of corner triplets. Our way of estimating a scale difference is based on the triplet pair TP_1. Figure 5.8 shows TP_1 in registering a pair of color and confocal images. The three corners C_t^i, C_t^j and C_t^k in the color image correspond to the three corners C_r^i, C_r^j and C_r^k in the confocal image. With the three corner pairs, the scale difference between the two images is estimated by averaging the length ratios between corresponding line segments in the two corner triplets, i.e.,

$$\sigma = \frac{1}{3} \times \left(\frac{|\overrightarrow{C_r^i C_r^j}|}{|\overrightarrow{C_t^i C_t^j}|} + \frac{|\overrightarrow{C_r^j C_r^k}|}{|\overrightarrow{C_t^j C_t^k}|} + \frac{|\overrightarrow{C_r^k C_r^i}|}{|\overrightarrow{C_t^k C_t^i}|}\right). \tag{5.11}$$

In the example shown in Figure 5.8, the ground-truth scale difference between the color and confocal images is 1:2.73, whereas the estimated scale difference is 1:2.82. We can see the estimated scale difference is quite close to the ground-truth one. The accuracy of scale estimation for all the tested image pairs will be illustrated in Section 5.6.2.

5.5.4 DEPAC: A Proposed Corner Descriptor

Curvature [3, 4, 6, 30] is an important representation of corners. The curvature of a corner describes how the edge pixels move along the contour of the corner in a small neighborhood. In order to better represent corners, we will propose a novel corner descriptor. Firstly, an example is given to illustrate the limitations of representing corners only using their curvatures. Figure 5.9 (a) and (b) show two corners and their contours that are extracted from a reference image and its target image in our tested image pairs. The two corners are not corresponding in terms of ground-truth locations. The curvatures of the two corners are very close as the edges in a small neighborhood are structurally very similar. However, the edge structures in a larger neighborhood are significantly more different. Based on this analysis, a novel corner descriptor is proposed in order to capture more edge information surrounding a corner as compared to its curvature. Note that only the edge pixels along the contour



(a) Corner (Reference) (b) Corner (Target)

(c) DEPAC of Corner (Reference) (d) DEPAC of Corner (Target)

Figure 5.9: Building the DEPAC Descriptor

where the corner is located are represented in the proposed corner descriptor, due to the fact that the number of edges may largely differ in the corresponding parts of multi-modal images. Thus, the proposed corner descriptor is called Distribution of Edge Pixels Along Contour (DEPAC).

Let C_r^i, C_t^j, $\Gamma(C_r^i)$ and $\Gamma(C_t^j)$ denote the two corners and their contours shown in Figure 5.9 (a) and (b). We illustrate how a DEPAC corner descriptor is built using C_r^i and $\Gamma(C_r^i)$ as follows.

i. Concentric circles are plotted by taking the corner as the center, as shown in Figure 5.9 (c). Let R denote the radius of the internal circle. The radius of a concentric circle is incremented by R, from inside to outside. In our implementations, R is set to five pixels.

ii. The main orientation of the corner, O_m, is defined by averaging the orientations

of two tangents [30]. In Figure 5.9 (c), the middle blue line denotes the main orientation.

iii. Orientation bins are defined at the two sides of the main orientation. As plotted by blue lines in Figure 5.9 (c), the four quantized orientations are $O_1 = O_m - 90°$, $O_2 = O_m - 45°$, $O_3 = O_m$ and $O_4 = O_m + 45°$ in an anticlockwise order. With four concentric circles and four quantized orientations, 16 sub-regions are defined in the neighborhood of the corner and each sub-region is denoted as (c,o), where $1 \leq c \leq 4$ and $1 \leq o \leq 4$.

iv. In the sub-region (c,o), the number of edge pixels along the contour is incremented by one if an edge pixel, P_e, satisfies

$$(c-1) \times R < d(P_e, C_r^i) \leq c \times R \tag{5.12}$$

and

$$O_o \leq \overrightarrow{P_e C_r^i} < O_{o+1}, \tag{5.13}$$

where $d(P_e, C_r^i)$ is the Euclidean distance between P_e and C_r^i, $1 \leq c \leq 4$ and $1 \leq o \leq 4$. The number of edge pixels computed for the sub-region (c,o) is denoted as $NEP_{c,o}$. For the corner C_r^i shown in Figure 5.9 (c), the number of edge pixels in each sub-region is listed in Table 5.2.

v. The number of edge pixels in each sub-region, $NEP_{c,o}$, is normalized into [0,1] by

$$NEP_{c,o} = \frac{NEP_{c,o}}{max\{NEP_{c,o}\}}. \tag{5.14}$$

Finally, the DEPAC descriptor is generated.

Table 5.2: Number of Edge Pixels in Each Sub-region for Corner C_r^i

orientation / **circle**	**1**	**2**	**3**	**4**
1	5	2	1	0
2	0	5	5	0
3	0	6	6	0
4	0	1	10	0

Table 5.3: Number of Edge Pixels in Each Sub-region for Corner C_t^j

orientation / circle	**1**	**2**	**3**	**4**
1	4	1	1	2
2	0	5	6	0
3	8	4	5	0
4	8	0	6	0

To compare the DEPAC descriptors built for the two corners, C_r^i and C_t^j, the number of edge pixels in each sub-region for C_t^j is listed in Table 5.3. We can clearly see that the two DEPAC descriptors are very different. Thus, our proposed DEPAC descriptor captures important edge information in the neighborhood of a corner.

It should be noted that scale invariance must be ensured in building DEPAC descriptors for corners in the reference and target images. Ideally, the size of concentric circles for building DEPAC descriptors should be in line with the actual scale difference between the reference and target images. To achieve scale invariance, the estimated scale difference σ, which has been discussed in Section 5.5.3, is used as

$$R_r = \sigma \times R_t, \tag{5.15}$$

where R_r and R_t denote the radius values of the internal circle for building DEPAC descriptors in the reference and target images, respectively.

5.5.5 Refining Localizations

As stated in Section 5.5.1, a triplet pair, TP_s, is selected from TP_1 and TP_2 by selecting the one with a higher NOP value. Let $C_r^i, C_r^j, C_r^k \mapsto C_t^i, C_t^j, C_t^k$ denotes TP_s. Based on our analysis, two corners of a match in this triplet pair might not be accurately corresponding. As shown in Figure 5.10, there is possibly an image pixel, C_t^x, in a small neighborhood of the corner C_t^i that leads to a more accurate match. This is very likely to occur in multi-modal images as there may be a localization error in detecting corners due to different amounts of noises at corresponding parts in the reference and target images.

The refinement of localizations is carried out by searching image pixels in an $r \times r$ window, where r is the width of the searching window. Note that the searching

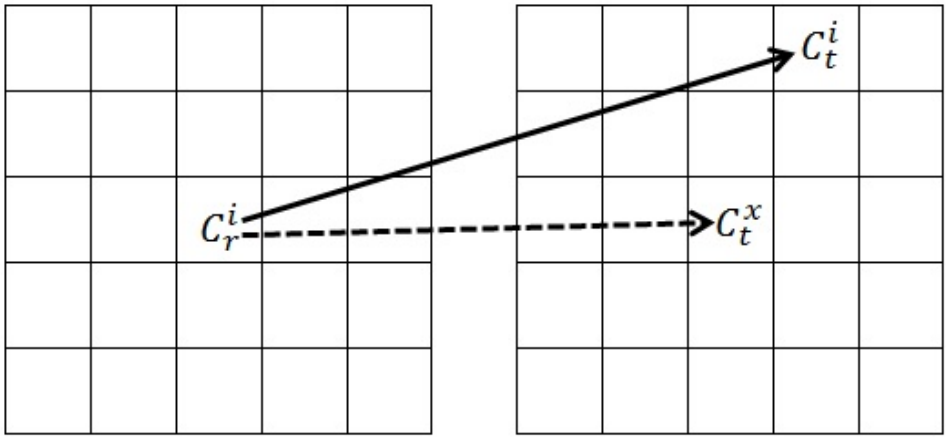


Figure 5.10: Refining Localizations

process is only performed in the target image while the corner localizations of the triplet pair in the reference image remain unchanged. As the searching window is set for each corner of the triplet in the target image, $(r \times r)^3 = r^6$ triplet pairs are additionally generated. If any triplet pair out of these r^6 pairs achieves a higher NOP, the triplet pair $C_r^i, C_r^j, C_r^k \mapsto C_t^i, C_t^j, C_t^k$ is accordingly updated. In our experiments, r is equivalent to five.

5.5.6 A Special Consideration

In COREG, spatial relationships between corners are used by representing and matching corner triplets. Where the number of corners is smaller than three, it is impossible to generate a corner triplet. Where this is the case, the registration process will be terminated. Thus, special consideration must be taken to ensure there are sufficient corners for generating corner triplets. In the Fast-CPDA corner detector [6], edges are detected using the Canny edge detector [16]. In the Canny edge detector [16], a high threshold and a low threshold are used to define strong and weak edge pixels respectively. In COREG, the high threshold for the Canny edge detector is empirically lowered to preserve more edges so that more corners are potentially detected, in the cases where the number of corners is smaller than three using the default threshold.

5.6 Performance Study

We will evaluate the proposed COREG from the following three aspects. First, we will measure the accuracy of the proposed way of estimating scale difference. In registering each image pair, the estimated scale difference will be compared with the ground-truth scale difference. Second, the registration performance of COREG will be evaluated against GI-PIIFD at various scale differences. Third, COREG is also compared with MOG-IS-SIFT and elastix [45]. MOG-IS-SIFT is a multi-modal image registration technique we have proposed in Chapter 3, while elastix is regarded as a benchmark in the category of intensity-based image registration techniques. Due to the overall poor performance of elastix and MOG-IS-SIFT in registering multi-modal microscopic images, the comparisons between elastix, MOG-IS-SIFT and COREG are only carried out at the 1X vs 1X scale difference.

