High Performance Image Analysis for Large Histological Datasets

DISSERTATION

Presented in Partial Fulfillment of the Requirements for

the Degree Doctor of Philosophy in the

Graduate School of The Ohio State University

By

Lee Cooper, M.S., B.S.

Graduate Program in Electrical and Computer Engineering

The Ohio State University

2009

Dissertation Committee:

Bradley Clymer, Co-Adviser

Kun Huang, Co-Adviser

Ashok Krishnamurthy

ABSTRACT

The convergence of emerging challenges in biological research and developments in imaging and computing technologies suggests that image analysis will play an important role in providing a better understanding of biological phenomenon. The ability of imaging to localize molecular information is a key capability in the post-genomic era and will be critical in discovering the roles of genes and the relationships that connect them. The scale of the data in these emerging challenges is daunting; high throughput microscopy can generate hundreds of gigabytes to terabytes of high-resolution imagery even for studies limited in scope to a single gene or interaction. In addition to the scale of the data, the analysis of microscopic image content presents significant problems for the state-of-the-art in image analysis.

This dissertation addresses two significant problems in the analysis of large histological images: reconstruction and tissue segmentation. The proposed methods form a framework that is intended to provide researchers with tools to explore and quantitatively analyze large image datasets.

The works on reconstruction address several problems in the reconstruction of tissue from sequences of serial sections using image registration. A scalable algorithm for non-rigid registration is presented that features a novel method for the matching small nondescript anatomical features using geometric reasoning. Methods for the nonrigid registration of images with different stains are presented for two application scenarios. Correlation

sharpness is proposed as a new measure for image similarity, and is used to map tumor suppressor gene expression to structure in mouse mammary tissues. An extended process of geometric reasoning based on the matching of cliques of anatomical features is presented and demonstrated for the nonrigid registration of immunohistochemical stain to hemotoxylin and eosin stain for human cancer images. Finally, a method for the incorporation of structural constraints into the reconstruction process is proposed and demonstrated on the reconstruction of ducts in mammary tissues.

The work on tissue segmentation focuses on the use of statistical geometrical methods to describe the spatial distributions of biologically meaningful elements such as nuclei in tissue. The two point correlation function is demonstrated to be an effective feature for the segmentation of tissues, and is shown to possess a peculiar low-dimensional distribution in feature space that permits unsupervised segmentation by robust methods. The relationship between two-point functions for proximal image regions is derived and used to accelerate computation, resulting in a 7-68x improvement over a naive FFT-based implementation.

In addition to the methods proposed for reconstruction and segmentation, a significant portion of this dissertation is devoted to applying high performance computing to enable the analysis of large datasets. In particular, multi-node parallelization as well as multi-core and general purpose computing on graphics processing are used to form a heterogeneous multiprocessor platform that is used to demonstrate the segmentation and reconstruction methods on images up to 62K $\times$ 23K in size.

To my parents Donald and Mary Mae Cooper for their love and support

ACKNOWLEDGMENTS

I have enjoyed my years at Ohio State, and have encountered many people along the way who have shaped my OSU experience and made this journey possible.

First and foremost I extend my thanks to my advisor Professor Kun Huang for his support and patience and also his insight into technical matters and the workings of academia. I would also like to thank Professor Bradley Clymer for his advice and invaluable help in navigating the complex OSU system. Thanks is also due to Professor Ashok Krishnamurthy and the staff at the Ohio Supercomputing Center (especially the help desk) for providing the computing infrastructure that made this research possible. There are a number of professors not on my committee who I would like to thank for guiding me. Professor Betty Lise Anderson for her career advice. Professors Manuel Ujaldón and Ümit Catalyürek for their computing expertise. A good part of the math department, too many to list, who challenged my thinking with their assignments.

Next I would like to thank my fellow students to thank for their technical discussions, advice, and commiseration. The guys from Biomedical Informatics, Olcay Sertel, Jeff Prescott, Selnur Erdal, and Jun Kong. Antonio Ruiz from Málaga, who managed to wade through my registration code. Special thanks to the IBGP students Leszek Rybaczyk and Kristin Keen-Circle from Kun's lab for their biology lessons.

This acknowledgment would not be complete without thanking my family for their love and support. My parents Donald and Mary Mae Cooper have been patient and understanding throughout the process of my education. In this difficult year the courage that my Mother showed was inspirational to me. I miss you and you will be in our thoughts on graduation day. I also want to give thanks to my big brother Steve Cooper who helped me keep my car on the road all these years. I have enjoyed all the good times in Detroit (many more to come!) and appreciate your sage advice.

VITA

March, 1978 ..... Born - Columbus, Ohio, USA

2003 ..... B.S. Electrical and Computer Engineering, Ohio State University

2005 ..... M.S. Electrical and Computer Engineering, Ohio State University

2005-present ..... Ph.D. Electrical and Computer Engineering, Ohio State University

FIELDS OF STUDY

Major Field: Electrical and Computer Engineering

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

Imaging will play a central role in addressing the emergent grand challenges of biology. With the human genome sequenced the post-genomic era has arrived, and the focus shifts to understanding the roles of genes and discovering the structures of the molecular networks that they regulate. Technologies such as microarray and ChIP-chip provide the opportunity to peer into this hidden world, however imaging is unique for its ability to localize molecular and genomic information. The need for locality is critical since tissues and even individual cells of the same type can be heterogeneous in the genetic sense. A realistic picture of a phenomenon such as cancer requires more than just observations of molecular behavior averaged over entire tissues, information with resolution at the scale of individual cells and beyond is required to understand intracellular regulation as well as the role of intercellular interactions.

The scale of the emerging problems in bioimaging is daunting. High throughput microscopy techniques enable scientists to generate hundreds of gigabytes or terabytes of high-resolution imagery even for an individual study that is limited in scope to a single gene or interaction. The manual analysis of these quantities of visual information is often beyond the capability of determined individuals. Additionally there are issues with regards

to inter and intra-reader variability. Dividing visual analysis tasks among multiple individuals is prone to introduce biases in the analysis outcome. Likewise the analysis of a single individual can vary significantly, especially when that individual is fatigued or overwhelmed by massive quantities of data. A more quantitative approach to image analysis is needed to address the new needs of bioimaging.

At the same time these grand challenges are emerging in biology, the state-of-the-art of automated image analysis is maturing. A wide variety of algorithms are available for commonly conceived problems such as tracking objects in a video sequence, segmentation of images into relevant regions, and the classification of image content. Microscopic images of tissues present unique challenges for the cast of image analysis algorithms. In the imaging sense their content is noisy, being characterized by many repeated and indistinguishable structures such as cells that produce an overall textural appearance. Performing segmentation or registration (matching) in this environment is typically difficult since many popular algorithms are not intended for the special case of microscopic imaging. Nevertheless successful automated image analysis has been demonstrated for many common problems in microscopic imaging.

In addition to the challenges posed by the content of microscopic images, their large size presents a significant challenge in terms of computation. High resolution images typically contain hundreds of millions or billions of pixels, resulting in individual color images that are several gigabytes each. A single study containing hundreds of images can easily push the scale of data into the terabyte range. Traditionally image analysis algorithms are not intended to address data on this scale, so a new collection of efficient algorithms is required. These new algorithms must strike a difficult compromise, being computationally feasible but also sufficiently complex to address the challenges of image content.

Hardware acceleration also has a role to play in making image analysis computationally tractable. Large parallel systems consisting of linked computing nodes offer a solution for analyses that can be parallelized. More recently, emerging architectures such as multicore processors and general purpose computing on graphics processors (GPGPU) provide solutions for the desktop end user who does not have access to the resources of a computing cluster.

The convergence of the challenges and available technology in biology, image analysis, and computing suggests that the time is ripe to develop systems for the quantitative analysis of bioimages. This convergence prompted me to develop the work described in this dissertation *High Performance Image Analysis for Large Histological Datasets*. In this work I have addressed two key problems in bioimage analysis: three dimensional reconstruction and tissue segmentation.

1.1 Problem Statement

One of the key problems in bioimage analysis is the acquisition of meaningful three dimensional (3D) information. Biological interactions unfold in the three dimensional space of tissues, and the analysis of individual two-dimensional images neglects off-plane information. Confocal and multi-photon fluorescence microscopes provide three dimensional image data but have limited penetration depths, far less than what is necessary to image large samples [1, 2]. Additionally, the staining techniques required for fluorescence imaging are difficult to administer. The use of immunofluorescent compounds, consisting of antibody-fluorophore complexes that bind to molecules of interest, require the introduction of antibodies into living tissue [3, 4]. An alternative approach allows fluorescent proteins such as GFP to be expressed natively in tissues, but requires the production of transgenic

animals and is still depth-limited [5–7]. Recently, fluorescent proteins have been developed that excite in the infra-red range, offering increased tissue penetration, however this technique is still new and it is not clear yet what depth limitations are for imaging at microscopic scale [8].

Another approach to garnering 3D information is the reconstruction of tissue from sequences of serial section images using *image registration* [9–14]. In this approach a specimen is stained and embedded with a material such as wax for rigidity. The prepared tissue is then sliced on a microtome and mounted to produce a sequence of slides that are digitized into a corresponding sequence of images. The images are then aligned (registered) to generate a volumetric dataset of the original tissue. This technique enables whole-tissue reconstructions of the tissue without depth limitations, however the process of reconstruction introduces several nontrivial challenges. The first challenge comes in accounting for the nonrigid distortions introduced by the sectioning process. As the tissue is sliced and mounted physical forces introduce relative distortions between slices. Due to the fragility of these slices the distortions are typically "nonrigid" or nonlinear in nature. When the sequence is aligned for reconstruction these nonrigid distortions must be corrected to recover a faithful representation of the original tissue. The second challenge is how to establish spatial correspondences between adjacent section images, given that matching the textural content of microscopic images is error prone, and that some natural difference in appearance is expected. The third and final challenge is how to address the computational aspects of the problem, given that section-scan images typically range into the tens-of-thousands of pixels in each dimension.

Another key problem in bioimage analysis is the identification of tissue boundaries. In order to calculate tissue volume fraction or investigate tissue layer morphologies the

boundaries of tissues must first be identified. This can be a tedious manual task since the differences between tissue may be subtle, or the boundaries may be relatively complex and difficult to trace. The image analysis approach to this problem is known as *image segmentation*, or in this particular instance *tissue segmentation* [15–21]. The automatic identification of tissue boundaries is also a challenging image analysis problem due to the peculiar content of microscopic images. The visual cues that distinguish one tissue from another include a broad range of criteria including color, shape, texture, and scale. The differences in these qualities for different tissues in the same sample may be relatively subtle. These points are illustrated in Figure 1.1, where the tissue boundary is shown for a small region of mouse placenta image. Developing a comprehensive segmentation algorithm that incorporates the multiple visual cue criteria and can distinguish subtle differences is a nontrivial task.

In the chapters that follow I present methods for the reconstruction and tissue segmentation problems for large microscopic images. The proposed algorithms fit into a framework that is intended to provide researchers in biology with the tools to explore and quantify large image datasets. The framework from the perspective of the biologist is presented in Figure 1.2. A genetic change is induced in an organism and the tissue of interest is harvested. The tissue is sectioned and the sections are mounted and digitized to produce a serial sequence of very large images. The sequence is used to reconstruct the tissue and segment the tissue layers to prepare for further analysis. The details of the proposed framework from an imaging perspective are described in Figure 1.3. The framework relies on a large number of image analysis components to produce the representations used for biological quantification and exploration.

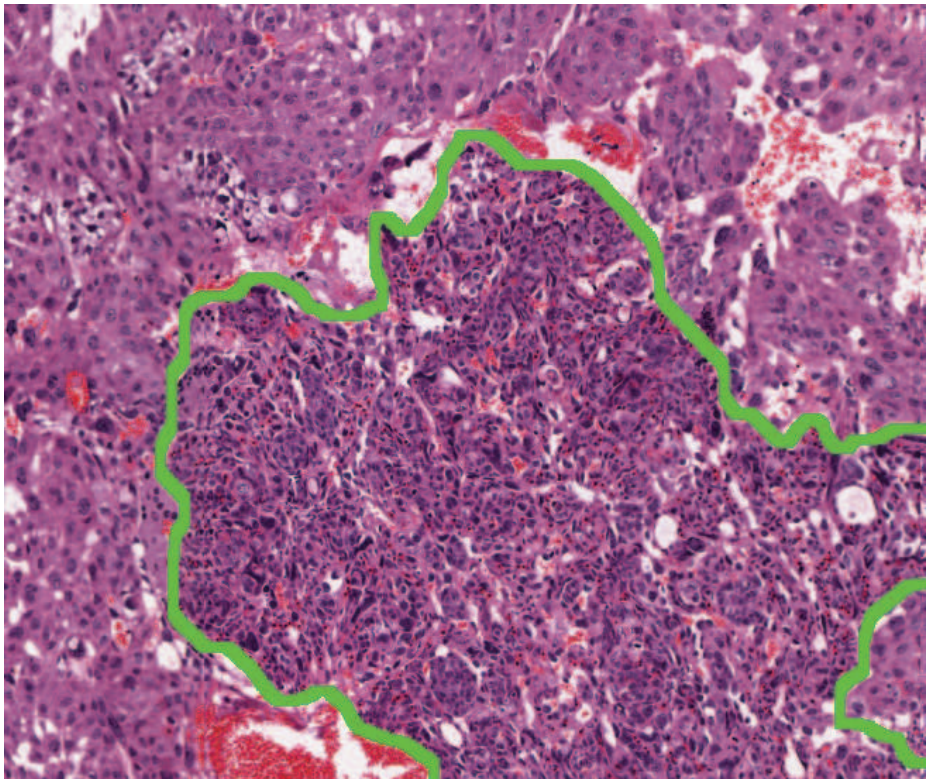


Figure 1.1: The visual cues that distinguish tissues include color, shape, texture, and scale. The green trace indicates the boundary between two different tissue layers with similar appearances.

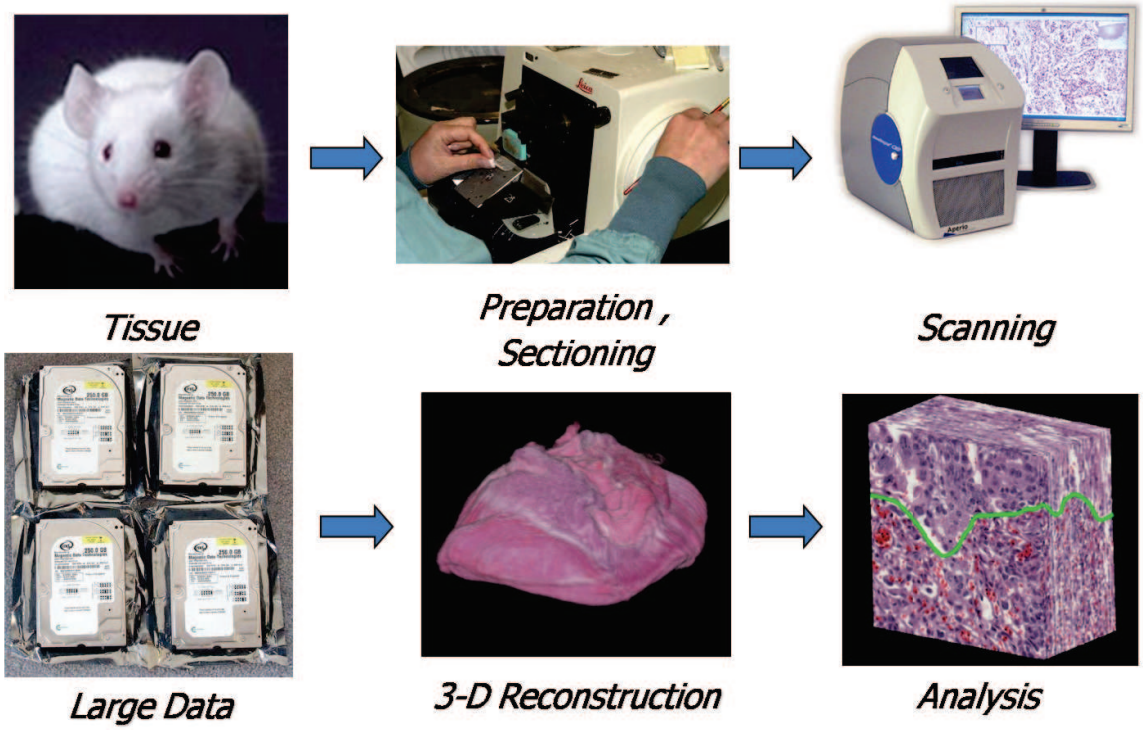


Figure 1.2: Imaging as a phenotyping tool for biologists. The proposed framework enables the exploration and analysis of sequences of large microscopic images.

1.2 Organization

This dissertation is organized into two parts: Chapters 2 through 7 address the problem of reconstructing tissues in 3D from serial image sequences. Chapters 8 and 9 address the problem of tissue segmentation.

Chapter 2 describes a scalable *two-stage algorithm* for the reconstruction of tissues from sequences of serial section images. A novel method based on the matching of representative microanatomical features is presented along with a precise and efficient refinement procedure for correcting nonrigid distortion. Chapter 3 describes the implementation of the two stage algorithm using general purpose computing on graphics processors (GPU).

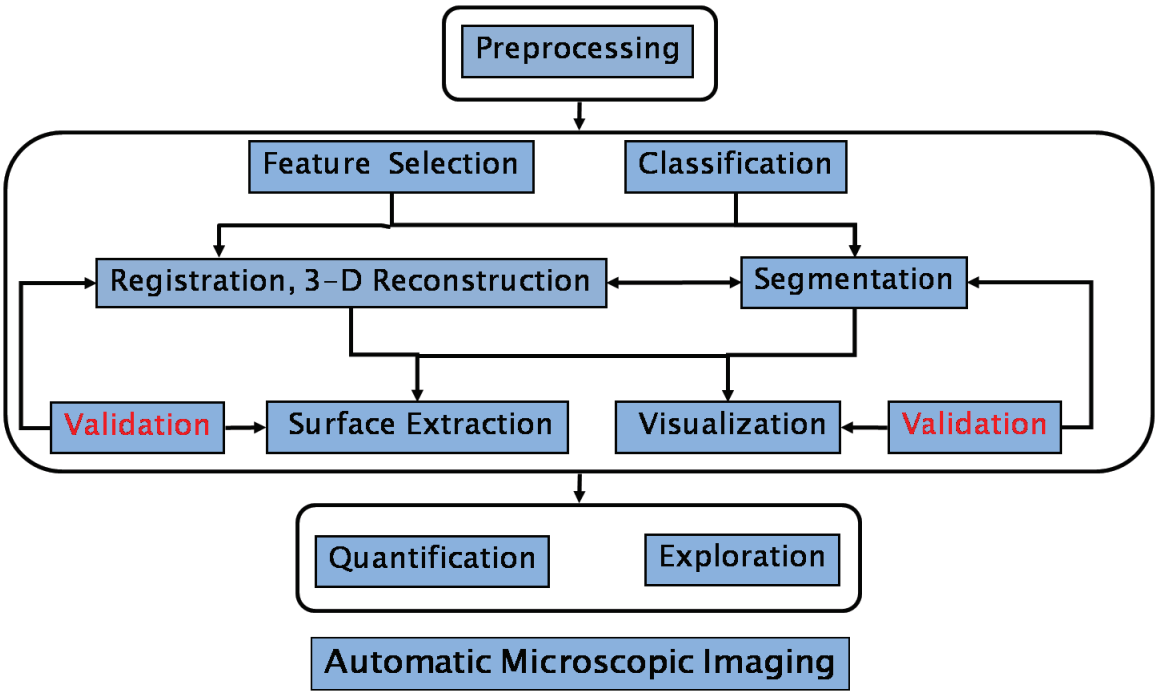


Figure 1.3: Image analysis components for the proposed framework. The framework depends on the interaction of a large number of components.

This implementation results in a 6.68x speedup using hardware that is commonly available on most desktop workstations. Chapter 4 extends these results to clusters of GPU-equipped computing nodes, producing a 49x speedup using 32 GPUs that is capable of registering 500 16K $\times$ 16K images in 3.7 hours. The problem of registering images of tissues with different stains is addressed in Chapter 5, where a novel metric of correlation sharpness is proposed for comparing intensity signals. Chapter 6 also addresses the problem of different stain registration, but in the scenario where intensity information is not sufficient for accurate matching. This chapter extends the work of the anatomical feature matching of the two stage algorithm to correct nonrigid distortion without intensity information. The final topic

on reconstruction is contained in Chapter 7 which proposes a method for the reconstruction of tissues under constraints on the structure of microanatomy.

The theoretical basis and procedure for the tissue segmentation method is described in Chapter 8. The *two-point correlation function* is described as a feature for the characterization of spatial distributions of cellular and subcellular components that distinguish tissues. Building on existing work, a deterministic method for two point function calculation is described and the two point features are demonstrated to possess peculiar low-dimensional distributions in feature space. Chapter 9 addresses the computational aspects of the proposed segmentation method. A theoretical shortcut based on the linearity of correlation is proposed for the calculation of two-point features that results in a 7-68x performance increase over a naive FFT-based method. A GPU implementation of this updating method nets another 11-16x for a total of 77-1088x improvement. This is extended to parallel computation on a cluster of computers for further gains.

CHAPTER 2

SCALABLE NONRIGID REGISTRATION FOR LARGE MICROSCOPIC IMAGES

In this chapter I present a novel scalable algorithm for the nonrigid registration of large microscopic images. The proposed algorithm address the shortcomings in the state-of-the-art for the registration of histological and microscopic images, specifically large image size, feature rich environment, and nonrigid distortion. The algorithm consists of two stages: initialization by rigid registration, and refinement by precise nonrigid registration. The initialization uses a novel approach that matches abundant anatomical features with rigid geometric constraints. The refinement uses normalized cross correlation to perform pixel-precision comparisons of local regions of intensity to establish more precise correspondences, calculated in the frequency domain using a high-performance FFT software library. The combination of initialization and refinement provides an approach to automatic sectioned image registration and reconstruction that is robust and easily parallelizeable.

The two-stage algorithm is demonstrated using a set of 100 16K $\times$ 16K microscopic images derived from a study on the role of the retinoblastoma gene [22]. Results show that the rigid initialization is comparable to a fully manual registration and that the nonrigid

refinement is capable of correcting the distortion encountered in serially sectioned microscopic images. The anatomical feature matching scheme used in rigid initialization is also demonstrated to be effective when tissue is partially occluded.

Execution times for a serial implementation are presented here along with a brief discussion on the computational aspects and parallelization of each stage. Chapters 3 and 4 treat these computational matters in further detail.

2.1 Introduction

An essential challenge for biologists in the post-genomic era is the understanding of gene functions and gene interactions. A critical element in this challenge is the ability to characterize the phenotypes associated with specific genotypes. Three dimensional (3D) morphologies of tissue structures at the cellular and sub-cellular scales are one aspect of phenotype that provides information key to the study of biological process such as the initiation of cancer in the tumor microenvironment, the development of organs, or the formation of neural networks in the brain. Nevertheless, existing techniques for obtaining high magnification 3D structures (e.g., confocal and multiphoton microscopy) from biomedical samples are rather limited. Therefore, a fundamental approach for 3D acquisition is to perform reconstruction by aligning multiple 2D images obtained from serial thin tissue sections via *image registration* [9–14, 22–37].

Image registration has been extensively studied in biomedical imaging, geological survey and computer vision [23, 24]. It can be formulated as an optimization problem of finding the optimal transformation T between two images I_1 and I_2 to maximize their similarity, ie,

$$T = \arg\max Similarity\left(I_1, T(I_2)\right). \tag{2.1}$$

Commonly used similarity/difference measures include mutual information (MI) [38], normalized cross correlation (NCC), and summed square difference (SSD) [23,24]. The transformation spaces include rigid transformation, which deals with only rotation and translation, and nonrigid transformation which compensates for scaling and deformations such as bending, stretching, shearing and warping [27, 39, 40]. In order to search for the optimal transformation, various searching procedures have been adopted such as Levenberg-Marquardt algorithm [41], EM algorithm [42], and geometric hashing [43]. Like any optimization process a good initialization is critical for a global optimum outcome. In many cases, a good rigid registration result serves as an ideal initialization for non-rigid registration [26]. For large images with conspicuous deformations, hierarchical multi-resolution registration methods have also been widely used in medical imaging applications [44, 45].

There are several major challenges for registering serial section microscopic images for 3D tissue reconstruction:

1. **Large image size.** High-resolution slide scanners are capable of generating images with resolutions of $0.23\mu m/pixel$ (with 40X objective lens), often producing images with hundreds of millions or even billions of pixels. The reconstruction of an individual tissue sample may involve hundreds of slides, and a full study may contains several samples with image data easily ranging in the Terabytes. While a multiscale approach can be applied to handle large images, it requires transformation of the free-floating image at each scale which is computationally nontrivial. For instance, this chapter deals with images sized $16K \times 16K$ pixels. With a scale factor of two, this implies transformation of an image containing around $8K \times 8K$ pixels prior to the final iteration at full resolution.

2. **Feature rich environment.** The textural quality of microscopic image content provides a unique challenge to the problems of feature selection and matching. Specifically, traditional feature detection schemes such as corner detection generate an overwhelming abundance of features that are similar in appearance making matching prone to error. In addition, at sub-micron resolutions a shift of one millimeter corresponds to thousands of pixels. The search for corresponding features is therefore computationally infeasible without a good initialization.

3. **Nonrigid distortion and local morphological differences.** The key challenge for image registration of serial section images is to compensate for distortion between consecutive images that is introduced by the sectioning process. Tissue sections are often extremely thin (3 to $5\mu m$) and delicate as a result. The preparation process (i.e., sectioning, staining, and mounting) can introduce a variety of nonrigid deformations including bending, shearing, stretching, and tearing. At micron resolutions, even minor deformations become conspicuous and may prove problematic when accuracy is critical to the end application. In order to compensate for such deformations, a *nonrigid registration* is essential and its success depends on establishing a large number of precise *feature correspondences* throughout the extent of the image. This precision requires comparison of intensity information and is very time consuming with popular similarity measures such as Mutual Information.

Motivated by several large-scale biomedical studies including developmental biology and breast cancer research, this chapter presents a scalable, efficient, and parallelizable image registration algorithm to address the above challenges. The *two-stage algorithm* consists of initialization by rigid registration followed with refinement by nonrigid registration. The *initialization stage* is a fast rigid registration algorithm based on the matching

of high-level features. This approach circumvents the issue of the presence of numerous and ambiguous local features, providing effective initialization for the next stage. In the *refinement stage*, nonrigid registration is achieved by precisely matching a large number of local intensity features using cross-correlation. This approach has the following advantages:

1. **Fast Rigid Registration for Initialization.** This algorithm uses conspicuous anatomical regions (e.g., blood vessels) as high-level features and the rigid transformation is derived using a voting scheme in the Euclidean transformation space. It is highly efficient and accurate for common histological images. In addition, it can accommodate arbitrary rotation and translations. This provides us with a good initialization for the nonrigid registration and the search space for point correspondence is significantly reduced.

2. **Feature Selection.** Point features for precise matching in nonrigid registration are selected based on neighborhood complexity rather than the presence of ambiguous content such as corners. This not only reduces computational burden but also allows the user to gain a more uniform distribution of features.

3. **Fast Normalized Cross-Correlation for Precise Matching.** Precise feature matching is based on the normalized cross-correlation (NCC) between local neighborhoods in each image. NCC calculation can be implemented efficiently using fast Fourier transform (FFT) resulting in a very fast execution as compared to measures like mutual information. This provides a significant advantage over mutual information that requires expensive calculation of joint histograms. Additionally, as compared with

other similarity measures like mutual information, NCC values have an intuitive interpretation which simplifies the selection of threshold parameters used to discriminate good matches from bad.

4. **Single Transformation Output.** For precise matching the Euclidean transformation parameters obtained from rigid initialization are used to locate and transform corresponding local neighborhoods to avoid applying an expensive rigid transformation to the entire image. Only one whole-image transform is necessary to generate the final registered result. This offers a significant advantage over multi-resolution or iterative optimization-based approaches that require a whole-image transformation at each iteration.

5. **Parallelization for Precise Matching.** The process of precise matching is embarrassingly parallel, lending itself to execution on multiple cores, sockets, or a computers in a cluster.

This chapter is organized as follows: In Section 2.2, the two-stage registration algorithm is presented along with a discussion on reconstruction applications. Section 2.3 discusses the workflow and computational aspects to prepare for Chapters 3 and 4 where high performance implementations are discussed. Results for the algorithm are presented in Section 2.5. In Section 2.6, existing approaches for image registration are reviewed with emphasis on large scale research projects that require the alignment of 2D microscopic slides for 3D reconstructions.

2.2 Two-Stage Scalable Registration

To address the specific challenges of nonrigid distortion, large image size, and feature rich content, an algorithm is proposed that consists of two stages: rigid initialization and nonrigid registration. Rigid initialization estimates the rough alignment of the base-float image pair from the consensus of correspondences between *high level features*, image regions that correspond to small, distinct, and anatomically significant features such as blood vessels or other ductal-type structures. The nonrigid stage seeks to refine the rigid initialization by establishing pixel-precision correspondences by matching areas of intensity information. The initialization reduces the search for matching in the nonrigid stage, resulting in a lower likelihood of erroneous matches and less computation. This combination provides an approach to the problem of automatic sectioned image registration and reconstruction that is robust, easily parallelizable, and scalable.

2.2.1 Fast Rigid Initialization

The basis of the rigid initialization stage is the matching of *high level features* or small regions that correspond to anatomically significant features such as blood vessels, mammary ducts, or small voids within the tissue boundary. This is a natural choice for features in microscopy images that has several advantages over the more primitive features generated by commonly used methods such as corner detection. First, the amount of high level features is relatively limited keeping the total number of possible matches reasonable. This is especially important when gross misalignment between images is possible and the range of search for feature matches cannot be limited. Second, the descriptions used to match these features such as shape, size, and eccentricity are invariant under rotation and translation and so the matching can accommodate the full range of misalignments. Third, the

feature descriptions are scalars and are fast and simple to compare once computed. Finally, many choices of high level features remain comparable even when the images to be registered have distinct stain types. This permits, for example, the alignment of an hematoxylin and eosin stained image with an immunohistochemically stained image. In contrast, performing corner detection on a typical microscopy image generates an overwhelming number of features due to the textural quality of the content. These features are relatively ambiguous, and their comparison requires the use of neighborhood intensity information and has to account for differences in orientation and also appearance if distinct stains are used.

High Level Feature Extraction

Extraction of high level features is a simple process as the features often correspond to contiguous regions of pixels with a common color characteristic. Color segmentation followed by morphological operations for cleanup usually suffice. The computational cost of these operations can be significantly reduced by performing the extraction on downsampled versions of the original images without compromising the quality of the final nonrigid result. The rigid estimate only serves as an initialization for the nonrigid stage and a matter of even tens of pixels difference is insignificant to the outcome of nonrigid stage. Figure 2.1 demonstrates the extraction process, showing an example from one of the Placenta test images.

Given the base image B, and float image F, their respective feature sets $\mathcal{B} = \{b_i\}$ and $\mathcal{F} = \{f_j\}$ are extracted according to the process described above. Each feature has associated with it a set of *descriptors* used for the matching processes, $\boldsymbol{b_i} = (\vec{\boldsymbol{x}}_i^b, s_i^b, e_i^b, \phi_i^b)$ and $\boldsymbol{f_j} = (\vec{\boldsymbol{x}}_j^f, s_j^f, e_j^f, \phi_j^f)$, where $\vec{\boldsymbol{x}} = (x, y)$ is the feature centroid, s the feature area in

pixels, e the feature eccentricity, and ϕ the feature semimajor axis orientation. These feature descriptions are computed using Matlab's Image Processing Toolbox (The Mathworks, Natick, MA).

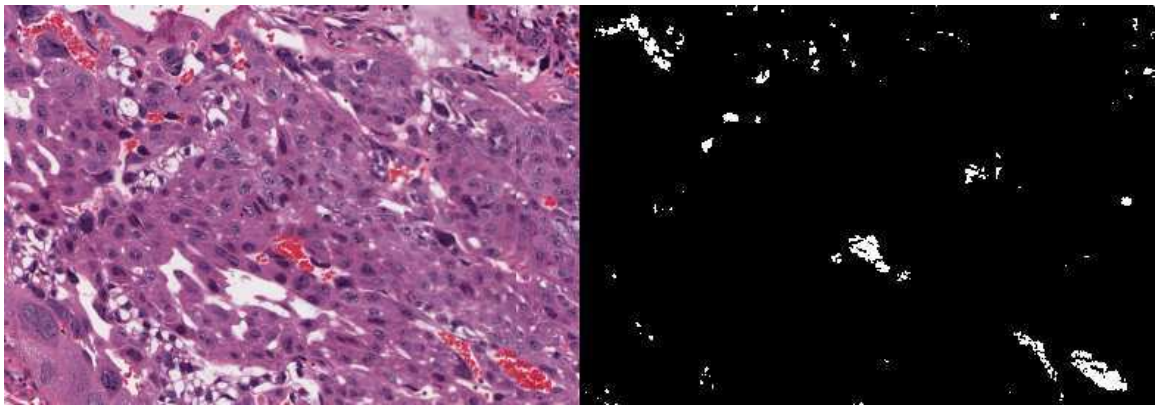


Figure 2.1: High level feature extraction. The binary image shows extracted features representing blood vessels. These features are extracted using color segmentation with morphological operations for cleaning up noise. Descriptions of centriod location, size, eccentricity, and major-axis orientation are calculated for each distinct feature.

High Level Feature Matching

The rigid initialization stage uses a scheme for matching high level features to establish spatial correspondences between the base and float images. The following describes the conventions used for feature matching. Matches between individual features are referred to as *match candidates* if their size and eccentricity descriptors are *consistent*. That is, given the feature sets $\mathcal{B}, \mathcal{F}$, a match candidate (b_i, f_j) is established if the descriptors of size s_i^b, s_j^f and eccentricity e_i^b, e_j^f are consistent within given percent difference thresholds ϵ_s, ϵ_e

$$(b_i, f_j) \Leftrightarrow \begin{cases} \frac{|s_i^b - s_j^f|}{min(s_i^b, s_j^f)} \leq \epsilon_s \\ \frac{|e_i^b - e_j^f|}{min(e_i^b, e_j^f)} \leq \epsilon_e \end{cases} . \tag{2.2}$$

Generating a model rigid transformation $(\tilde{\theta}, \tilde{T_x}, \tilde{T_y})$ requires, at minimum, a pair of match candidates. To identify models originating from coherent pairs of match candidates, geometric consistency criteria are used to ensure consistent intra-image distances between feature centroids and also consistent feature orientations. For a pair of match candidates to form a *candidate pair*, $\{(b_i, f_j), (b_k, f_l)\}$, the intra-image centroid-to-centroid distances between features b_i, b_k and f_j, f_l are required to be consistent within the percent difference threshold $\epsilon_{\vec{x}}$ (see Figure 2.2). Additionally, for the initialization stage, the orientations of the feature semimajor axes must be consistent with the model transformation angle $\tilde{\theta}$

$$\{(b_i, f_j), (b_k, f_l)\} \Leftrightarrow \begin{cases} \frac{|\|\vec{x}_i^b - \vec{x}_k^b\|_2 - \|\vec{x}_j^f - \vec{x}_l^f\|_2|}{min(\|\vec{x}_i^b - \vec{x}_k^b\|_2, \|\vec{x}_j^f - \vec{x}_l^f\|_2)} \leq \epsilon_{\vec{x}} \\ |\phi_i^b - \phi_j^f - \tilde{\theta}| < \epsilon_\phi \\ |\phi_k^b - \phi_l^f - \tilde{\theta}| < \epsilon_\phi \end{cases} . \tag{2.3}$$

The model transformation $(\tilde{\theta}, \tilde{T_x}, \tilde{T_y})$ for the candidate pair $\{(b_i, f_j), (b_k, f_l)\}$ is calculated by first solving for the angle $\tilde{\theta} = tan^{-1}((y_i^f - y_k^f)/(x_i^f - x_k^f)) - tan^{-1}((y_j^b - y_l^b)/(x_j^b - x_l^b))$, corrected to the interval $[-\pi, \pi]$. The translation components $\tilde{T_x}, \tilde{T_y}$ are calculated using $\tilde{\theta}$ and least squares. Typical values for percent difference tolerances $\epsilon_s, \epsilon_e, \epsilon_{\vec{x}}$ are 0.1-0.2, and $5 - 10^\circ$ for the orientation threshold ϵ_ϕ.

The match candidate and candidate pair concepts are illustrated in in Figure 2.2 and the algorithm is summarized in Algorithm 1.

Histogram Voting

Determining an estimate for rigid registration from a set of feature matches requires a method that is robust to erroneous matchings. This is especially true in microscope images

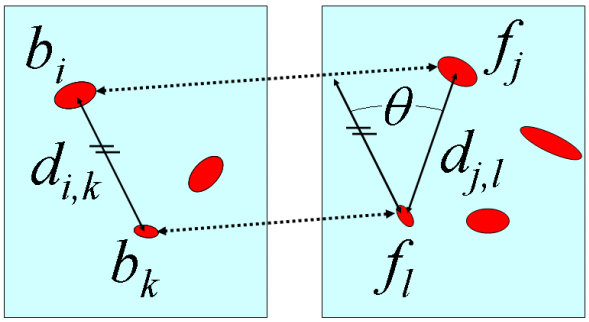


Figure 2.2: High level feature matching. Features are matched between the base and float images based on size and eccentricity to form *match candidates* $(b_i, f_j), (b_k, f_l)$. Intra-image distances $d_{i,k}, d_{j,l}$, between pairs of match candidates are compared to identify *candidate pairs*. A model rigid transformation, $(\tilde{\theta}, \tilde{T}_x, \tilde{T}_y)$, is defined for candidate pairs with consistent distances.

where many features are indistinguishable, and a substantial amount of mismatches are inevitable. The fundamental idea of the method presented in [33] is the recognition that any candidate pair $\{(b_i, f_j), (b_k, f_l)\}$ defines a model rigid transformation $(\tilde{\theta}, \tilde{T}_x, \tilde{T}_y)$, and for candidate matches and candidate pairs chosen using the criteria described in Section 2.2.1, a large portion of the concomitant model transformations will concentrate around the desired parameters in the Euclidean transformation space.

With a set of model transformations identified from consistent candidate pairs, a histogram voting scheme is used to estimate the initialization parameters (θ, T_x, T_y). First, θ is estimated by counting the models in the w_θ-interval centered at at each $\tilde{\theta}$, taking θ as the $\tilde{\theta}$ with the largest w_θ-interval count. The models that fall within this maximum count w_θ-interval are then selected and used to estimate the translation parameters. Interval counting is then applied with w_T-intervals centered at each of $\tilde{T}_x, \tilde{T}_y$ from the selected models to identify T. This algorithm is summarized in Algorithm 2. Although counting with intervals centered at each model adds computation, this prevents the possibility of splitting the

Algorithm 1 Rigid Feature Matching

input: Feature sets $\mathcal{B}$ and $\mathcal{F}$, thresholds $\epsilon_s, \epsilon_e, \epsilon_{\vec{x}}, \epsilon_\phi$.
initialize match candidates $\mathcal{M} = \{\}$
for each $b_i \in \mathcal{B}$
    for each $f_j \in \mathcal{F}$
        compare s_i^b, s_j^f, and e_i^b, e_j^f with ϵ_s, ϵ_e
        if b_i, f_j consistent **then**
            $\mathcal{M} = \mathcal{M} \cup \{(b_i, f_j)\}$
    end
end
initialize match pairs $\mathcal{P} = \{\}$
for each $(b_i, f_j) \in \mathcal{M}$
    for each $(b_k, f_l) \in \mathcal{M}, k \neq i, l \neq j$
        if $\frac{|\|\vec{x}_i^b - \vec{x}_k^b\|_2 - \|\vec{x}_j^f - \vec{x}_l^f\|_2|}{min(\|\vec{x}_i^b - \vec{x}_k^b\|_2, \|\vec{x}_j^f - \vec{x}_l^f\|_2)} \leq \epsilon_{\vec{x}}$ **then**
            compute model transformation $(\tilde{\theta}, \tilde{T})$
            if $|\phi_i^b - \phi_j^f - \tilde{\theta}|, |\phi_k^b - \phi_l^f - \tilde{\theta}| < \epsilon_\phi$ **then**
                $\mathcal{P} = \mathcal{P} \cup \{(b_i, f_j), (b_k, f_l)\}$
    end
end
output: $\mathcal{P}$

mode with an arbitrarily placed histogram bin boundary, which might allow another interval to emerge as the maximum. An example result for histogram voting is presented in Figure 2.3. Interval sizes for histogram voting typically range from $0.5 - 1^\circ$ for θ and from 30-50 pixels for T_x, T_y. Parameter choices for the placenta dataset are described in Table 2.1 of Section 2.5.