The tested data sets are the same as those used in Chapter 3. Details about the data sets can be found in Section 3.4.1.2 of Chapter 3. The four data sets include 40 image pairs which have similar scales and we call them the base image pairs. With these base image pairs, we have manually generated corresponding image pairs which have scale differences of 1.5, 2, 3 and 4 times, respectively. Thus, five patterns of scale differences are tested. For the referencing purpose, the five patterns are called 1X vs 1X, 1X vs 1.5X, 1X vs 2X, 1X vs 3X and 1X vs 4X, respectively. Here, X is equivalent to times with regard to a scale difference.

With regard to registering multi-modal microscopic image pairs, whether DSS (the technique we have proposed in Chapter 4) is used should be taken into consideration. When DSS is used together with GI-PIIFD, COREG, elastix and MOG-IS-SIFT, the four techniques are called DSS-GI-PIIFD, DSS-COREG, DSS-elastix and DSS-MOG-IS-SIFT respectively.

5.6.1 Evaluation Metric

To carry out quantitative performance comparisons, average registration error [109] is used to measure the overlap error after aligning the reference and target images with

the estimated transformation. Average registration error is defined as

$$\varepsilon = \frac{1}{H \times W} \sum_{x=1}^{W} \sum_{y=1}^{H} \|T_e(x,y) - T_g(x,y)\|, \tag{5.16}$$

where H and W are the height and width of the reference image, T_g is the ground-truth transformation and T_e is the estimated transformation. The smaller the ε value is, the better the registration performance will be. For the referencing purpose, average registration error is called ARE in short.

5.6.2 Accuracy of Scale Estimation

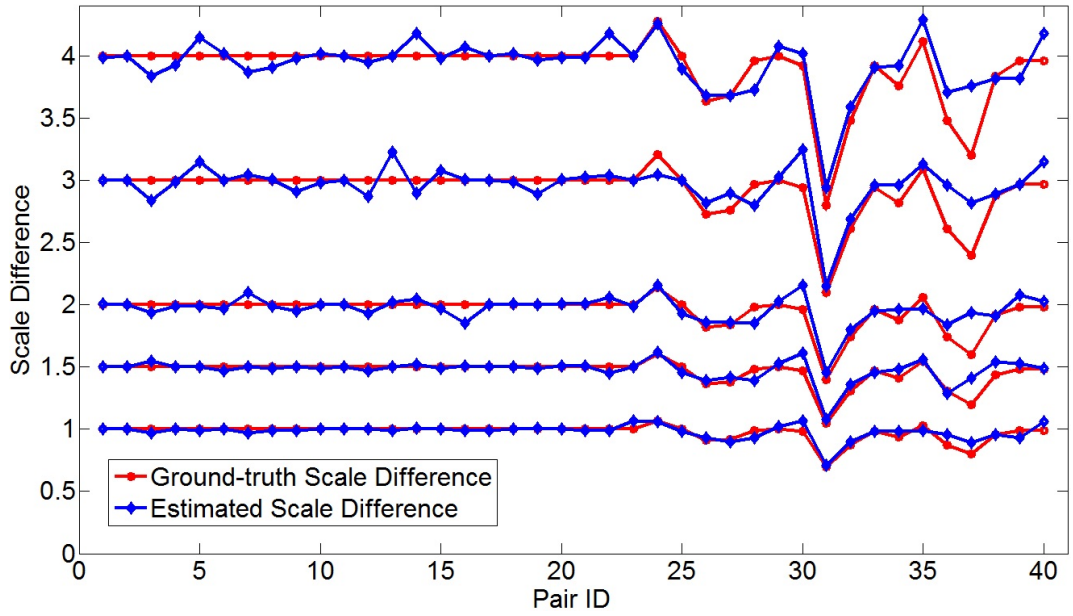


Figure 5.11: Comparing Estimated and Ground-truth Scale Differences

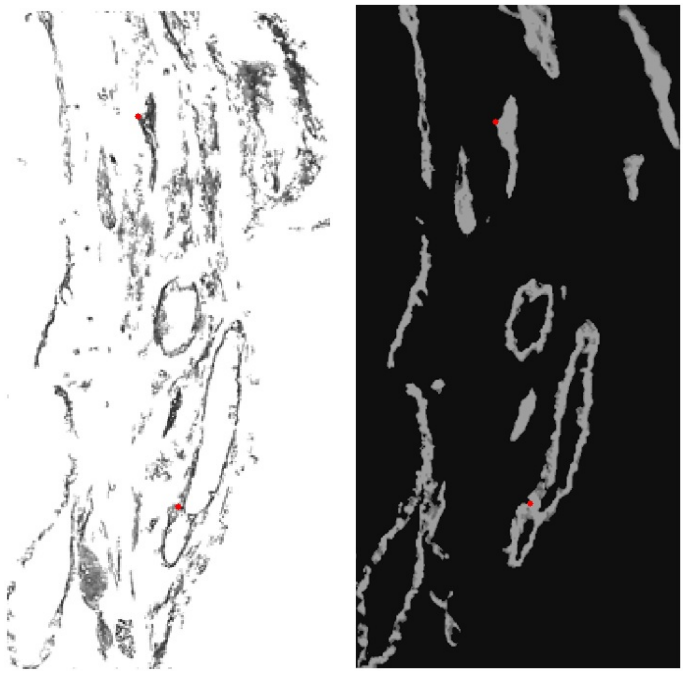
As discussed in Section 5.3, achieving scale invariance is of great importance in the process of image registration. In our proposed COREG, the reference and target images are resized using the estimated scale difference. If the estimated scale difference is close to the ground-truth scale difference, the reference and target images will have similar scales after being resized. Here, the accuracy of scale estimation is measured by an error which deviates from the ground-truth scale difference. Let σ_e and σ_g denote the estimated scale difference and the ground-truth scale difference

respectively. The error of estimating a scale difference is defined as

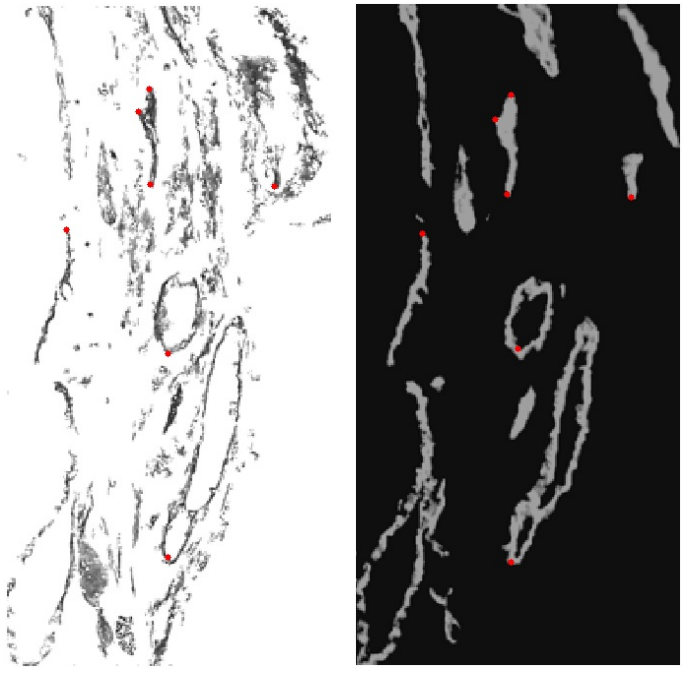
$$\varepsilon_s = \frac{|\sigma_e - \sigma_g|}{\sigma_g}\%. \tag{5.17}$$

Figure 5.11 compares the estimated and ground-truth scale differences for 40 image pairs at five patterns of scale differences. Note that DSS which has been proposed in Chapter 4 is performed on the 16 microscopic image pairs. It can be seen in Figure 5.11 that the estimated scale difference is in many cases close to the ground-truth scale difference. With the measure defined in Equation 5.17 for accuracy of scale estimation, a threshold is set to 5%. For these 40 pairs, with five patterns of scale differences from 1X vs 1X to 1X vs 4X, ε_s is below 5% in 33, 36, 35, 34 and 36 pairs, respectively.

Moreover, the advantage of using our proposed scale estimation can be seen in terms of initial mapping of feature points. In Figure 5.6 (c) and (d), we have presented the correspondences in initial mappings of PIIFD keypoints where there are only two correspondences. By comparison, Figure 5.12 shows correspondences which appear in initial mappings of corners before and after using scale estimation, when registering a pair of color and confocal images at the 1X vs 3X scale difference. Note that only curvature similarity is used to determine initial mappings of corners in obtaining the results in Figure 5.12. As shown in Figure 5.12 (a) and (b), only two correspondences out of eight appear in the initial mappings before using scale estimation. Therefore, there is no chance to compare and match a pair of corner triplets where all three corner pairs are truly matched. In contrast, there exist seven correspondences out of 19 in initial mappings of corners after using scale estimation, as shown in Figure 5.12 (c) and (d). The correspondences in initial mapping of corners shown in Figure 5.12 (c) and (d) provide a good basis for the following matching of corner triplets, thereby leading to an effective registration.