Feature Matching in Partial Common Tissue Scenario

In many microscopy applications, a pair of images that are to be registered may share only a portion of their tissue content. The harsh sectioning and mounting process may remove or separate part of one sample, or there may be multiple samples mounted per slide. Registering the pair via an optimization method such as Maximum Mutual Information may

Algorithm 2 Rigid Voting

input: Candidate pairs $\mathcal{P}$, interval sizes ω_θ, ω_T.
for each $p_i \in \mathcal{P}$
    $\Theta_i = \{p_j \in \mathcal{P}, j \neq i : |\tilde{\theta}_j - \tilde{\theta}_i| \leq \frac{\omega_\theta}{2}\}$
    $t(i) = |\Theta_i|$
end
$\alpha = \arg\max_i t(i)$
$\theta = \tilde{\theta}_\alpha$
for each $p_i \in \Theta_\alpha$
    $x(i) = |\{p_j \in \Theta_\alpha, j \neq i : |\tilde{T}_{x_j} - \tilde{T}_{x_i}| \leq \frac{\omega_T}{2}\}|$
    $y(i) = |\{p_j \in \Theta_\alpha, j \neq i : |\tilde{T}_{y_j} - \tilde{T}_{y_i}| \leq \frac{\omega_T}{2}\}|$
end
$\beta = \arg\max_i x(i), T_x = \tilde{T}_{x_\beta}$
$\gamma = \arg\max_i y(i), T_y = \tilde{T}_{y_\gamma}$
output: (θ, T_x, T_y)

be troublesome in this scenario, since the optimum position may be obscured by the lack of similarity of the overall image when the common areas are aligned.

Feature matching provides a means to establish correspondences between common tissue regions of disparate images. Regardless of the matching criteria used, the set of correct matches from the common areas will undoubtedly be accompanied by a significant number of erroneous coincidental feature mismatches from the non-common areas. A method for recovering the alignment of the image pair must distinguish the signal of correct matches from the noise of the erroneous matches. High level feature matching with histogram voting has demonstrated some capability of successfully recovering alignment in this scenario, as is demonstrated in Section 2.5.

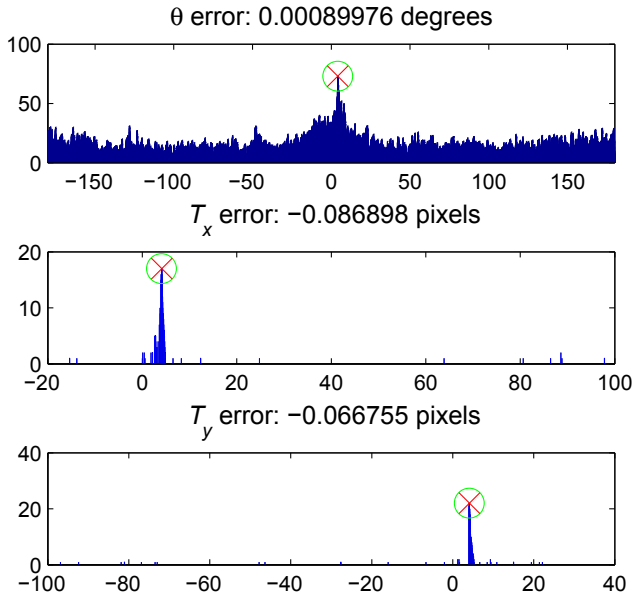


Figure 2.3: Sample histogram voting result for rigid initialization of placenta image pair. Manual parameter results are shown in red and automatic results in green. Errors between manual and automatic parameter estimates are indicated for each parameter. The images used for this example were approximately $16K \times 16K$ pixels in size.

2.2.2 Nonrigid Registration

Correcting nonrigid distortion to the accuracy necessary for end applications in quantitative phenotyping requires establishing a large number of precise spatial correspondences between base and float image pairs. The desired pixel-level precision suggests that assignment of correspondences between representative features, such as high level features, is not accurate enough. Instead, direct comparison of intensity information is needed which introduces the problem of computational burden. These considerations are addressed in an approach to extraction and matching of *intensity features* that compares small tile regions between the base and float images in an efficient manner, using the rigid initialization parameters to align them and Fast Fourier Transform to compute their cross-correlations for

matching. The implementation of this approach using graphics processing units (GPU) has been presented previously [46], here this approach is explained in detail with a focus on parallel implementation using CPU clusters.

Intensity Feature Extraction

The primary issue in extracting intensity features is the selection of unambiguous regions that are likely to produce accurate matches. This issue is especially important for matching in nonrigid registration since using the aggregate of matching results to make inference about the quality of any single match is difficult due to the freedom and subtlety of nonrigid distortions. In this sense the matchings at this stage are local: the only information available to judge their quality comes from the individual intensity regions themselves.

Good candidate regions for matching have rich content, a mixture of different tissues or tissue and background that forms a distinctive appearance. Often these regions will coincide with blood vessels, ducts, or other content with distinctive shape. Regions containing uniform tissue are not good candidates for matching, as accurate matchings are unlikely due to the textural quality of content and the natural morphological differences between sections. A simple way to enforce this quality in selected intensity features is to choose templates whose variance meets a certain minimum threshold. That is, for any feature point p with coordinates $\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right]$ centered in the $W \times W$-pixel window a variance condition must be met

$$\frac{1}{W^2 - 1} \sum_{i,j} (t(i,j) - \bar{t})^2 \geq \sigma^2 \tag{2.4}$$

where t is the *template*, a grayscale representation of the p-centered pixel window with mean value $\bar{t}$, and σ^2 a significance threshold. There are cases where the variance threshold

can be met and an ambiguous matching result can occur (consider matching a two templates with upper half white and lower half black) although these cases are uncommon in natural images.

Another important issue in intensity feature matching is the spatial distribution of features. The correspondences resulting from the matching of intensity features forms the set of control points for a nonrigid transformation of the float image. These correspondences should be fairly distributed throughout areas of interest in order to produce a result that conforms in the areas where further analysis on the registered result will take place. To keep the total number of features reasonable and attempt an even spatial distribution, features are sampled uniformly over the image with a $W \times W$ tiling. For example, in the $16K \times 16K$ placenta images a tiling would typically fall in the range of 150-350 pixels to generate a total of 2025-11236 possible features, the large majority of which are discarded due to insufficient variance.

Intensity Feature Matching

For a selected feature point p_1 with coordinate $\left[\begin{smallmatrix} x_1 \\ y_1 \end{smallmatrix}\right]$ in the base image, a $N \times N$-pixel window is taken centered at p_1. This window is converted to grayscale and rotated by the angle θ obtained from the initialization stage. The central $W_1 \times W_1$-pixel patch is then used as the $p1$ template for identifying p_2, the correspondence point of p_1 in the float image. N is calculated from θ and W_1, taken just large enough to accommodate the rotated template.

The coordinate p_2 is estimated using the rigid initialization stage estimates for $\theta, \boldsymbol{T} = [T_x, T_y]^T$

$$p_2' = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \boldsymbol{T}. \tag{2.5}$$

A $W_2 \times W_2$-pixel tile ($W_2 > W_1$) centered at p_2', designated as the *search window*, is taken from the float image.

Commonly used similarity measures for intensity information other than NCC include summed square of difference (SSD) and mutual information (MI). SSD is not a good choice for microscopic images since the content tends to be discrete (e.g., sharp boundaries between cell nucleus and cytoplasm and cell membrane). MI is commonly used as a metric in gradient search strategies but the cost of joint-histogram computation makes its use in exhaustive search prohibitively expensive. We choose NCC since it is not only robust in identifying structural similarity but also highly efficient when implemented with fast Fourier transform. Furthermore, NCC values have an intuitive interpretation, making threshold parameter selection easier.

The NCC between the template and search window is computed as the quotient of covariance and individual variances

$$\rho(u,v) = \sum_{x,y} \frac{\{t(x-u,y-v) - \bar{t}\}\{s(x,y) - \bar{s}_{u,v}\}}{(\{t(x-u,y-u) - \bar{t}\}^2\{s(x,y) - \bar{s}_{u,v}\}^2)^{\frac{1}{2}}}, \tag{2.6}$$

where $\bar{t}$ is the template mean and $\bar{s}_{u,v}$ is the mean of the search window portion overlapping the template at offset (u,v). The center of the template offset location at the maximum NCC result is taken as p_2

$$(m,n) = \arg\max_{u,v} \rho(u,v) \tag{2.7}$$

$$p_2 = \begin{bmatrix} m \\ n \end{bmatrix} + p_2' + \begin{bmatrix} \frac{W_1 - W_2}{2} \\ \frac{W_1 - W_2}{2} \end{bmatrix} \tag{2.8}$$

If $\rho(m,n)$ value exceeds a threshold (usually 0.8 or greater), then the match is considered successful and $p1$, $p2$ are recorded as a correspondence. The intensity feature matching process is demonstrated graphically in Figure 2.4. A selection of sample matches

is presented in Figure 2.5. The algorithm for intensity feature extraction and matching is summarized in Algorithm 3.

The choice of W_1 and W_2 is based on the severity of the deformation as well as computational capacity. Empirically $W_2 = 2W_1$, however cases with large deformation may require a larger search area. Demonstration of the effect of tile size on execution time is demonstrated in section 2.5.

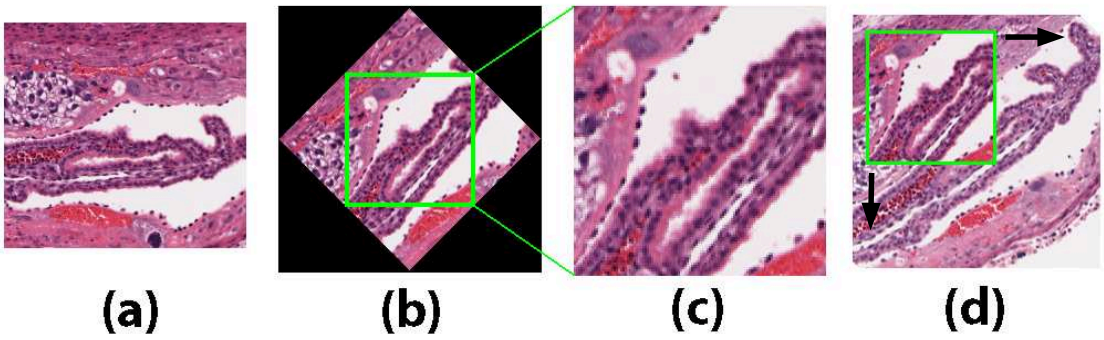


Figure 2.4: Intensity feature matching. (a) A template region from the base image meeting the variance condition is identified. (b) The region containing the rotated template is selected and rotated. The size N of the bounding box for the rotated template is calculated from θ. (c) The center $W_1 \times W_1$ portion of the rotated template area is extracted. (d) The normalized cross correlation between (c) and the corresponding search area within the float image is computed at all offsets with full overlap.

The large number of features that exist within a typical dataset makes efficient computation of NCC critical. Additionally, rather than using a search strategy NCC is computed between template and search window pairs at all spatial offsets to avoid the problem of local minima. For calculating normalizing factors in the denominator of Equation 2.6 the method of running sums is used as presented in [47]. This avoids the expensive local calculations of search window mean and variance for the template overlap region as the template

Algorithm 3 Intensity Feature Extraction and Matching

input: Rigid initialization estimate (θ, T_x, T_y), feature window size W_1, search window size W_2, variance threshold σ_2, NCC threshold τ.
initialize correspondences $\Omega = \{\}$
tile base image B into $W_1 \times W_1$ tiles t_i, centered at p_i
for each t_i
  if $variance(t_i) \geq \sigma_2$ **then**
    compute $N(\theta)$
    take p_i-centered $N \times N$ tile from B
    Rotate $N \times N$ tile by θ,
    extract center $W_1 \times W_1$ portion, t_i
    $q_i' = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} p_i + \boldsymbol{T}.$
    take q_i-centered $W_2 \times W_2$ tile from F, s_i
    compute $\rho(u, v) = NCC(t_i, s_i)$
    $(m, n) = \arg\max_{u,v} \rho(u, v)$
    $q_i = \begin{bmatrix} u \\ v \end{bmatrix} + q_i' + \begin{bmatrix} \frac{W_1 - W_2}{2} \\ \frac{W_1 - W_2}{2} \end{bmatrix}$
    if $\rho(m, n) \geq \tau$ **then** $\Omega = \Omega \cup \{(p_i, q_i)\}$
end
output: Ω

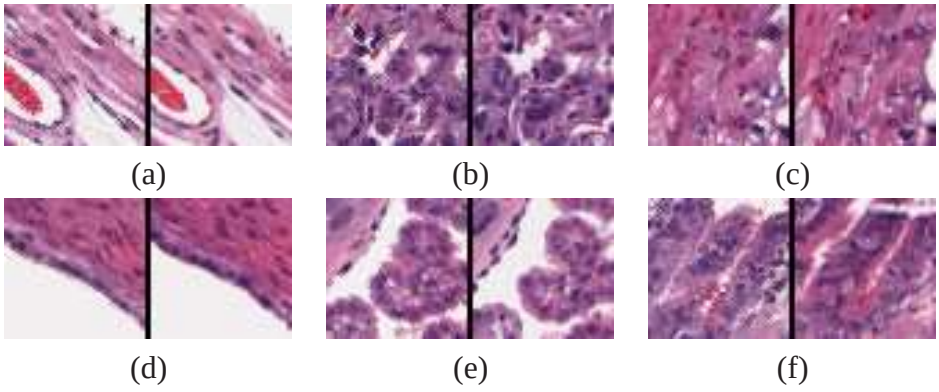


Figure 2.5: (a-f) Sample intensity feature matches.

is shifted through each of $(W_1 + W_2 - 1)^2$ positions, reducing the operation count from $3W_2^2(W_1 - W_2 + 1)^2$ to approximately $3W_1^2$. The unnormalized cross-correlation from the numerator of Equation 2.6 is calculated via the convolution theorem of the Discrete Fourier Transform that relates the product of DFT spectra to circular convolution in the spatial domain. To compute cross correlation ordinary convolution is required so t and s are padded with zeros to size $W_1 + W_2 - 1$ prior to forward transform to ensure that the circular overlap portions of the convolution result are null.

2.2.3 Image Transformation

The collection of point correspondences generated by the precise matching process provides the information needed to form a mapping that transforms the float image into conformation with the base. A variety of nonrigid mappings are used in practice, differing in computational burden, robustness to erroneous correspondences, and existence of inverse form.

The polynomial transformation

In choosing a transformation type a transformation is desired that is capable of correcting complex distortions, that is robust to matching errors, that admits a closed inverse form, and is computationally reasonable to calculate and apply. Of the commonly used nonrigid mapping types such as thin-plate spline, local weighted mean, affine, polynomial, and piece-wise variations, the polynomial mapping variety is chosen. Thin plate spline provides a minimum energy solution which is appealing for problems involving physical deformation, however perfect conformity at correspondence locations can potentially cause large distortion in other areas and excess error if an erroneous correspondence exists. The lack of an explicit inverse form means the transformed image is calculated in a forward direction, likely leaving holes in the transformed result. Methods such as gradient search can be used to overcome the inverse problem but at the cost of added computation which can become astronomical when applied at each pixel in a gigapixel image. Kernel-based methods such as local weighted mean require a uniform distribution of correspondences. Given the heterogeneity of tissue features this distribution cannot always be guaranteed.

Polynomial warping admits an inverse form, is fast in application, and has been demonstrated as capable of satisfactorily correcting the distortion encountered in sectioned images [46]. Polynomial warping parameters can be calculated using least squares or least squares variants which can mitigate the effect of matching errors. Affine mapping offers similar benefits but is more limited in the complexity of the warpings it can represent.

Second degree polynomials are used for the nonrigid results presented in this chapter. Specifically, for a point (x, y) in the base image, the coordinate (x', y') of its correspondence in the float image is

$$\begin{cases} x' = a_1x^2 + b_1xy + c_1y^2 + d_1x + e_1y + f_1, \\ y' = a_2x^2 + b_2xy + c_2y^2 + d_2x + e_2y + f_2, \end{cases} \tag{2.9}$$

Since each pair of matched point correspondences provide two equations, at least six pairs of point correspondences are needed to solve for the coefficients in (2.9). In practice, a much larger number of point correspondences is obtained.

2.2.4 3D reconstruction

For 3D tissue reconstruction applications, where a sequence of images is to be registered together, the matching process is applied successively to each ordered pair in the sequence. Images are transformed starting at one end of the sequence, and at each step the transformations from prior image pairs are propagated through the match coordinates in order to achieve a coherent set of transformed images. Figure 2.10 in Section 2.5 shows a reconstruction result generated from a sample set of mouse placenta images. The improvement with respect to reconstruction quality that is provided by nonrigid registration is demonstrated in Figure 2.11.

2.3 Workflow and Computational Aspects

In this section the parallelization and computational considerations of the two stage algorithm are described to illustrate its potential to address large images. Chapters 3 and 4 treat these issues in greater detail in discussions on GPU and parallel implementations of the two stage algorithm.

The workflow for the two stage algorithm is summarized in Figure 2.6. With the exception of nonrigid transformation, the CPU-bound operations of each stage are easily parallelizable. The overwhelming majority of computation is concentrated in the normalized cross correlation calculations of intensity feature matching. For example, consider that the rigid initialization stage for a typical $16K \times 16K$ mouse placenta image executes in four seconds where the nonrigid stage executes in 60-90 seconds (depending on intensity feature and search window sizes). Referring to Table 2.3 containing a summary of serial execution times, 90+% of the time in the nonrigid stage is spent in extracting and matching intensity features, with upwards of 73% spent computing FFTs for normalized cross correlation calculations. Therefore efforts can be focused on the nonrigid stage, primarily on distributing and accelerating intensity feature extraction and matching operations.

2.3.1 Rigid Initialization Stage

Due to the modest computational requirements it is typically not necessary to reduce the execution time of the rigid initialization stage. If a real-time response is desired though then the operations of this stage are all parallelizable. The extraction of high level features uses a color segmentation followed with morphological operations for cleanup, both are independent and local operations that can also be pipelined with reading source images from disk. The matching of high level features and histogram voting consist of the simple search procedures detailed in Algorithm Tables 1 and 2. These searches can be carried out straightforwardly in parallel as well.

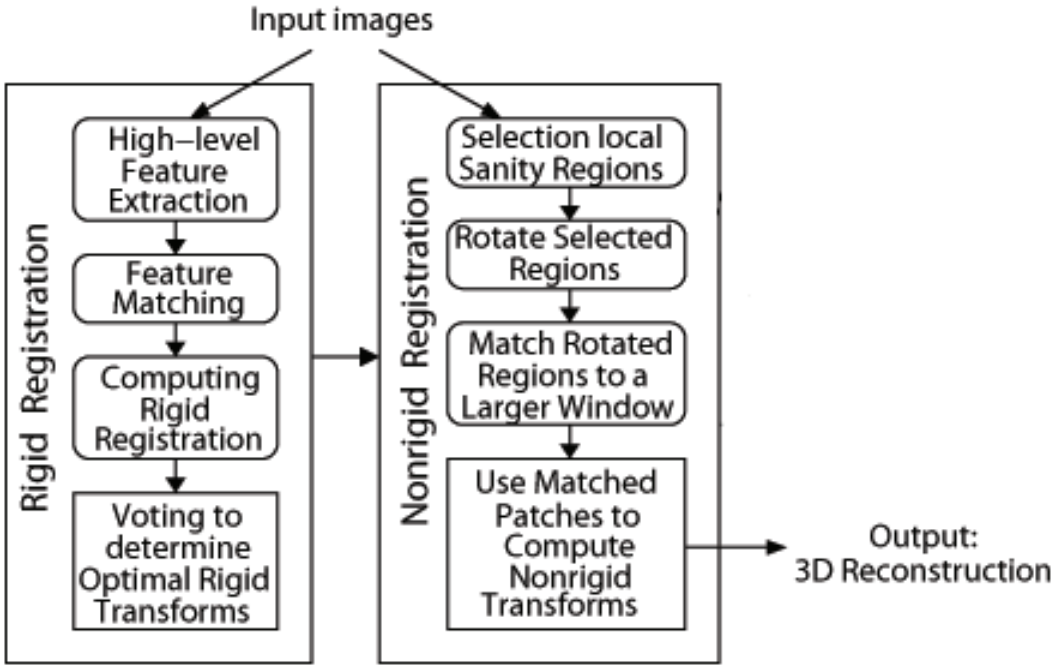


Figure 2.6: Workflow of the two stage nonrigid registration algorithm. Rounded items indicate operations that can be carried out simply in parallel. The most computationally demanding phase is the intensity feature matching portion, consisting of two forward FFTs and one inverse.

2.3.2 Nonrigid Stage

Where the primary effort is focused on improving intensity feature extraction and matching performance, the performance of reading images from disk and grayscale conversion can be improved as well.

Given the large size of microscope images, some in excess of 10 GB, reading from disk and decoding compressed images requires considerable time. A parallel file system may be employed to reduce the time spent reading from disk, although this requires distributing large amounts of data over network and complicates later implementation steps since the data will be distributed among several nodes rather than a single head node. A more simple

approach is to hide a portion of the read and decode time by overlapping reading/decoding with grayscale conversion, using the a head node to read/decode incrementally and asynchronous communication to distribute grayscale conversion of incremental read results to worker nodes.