(a) Color (1X) (b) Confocal (3X)

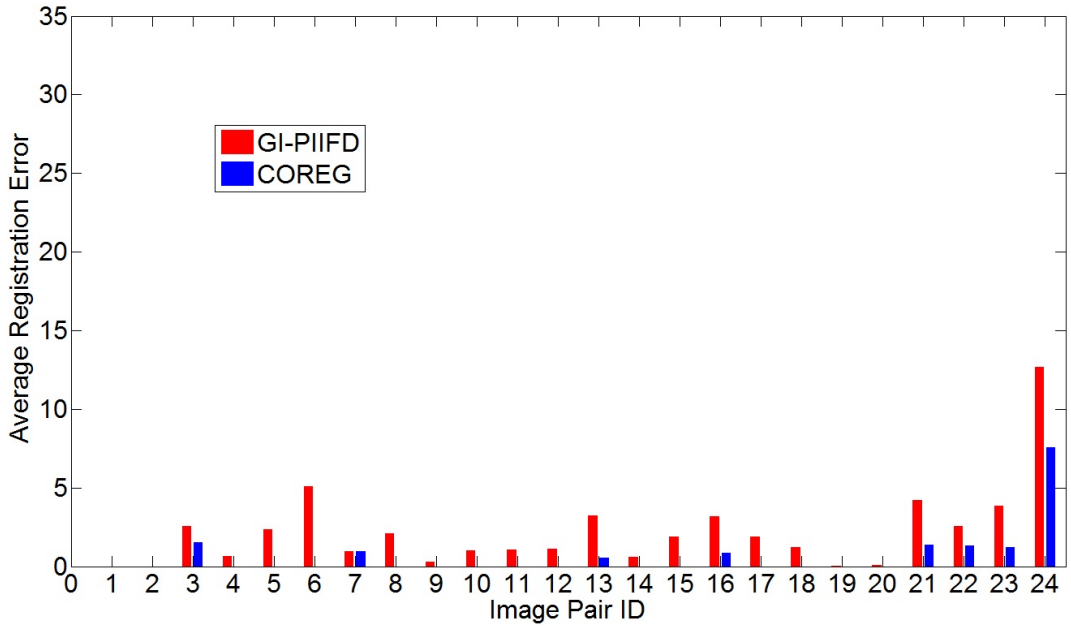


(c) Color (1X) (d) Confocal (3X)

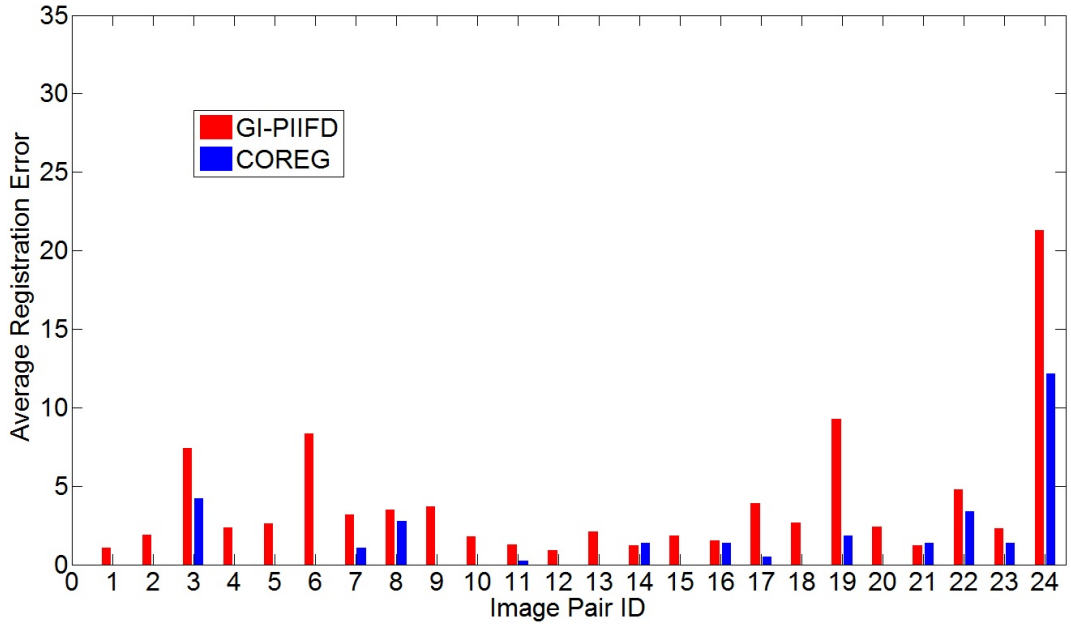
Figure 5.12: Number of Correspondences in Initial Mappings before and after Scale Estimation. (a) and (b): before using scale estimation; (c) and (d): after using scale estimation

5.6.3 Performance Comparisons

5.6.3.1 GI-PIIFD vs COREG on Non-Microscopic Images

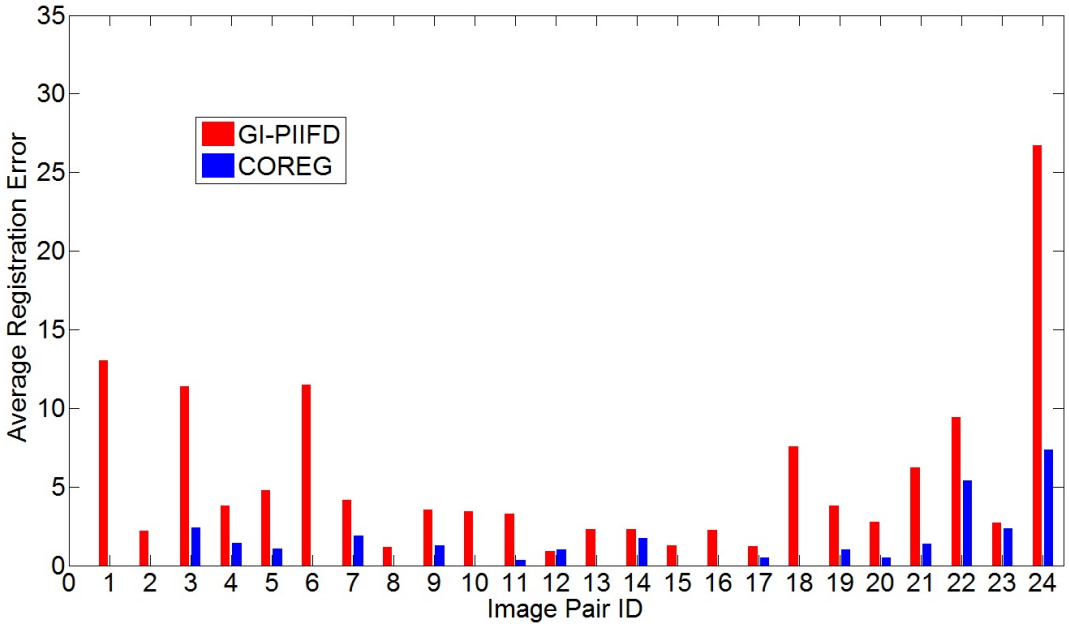


(a) 1X vs 1X

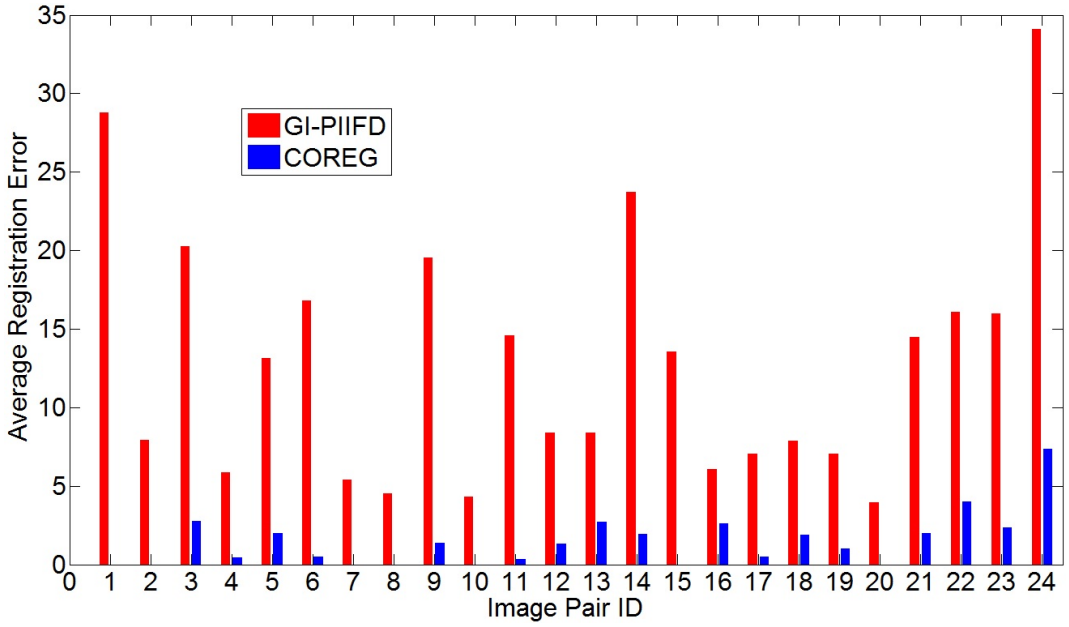


(b) 1X vs 1.5X

Figure 5.13: ARE Comparisons between COREG and GI-PIIFD for Non-microscopic Image Pairs (Part 1)



(a) 1X vs 2X



(b) 1X vs 3X

Figure 5.14: ARE Comparisons between COREG and GI-PIIFD for Non-microscopic Image Pairs (Part 2). The legend is the same as that in Figure 5.13.

GI-PIIFD and COREG are compared in terms of ARE across five patterns of scale differences, i.e. 1X vs 1X, 1X vs 1.5X, 1X vs 2X, 1X vs 3X and 1X vs 4X, as shown in Figures 5.13, 5.14 and 5.15. At the 1X vs 1X scale difference, ARE is generally

small using either GI-PIIFD or COREG except in registering pair 24, as shown in Figure 5.13 (a). Regarding pair 24, we have shown the image pair in Section 3.4.4 of Chapter 3. In pair 24, the objects are very unclear and content differences are very large. Still, COREG significantly improves the registration performance over GI-PIIFD in registering pair 24 across various scale differences. On average, the ARE for GI-PIIFD is 2.18, whereas COREG achieves an ARE of 0.64, in registering the 24 image pairs at the 1X vs 1X scale difference.