With the grayscale base and float images in memory, the next step is to determine which template regions will serve as candidates for intensity feature matching. The base image can be divided evenly among the worker nodes that then compute the variances of the $W_1 \times W_1$ template sized tiling of their portions and return the results.

With a set of candidate intensity feature regions identified, what remains is to rotate them, extract their templates, and perform the correlations between the templates and their corresponding search areas. The candidate features are evenly divided among the worker nodes, who rotate them, extract their templates, and perform the correlations between template and search, returning the maximum correlation values and corresponding coordinates for each feature. The base image is stored in column-major format, so to keep communication to a minimum the candidate intensity feature regions are buffered and the remainder of the image is discarded. Asynchronous communication is used to keep the head node busy while send operations post. The search windows, taken from the float image, are handled in a similar manner. However, since the search windows for distinct features can overlap significantly they are not individually buffered, rather their union is buffered as a whole.

The Discrete Fourier Transforms necessary for calculating correlations on CPU are performed using the FFT library FFTW [48]. The 2D-DFT dimensions are critical for performance, ideally the size of the padded transform $W_1 + W_2 - 1$ is a power of two or a small prime factor. For the cases when this size rule cannot be obeyed, FFTW provides a simple mechanism called a *plan* that specifies an optimized plan of execution for the

transformation. This plan is precomputed and subsequently reused, resulting in a one-time cost. For example, with a template size $W_1 = 350$ and a search window size $W_2 = 700$, FFTW takes around 0.7 seconds to compute the two 1049×1049 forward transforms without planning, whereas with plan the computation takes only 0.32 seconds with a six second one-time penalty (a cost which can later be amortized by loading the plan from disk at runtime on subsequent transforms of the same size).

2.3.3 Nonrigid Transformation

The topic of high performance image transformations has been addressed previously [49–51]. Most focus on optimizing the use of cache and/or parallelization. Efficient distributed transformations are possible for many transformation types but often result in complex implementations due to spatial dependencies and communication requirements. For these reasons the focus of high performance computing for registration in this dissertation is restricted to the problems of establishing correspondences through feature matching.

2.4 Experimental Setup

The results of this chapter were computed with a fully serial implementation. Multiple cores or sockets were not used and no effort was made at acceleration other than using FFTW library to compute normalized cross correlations. Chapter 3 presents the results of [46] where a more sophisticated single node implementation uses graphics processors to accelerate FFT operations and PThreads to access multiple sockets and graphics processors. The parallel implementation described above is presented in further detail in Chapter 4.

Table 2.1: Summary of test parameter values for rigid initialization stage.

Parameter Description	Value
Size similarity (ϵ_s)	0.1
Eccentricity tolerance (ϵ_e)	0.1
Distance tolerance ($\epsilon_{\vec{x}}$)	0.1
Orientation tolerance (ϵ_ϕ)	5°
Voting interval for θ (ω_θ)	0.5°
Voting interval for T (ω_T)	30

2.4.1 Benchmark Dataset and Parameters

The two stage registration algorithm was applied to a set of mouse placenta images from a morphometric study on the role of retinoblastoma gene [22]. The goal in this study was the reconstruction of 3D tissue models to study microanatomy. A total of 100 images (99 pairs) were used, averaging 16K$\times$16K pixels in size and 730 MB each in uncompressed RGB form.

All image pairs in the dataset were run in the rigid initialization stage with the parameter values described in Table 2.1. The nonrigid stage was evaluated with a variety of values for W_1 and W_2, chosen to cover both optimal and sub-optimal cases for DFT size and to demonstrate the effect of parameter size on execution time performance. These parameter sets are summarized in Table 2.2.

2.4.2 Hardware

Experiments were run on a single node of the BALE Visualization Cluster at the Ohio Supercomputer Center, a General Purpose Graphics Processor Unit (GPGPU) equipped computing cluster. The Visualization Cluster contains 16 nodes, each equipped with dual-socket $\times$ dual-core AMD Opteron 2218 CPUs and 8GB DDR2 DRAM running at 667

Table 2.2: Summary of test parameters values for the nonrigid stage. Parameters for template size W_1 and search window size W_2 were chosen to reflect a realistic range that demonstrate effect on performance of size and optimality with respect to FFT.

Window size:	Small	Medium	Large
Template W_1 (pixels)	171	250	342
Search W_2 (pixels)	342	500	683
Aggregate ($W_1 + W_2 - 1$)	512	749	1024

MHz. All nodes are connected by Infiniband and include 750 GB, 7200 RPM local SATA II disks with 16 MB cache.

2.5 Experimental Results

2.5.1 Automatic Rigid Initialization vs. Manual Rigid Registration

The accuracy of rigid initialization is critical to the nonrigid stage. Most critical is the estimate of angular offset between the image pair. Where offsets in translation can be accounted for by increasing search window size, offsets in angle result in poor comparison of intensity features via NCC.

To demonstrate the accuracy of rigid initialization, the 99 image pairs were manually registered by selecting four control point pairs for each image pair. Control points were selected uniformly throughout the extent of the tissue area, taken at unambiguous visual features to get as close as possible to pixel precision in correspondence. For each image pair, the manual rigid estimates (θ, T_x, T_y) were calculated from the manually selected control points and compared to their corresponding estimates generated by the rigid initialization stage. Figure 2.7 shows the comparison errors between the manual and automatic results. For θ estimates, the automatic results are acceptable in 92 of 99 cases, falling within $\pm 4^\circ$

of their manual counterparts. For the translation estimates, T_x, T_y, most automatic results fall within 100 pixels of their manual counterparts, and all are within 450 pixels and so are easily accommodated by reasonable search window sizes.

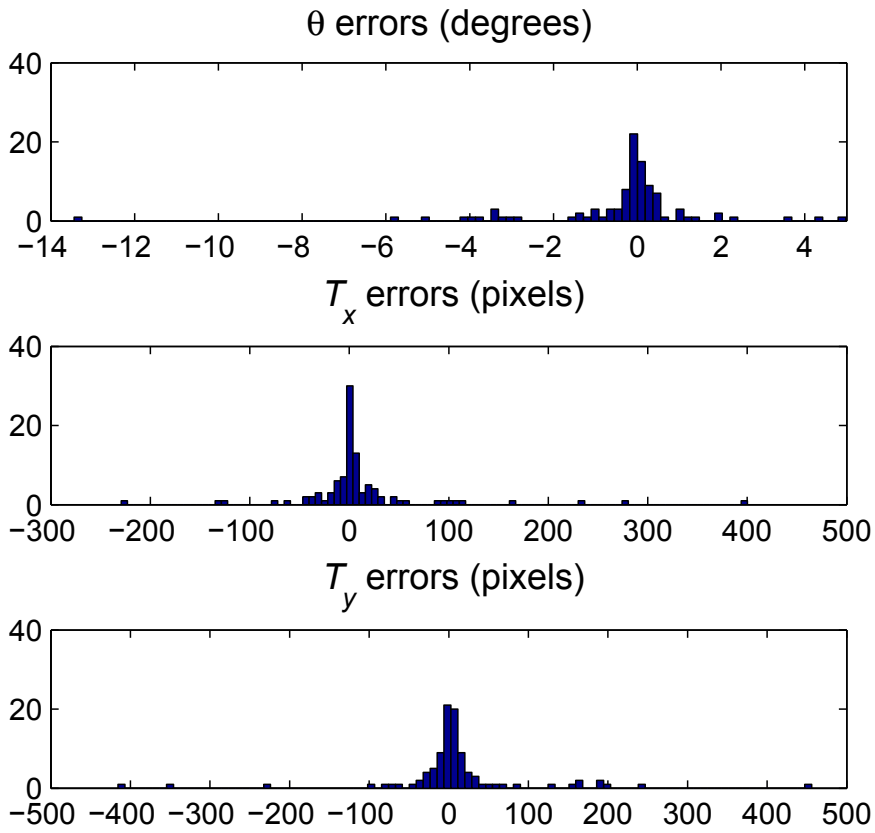


Figure 2.7: Histogram of errors between manual rigid and automatic rigid registrations. Automatic results are acceptable as input for the nonrigid stage in 93 of 99 cases.

The rigid initialization compares well to manual rigid registration, however a manual registration is not implicitly superior in terms of the resulting similarity between the registered base and float images. To objectively compare the quality of registrations between the automatic and manual methods, the manual and rigid initialization registrations from the prior experiment were used to transform the 99 image pairs, and the Normalized Mutual

Information (NMI) was calculated for both cases for each image pair. NMI is a popular similarity measure commonly used in image registration [52]. NMI is defined as the ratio of the sum of individual image entropies, $H(B), H(F)$, and the joint entropy of the base and float images, $H(B, F)$

$$H(B, F) = \frac{H(B) + H(F)}{H(B, F)}, \tag{2.10}$$

and is calculated via joint and individual histograms of grayscale image conversions. Figure 2.8 shows the results of the NMI calculations for the manual and automatic rigid registrations. Both methods are comparable, with the automatic registrations having greater NMI in 23 of 99 cases, with a maximum difference of less than -0.085 normalized bits.

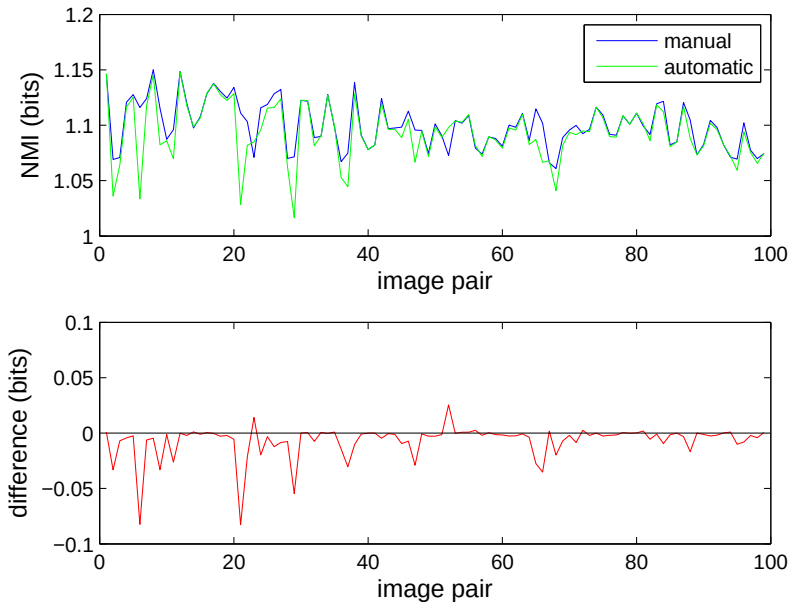


Figure 2.8: Comparison of manual and automatic rigid registration quality. Image pairs were registered using both manual rigid and automatic rigid methods and the normalized mutual information was calculated in each case. In terms of NMI, the manual and automatic rigid registrations are comparable. The automatic registrations have greater NMI in 23 of 99 cases, the maximum difference is less than -0.085 normalized bits.

2.5.2 Partial Common Tissue Simulation

To demonstrate the capability of the high level feature matching approach in a partial common tissue scenario, high level features were discarded from each of the 99 image pairs and the rigid initialization stage was re-applied to the truncated feature set images. For each base-float pair, the bounding box of the base image feature set was calculated, and the features in the rightmost 1/3 of this area were discarded. The manual registrations were used to identify the corresponding leftmost 1/3 in the float image, and those features were discarded as well. This left roughly 1/2 of the features common between the base float pair, depending on spatial distribution, effectively increasing the signal to noise ratio for the input to the rigid initialization stage. This is demonstrated in Figure 2.9. The rigid initialization parameter estimates for these modified images were compared to the manual registration parameters from the original images, and were found to be acceptable in 37 of 99 cases, falling within four degrees for θ and several hundred pixels for T_x, T_y.

2.5.3 Visualization of Nonrigid Registration Results

A sample 3D reconstruction from 50 placenta slides is presented in Figure 2.10. Due to the absence of ground truth, evaluation beyond visual inspection of nonrigid registration quality is difficult. Differences in morphology between adjacent sections can mask small but significant differences in quality regardless of the choice of evaluation metric. Figure 2.11 (d) and (e) demonstrates the improvement of nonrigid registration over rigid alone, where no coherent structures are apparent in the reconstruction (Figure 2.11(d)), preventing morphometric analyses of the volume. This improvement is also demonstrated in 2D in Figure 2.11, where difference images between the base and float are shown for the nonrigid and rigid-only cases.

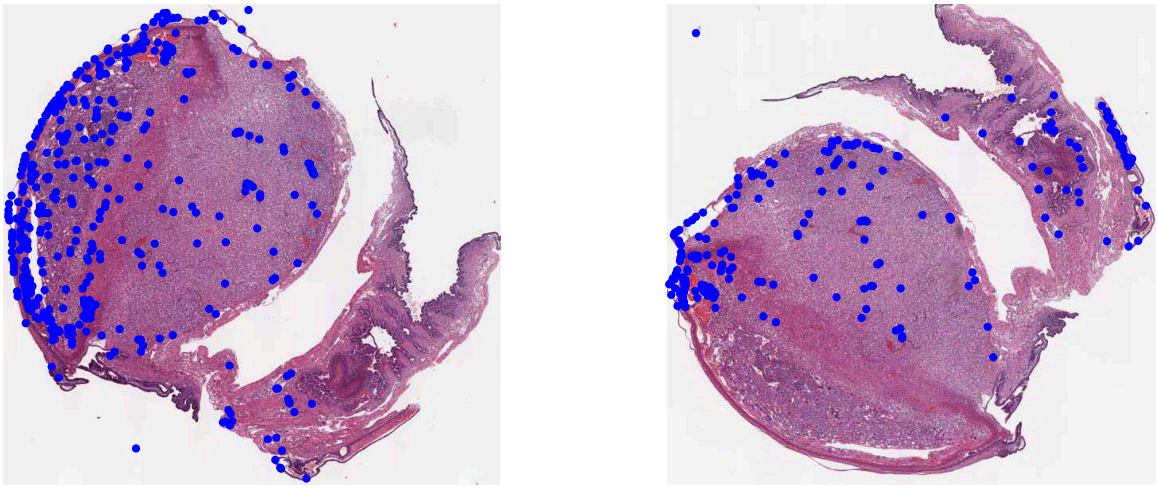


Figure 2.9: Partial common tissue simulation. Due to a feature matching approach, the rigid initialization stage is capable of recovering base-float alignment in the scenario where only part of the tissue is common between both images. To simulate this scenario, features were selected in the 2/3 left portion of tissue area of the base image. Using the manual rigid registrations, features from the float image are taken from the corresponding opposite 2/3 of tissue area, so that only 1/3 of the tissue area is common to both images. The rigid initialization results on these modified image pairs are acceptable in 37 of 99 cases.

2.5.4 Performance Results

The experiments from Table 2.2 were performed on the benchmark dataset using a serial implementation run on a single node configuration. A breakdown for the single node configuration execution time spent between loading from disk, grayscale conversion, and intensity feature extraction and matching is presented in Table 2.3. For each window size configuration, at least 90 percent of the total execution time is consumed by intensity feature extraction and matching. Since intensity feature extraction and matching are so demanding, and are consequently the focus of the high performance implementation effort, from this point forward all references to execution times are limited to only this portion of the nonrigid stage.

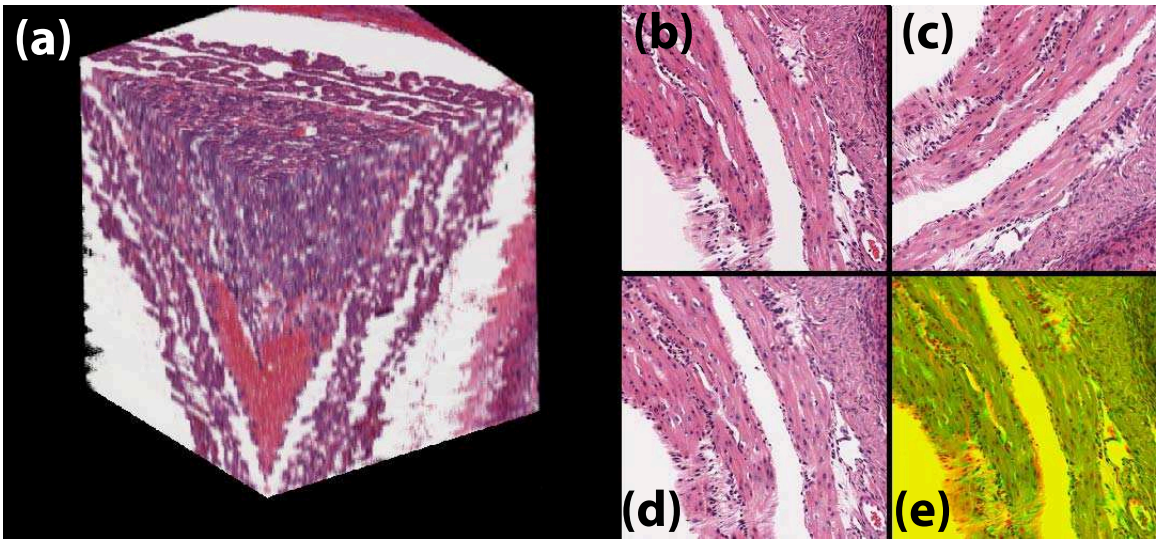


Figure 2.10: (a) A sample 3D reconstruction of mouse placenta. Only a fraction of the reconstructed volume is shown at high resolution due to the memory limitations of the rendering software; (b-e) Registration of mouse placenta images: (b) a 1000×1000-pixel patch from the base image; (c) Corresponding 1000×1000-pixel patch taken from the float image; (d) Patch from (c) after nonrigid transformation of the float image; (e) Overlay between between (b) and (d) with the grayscale representations embedded in the red and green channels respectively. Small areas of intense green or red indicate morphological differences between sections.

The number of intensity features extracted within each image varies significantly due to content. Table 2.4 summarizes the number of intensity features extracted per image in the dataset. The percentage of intensity features selected ranges from 10% to 30% of the total image area, with those percentages varying slightly with the value of W_1.

Execution times for the single node configuration running on a single Opteron CPU are presented in Figure 2.12. The small and large parameter sets both fulfill the optimal DFT size conditions, where the medium size is not compliant. The effect on execution time is apparent: For CPU with the FFTW library, the average time for the case of 295 seconds for the medium size versus 58 seconds for the small and 91 seconds for the large.

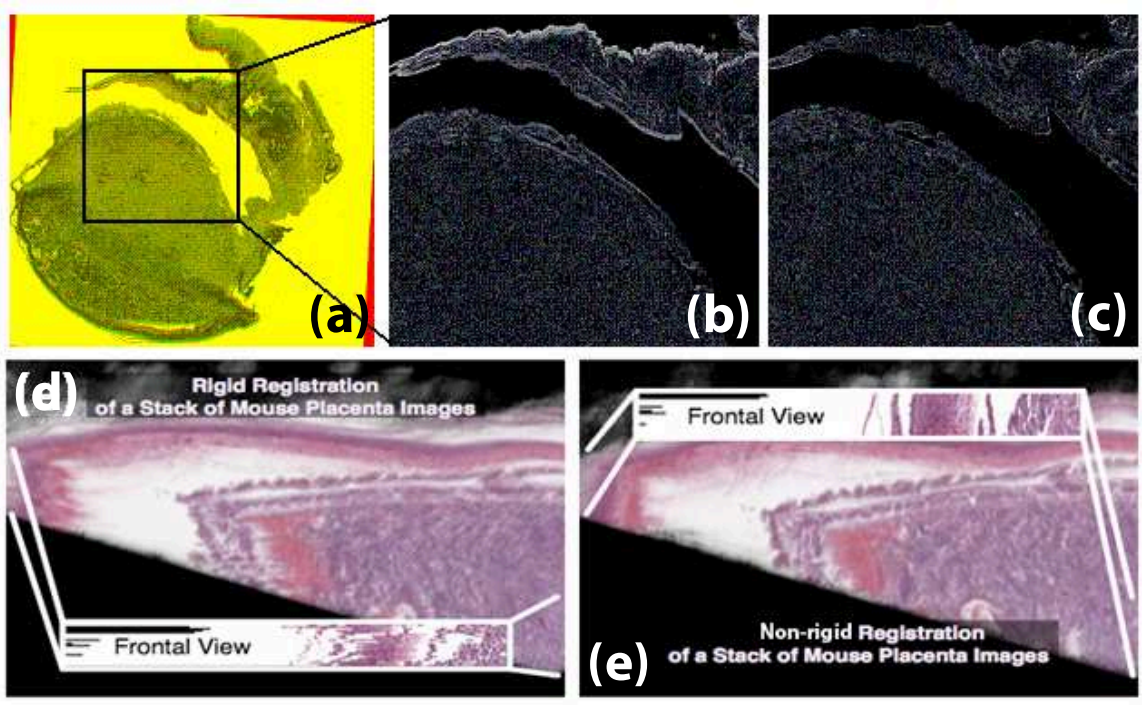


Figure 2.11: (a) Overlay of base and float placenta images after rigid registration; (b) High-resolution differenced patch from (a); (c) High-resolution differenced patch from same area as (b) following nonrigid registration; (d)-(e) Rendering of an edge view of placenta reconstruction, the frontal views represent virtual cross-sections of the reconstructed tissue; (d) with rigid registration alone, no coherent structures are apparent in the frontal view; (e) nonrigid registration corrects the structural distortions apparent in (d) and the reconstructed volume is then suitable for further analysis.