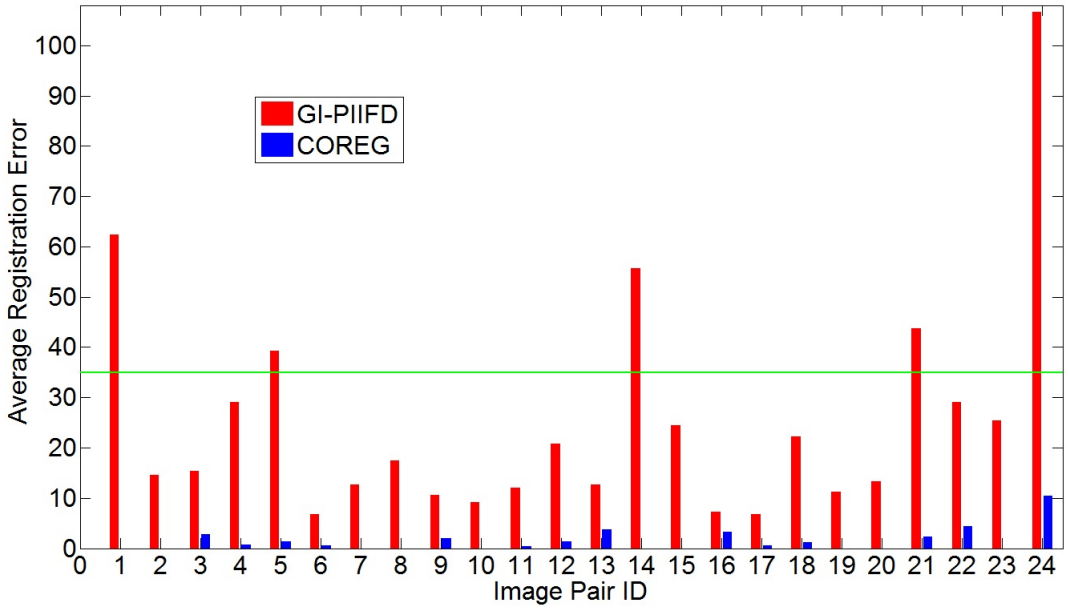


Figure 5.15: ARE Comparisons between COREG and GI-PIIFD for Non-microscopic Image Pairs (Part 3). The legend is the same as that in Figure 5.13. The scale difference is 1X vs 4X.

For the other four patterns of scale differences, GI-PIIFD and COREG are compared as shown in Figures 5.13 (b), 5.14 and 5.15. Please note that we have used 35 as the upper limit on the y axis in Figures 5.13 and 5.14. In Figure 5.15, the upper limit on the y axis is much higher. For a better comparison across all the five patterns of scale differences in Figures 5.13, 5.14 and 5.15, a horizontal line (35 at y axis) is plotted in Figure 5.15. As the scale difference increases, GI-PIIFD performs increasingly poor, whereas COREG is much more robust. In other words, the advantage of COREG over GI-PIIFD is more significant as the scale difference increases. Table 5.4 compares average ARE values between GI-PIIFD and COREG for the five patterns of scale differences. Note that the special consideration described in Section 5.5.6 is taken for

registering image pair 11 across the five patterns of scale differences. In registering image pair 11, the high threshold for the Canny edge detector is lowered from 0.35 to 0.25.

Table 5.4: Average ARE of Each Pattern of Scale Difference for Non-microscopic Images

Scale Difference	GI-PIIFD	COREG
1X vs 1X	2.18	0.64
1X vs 1.5X	3.85	1.32
1X vs 2X	5.48	1.23
1X vs 3X	12.83	1.46
1X vs 4X	25.36	1.45

5.6.3.2 GI-PIIFD vs COREG on Microscopic Images

In registering multi-modal microscopic images, comparisons will be made between four techniques, i.e. GI-PIIFD, COREG, DSS-GI-PIIFD and DSS-COREG, at five patterns of scale differences as shown in Figures 5.16, 5.17 and 5.18. Note that there are a few failures when using GI-PIIFD or COREG for registration, and these failures are marked with an $\times$. There are two different scenarios for these failures as follows. First, there is no corresponding corners which have been detected in a pair of color and confocal images. Second, there are a number of corresponding corners, however there is no correspondence in initial mappings of corners. For both scenarios, NOP values are all zero, indicating that there is no overlapped edge between edges of color and confocal images.

The ARE comparisons shown in Figures 5.16, 5.17 and 5.18 can be analyzed from different perspectives with the following conclusions.

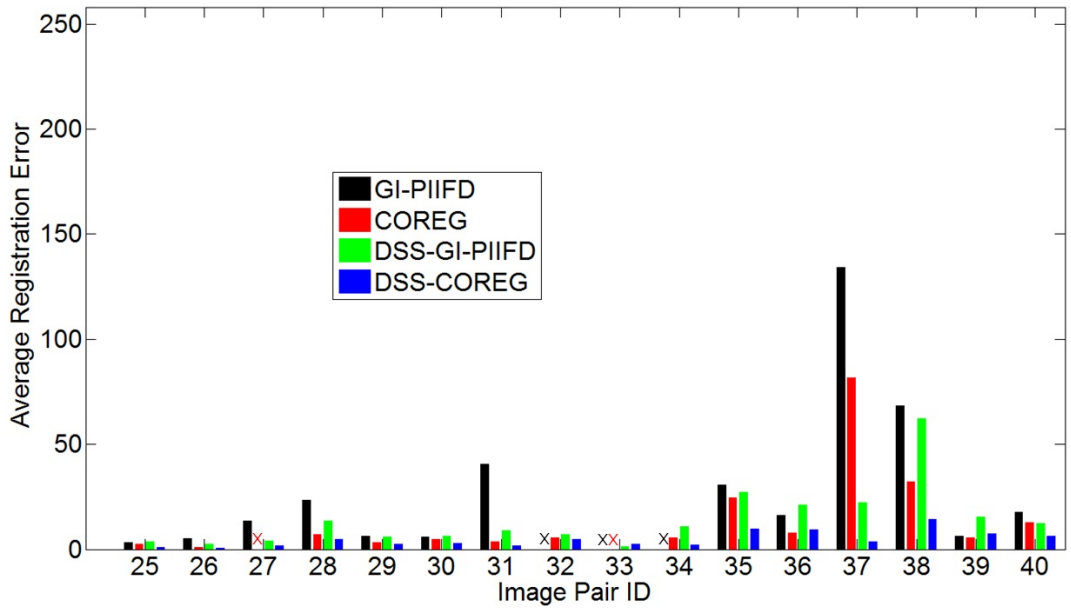
i. Across the five patterns of scale differences, COREG outperforms GI-PIIFD in 96.77% of cases, and DSS-COREG outperforms DSS-GI-PIIFD in all cases.

ii. DSS-COREG performs better than COREG in most cases. This simply verifies the significance of DSS which has been proposed in Chapter 4.

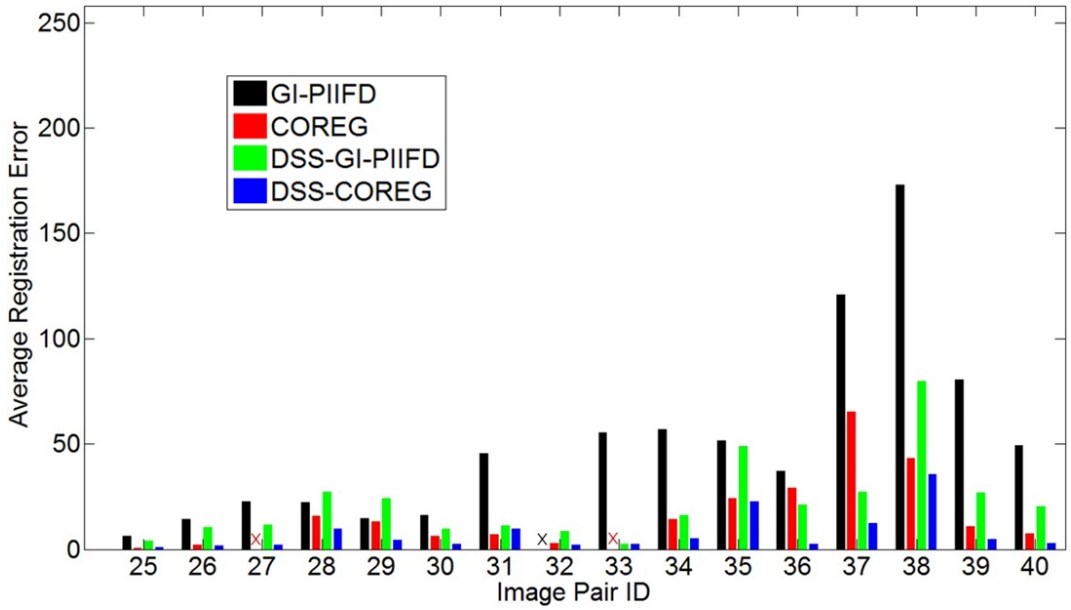
iii. As the scale difference increases, COREG shows increasing advantages over GI-PIIFD, and DSS-COREG shows increasing advantages over DSS-GI-PIIFD.

Therefore, the proposed COREG significantly improves the robustness to scale differences.

Overall, DSS-COREG achieves the best performance out of the four techniques.

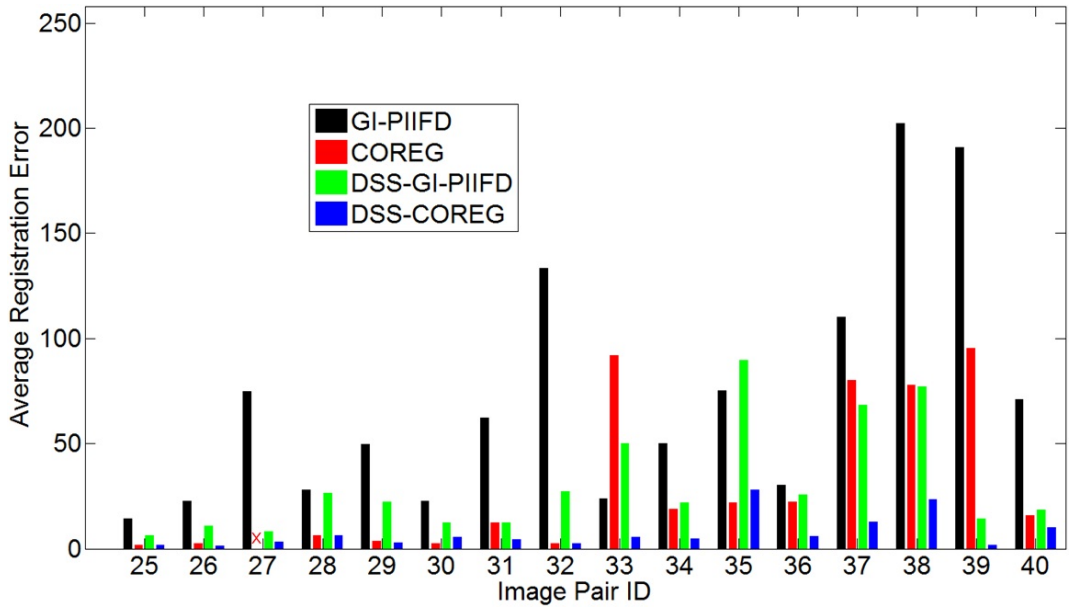


(a) 1X vs 1X

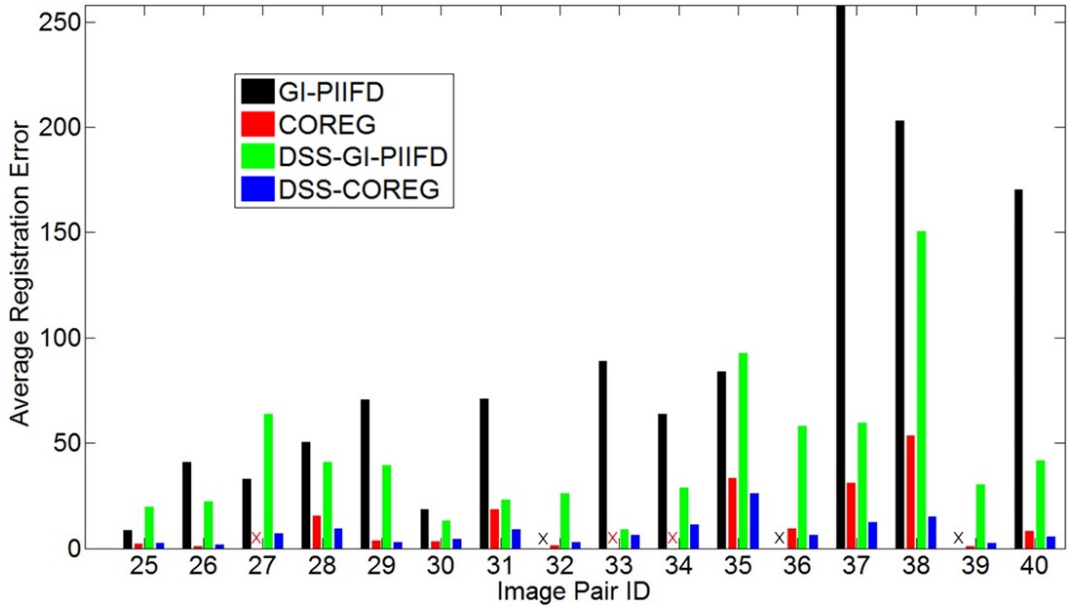


(b) 1X vs 1.5X

Figure 5.16: ARE Comparisons between COREG and GI-PIIFD for Microscopic Image Pairs (Part 1)



(a) 1X vs 2X



(b) 1X vs 3X

Figure 5.17: ARE Comparisons between COREG and GI-PIIFD for Microscopic Image Pairs (Part 2). The legend for the four techniques is the same as that in Figure 5.16.

Although DSS-COREG achieves the best performance among the compared four techniques, we have found that DSS-COREG has difficulties in registering very challenging image pairs. Considering all the five patterns of scale differences, the ARE values achieved by DSS-COREG for pairs 35 and 38 are relatively high as compared

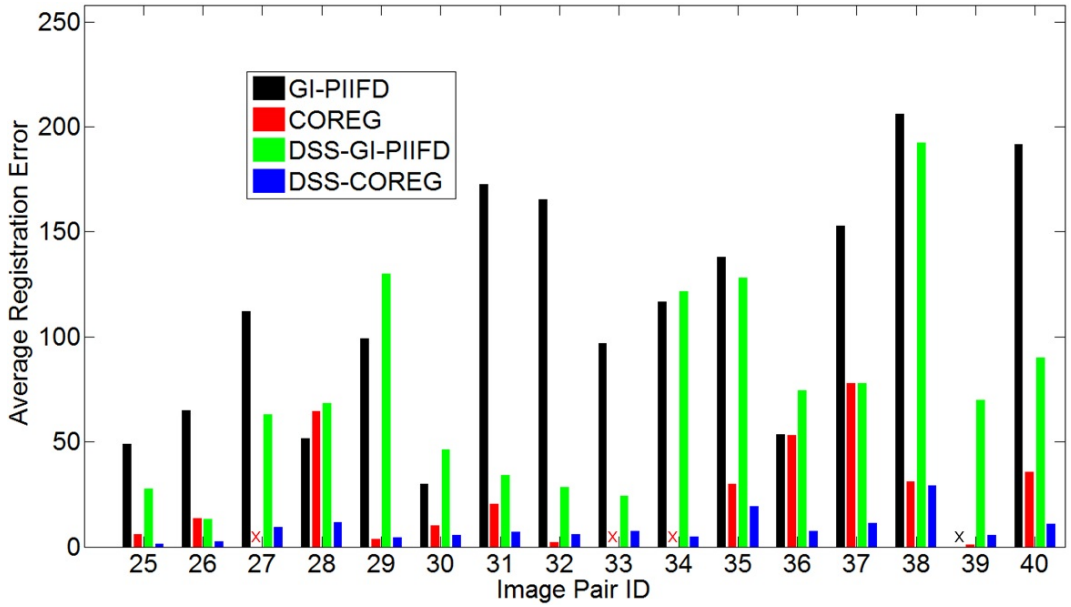
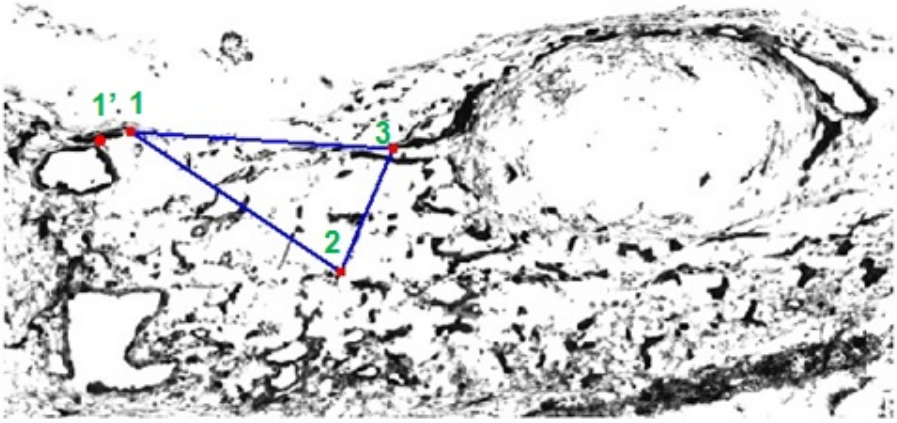


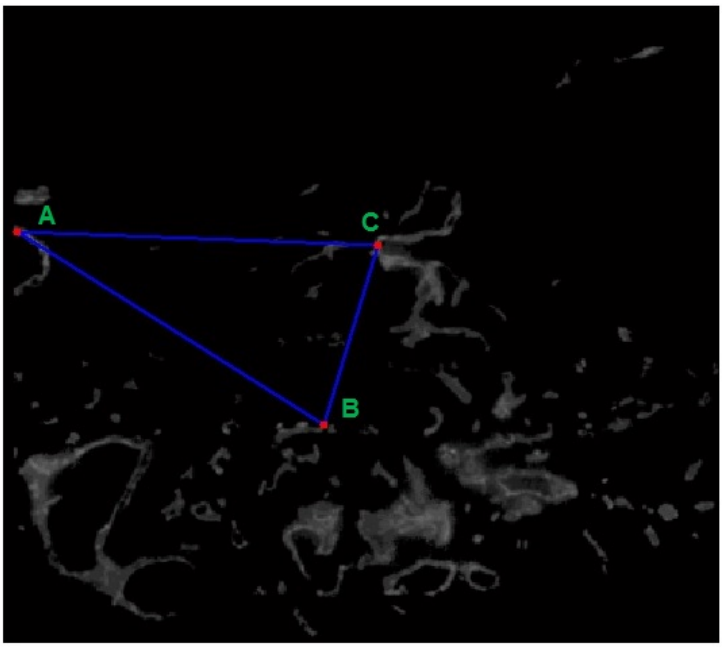
Figure 5.18: ARE Comparisons between COREG and GI-PIIFD for Microscopic Image Pairs (Part 3). The legend for the four techniques is the same as that in Figure 5.16. The scale difference in a pair of color and confocal images is 1X vs 4X.

to the other microscopic image pairs. For instance, DSS-COREG achieves an ARE of 14.25 in registering pair 38 at the 1X vs 1X scale difference, as shown in Figure 5.16(a). Figure 5.19 shows the pair of corner triplets for pair 38 which leads to the maximum NOP value. In the pair of corner triplets shown in Figure 5.19, corners 1, 2 and 3 in the color image correspond to corners A, B and C respectively. Corner pairs 2↦B and 3↦C are relatively accurate as compared to corner pair 1↦A. We have found that there is a corner in the color image, as marked with 1′ in Figure 5.19(a), which better match corner A in the confocal image. According to how COREG works, the pair of corner triplet (1, 2, 3)↦(A, B, C) leads to a higher NOP value as compared to (1′, 2, 3)↦(A, B, C). Based on our analysis, it is possible that the best pair of corner triplets is suppressed in registering an image pair where content differences are very large and edge structures are complex and chaotic.

Also, Table 5.5 clearly compares average ARE values for the four techniques at each of the five patterns of scale differences. We can see that DSS-COREG achieves an average ARE which is below 9.00 even at the 1X vs 4X scale difference. By comparing ARE values achieved by COREG and DSS-COREG, we can clearly conclude that both DSS and COREG contribute to the final registration performance.