Doubling window sizes in the optimal cases from large to small only increments execution times by 60%, where moving from the small optimal size to the non-compliant medium size increases execution times by nearly 410%.

Taking the single node configuration as a departure point, the total execution times for the entire dataset are 1.59, 8.10, and 2.51 hours for the small, medium, and large window sizes respectively.

Table 2.3: Average percentage of execution time for elements of the nonrigid stage over all image pairs as executed on a single node (serial) configuration.

Window sizes: (W_1,W_2)	Loading	Grayscale Conversion	Intensity Feature Extraction & Matching	FFT
Small (171,342)	9.3%	3.8%	90.7%	73.4%
Medium (250,500)	1.5%	0.6%	98.5%	95.4%
Large (342,683)	6.8%	2.8%	93.2%	78.5%

Table 2.4: Intensity feature distribution per image. The number of intensity features extracted for each image differs due to content and the value of W_1.

	Number of features extracted		
Statistic: (W_1,W_2)	Small (171,342)	Medium (250,500)	Large (342,683)
Maximum	2121	1105	657
Minimum	676	358	207
Average	1241	656	392

2.6 Related Work

Image registration has been extensively studied in many applications including biomedical imaging, geological survey and computer vision. Since it is not possible to provide a complete list of literatures on registration, this section focuses on the recent works in 3D reconstruction of biological samples at microscopic resolution and HPC solutions for image registration.

Besides these works on the topic of registration, the rigid registration algorithm presented in this chapter shares some similarities with the geometric hashing algorithm [43, 53]. However, since the rigid registration algorithm is focused on Euclidean transformation,

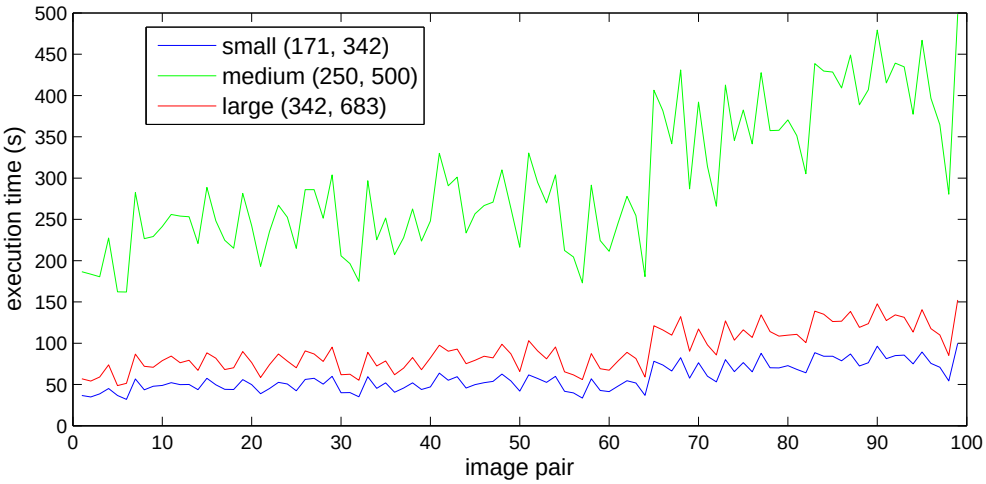


Figure 2.12: Execution times for single node (serial) configuration.

the search for matches can be performed directly and exhaustively with voting in the space of transformation parameters instead of working in the space of feature representations. Voting over all possible matches yields very accurate estimates of rigid transformation.

2.6.1 Registering microscopic images for 3D reconstruction in biomedical research

There have been many works focusing on acquiring the capability for analyzing large microscopic image sets in 3D space. In [14] and [31], the authors used stacks of confocal microscopic images to develop a 3D atlas for brains of various insects including honeybee and fruit fly. Both research groups focus on developing a consensus 3D model (atlas) for all key functional modules of insect brains. In [54], gene expression patterns in whole fly embryos are studied in 3D space using stacks of confocal microscopic images. In the Edinburgh Mouse Atlas Project (EMAP), 2D and 3D image registration algorithms have been developed to map the histological images with 3D optical tomography images of the

mouse embryo [28]. Similarly, in [25] the authors presented a workflow for observing the 3D distribution of expression patterns of genes in mouse embryo. In [26], the authors built 3D models for human cervical cancer samples using stacks of histological images in clinical settings. A complete study on registering large microscopic images of mouse brain sections was presented in [37].

2.7 Discussion and Conclusions

The next generation of automated microscope imaging applications, such as quantitative phenotyping, require the analysis of extremely large image datasets, making scalability and parallelization of algorithms essential.

This paper presents a fast, scalable, and parallelizable algorithm for image registration that is capable of correcting the nonrigid distortions of sectioned microscope images. Rigid initialization follows a simply reasoned process of matching high level features that are quickly and easily extracted through standard image processing techniques. Nonrigid registration refines the result of rigid initialization, using the estimates of rigid initialization to match intensity features using a fast FFT-based implementation of normalized cross-correlation.

The rigid initialization approach is based on the matching of high level features, using feature descriptions and geometric constraints to identify candidate pairs of feature matches. Histogram voting on model rigid transformations computed from the candidate pairs leverages the predominant presence of correct matches to produce estimates for rigid alignment of the feature sets. The rigid initialization performs well when compared to a manual registrations based on four control point pairs. Using normalized mutual information (NMI) to compare the registration outcomes, the automatic rigid registrations are

comparable to manual registrations, having greater NMI in 23 of 99 cases. Overall, the process of high level feature matching and histogram voting yield high accuracy initialization for the nonrigid stage in most cases (92 of 99 instances), which significantly reduces the computational burden for handling images with hundreds of millions or billions of pixels.

The registration framework presented here is part of an effort in designing a microscopic phenotyping system for biomedical research. One of the goals of this system is to build realistic 3D models for biological samples at micron resolution. The effectiveness of the nonrigid registration algorithm presented here is demonstrated by the success in building the 3D models for the samples (mouse placenta) with microanatomical structures clearly reconstructed. This framework is of great importance in helping biologists to characterize the changes in tissue morphology at the microscopic level induced by various genetic perturbations (e.g., gene knockout).

Two advantages of high level feature matching are being pursued in further applications. Firstly, high level feature matching enables the registration of images of different modalities such as microscopic images with different stain types. This turns out to be important for many studies in pathology where serial histological sections are stained for different proteins and overlaid (registered) to study the co-expression of multiple genes. A system is currently being developed for registering images with different stain types based on the work presented here. The second advantage of the high level feature matching approach is the capability to register images with only partial overlap. When simulating a partial tissue overlap scenario, the outcome is acceptable as input to the nonrigid stage in 37 of 99 instances, demonstrating a capability that could possibly extend to registration of images in occlusion scenarios in more general applications.

CHAPTER 3

NON-RIGID REGISTRATION FOR LARGE SET OF MICROSCOPIC IMAGES ON GRAPHICS PROCESSORS

The two stage algorithm provides a fast and parallelizable method for reconstructing tissue from large microscopic image sequences. The time necessary to resolve correspondences for nonrigid registration is not unreasonable at 1.6 hours for a sequence of 100 16K$\times$16K images. In practice, however, biological studies may require the analysis of thousands of such images, or images of much larger sizes, easily extending execution time from the scale of hours to days.

The primary bottleneck in the two stage algorithm is the precision matching of intensity features. As indicated in the previous chapter, up to 94% of execution time is spent computing the FFTs used to implement normalized cross-correlation. The first matter then in reducing execution time is the acceleration of these FFT calculations.

In this chapter I present a method for the hardware acceleration of FFT calculations using graphics processors (GPUs) and multi-socket processing on a single compute node. The features of GPUs are combined with multi-socket programming to achieve speed-up factors of up to 4.11x on a single GPU and 6.68x on a pair of GPUs using CUDA and pthreads versus a fully serial C++ CPU implementation. Execution results are shown for a benchmark composed of large-scale images derived from two different sources: Genetic

studies ($16K \times 16K$ pixels) and breast cancer tumors ($23K \times 62K$ pixels). It takes more than 12 hours for the genetic case in C++ to register a typical sample composed of 500 consecutive slides, which was reduced to less than 2 hours using two GPUs, in addition to a very promising scalability for extending those gains easily on a large number of GPUs.

3.1 Introduction

This chapter describes a high-performance computing approach for single-node processing of the two-stage algorithm, based on the work of [46]. Multi-socket parallelization enables multiple GPUs to simultaneously calculate the FFTs used to generate correspondences between microscopic images at the scale of hundreds of millions to billions of pixels. The primary advantage of this approach is the computing capacity of the GPU which has become a cost-effective parallel platform to implement grand-challenge biomedical applications [55, 56]. CUDA (Compute Unified Device Architecture) offers an alternate programming model to the underlying parallel graphics processor without requiring a deep knowledge about rendering or graphics. The interface uses standard C code with parallel features to transform the GPU technology to massive parallel processors for commodity PCs.

The results of this method are demonstrated by comparing serial and multi-socket parallel implementations with both CPU and GPU, using a variety of parameter choices to explore the efficiency and scalability of the approach (see Table 3.4.) The benchmark of image datasets (see Table 3.5) are taken from two quantitative phenotyping projects. The first project is a morphometric study on the role of the retinoblastoma gene (a well-known tumor suppressor) in mouse placenta development. In this study, three control placentas and three mutant placentas with Rb gene deletion were obtained. Each sample was sliced into

$5\mu m$ sections and each section was stained using standard hematoxylin and eosin staining. The stained sections were digitized using an Aperio ScanScope high resolution scanner with a $20X$ objective lens which produces a resolution of $0.46\mu m$/pixel. The six samples yielded more than 3,000 images with typical dimensions $16K \times 16K$ pixels for a total of more than three terabytes (uncompressed) of data. The second project is part of ongoing work studying the breast cancer tumor microenvironment in mice. Images from this study are typically $23K \times 62K$ pixels and around four gigabytes in uncompressed form.

This chapter is organized as follows: A summary of GPU architecture is provided in Section 3.2. Descriptions of the two-stage algorithm GPU implementations are provided in Section 3.3. The experimental setup is presented in Section 3.4. Performance results and analysis are contained in Section 3.5. The chapter concludes in Section 3.6 with a discussion on related work.

3.2 GPU Architecture and CUDA

The performance of algorithms on GPUs depends on how well they can exploit parallelism, closer memory, bus bandwidth, and GFLOPS.

Parallelism: Programs running on GPU are decomposed into threads and are executed on a massively parallel multiprocessor composed of 128 cores or stream processors (see central row in Figure 3.1).

Memory access: Data is stored on L1 caches, L2 caches and video memory (see lower rows in Figure 3.1), with closer memory being faster. Spatial locality is best exploited by caches, which are around a thousand times larger on the CPU, whereas temporal locality benefits the GPU, whose architectural rationale and programming model are inspired by the producer/consumer paradigm.

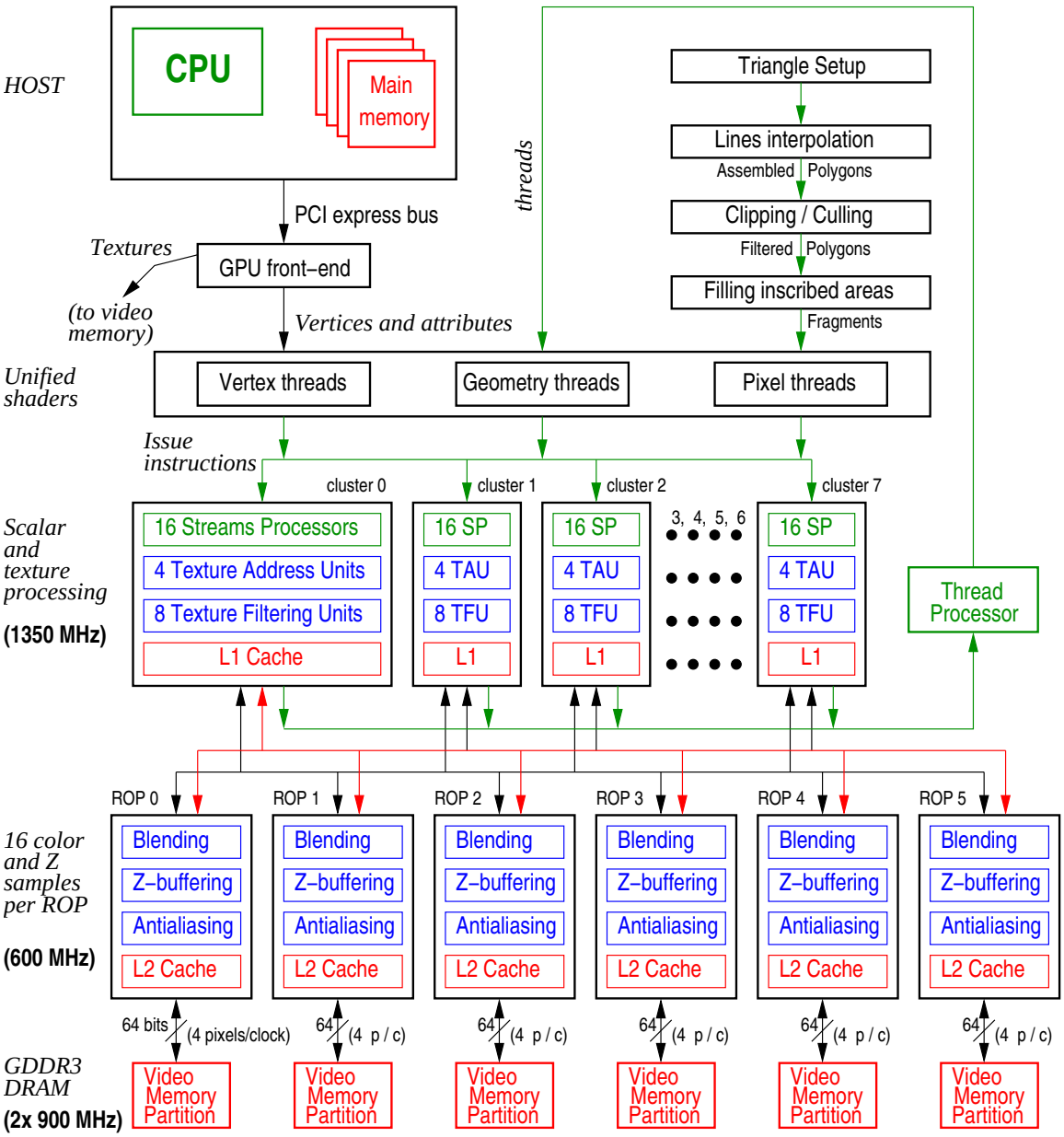


Figure 3.1: The block diagram of the Nvidia G80 architecture, the GPU used for experiments. The program, decomposed in threads, is executed on 128 streams processors (central row). The data are stored on L1 caches, L2 caches and video memory (lower rows).

Bus bandwidth: A state-of-the-art 2007 graphics card delivers a peak performance memory bandwidth around 80 GB/sec., as compared to 10 GB/sec. for CPU. This is mainly due to its wider data path (384 bits, decomposed into six partitions of 64 bits in Figure 3.1).

Computational units: The GPU capacity for floating-point operations exceeds 500 GFLOPS, in contrast with around 10 GFLOPS for a 2007 state-of-the-art CPU. This advantage is a result of design for the color and position interpolations that are required for performance graphics applications.

The outstanding features of the GPU and CPU are combined to create a bi-processor platform that balances workload and enhances the execution of the nonrigid registration. The rest of this section focuses on the GPU implementation.

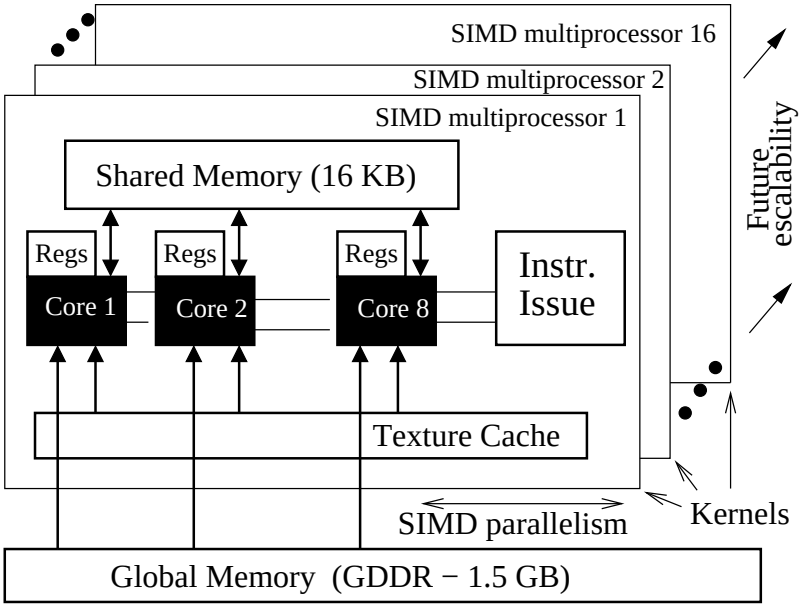


Figure 3.2: The CUDA hardware interface for the GPU.

3.2.1 The CUDA programming model

The CUDA (Compute Unified Device Architecture) [57] programming interface consists of a set of library functions which can be coded as an extension of the C language. The CUDA compiler generates executable code for the GPU, which is seen as a multicore

processor resource by the CPU. CUDA is designed for generic computing and hence it does not suffer from constraints when accessing memory, though the access times vary for different types of memory.

Computation Paradigm

General-purpose on GPUs (GPGPU) [58] is designed to follow the general flow of the graphics pipeline (consisting of vertex, geometry and pixel processors - see Figure 3.1), with each iteration of the solution being one rendering pass. The CUDA hardware interface (see Figure 3.2) attempts to hide all these notions by presenting a program as a collection of threads running in parallel. The elements for this approach are:

- A **warp** is a collection of threads that can actually run concurrently (no time sharing) on all of the multiprocessors. The size of the warp (32 on the G80 GPU) is less than total available cores (128 on G80) due to memory access limitations. The programmer decides the number of threads to be executed, but if there are more threads than the warp size, they are time-shared.

- A **block** is a group of threads that are mapped to a single multiprocessor. Since each multiprocessor has multiple cores (8 on the G80) and a shared memory, threads in a block are executed together and can efficiently share memory. All threads of a block executing on a single multiprocessor divide its resources equally amongst themselves, with each thread and block having a unique ID accessed during its execution to process different sets of data in a SIMD (Single Instruction Multiple Data) fashion.

- A **kernel** is the core code to be executed on each thread, which performs on different sets of data using its ID. The CUDA programming model does not allow you to

select a different kernel to be executed on each of the multiprocessors. The hardware architecture, however, allows multiple instruction sets to be executed on different multiprocessors, so this may be simulated using conditionals.

- A **grid** is a collection of all blocks in a single execution. That way, a program is decomposed into kernels, each implemented through a grid which is composed of blocks consisting of threads (see Figure 3.3).