(a) Corner Triplet (Color)



(b) Corner Triplet (Confocal)

Figure 5.19: Triplet Pair Determined by DSS-COREG in Registering Image Pair 38

Table 5.5: Average ARE of Each Pattern of Scale Difference for Microscopic Images

Scale Difference	GI-PIIFD	COREG	DSS-GI-PIIFD	DSS-COREG
1X vs 1X	29.84	15.50	14.02	4.70
1X vs 1.5X	52.93	18.38	21.87	7.57
1X vs 2X	72.45	30.38	30.73	7.49
1X vs 3X	88.25	16.22	44.90	7.78
1X vs 4X	114.49	28.94	74.28	8.91

5.6.3.3 Other Comparisons

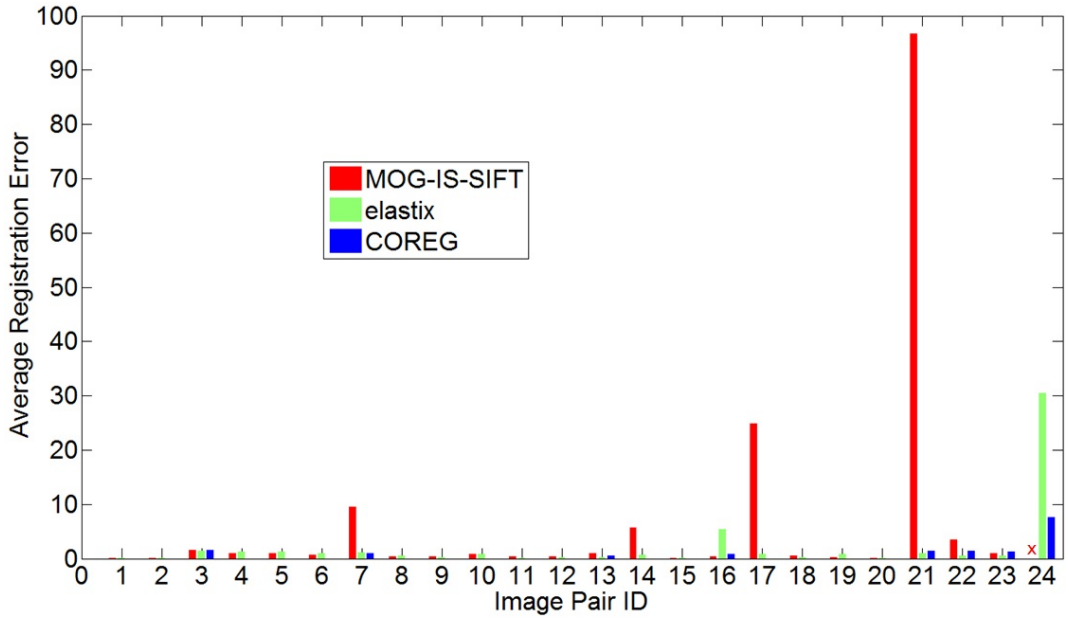


Figure 5.20: ARE Comparisons between MOG-IS-SIFT, elastix and COREG in Registering Non-microscopic Pairs

In our experiments, we have also compared the proposed COREG with MOG-IS-SIFT and elastix [45], where MOG-IS-SIFT has been proposed in Chapter 3 and elastix is a popular registration technique based on mutual information. The three techniques, i. e. MOG-IS-SIFT, elastix and COREG, are compared in terms of ARE in Figure 5.20 and Table 5.6 which are for non-microscopic and microscopic image pairs respectively. As shown in Figure 5.20 and Table 5.6, there are a few failures when MOG-IS-SIFT is used for registration. A failure in using MOG-IS-SIFT indicates that there is no keypoint match or that the number of keypoint matches is smaller than three which is insufficient for estimating a transformation in an image pair.

Figure 5.20 compares the three techniques in registering non-microscopic image pairs. The average ARE values for MOG-IS-SIFT, elastix and COREG are 6.51, 0.79 and 0.34 respectively. Table 5.6 presents the ARE values for the three techniques in registering microscopic image pairs. Note that, we have used microscopic images which have been processed by DSS for the three techniques MOG-IS-SIFT, elastix and COREG. As shown in Table 5.6, the techniques compared are accordingly called DSS-MOG-IS-SIFT, DSS-elastix and DSS-COREG. It is clear that DSS-MOG-IS-SIFT

performs very poorly and cannot achieve any effective registration. DSS-COREG outperforms DSS-elastix in registering 15 microscopic pairs out of all 16 ones. On average, ARE values are 19.76 and 4.70 for DSS-elastix and DSS-COREG respectively. Overall, DSS-COREG achieves a lot better performance than DSS-MOG-IS-SIFT and DSS-elastix. Note that we have only compared the three techniques at the 1X vs 1X scale difference, and the advantage of DSS-COREG over DSS-MOG-IS-SIFT and DSS-elastix is already very clear.

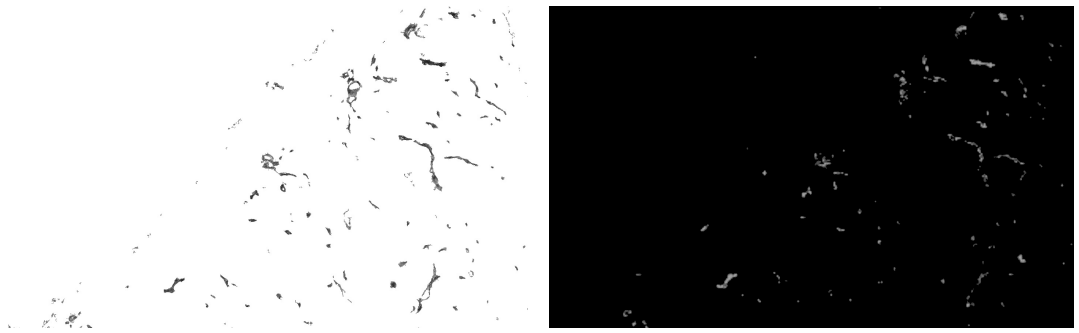
Table 5.6: ARE Comparisons between DSS-MOG-IS-SIFT, DSS-elastix and DSS-COREG for Microscopic Images

Pair ID	DSS-MOG-IS-SIFT	DSS-elastix	DSS-COREG
25	54.39	8.14	1.04
26	387.81	16.85	0.46
27	118.99	31.52	1.54
28	185.64	25.36	4.88
29	11.58	11.14	2.38
30	X	9.28	2.93
31	X	23.76	1.62
32	67.84	23.61	4.89
33	719.13	6.49	2.40
34	111.27	11.80	2.18
35	339.08	35.98	9.89
36	X	6.24	9.38
37	X	21.74	3.59
38	547.55	59.61	14.25
39	X	8.36	7.55
40	403.57	16.21	6.26

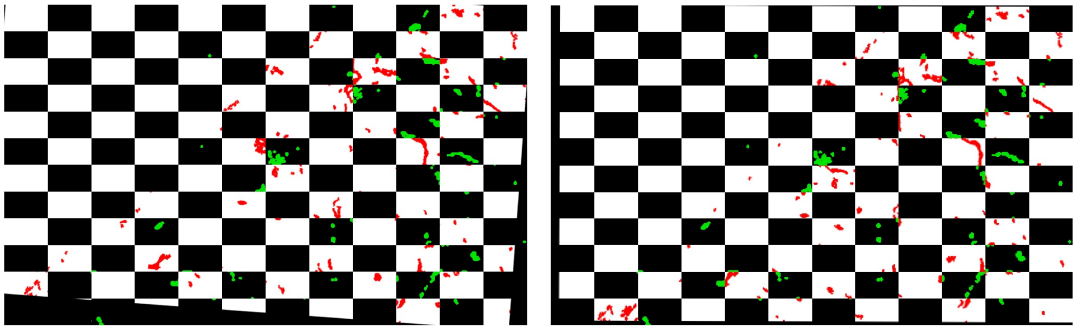
[a] X indicates a registration failure.
[b] The scale difference is 1X vs 1X in a pair of color and confocal images.

Overall, as Figure 5.20 and Table 5.6 indicate, MOG-IS-SIFT and elastix perform much more poorly than COREG, which can be explained as follows.

- MOG-IS-SIFT builds gradient-based descriptors. In registering multi-modal images with large content differences, MOG-IS-SIFT descriptors are not sufficiently discriminative, resulting in a poor registration performance.
- As an intensity-based image registration technique, elastix is sensitive to intensity variations across multi-modal images. More specifically, the similarity metric between the reference and target images is based on mutual information.



(a) Color Image (b) Confocal Image



(c) Alignment (DSS-GI-PIIFD) (d) Alignment (DSS-COREG)

(e) Part of Color Image (f) Part of Confocal Image

(g) Part of Alignment (DSS-GI-PIIFD) (h) Part of Alignment (DSS-COREG)

Figure 5.21: Alignment Images Using Checkerboard. (a) and (b) are the color and confocal images which have been processed by DSS.

The larger the content differences between images are, the less accurate the similarity metric is. Thus, the registration accuracy of DSS-elastix is low in registering multi-modal microscopic images.

5.6.3.4 An Alignment Example

In addition to evaluating the proposed COREG in terms of ARE, an alignment example is also given for a visual comparison, as shown in Figure 5.21. In this example, the alignments achieved by DSS-GI-PIIFD and DSS-COREG are compared using checkerboard images. To generate an aligned image, an estimated transformation is used to transform a color image onto its corresponding confocal image. The transformed color image and confocal image are displayed in an alternate way using the checkerboard format. To better identify alignments of image structures, the foregrounds of the color and confocal images are displayed using red and green colors respectively in the checkerboard image. In the example shown in Figure 5.21, the actual scale difference between the color and confocal images is 1:3.76. The ARE values achieved by DSS-GI-PIIFD and DSS-COREG are 121.61 and 4.87 respectively. To easily compare alignments achieved by DSS-GI-PIIFD and DSS-COREG, a small area of corresponding parts is extracted from the color and confocal images, as shown in Figure 5.21 (e) and (f). Clearly, Figure 5.21 (h) shows a much better alignment as compared to Figure 5.21 (g). Thus, DSS-COREG significantly improves the registration performance over DSS-GI-PIIFD.

5.6.4 Efficiency Comparison between GI-PIIFD and COREG

Although our focus is on improving the registration accuracy, we now give a rough efficiency comparison between GI-PIIFD and COREG as follows.

i. In registering image pairs with the same or similar scales, GI-PIIFD is a little faster than COREG.

   There are two main reasons why COREG is less efficient than GI-PIIFD. First, two rounds of matching corner triplets are needed in GOREG, while there is only one round in GI-PIIFD. Second, compared with GI-PIIFD, additional time is needed

in COREG for refining localization which has been discussed in Section 5.5.5. However, COREG is more efficient in building local descriptors than GI-PIIFD. The local descriptor in GI-PIIFD is 128-dimensional, whereas only the curvature and 16-dimensional DEPAC descriptor are used for describing corners in COREG.

ii. As the scale difference in an image pair increases, COREG achieves comparable or even higher efficiency than GI-PIIFD.

When the scale difference increases, the space of geometric transformations is larger and larger. Accordingly, more and more time is needed in comparing corner triplets. In COREG, the reference and target images have similar scales after applying the estimated scale difference. Thus, the second round of comparing corner triplets in COREG is much faster than the first round.

5.7 Summary

Following the work in Chapter 4 with regard to structural similarity in multi-modal microscopic images, this chapter has focused on content and scale differences in these images. In order to effectively register these images, we have presented COREG which is an image registration technique based on corners. Without the loss of generality, COREG is suited for registering all kinds of multi-modal images. To address content differences, we have explored curvatures of corners and have proposed a novel corner descriptor for feature representations. In addition, we have proposed a new way of estimating the scale difference between the reference and target images. The scale estimation is achieved with the assistance of a pair of corner triplets which leads to optimal transformation between the reference and target images. Experimental results have shown that our proposed COREG achieves greater robustness in both content difference and scale differences as compared to the latest existing technique [49].

Chapter 6

Conclusions and Future Work

Multi-modal image registration is of great importance in various applications, especially in medical image analysis. Due to substantial visual differences between corresponding parts across multi-modal images, it is challenging to achieve effective registration. Motivated by achieving effective registration of multi-modal microscopic images, the thesis has been dedicated to multi-modal image registration. The main contributions of the thesis are as follows.

i. We have analyzed the utilization of two types of gradient information, i.e. gradient magnitudes (GM) and gradient occurrences (GO), in building and matching SIFT-like descriptors. After identifying the limitations of only utilizing either of the two types of gradient information, we have proposed a technique called MOG (Magnitudes and Occurrences of Gradient) to take into consideration both GM and GO. MOG increases the discrimination of SIFT-like descriptors, thereby achieving more effective registration.

ii. The structural similarity in multi-modal microscopic images is very low, which hinders existing registration techniques from achieving a satisfactory performance. To detect the intrinsic structural similarity in these images, DSS (Detector of Structural Similarity) has been proposed. After performing DSS on these images, the structural similarity has been significantly increased. To evaluate DSS, we have used existing multi-modal image registration techniques on original microscopic images and the images after applying DSS. Experimental results show that DSS has substantially increased the registration performance.

iii. After applying DSS on multi-modal microscopic images, there are still large content differences. The latest multi-modal image registration technique called

GI-PIIFD cannot effectively register these images, and performs increasingly worse as the scale difference in an image pair increases. To achieve greater robustness to both content differences and scale differences, we have proposed COREG (COrner based REGistration). Without loss of generality, COREG is applicable to registering various kinds of multi-modal images. Experimental results show that COREG is more robust than GI-PIIFD in both content differences and scale differences.

Based on the research in the thesis, we suggest potential future work as follows.

i. The proposed MOG can be further improved by ranking the distance ratio between the nearest neighbor and the second nearest neighbor in matching descriptors. A lower distance ratio in a keypoint match indicates that the match is more discriminative. The highly-ranked keypoint matches in terms of the distance ratio are likely to improve the final registration accuracy.

ii. MOG can be generalized and applied to various applications in which SIFT or one of its variants is used. MOG has been proposed based on the original SIFT in registering mono-modal images. Nonetheless, we believe that if MOG is built on a variant of SIFT, such as PCA-SIFT [43] and GLOH [66], the effectiveness of descriptors will be most likely to be increased.

iii. The self-similarity concept [32, 33, 93] can be incorporated into our proposed COREG. Specifically, we will take into account the relationship or similarity between corner descriptors in representing corner triplets. By doing so, both geometric relationship and descriptor relationship between corners are used, thereby enhancing the possibility of finding truly-matched corner triplets. Based on COREG, the registration performance would potentially be further increased.

iv. We will explore the possibility of making COREG robust to deformable transformations [34, 99].

Publications during My PhD Period

1. G. Lv, S. W. Teng and G. Lu, "A Novel Multi-modal Image Registration Method based on Corners", *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2014.

2. G. Lv, S. W. Teng, G. Lu and M. Lackmann, "Detection of Structural Similarity for Multimodal Microscopic Image Registration", in *Proc. International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2013.

3. G. Lv, S. W. Teng, G. Lu and M. Lackmann, "Maximizing structural similarity in multimodal biomedical microscopic images for effective registration", in *Proc. International Conference on Multimedia and Expo (ICME)*, 2013.

4. G. Lv, M. T. Hossain, S. W. Teng, G. Lu and M. Lackmann, "Improving SIFT's performance by incorporating appropriate gradient information", in *Proc. International Conference on Image and Vision Computing New Zealand (IVCNZ)*, 2011.

5. M. T. Hossain, G. Lv, S. W. Teng, G. Lu and M. Lackmann, "Improved Symmetric-SIFT for Multi-modal Image Registration", in *Proc. International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2011.

Bibliography

[1] *Image Processing, IDL Version 7.1, May 2009*. 90

[2] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 510–517, 2012. 31

[3] M. Awrangjeb and G. Lu. An improved cuvature scale-space corner detector and a robust corner matching technique for transformed image identification. *IEEE Transactions on Image Processing*, 17(12):2425–2441, 2008. 16, 124

[4] M. Awrangjeb and G. Lu. Robust image corner detection based on the chord-to-point distance accumulation technique. *IEEE Transactions on Multimedia*, 10(6):1059–1072, 2008. 16, 20, 21, 22, 119, 124

[5] M. Awrangjeb, G. Lu, and C. S. Fraser. Performance comparisons of contour-based corner detectors. *IEEE Transactions on Image Processing*, 21(9):4167–4179, 2012. 16

[6] M. Awrangjeb, G. Lu, C. S. Fraser, and M. Ravanbakhsh. A fast corner detector based on the chord-to-point distance accumulation technique. In *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 519–525, 2009. xvii, 15, 16, 20, 22, 111, 118, 119, 120, 124, 128

[7] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981. 47

[8] I. N. Bankman. *Handbook of Medical Imaging*, Academic Press, 2000. 2

[9] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. 17, 26, 27, 31

[10] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, pages I–404–417, 2006. 17, 26, 27, 31

[11] J. Beis and D. G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1000–1006, 1997. 38

[12] O. Bretscher. *Linear Algebar with Applications*, 2005. 45

[13] L. G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992. 1

[14] A. Buades, B. coll, and J. M. Morel. A non-local algorithm for image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages II–60–65, 2005. 39

[15] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *European Conference on Computer Vision (ECCV)*, pages 778–792, 2010. 31

[16] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986. 15, 20, 128

[17] J. Chen and J. Tian et al. A partial intensity invariant feature descriptor for multimodal retinal image registration. *IEEE Transactions on Biomedical Engineering*, 37(7):1707–1718, 2010. xvii, 29, 37, 42, 49, 80, 100, 111, 112, 116

[18] J. Chen and S. Shan et al. Wld: A robust local image descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1705–1720, 2010. 29

[19] J. Chen and J. Tian. Real-time multi-modal rigid registration based on a novel symmetric-sift descriptor. *Progress in Natural Science*, 19:643–651, 2009. 33, 36, 50, 51, 52, 80, 85, 115

[20] M. Deshmukh and U. Bhosle. A survey of image registration. *International Journal of Image Processing*, 5(3):245–269, 2011. 8

[21] T. Dickscheid, F. Schindler, and W. Förstner. Coding images with local features. *International Journal of Computer Vision*, 94:154–174, 2011. 17

[22] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972. 46

[23] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 25, 43, 47, 72

[24] W. Forstner. A framework for low level feature extraction. In *Europen Conference on Computer Vision (ECCV)*, pages II–383–394, 1994. 15

[25] S. Gauglitz, T. Hollerer, and M. Turk. Evaluation of interest point detectors and feature descriptors for visual tracking. *International Journal of Computer Vision*, 94:335–360, 2011. 15

[26] A. Hafiane and B. Zavidovique. Local relational string and mutual matching for imge retrieval. *Information Processing and Management*, 44(3):1201–1213, 2008. 30

[27] J. H. Han and T. T. Poston. Chord-to-point distance accumulation and planar curvature: a new approach to discrete curvature. *Pattern Recognition Letter*, 22. 20, 21, 22, 119

[28] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988. 15, 37, 111, 118

[29] D. He and N. Cercone. Local triplet pattern for content-based image retrieval. In *International Conference on Information Analysis and Recognition*, pages 229–238, 2009. 30

[30] X. C. He and N. H. C. Yung. Corner detector based on global and local curvature properties. *Optical Engineering*, 47(5):1–12, 2008. 16, 22, 124, 126

[31] M. Heikkilä, M. Pietikäinen, and C. Schmid. Description of interest regions with local binary patterns. *Pattern Recognition*, 42(3):425–436, 2009. 30

[32] M. P. Heinrich and M. Jenkinson et al. Non-local shape descriptor: A new similarity metric for deformable multi-modal registration. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 541–548, 2011. 2, 40, 80, 147

[33] M. P. Heinrich and M. Jenkinson et al. Mind: Modality indepdent neighbourhood descriptor for multi-modal deformable registration. *Medical Image Analysis*, 16(7):1423–1435, 2012. 41, 42, 80, 147

[34] M. Holden. A review of geometric transformations for nonrigid body registration. *IEEE Transactions on Medical Imaging*, 27(1):111–128, 2008. 8, 9, 10, 147

[35] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17. 27

[36] J. P. Hornak. *The Basics of MRI*. Available at: http://www.cis.rit.edu/htbooks/mri/. 55, 64

[37] M. T. Hossain. *An Effective Technique for Multi-modal Image Registration*, 2012. PhD Thesis, Monash University. 14, 29, 30, 33, 36, 38, 47

[38] M. T. Hossain, G. Lv, S. W. Teng, G. Lu, and M. Lackmann. Improved symmetric-sift for multi-modal image registration. In *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 197–202, 2011. 33, 36, 38, 50, 80

[39] M. T. Hossain, S. W. Teng, G. Lu, and M. Lackmann. An enhancement to sift-based techniques for image registration. In *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 166–171, 2010. 36, 50, 52, 53, 80