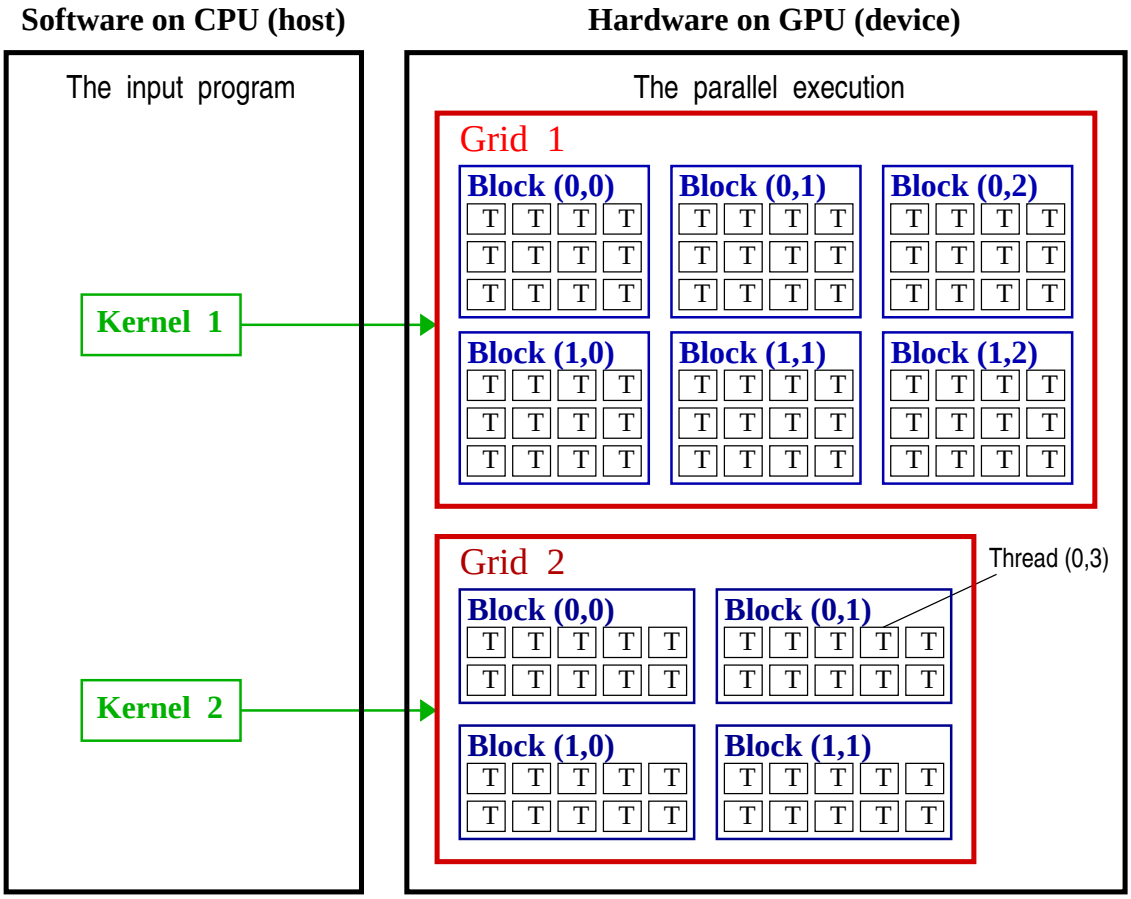


Figure 3.3: The CUDA programming model. In this example, a program is decomposed into two kernels, each implemented through a grid, with the first grid composed of 2x3 blocks, each containing 3x4 threads executed in a SIMD fashion.

Table 3.1: Major limitations for the CUDA programming model on the Nvidia G80 GPU used during the experimental study. The last column assesses its importance according to the impact on the programmer's job and overall performance.

Parameter	Limit	Impact
Multiprocessors per GPU	16	Low
Processors / Multiprocessor	8	Low
Threads / Warp	32	Low
Thread Blocks / Multiprocessor	8	Medium
Threads / Block	512	Medium
Threads / Multiprocessor	768	High
32-bit registers / Multiprocessor	8192	High
Shared Memory / Multiprocessor	16 KB	High

A single block should contain 128-256 threads for an efficient execution. The maximum possible thread total is 512. Other hardware limitations are listed on Table 3.1, where they have been ranked according to impact on the programmer's job and overall performance based on experience.

Memory and registers

In CUDA, all threads can access any memory location, but as expected, performance will increase with the use of closer shared memory whenever data to be collectively read by threads within a block belong to different memory banks. The use of shared memory is explicit within a thread and cannot exceed 16 Kbytes. Optimizations using shared memory may speed-up the code up to a 10x factor for vector operations, and latency hiding up to 2.5x [59]. Other performance issues are summarized in the last two rows of Table 3.2.

The role of 32-bit registers becomes more important as a limiting factor for the amount of parallelism that can be exploited, rather than as the conventional mechanism to hide

Table 3.2: Constraints in memory addressing (first five rows) and maximum performance (last two rows) reached by the CUDA programming model in its latest version (1.1, as of December 2007).

Parameter	Value
Constant memory / multiprocessor	64 KB.
Maximum sizes of each dimension of a block	512x12x64
Maximum sizes of each dimension of a grid	64K x 64K x 1
CUDA maximum memory pitch	256 KB.
CUDA texture alignment	256 bytes
Geometrical performance	$3*10^8$ triangles/sc.
Fill-rate (textural performance)	$192*10^8$ texels/sc.

memory latency. A multiprocessor contains 8192 registers, each owned exclusively by a thread. Registers should be split among the threads so that the number of threads created reaches the maximum occupancy on each multiprocessor given the constraints outlined in Tables 3.1 and 3.2. For example, if a thread consumes 10 or fewer registers then an implied 819 threads may be used, but only 768 are allowed on a multiprocessor and only 512 are allowed for a block: A possible solution is to build 3 blocks of 256 threads each. Reversely, if a thread consumes 16 registers, a maximum of 512 threads is allowed (512x16=8192), and all threads may belong to a single block.

Developing in CUDA

A typical CUDA development cycle is as follows. First, the code is compiled using a special CUDA compiler that outputs the hardware resources (registers and local shared memory) that are consumed by the kernel. Using these values, the programmer determines the number of threads and blocks that are needed to use a multiprocessor efficiently. If a satisfactory efficiency cannot be achieved, the code needs to be revised to reduce the memory

foot print (registers and local shared memory). Due to the high FLOPS performance of the streaming processor, memory access becomes the bottleneck in the registration algorithm.

3.3 Image registration on the GPU

The workflow for the registration algorithm is summarized in Figure 3.4. From a performance perspective, the most interesting phase is the set of Fast Fourier Transforms (FFTs) used to compute the normalized cross-correlations for precise point matching in nonrigid registration, since they entail most of the execution time. For example, in experiments registering a pair of 23K x 62K images on the CPU, represented in Figure 3.5, more than 60% of the total running time is spent in computing normalized cross-correlation.

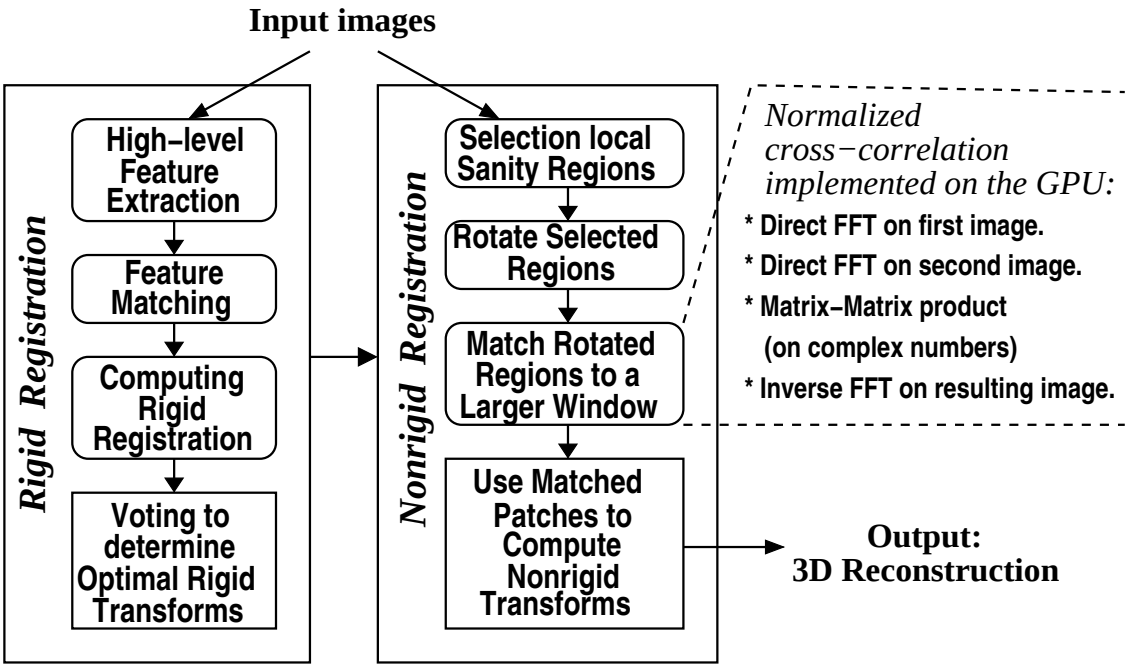


Figure 3.4: The workflow for the two stage image registration algorithm as implemented on GPU. Rounded boxes are independent local operations that can be straightforwardly carried out in parallel. The most computationally demanding phase is selected to run on the GPU for a much faster execution.

Table 3.3: Percentage weight on average for each of the computational stages before and after porting to the GPU.

Computational phase	CPU	GPU
Two forward FFTs	68%	74%
Point-wise multiplication	3%	2%
One inverse FFT	29%	24%

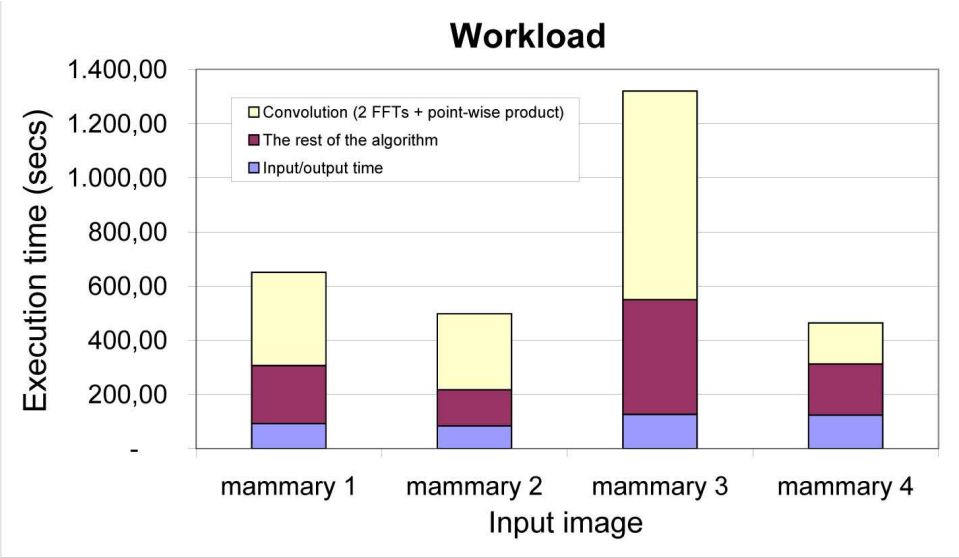


Figure 3.5: Workload of each phase of the two stage registration algorithm.

This process is optimized by implementation on the GPU (see Section 3.3.1 below), including the two forward FFTs and the subsequent inverse. The point-wise multiplication of FFT spectra which is required between the forward and inverse transforms was also implemented on the GPU to save data movement between processors and to take advantage of higher arithmetic intensity versus computation on the CPU (see Table 3.6). Table 3.3 depicts the percentage weight for these operations on each platform.

The remaining parts of the registration algorithm including voting, variance calculations, and simple transformations (e.g. rotating) did not show any significant speed-up on the GPU for three major reasons:

1. They were already computationally cheap on the CPU.

2. More importantly, it was remarkable how much time was required to ship code and data back and forth between the CPU and the GPU through the memory bus, hyper-transport link, and PCI-express controller (see Figure 3.7). This cost could not be amortized during the subsequent computation despite the high GFLOPS rate.

3. Most of these operations contain conditionals and are not arithmetic intensive, which makes them more appropriate for the CPU processor. Additionally, this enables the bi-processor platform to achieve a more balanced execution.

3.3.1 Normalized cross-correlation using CUDA

Normalized cross-correlation can be efficiently implemented on the GPU using the CUDA programming model. The computation strategy is based on the theorem that circular convolution in geometric space amounts to point-wise multiplication in discrete frequency space. This way, using the CUFFT library [60] as an efficient direct/inverse Fast Fourier Transform implementation, Fourier-based correlation can be more efficient than a straightforward spatial domain implementation, and permits leveraging of the floating-point power and parallelism of the GPU without having to develop a custom GPU-based implementation.

The FFT is a highly parallel ”divide and conquer” algorithm for the computation of the Discrete Fourier Transform of single or multidimensional signals. The convolution theorem

applies to an image (search window) and convolution kernel (template window) that share the same sizes. In cases where the image is bigger than the kernel, such as the matching of a template within a larger search area, the kernel has to be expanded to the image size as shown in Figure 3.6. Also, ordinary convolution requires the template and search windows to be padded with zeros on the bottom and right borders as anticipated in Section 2.2.2.

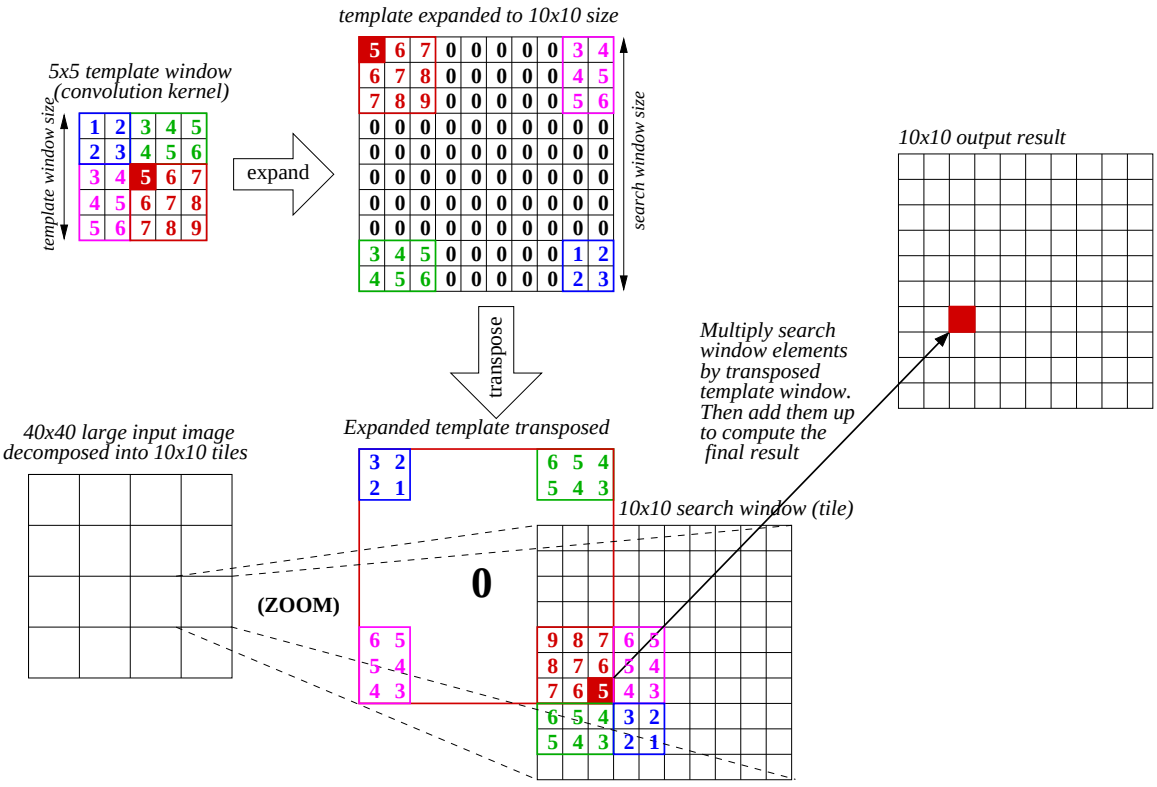


Figure 3.6: The computation of FFT-based normalized cross-correlation. The template window has to be expanded to the search window size and convolution with the expanded kernel is equivalent to the one with the initial kernel. The example is shown for a large image having 40x40 pixels and decomposed into 4x4 tiles, thus resulting a search window of 10x10 pixels. The template window has 5x5 pixels, half of the search window size as in the registration algorithm.

Table 3.4: Template (feature) and search window sizes (in pixels). An evaluation about whether those sizes contribute to perform further optimizations in the corresponding CPU and GPU codes is included, considering the libraries used during the implementation: FFTW on the CPU and CUFFT on the GPU. (*) This slot is partially in favour of the GPU because 749 is a multiple of seven, a small prime number.

Input image:	Placenta: 16K x 16K			Mammary: 23K x 62K		
Window size:	Small	Medium	Large	Small	Medium	Large
Template window (in pixels)	171	250	342	342	500	683
Search window (in pixels)	342	500	683	683	1000	1366
Aggregate (template+search-1)	512	749	1024	1024	1499	2048
CPU friendly (FFTW library)	Yes	No	Yes	Yes	No	Yes
GPU friendly (CUFFT library)	Yes	(*)	Yes	Yes	No	Yes

The 2D-FFT dimensions are fundamental in CUDA for optimizing performance. When the template and search window are multiples of either a power of two or a small prime factor, the memory footprint generated by the CUDA algorithm minimizes conflicts accessing banks on shared memory and performance increases. For the counterpart C++ implementation on the CPU the FFTW [48] was used, one of the most popular and efficient CPU-based FFT libraries, for a fair comparison with the GPU results. FFTW also favours certain 2D-FFT dimensions, and the optimal cases arise when the sum of the template window and the search window sizes minus one is a power of two. With a careful selection of FFT dimensions, a benchmark was created that fulfills most of these rules on both CPU and GPU implementations. Table 3.4 summarizes all sizes selected for experimental evaluation and evaluates their adequacy for each type of processor.

For the cases in which the data size cannot fulfill the previous rules, FFTW and CUFFT provide a simple configuration mechanism called a *plan* that completely specifies the optimal - that is, the minimum floating-point operation - plan of execution for a particular

FFT size and data type. The advantage of this approach is that once the user creates a plan, the library stores on file whatever state is needed to execute the plan multiple times, thus avoiding the penalty of carefully planning the transforms at run-time. For example, with a template window equal to 350x350 pixels and a search window equal to 700x700 pixels, FFTW takes around 0.7 seconds, whereas the pre-planned computation takes only 0.32 seconds with a previous 6 seconds penalty required to pre-compute the plan (a cost which can later be amortized by loading the plan at run-time on subsequent 2D transforms of the same size).

3.4 Experimental Setup

3.4.1 Input data set

The multi-socket GPU implementation was applied to a series of microscopic images derived from consecutive sections of (1) mouse placenta for a morphometric study on the role of the retinoblastoma gene and (2) mammary gland for studying the breast cancer tumor microenvironment [22]. For details about these sets of images, see Table 3.5. The goal in both cases is to reconstruct 3D tissue models for the study of microanatomy.

Table 3.5: The set of images used as input data sets for our registration algorithm.

Field of study	Research area and biomedical goals	Mouse source	Computational workload	Image size (pixels)	Number of slides
Genetic	Role of a gene	Placenta	Medium	16K x 16K	100
Oncology	Breast cancer tumor	Mammary	Large	23K x 62K	4

Table 3.6: Summary of the major features of the high-end GPU from Nvidia.

GPU feature	Value
Model	Quadro FX 5600
Core clock frequency	600 MHz
Stream processors clock	1.35 GHz
Manuf. technology	90 nm

Video memory feature	Value
Clock frequency	1.6 GHz
Bus width	384 bits
Bandwidth	76.8 GB/sc
Memory size	1.5 GB

3.4.2 Hardware

The multisocket GPU implementation was executed on a GPGPU visualization node where the features of dual-core AMD Opteron 2218 CPU are combined with dual-socket high-end Nvidia Quadro FX 5600 GPU (see Figure 3.7). The CPU is endowed with 4 GB of DDR2 DRAM running at 667 MHz, whereas each of the dual GPUs contains 1.5 GB of on-board GDDR3 DRAM at 1600 MHz (see remaining features in Table 3.6). This leads to a total available DRAM memory of 7 GB. The system is completed with a 750 GB, 7200 RPM local SATA II hard disk with 16 MB cache and an InfiniBand card for communication purposes.

In the experiments, the time for reading the input images from file is not considered. This time can be partially hidden by overlapping I/O communications with internal computations on the GPUs due to the asynchronous communications supported within CUDA 1.1. In addition, it has been observed that shared I/O due to other cluster users slightly affects the computational time. To minimize this variation, several runs were performed for each experiments, taking the average among all of them.