[40] P. Hough. Machine analysis of bubble chamber pictures. In *International Conference on High Energy Accelerators and Instrumentation*, pages 554–558, 1959. 46

[41] T. Kadir and M. Brady. Saliency, scale and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001. 16, 17

[42] K. R. Katikireddy and F. O'Sullivan. Immunohistochemical and immunofluorescence procedures for protein analysis. *Methods in Molecular Biology*, 784. 83

[43] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages I–506–513, 2004. 25, 27, 79, 147

[44] A. Kelman, M. Sofka, and C. Stewart. Keypoints descriptors for matching across multiple image modalities and non-linear intensity variations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–7, 2007. 32

[45] S. Klein and M. Staring et al. elastix: A toolbox for intensity-based medical image registration. *IEEE Transactions on Medical Imaging*, 29(1):196–205, 2010. 10, 12, 13, 14, 111, 129, 141

[46] W. Kosiński, P. Michalak, and P. Gut. Robust image registration based on mutual information measure. *Journal of Signal and Information Processing*, 3(2):175–178, 2012. 2

[47] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *International Conference on Computer Vision (ICCV)*, pages 2548–2555, 2011. 31, 32

[48] J. P. Lewis. Fast normalized cross-correlation. *Vision Interface*, 10(1):120–123, 1995. 41

[49] Y. Li and R. Stevenson. Incorporating global information in feature-based multimodal image registration. *Journal of Electronic Imaging*, 23(2):023013–1–14, 2014. 5, 8, 12, 13, 14, 42, 43, 44, 49, 111, 116, 119, 120, 122, 145

[50] T. Lindeberg. *Scale-Space Theory in Computer Vision, Springer, 1994.* 115

[51] T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision*, 11(3):283–318, 1993. 16, 17

[52] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):978–994, 2011. 27

[53] J. Liu, G. Zeng, and J. Fan. Fast local self-similarity for describing interest regions. *Pattern Recognition Letters*, 33(9):1224–1235, 2012. 40

[54] K. Lkeuchi. *Computer Vision: A Reference Guide, Springer, 2014*. 115

[55] D. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1150–1157, 1999. 19

[56] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 4, 17, 19, 23, 27, 28, 37, 42, 45, 47, 50, 51, 80, 115

[57] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Conference on Artificial Intelligence*, pages 674–679, 1981. 27

[58] R. Ma, J. Chen, and Z. Su. Mi-sift: Mirror and inversion invariant generalization for sift descriptor. In *International Conference on Image and Video Retrieval*, pages 228–235, 2010. 35

[59] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967. 99

[60] J. B. A. Maintz and M. A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998. 2, 7, 9

[61] S. Maji and J. Malik. Object detection using a max-margin hough transform. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1038–1045, 2009. 47

[62] 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. 17

[63] K. Mikolajczyk and T. Tuytelaars et al. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2):43–72, 2005. 18

[64] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *International Conference on Computer Vision (ICCV)*, pages I–525–531, 2001. 15

[65] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004. 16, 17

[66] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. 11, 26, 27, 44, 50, 59, 79, 80, 147

[67] F. Mokhtarian and R. Suomela. Robust image corner detection through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1376–1381, 1998. 16, 22

[68] J. Morel and G. Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM Journal of Imaging Science*, 2(2):438–469, 2009. 27, 50, 80

[69] G. Mori, X. Ren, A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages II–326–333, 2004. 17

[70] E. Mortensen, H. Deng, and L. Shapiro. A sift descriptor with global context. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages I–184–190, 2005. 26, 50, 80

[71] A. Myronenko and X. Song. Intensity-based image registration by minimizing residual complexity. *IEEE Transactions on Medical Imaging*, 29(11):1882–1891, 2010. 10

[72] D. D. Nigris, D. L. Collins, and T. Arbel. Multi-modal image registration based on gradient orientations of minimal uncertainty. *IEEE Transactions on Medical Imaging*, 31(12):2343–2354, 2012. 10

[73] D. D. Nigris and L. Mercier et al. Hierarchical multimodal image registration based on adaptive local mutual information. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages II–643–651, 2010. 10

[74] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002. 29

[75] G. Olague and B. Hernandez. A new accurate and flexible model based multi-corner detector for measurement and recognition. *Pattern Recognition Letters*, 26(1):27–41, 2005. 16

[76] S. Oldridge, G. Miller, and S. Fels. Mapping the problem space of image registration. In *Canadian Conference on Computer and Robot Vision*, pages 309–315, 2011. 1

[77] F. P. Oliveira and J. M. Tavares. Medical image registration: a review. *Computer methods in biomechanics and biomedical engineering*, 17(2):73–93, 2014. 7, 9

[78] J. Orchard. Efficient least squares multimodal registration with a globally exhaustive alignment search. *IEEE Transactions on Image Processing*, 16(10):2526–2534, 2007. 10

[79] N. Otsu. A threshold selection method from gray-level histogram. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-9(1):62–66, 1979. 90

[80] Visual Geometry Group from Department of Engineering Science Oxford University. *Affine Covariant Regions Datasets*. Available at: http://www.robots.ox.ac.uk/~vgg/data/data-aff.html. xix, 30, 31, 32, 40, 59, 65, 66

[81] S. W. Paddock. *Confocal Microscopy*, Oxford University Press, 2001. 81, 85

[82] J. B. Pawley. *Handbook of Biological Confocal Microscopy*, Plenum Press, 1995. 85

[83] G. V. Pedrosa and C. A. Z. Barcelos. Anisotropic diffusion for effective shape corner point detection. *Pattern Recognition Letter*, 31(12):1658–1664, 2010. 16

[84] R. Raguram, J. M. Frahm, and M. Pollefeys. A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. In *European Conference on Computer Vision (ECCV)*, pages II–500–513, 2008. 48

[85] J. A. Ramos-Vara. Technical aspects of immunohistochemistry. *Veterinary Patholology*, 42(4):405–426, 2005. 83

[86] X. Ren and J. Malik. Learning a classification model for segmentation. In *International Conference on Computer Vision (ICCV)*, pages I–10–17, 2003. 17

[87] A. Rosenfeld and E. Johnston. Angle detection on digital curves. *IEEE Transactions on Computers*, C-22(9):875–878, 1973. 16

[88] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision (ECCV)*, pages 430–443, 2006. 16

[89] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):105–119, 2010. 16, 31

[90] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: an efficient alternative to sift and surf. In *International Conference on Computer Vision (ICCV)*, pages 2564–2571, 2011. 31

[91] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000. 16, 17

[92] K. Reddy Sekhar and M. Mahesh. Performance evaluation of corner detectors: A survey. *International Journal of Computer Science and Mobile Computing*, 2(10):226–233, 2013. 15

[93] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. 39, 40, 147

[94] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 18

[95] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, 1994. 15

[96] E. D. Sinzinger. A model-based approach to junction detection using radial energy. *Pattern Recognition*, 41(2):494–505, 2008. 16

[97] S. M. Smith and J. M. Brady. Suan - a new apprach to low level image processing. *International Journal of Computer Vision*, 23(34):45–78, 1997. 15

[98] Z. Song, S. Zhou, and J. Guan. A novel image registration algorithm for remote sensing under affine transformation. *IEEE Transactions on Geoscience and Remote Sensing*, 52(8):4895–4912, 2014. 8

[99] A. Sotiras, C. Davatzikos, and N. Paragios. Deformable meical image registration: A survey. *IEEE Transactions on Medical Imaging*, 32(7):1153–1190, 2013. 1, 9, 10, 147

[100] M. Staring and U. A. van der Heide et al. Registration of cervical mri using multifeature mutual information. *IEEE Transactions on Medical Imaging*, 28(9):1412–1421, 2009. 10

[101] R. Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, 2(1):1–104, 2006. 9

[102] D. M. Tsai, H. T. Hou, and H. J. Su. Boundary-based corner detection using eigenvalues of covariance matrices. *Pattern Recognition Letter*, 20(1):31–40, 1999. 16

[103] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *British Machine Vision Conference (BMCV)*, pages 412–425, 2000. 17, 18

[104] T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004. 17, 18

[105] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2008. 15, 18

[106] C. Wachinger and N. Navab. Entropy and laplacian images: Structural representations for multi-modal registration. *Medical Image Analysis*, 16(1):1–17, 2012. 2, 41, 80

[107] M. Wainwright. *Photosensitisers in Biomedicine*, March, 2009. 85

[108] E. Weisstein. *Shear*. From MathWorld–A Wolfram Web Resource: http://mathworld.wolfram.com/Shear.html. 8

[109] M. Xia and B. Liu. Image registration by 'super-curves'. *IEEE Transactions on Image Processing*, 13(5):720–732, 2004. 129

[110] Z. Xiong and Y. Zhang. A critical reviw of image registration methods. *International Journal of Image and Data Fusion*, 1(2):137–158, 2010. 7, 8

[111] G. Yang, C. Stewart, M. Sofka, and C. Tsai. Registration of challenging image pairs: Initialization, estimation, and decision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1973–1989, 2007. 14, 28, 61, 65

[112] G. Yu and J. Morel. A fully affine invariant image comparison method. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1597–1600, 2009. 27

[113] S. Zhang and Q. Tian et al. Edge-sift: Discriminative binary descriptor for scalable partial-duplicate mobile search. *IEEE Transactions on Image Processing*, 22(7):2889–2902, 2013. 27

[114] X. Zhang and H. Wang et al. Robust image corner detection based on scale evolution difference of planar curves. *Pattern Recognition Letter*, 30(1):449–455, 2009. 16

[115] X. Zhang and H. Wang et al. Corner detection based on gradient correlation matrices of planar curves. *Pattern Recognition*, 43(4):1207–1223, 2010. 16

[116] X. Zhang and M. Lei et al. Multi-scale curvature product for robust image corner detection in curvature scale space. *Pattern Recognition Letter*, 28(1):545–554, 2007. 16

[117] D. Zhao, Z. Lin, and X. Tang. Laplacian PCA and its applications. In *International Conference on Computer Vision (ICCV)*, pages 1–8, 2007. 25

[118] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003. 1