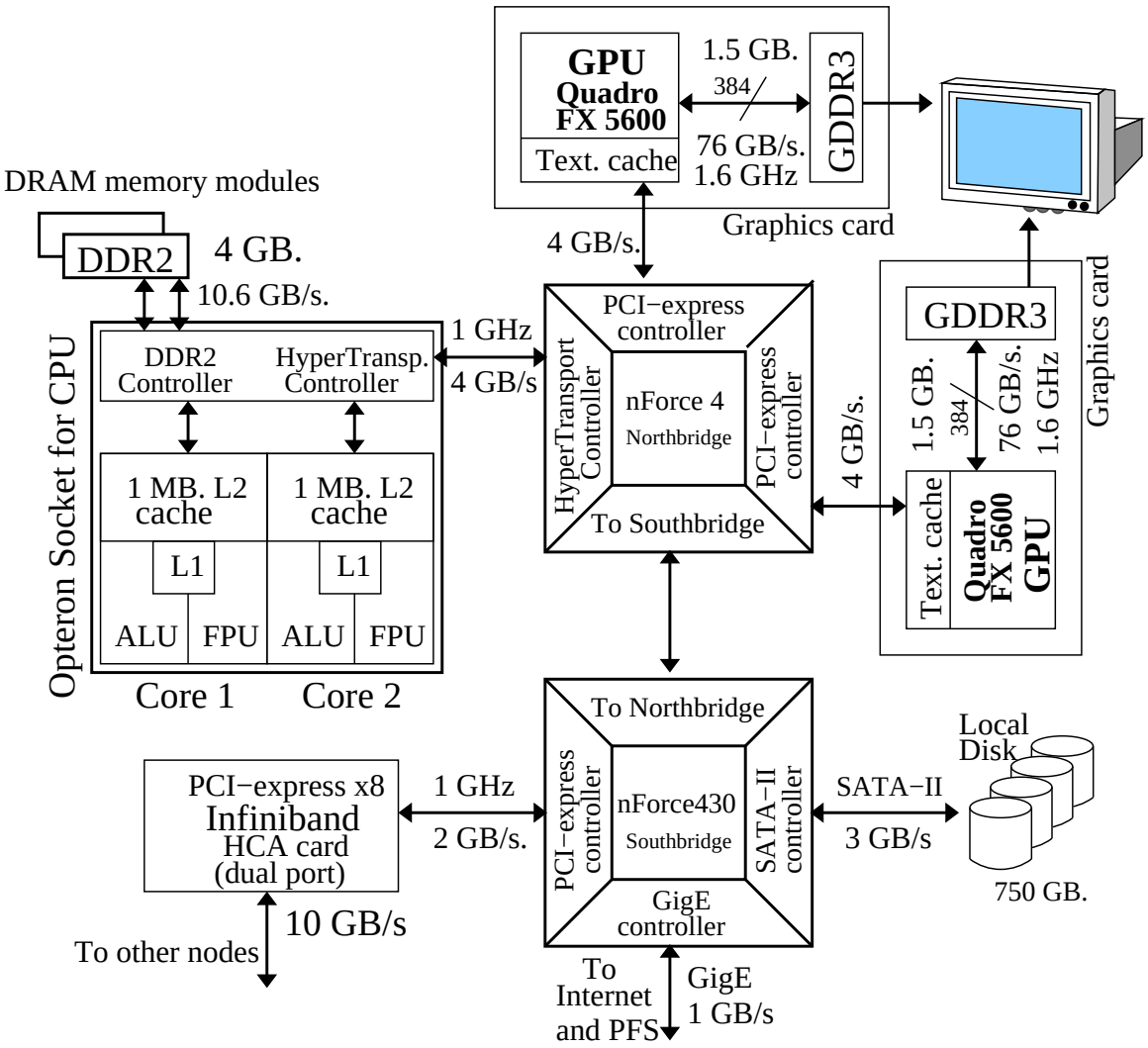


Figure 3.7: The block diagram for a single computing node, which integrates one CPU and two GPUs.

3.4.3 Software

The GPU was programmed using the CUDA Programming Toolkit, version 1.1 (December, 2007), and for the cases where we used two GPUs, *pthreads* were used to run the code on each GPU.

On the CPU side, we used the Microsoft Visual Studio 2005 8.0 C++ compiler. Matlab 7.1 was also used to validate the results from our implementation as well as to provide the departure sequential execution time.

3.5 Empirical Results

A broad number of experiments were conducted on one hundred images in the placenta image set and four images for the mammary image set as reflected in Table 3.5.

3.5.1 Characterizing the workload

A preliminary issue to mention is that the execution time for each slide within the same working image set experiences variations due to the content and consequentially the different number of features processed. As described in Section 2.2.2, the variance is computed on a 200×200-pixel window to retain only feature points that are meaningful. This may lead images of similar sizes to produce different workloads based on their contents (the more homogeneous an image is, the less computation required). Table 3.7 summarizes the number of features extracted for each input image belonging to the mammary data set as well as the total and computational time required for the registration algorithm to be completed on an Opteron CPU.

The percentage of features processed ranges from 4% to 30% of total image area, with those percentages varying slightly when using small, medium or large window sizes (see

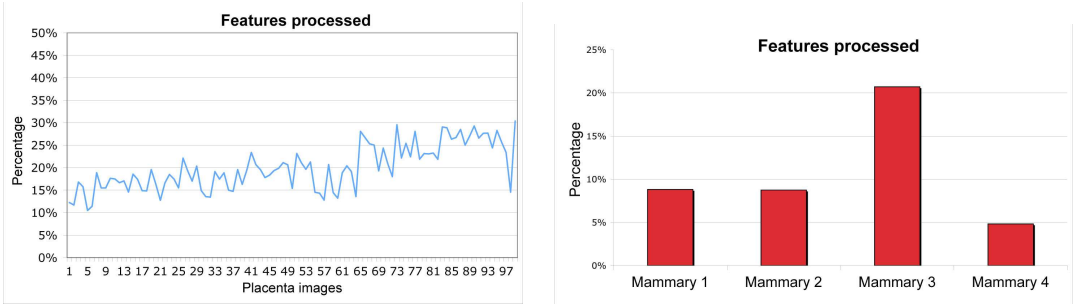
Table 3.7: Workload breakdown on single CPU for mammary image set. The number of features extracted for each input image within the mammary data set differs due to content. Execution times in the last two columns represent the large case.

	Number of features extracted			Workload on CPU (in seconds)	
Window size: (template,search)	Small (342,683)	Medium (500, 1000)	Large (683, 1366)	Execution time with I/O	Execution time without I/O
Mammary 1	1196	655	384	650.86	558.54 (85%)
Mammary 2	1048	568	312	497.83	414.17 (83%)
Mammary 3	3119	1528	854	1320.01	1192.69 (90%)
Mammary 4	690	322	168	463.77	340.62 (73%)

Table 3.4). However they may consider as stable for each image if the smaller window size is selected as the most representative (higher search resolution). Under this assumption, Figure 3.8 provides details about the percentage of features processed for the placenta and mammary image sets: For the placenta images the minimum percentage corresponds to image 5 with 10.48% and the maximum to image 99 with 30.38%, and a total average of 19.88%. For the mammary gland images the minimum percentage is 4.82% by image 4, with a maximum of 20.71% by image 3, and an average of 10.77%. According to our definition of feature, the placenta image set containts nearly double the density of meaningful information. While the mammary gland set is a larger image, it represents a higher rate of sparsity.

3.5.2 Execution times on the CPU

Figure 3.9 presents the execution time for the registration algorithm depicted in Figure 3.4 when it is entirely computed on the CPU using the FFTW library. The results for the placenta image set are shown on the left, mammary on the right. Within each case, experiments were run for three different template and search windows (see Table 3.4):



(a) Placenta image set.

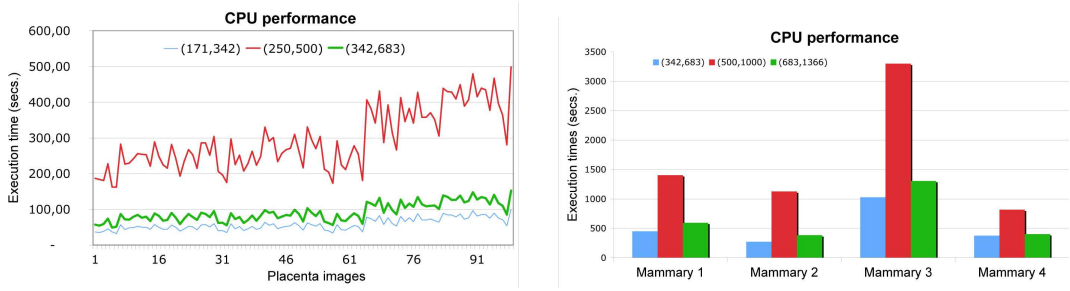
(b) Mammary image set.

Figure 3.8: Percentage of features processed per image on each input image set. The small template and search window size was selected as the most representative.

small (blue, leftmost), medium (red, center) and large (yellow, rightmost). According to the details provided by the FFTW library, the small and large sizes fulfill optimal conditions, whereas the medium size breaks all rules (from now on, this case will be referred to as *non-compliant*). This has a major impact on the execution time, with an average time for the placenta case of 294.57 seconds using the medium size versus 57.97 seconds in the small case and 91.33 seconds in the large one. This results in an increment of 57% when the windows are doubling size within optimal conditions and an additional 222% when using non-compliant sizes. Mammary offers a similar behavior, though the last two overheads are reduced to 26% and 147% respectively.

3.5.3 Execution times on the GPU

Figure 3.10 shows execution times for the registration algorithm when the GPU helps the CPU by computing the FFT-based cross-correlation using CUDA. The left side represents the placenta image set and the right side the mammary image set, with the legend differentiating the small, medium and large window size cases (see Table 3.4). This time,



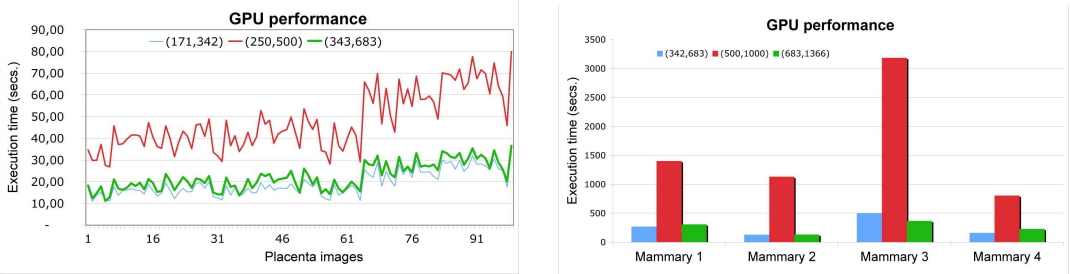
(a) Placenta image set (b) Mammary image set.

Figure 3.9: Execution times on the CPU Opteron for the registration algorithm on a pair of images under different image sets and window sizes. The first pair of numbers on chart legends corresponds to the small window sizes (template and search window, respectively), then medium and finally large sizes. For placenta, average times are 57.97 seconds (small), 294.57 seconds (medium) and 91.33 seconds (large). For mammary, average times are 530.41 seconds (small windows), 1660.91 seconds (medium) and 669.96 seconds (large).

the small and large sizes fulfill all conditions imposed by the CUFFT library and also the medium search window size of 749 pixels satisfies being a multiple of a small prime number (7). Nevertheless, its overhead is still significant. The average times for the placenta case are 19.27 seconds (small), 47.80 (medium) and 22.22 seconds (large), and the slowdown is of 15% when the windows are doubling size within optimal conditions and an additional 115% for the non-compliant case. For mammary, the large sizes perform slightly better than the small ones, and the non-compliant overhead (medium size) reaches the top: 531%.

3.5.4 CPU-GPU comparison

The central row in Table 3.8 reports the average speed-up factors on the GPU when helping to compute the FFT-based cross-correlation using CUDA. Gains are unstable for the non-compliant cases, and the most realistic results are the small and large cases where



(a) Placenta image set.

(b) Mammary image set.

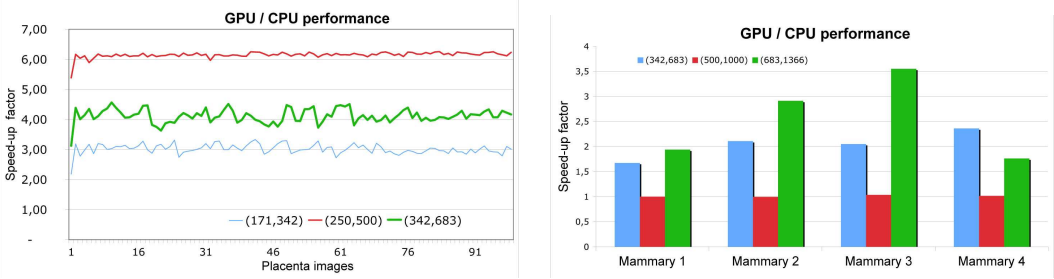
Figure 3.10: Execution times on the GPU Quadro for the registration algorithm on a pair of images under different image sets and window sizes. The first pair of numbers on chart legends corresponds to the small window sizes (template and search window, respectively), then medium and finally large sizes. For placenta, average times are 19.27 seconds (small), 47.80 seconds (medium) and 22.22 seconds (large). For mammary, average times are 264.09 seconds (small windows), 1629.72 seconds (medium) and 257.95 seconds (large).

window sizes strictly follow the guidelines provided by the FFTW and CUFFT libraries. For the placenta image set, small windows produce a three times acceleration factor and large windows extend gains to reach 4.11x. For the mammary image set, those gains are more modest: 2.00x and 2.59x, respectively.

Figure 3.11 demonstrates that the improvement factor on the GPU depends much more on the input image when using mammary rather than placenta, where numbers are more consistent. Additionally, gains are more volatile when increasing the window sizes. This is because the image contents become more heterogeneous on a larger search, showing also higher disparities among images. This effect is corroborated in Figure 3.8.

3.5.5 Parallelism and scalability on the GPU

The GPU has gained popularity as an outstanding scalable architecture over the past decade, being able to succeed in its goal of sustaining performance doublings every six



(a) Placenta image set. (b) Mammary image set.

Figure 3.11: Comparison between the GPU and CPU execution time in terms of GPU speed-up factor. When the window sizes increase, times are more irregular in (b). The first pair of numbers on chart legends corresponds to the small window sizes (template and search window, respectively), then medium and finally large sizes. For placenta, the average speed-up is 3.00x (small), 6.16x (medium) and 4.11x (large). For mammary, the average speed-up is 2.00x (small windows), 1.01x (medium) and 2.59x (large).

Table 3.8: Execution times (in seconds) and speed-up factors for the different implementations developed for computing our registration algorithm on a pair of images with maximum performance. The average of all 100 and 4 runs is reported for the placenta and mammary image sets. Boxed numbers highlight the GPU speed-up under the most typical scenarios.

Input image set:	Placenta: 16K x 16K			Mammary: 23K x 62K		
Window size:	Small	Medium	Large	Small	Medium	Large
(template,search)	(171,342)	(250,500)	(342,683)	(342,683)	(500,1000)	(683,1366)
CPU exec. time	57.97	294.57	91.33	530.41	1660.91	669.96
GPU exec. time	19.27	47.80	22.22	264.09	1629.72	257.95
GPU speed-up	**3.00x**	**6.16x**	**4.11x**	**2.00x**	**1.01x**	**2.59x**
2 GPUs time	13.13	26.05	13.66	225.17	837.51	234.62
2 GPU / 1 GPU	**1.46x**	**1.83x**	**1.62x**	**1.17x**	**1.94x**	**1.09x**
2 GPU / 1 CPU	**4.41x**	**11.30x**	**6.68x**	**2.57x**	**1.98x**	**2.85x**

months. In addition to this intra-chip trend, other initiatives like SLI from Nvidia and Crossfire from ATI have emerged to explore inter-chip parallelism (SMP - Symmetric Multi-Processing). The initiative has achieved a remarkable success within the video-game industry, but so far has not been explored for general-purpose computing to our knowledge.

This section evaluates the performance of our registration algorithm on a pair of GPUs when applying SMP parallelism. Our programming techniques are straightforwardly extensible to higher number of graphics cards, and the methods used for partitioning the problem guarantees excellent scalability beyond that point. Nevertheless, in this ambitious project a warning against the critical role assumed by the input/output system is necessary: Dozens or even hundreds of GPUs working in parallel can find an easy way of distributing different search windows efficiently when working on large-scale input images, but there must be a high-performance file system able to read the image tiles in parallel at a sustainable bandwidth high enough to provide data to be processed over the Teraflop rate. During experiments this bottleneck was not investigated on a larger number of GPUs. Table 3.7 quantifies in its last two columns the execution time (including input/output) and the computational time (excluding I/O) to reveal that I/O is responsible of 10-20% of the total execution time. This time has not been included in our subsequent analysis since it is the same for both the CPU and the GPU-optimized versions of our registration algorithm, and I/O is out of the scope of this work. This implicitly assumes that image data are available in DRAM memory or that they can be retrieved efficiently from file using either a parallel file system or a RAID system.

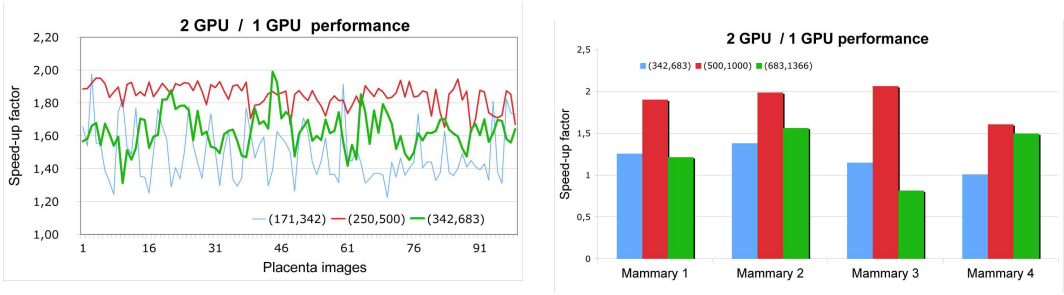
Once data reaches the CPU, there are two basic ways of distributing the workload among multiple GPUs in our registration algorithm: BLOCK or CYCLIC. For the particular case of a pair of GPUs (but without losing any generality), BLOCK assigns the

upper half of an image to the first GPU and its lower half to the second GPU. CYCLIC, on the contrary, numbers image tiles and assign even tiles to the first GPU and odd tiles to the second GPU. Because interesting image features tend to be spatially concentrated, BLOCK presents higher potential risk for an unbalanced data partitioning, so CYCLIC was selected for all experiments.

Table 3.9: Number of windows processed and discarded for each image within the mammary image set on each GPU under the two GPUs parallel execution. Workload unbalance and execution time are shown in the last two columns. The search window size here is 684x684 pixels.

Input image	Graphics processor	Number of windows tested	Number of windows processed/discarded	Workload unbalance	Execution time (secs.)
Mammary 1	GPU 1	1672	196/1476	4.08%	260.41
	GPU 2	1672	188/1484		
Mammary 2	GPU 1	1496	158/1338	2.53%	101.32
	GPU 2	1496	154/1342		
Mammary 3	GPU 1	1872	428/1444	2.76%	522.43
	GPU 2	1911	426/1485		
Mammary 4	GPU 1	1786	78/1708	13.33%	225.37
	GPU 2	1786	90/1696		

The parallelization method works the following way: a thread is created for each image region (tile) which computes the variance on a given CPU to assess whether it is worth computing. If the tile passes this test, it is sent to a predetermined GPU to compute the normalized cross-correlation and search for features. Table 3.9 outlines the number of tiles processed and discarded on each GPU depending on the input image used from the mammary data set. Workload unbalances range from 2.76% on image 2 to 13.33% on



(a) Placenta image set.

(b) Mammary image set.

Figure 3.12: GPU scalability. Improvement factor when enabling a second GPU. The first pair of numbers on chart legends corresponds to the small window sizes (template and search window, respectively), then medium and finally large sizes. For placenta, the average speed-up is 1.46x (small), 1.83x (medium) and 1.62x (large). For mammary, the average speed-up is 1.17x (small windows), 1.94x (medium) and 1.09x (large).

image 4, always growing for lower number of tiles to process (sparsity rate of the input image).

Finally, Figure 3.12 shows that gains produced when enabling a second GPU are very diverse, starting with 30-50% on small window sizes, continuing with 60% on large window sizes and ending with an optimal scalability (100% gain) on medium sizes. Those gains are proportional to the computational workload, showing that the GPU is a more scalable processor when it can exploit its arithmetic intensity. In other words, GFLOPS are not limited by data shortages coming from insufficient bandwidth between the video memory and the GPU.

3.5.6 Summary and conclusions

Several conclusions can be drawn from our experimental analysis:

1. The placenta image set shows higher speed-up factors on the graphics platform. This is because the images have a larger portion of meaningful content, leading to a denser

workload which exploits its arithmetic intensity and memory bandwidth better. Also, a lower number of features processed means higher presence of conditionals in the code, one of the most harmful instructions for GPU performance.

2. The placenta image set is more scalable on multiple GPUs, and gains are more stable among different window sizes. The higher sparsity of the mammary images plays a negative role in the workload distribution, introducing unbalances and preventing parallelism from being fully exploited.

Overall, the GPU achieves a 3-4 speed-up factor in the most typical scenarios (boxed slots in Table 3.8) versus the CPU, and a pair of GPUs show a satisfactory scalability but unstable gains under different image sets and window sizes.

3.6 Related Work

Large scale image registration has many applications in both biomedical research [26, 37, 61] and geophysics [62]. However, there are currently few works addressing image registration algorithms intended to run efficiently on high performance computing (HPC) environments.

The work on parallel image registration on multicomputers is limited [37] and is restricted to either large computer clusters [63–65] or IBM cell clusters [66]. Clusters of GPUs have been used to implement other heavy workload tasks [67], mostly within the simulation and visualization fields. For example, numerical methods for finite element computations used in 3D interactive simulations [68], and nuclear, gas dispersion and heat shimmering simulations [69].

On the other hand, commodity graphics hardware has become a cost-effective parallel platform to implement biomedical applications in general [55]. One work similar to the

application in this dissertation on the registration of small radiological images [70], and others within the fields of data mining [71], image segmentation and clustering [72] have applied commodity graphics hardware solutions. Those efforts have reported performance gains of more than ten times [56] but were mostly implemented using shaders with the Cg language [56].

The present work enhances the graphics implementation through CUDA [57] which exploits parallelism to a wide variety of layers. The combined implementation of CPU and GPU on a bi-processor platform is one step ahead in performance and provides the first parallel processing solution on large microscopic images for users without requiring an expensive multiprocessor.

In 2004 it was reported that on real numbers, the MxM product may run slower on the GPU due to the lack of high bandwidth access to cached data [73]. The same set of operations that is described for the correlation phase (two direct FFTs, point-wise multiplication in frequency space, and a inverse FFT) took 0.625 seconds on a 2003 Intel Xeon CPU for a 1024x1024 matrix, versus 2.7 seconds on a counterpart GeForce 5 GPU [74]. This situation is reversed in 2008 for two major reasons:

1. On the software side, the CUDA programming model makes explicit the use of shared memory, which overcomes the lack of high bandwidth access to closer data.

2. On the hardware side, the higher scalability of the GPU is exploited, doubling performance every six months during the present decade versus the 18 month period that takes the CPU that achievement [59].

GPUs for general-purpose computations are an emerging field evolving quickly within computer architecture. Tesla [75] is the latest and more powerful contribution from Nvidia

to this area, offering multiple GPUs without video connectors into either a board or a desk-side box to reach near supercomputer levels of single-precision floating-point operations at a cost starting around $1500 (a price similar to the Quadro FX 5600 used during experiments). At a lower price range, there have been recent announcements on double precision graphics architectures from Nvidia (GeForce 9 Series) and ATI (FireStream - see [76]) to provide a definitive solution to software requiring high-precision arithmetic in floating-point operations.

3.7 Discussion and Conclusions

With the advances in imaging hardware, tasks like the nonrigid registration of large images with billions of pixels become increasingly popular, evolving towards computationally demanding algorithms for which parallel and scalable solutions become essential. Within this scope, the contribution of the work in this chapter is twofold:

- First, the two stage algorithm provides a parallelizable method for registration which has been successfully applied to biomedical studies for reconstructing the 3-D structures of biological specimens with micron resolution. While the algorithm is motivated by biomedical applications, the principle of using high-level region features for rigid registration and using uniform sampling for nonrigid feature matching are ubiquitous for other applications.

- Second, a computational framework has been developed to expedite execution using graphics processors. A solid heterogeneous and cooperative multiprocessor platform is established using an AMD Opteron CPU and a pair of Nvidia Quadro GPUs, where the best features of each processor are fully exploited for applying higher degree of parallelism at a variety of levels: Multi-task for simultaneous executions of CPU and

GPU codes, SMP (Symmetric MultiProcessing) for multicard GPUs using pthreads, and SIMD (Simple Instruction Multiple Data) for the 128 stream processors of the GPU using CUDA.

The CUDA programming model exploits all the capabilities of the GPU as a massively parallel co-processor to achieve a remarkable speed-up factor as opposed to an expensive supercomputer. Experimental numbers show the success of these techniques, first by decreasing the execution time a 2-4x factor on a single GPU and later extending those gains to a pair of GPUs. For the genetic studies of a mouse placenta sample composed of 500 slides of $16K \times 16K$ pixels each, it takes more than 12 hours for serial C++ code to accomplish the registration process. This was reduced to less than 2 hours using two GPUs, and in addition, promising scalability was demonstrated for extending those gains easily on a large number of GPUs.

Overall, this study provides an illustrative example for how a graphics architecture in conjunction with its CUDA programming model may assist non-computer scientists by adapting grand-challenge biomedical applications to provide almost real-time response to pathologists in computer-aided methods.

CHAPTER 4

PARALLEL AUTOMATIC REGISTRATION OF LARGE SCALE MICROSCOPIC IMAGES ON MULTIPROCESSOR CPUS AND GPUS

During the present decade, emerging architectures including multicore CPUs and graphics processing units (GPUs) have gained popularity for their ability to deploy high computational power at a low cost. The effectiveness of emerging architectures was demonstrated in the previous chapter where GPUs with multi-socket parallelism were used to accelerate the two stage registration algorithm on a single computing node.

In this chapter I introduce another level of parallelism to the two stage high performance implementation, extending the single node methods for simultaneous execution to multiple nodes. Parallelization techniques from multiple levels are combined on a cooperative cluster of multicore CPUs and multisocket GPUs to apply their joint computational power to further reduce execution time for the two stage algorithm. As before, the two stage algorithm is analyzed to identify those parts that are more favorable to the CPU or GPU execution models and decomposed accordingly.

Performance results are presented for both the mouse placenta (16K $\times$ 16K pixels) and mouse mammary tumor (23K $\times$ 62K pixels) image datasets. Execution times are provided for different multi-node, multi-socket and multi-core configurations to provide

performance insights about the most effective approach. For a mouse mammary sample composed of 500 slides, more than 181 hours are required for a fully serial C++ code to accomplish the registration process on a high-end CPU. This time was reduced to less than 50 hours using a single GPU on a single node, and to 3.7 hours for a total speedup of 49× when 32 CPUs and GPUs participate in the cooperative environment.

4.1 Introduction

Light microscopy offers the desired field range and magnification for the study of many complex biological phenomenon, but acquiring 3-D information requires the reconstruction of sequences of very large images, often with hundreds of millions or billions of pixels each. The challenges of image size, rich feature environment, and nonrigid distortion and local morphological differences as described in Chapter 2.1 are addressed through solutions at both the software and hardware layers. The software layer solutions of fast rigid registration, refinement using correlations calculated with FFT, and single transform output were a good departure point but not sufficient by themselves for large scale applications. Chapter 3 introduced hardware layer solutions, using GPUs and multi-socket parallelism to accelerate the FFT calculations that bottleneck the two stage algorithm.

This chapter extends the hardware layer effort to multiple nodes, using node-level parallelism to address additional portions of the refinement stage beyond FFT calculation. Implementation issues and performance are studied on various high performance computing environments. Specifically, effort is split between two areas:

- **Parallel systems** based on a cluster of multisocket and multicore CPUs programmed using MPI (Message Passing Interface) [77].

- **GPUs** reoriented to general-purpose computing using CUDA (Compute Unified Device Architecture) [57].

Since each approach presents unique features for a high performance execution, the goal is to find cooperative scenarios where each resource is utilized at its peak while overcoming the weaknesses of the other. This way, performance can be compared on single nodes, CPU clusters, GPU clusters and a mixture of CPUs and GPUs in a cooperative execution.

This chapter is organized as follows: Section 4.2 provides details on the software tools and computing cluster hardware. Section 4.3 explains the implementation of the two stage algorithm on multiple nodes. Results and discussion are provided in Section 4.4. Conclusions about the high performance implementation are presented in Section 4.5.

4.2 Hardware and programming tools

4.2.1 The multiprocessor system at a glance

The two stage registration algorithm was implemented on a GPU equipped cluster, the BALE system at the Ohio Supercomputer Center (see Figure 4.1). The BALE supercomputer is endowed with 55 workstation nodes based on a dual-core Athlon 64 X2 architecture with integrated graphics card and 16 visualization nodes enhanced with dual-socket x dual-core AMD Opteron 2218 CPUs and dual-card Nvidia Quadro FX 5600 GPUs. All of these nodes are interconnected with Infiniband, and include a 750 GB, 7200 RPM local SATA II hard disk with 16 MB cache.

Experiments were run on the sixteen visualization nodes, where each node has 8 GB of DDR2 DRAM running at 667 MHz on the CPU side and 2x1.5 GB of on-board GDDR3

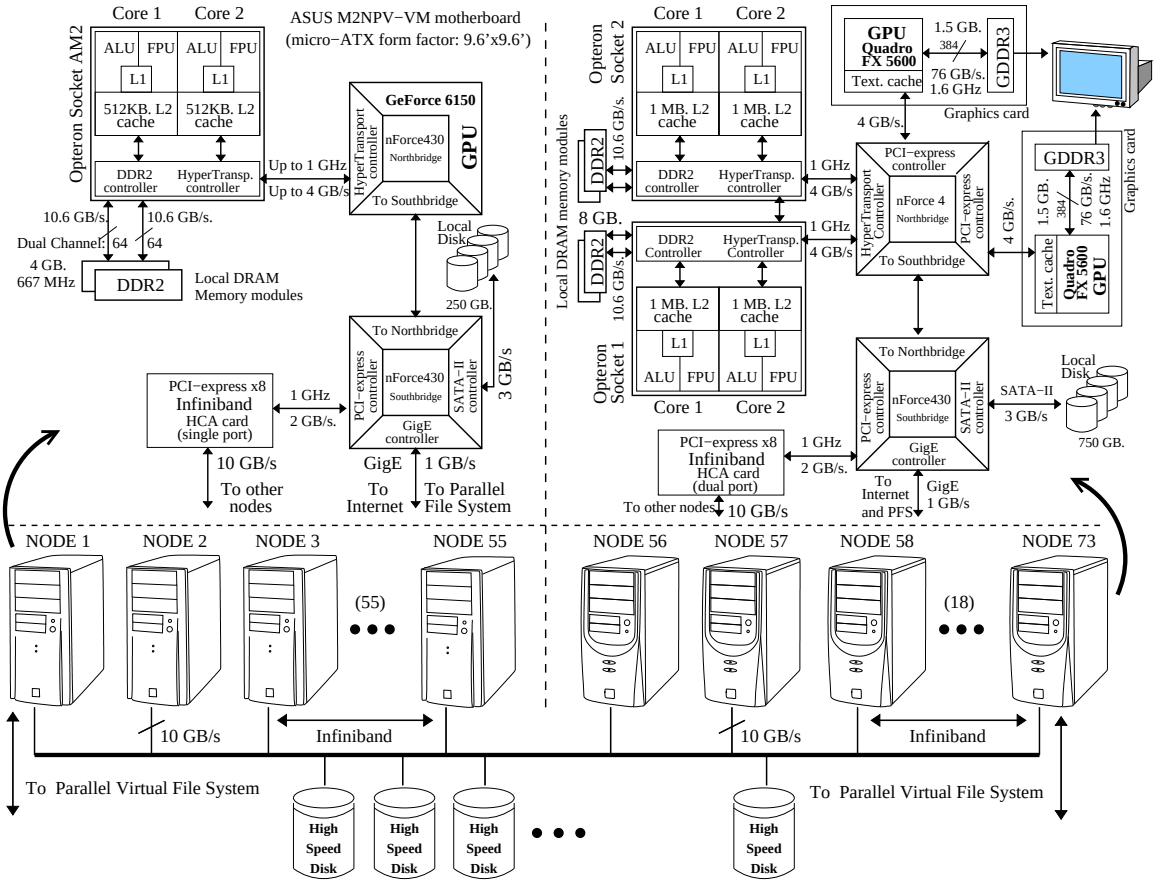


Figure 4.1: The BALE supercomputer at a glance.

DRAM running at 1600 MHz on the GPU side, for a total of 11GB available DRAM per node.

4.2.2 The CPUs: AMD Opteron X2 2218

On the CPU side, each BALE node of the visualization cluster consists of two Opterons X2 2218 composed of dual-core processors running at 2.6 GHz (see Table 4.1). Each core can fetch and decode three x86 instructions per cycle and execute 6 micro-ops per cycle.

Table 4.1: Summary of the major features of the high-performance multiprocessor nodes .

Hardware feature	CPU	GPU
Commercial model	AMD Opteron 2218	Quadro FX 5600
Clock frequency	2600 MHz	600 MHz
Sockets (SMP)	Dual	Dual
Cores (per socket)	Dual	128 stream procs.
Cache size (L1 & L2)	2x2x1MB.	2x 392 KB.
Cache latency (L1, L2)	3, 9 cycles	10 cycles
DRAM capacity	8 GB DDR2	2x 1.5 GB GDDR3
DRAM latency	138 cycles	200 cycles
DRAM data bandwidth	2x 10.8 GB/s	2x 76.8 GB/s
Peak processing power	2x 2x 4.4 GFLOPS	2x 330 GFLOPS

The cores support 128 bits SSE instructions in a half-pumped fashion, for a peak double-precision performance of 4.4 GFLOPS per core, 8.8 GFLOPS per socket, 17.6 GFLOPS per node and 35.2 GFLOPS in simple precision, for a total aggregate of 563.2 GFLOPS for the 16 visualization nodes in 32-bits arithmetic.

The Opterons contain two cores, each with a pair of 64 KB 2-way set associative L1 caches, a 1 MB 4-way L2 cache, and a dual-channel DDR2-667 memory controller as well as a single HyperTransport link to access the cache and memory of the other socket. Each socket can thus deliver 10.6 GB/s for an aggregate memory bandwidth of 21.3 GB/s per node.

4.2.3 The GPUs: Nvidia Quadro FX 5600

Figure 4.2 shows an outline of the G80 architecture, the baseline for the Nvidia Quadro FX 5600 GPU. Further details of the G80 architecture are provided in Table 4.2. Vertices and their attributes are the input to unified shaders (vertex, geometry and pixel), and later

processing is organized using multiple functional units working on data groups. Unified shaders are executed on 8 separated clusters, each containing 16 stream processors, 4 texture address units and 8 texture filtering units, together with a small L1 cache. This part is built on a hardwired design for a much faster clock frequency than the rest of the silicon area (1350 MHz versus 600 MHz), leading to a peak processing power of one third of a TFLOP. In the final stages of the graphics pipeline six partitions are responsible for the antialiasing, z-buffer and blending, whose results are finally written into video memory.

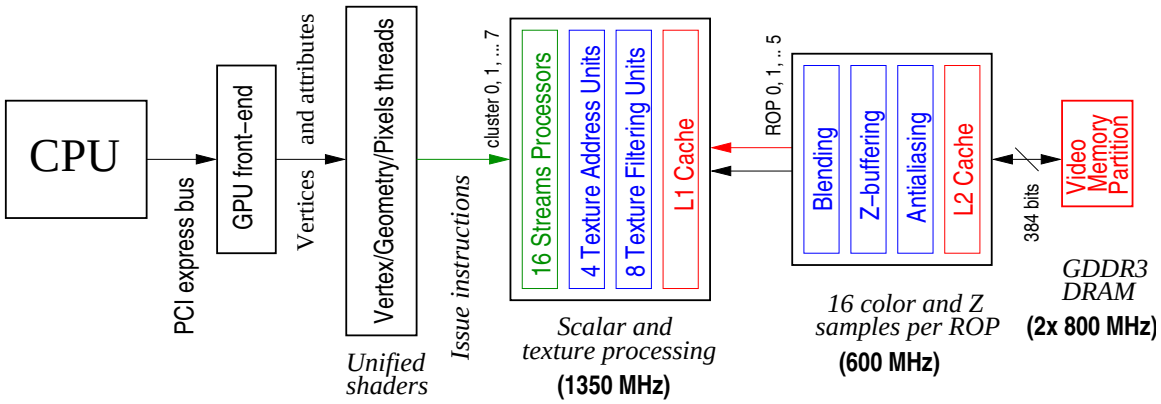


Figure 4.2: The graphics pipeline of the Nvidia G80 architecture.

From a graphics viewpoint, the G80 can be seen as a 4-stage graphics pipeline for shading, texturing, rasterizing and coloring. As a parallel architecture, however, the G80 becomes a SIMD processor equipped with 128 cores, and CUDA is the programming interface to use it for general purpose computing. From the CUDA perspective, cores are organized into 16 multi-processors (each cluster becomes 2 multi-processors with 8 cores), each having a set of 32-bit registers, constants and texture caches along with 16 KB of shared memory. At any given cycle, each core executes the same instruction on different

Table 4.2: Summary of the major features of our high-performance graphics card, the Nvidia Quadro FX 5600, together with its limitations when programmed with CUDA.

GPU feature	Value
Model	G80GL
Core clock frequency	600 MHz
Stream processors clock	1.35 GHz
Manufacturing technology	90 nm
Memory feature	**Value**
Memory clock	1.6 GHz
Bus width	384 bits
Bandwidth	76.8 GB/s
Size	1.5 GB
CUDA feature	**Value**
Constant memory	64 Kbytes
Shared memory per multiprocessor	16 Kbytes
32-bit registers per multiprocessor	8192
Max. no. threads per block	512 bytes

data, and communication between multiprocessors is performed through global memory (see Figure 3.2). The features for the Nvidia Quadro FX 5600 GPU are summarized in the last column of the Table 4.1, and the most important parameters for its programming with CUDA are given in Table 3.6.

4.2.4 CPU-GPU comparison

Four key issues are considered for maximizing performance of algorithms running on CPUs and GPUs:

1. **Parallelism:** CPUs are more popular on high-level parallelism like multi-nodes and multi-sockets (SMP - Symmetric MultiProcessing). GPUs are more aggresive on

inner parallelism, like multicores (128 cores or stream processors in the G80 architecture), SIMD (Single Instruction Multiple Data) and ILP (Instruction-Level Parallelism).

2. **Computational power:** The GPU capabilities for floating-point operations exceed 500 GFLOPS, in contrast with 10 GFLOPS for a 2007 state-of-the-art CPU. This advantage is a result of design for the color and position interpolations that are required for performance graphics applications.

3. **Memory access**: Spatial locality is best exploited with cache memory, which is around a thousand times larger in the CPU. Temporal locality, on the other hand, benefits the GPU, whose architectural rationale and programming model are inspired by the producer/consumer paradigm.

4. **Bus bandwidth**: A state-of-the-art 2007 graphics card delivers a peak performance exceeding 80 GB/sec. of memory bandwidth, as compared to 10 GB/sec. for the CPU. This is mainly due to its wider data path (384 bits, decomposed into six partitions of 64 bits).

4.2.5 Layers of parallelism

These features are combined to create a cooperative multi-processor platform where all the granularities of parallelism inherent in the architecture meet and are fully exploited in the two stage algorithm at different layers:

1. **Multi-node:** The outer layer, where MPI is used for data partitioning and communication across nodes.

2. **SMP:** The motherboard or inter-CPU layer, where MPI is also used for mapping processes to processors sharing the on-board DRAM memory.

3. **Multi-core:** The thread or intra-CPU layer, where pthreads are used as a software mechanism to decompose the program according to the number of cores available. Some multicore architectures have distributed all cache layers, while others share the most outer one.

4. **SIMD:** Used within CUDA to fully occupy the 128 stream processors of the GPU with a single code. These processors are grouped into 8 clusters of 16 cores sharing 16 KB of an internal shared memory.

5. **ILP:** The innermost layer, enabled by setting up CUDA blocks of computational threads on the GPU. These blocks partition the internal register data set available in the graphics processor.

4.2.6 Programming tools

Programming tools involved in the parallelization effort include MPI, Pthreads, C++, Matlab and CUDA.

- **MPI** Message Passing Interface is used for programming the BALE multiprocessor, or inter-node allocation and communication [77]. The MPI routines are callable from C++ code.

- **Pthreads** POSIX Threads are used for programming the multicore CPUs more explicitly [78]. This is an API for creating and manipulating threads which consists of a set of C programming language types and procedure calls.

- **C++** Microsoft Visual Studio 2005 8.0 C++ compiler was used for programming the CPU code. Multimedia extensions were enabled directly through HAL layer without any specific library in between.

- **CUDA** Finally, the GPU was programmed using the **CUDA** Programming Toolkit, version 1.1 (December, 2007).

4.3 Multiple Node Implementation

The two stage algorithm as implemented on the multiple node system is summarized in Figure 4.3. As with the single node implementation, the most demanding task of computing FFTs is carried out on GPU. In addition to the FFT calculations, other procedures of the nonrigid refinement stage are effectively parallelized at the node-level:

1. The process begins as the head node/socket loads the image pair from disk and distributes the RGB pixels to worker nodes/sockets for grayscale conversion. Multiple-buffering is used along with asynchronous communication to amortize disk operation.

2. With the grayscale representation of the image pair residing on the worker nodes, each node performs the $W \times W$ tile variance calculations on its portion of the base image, reporting the variance calculations along with the grayscale conversion results to the head node.

3. The qualified intensity features that meet the σ^2 threshold are evenly distributed among the nodes (including the head node) along with the corresponding portions of the grayscale images. The nodes compute the normalized cross correlation ρ and report the maximum correlation for each intensity feature along with the maximum

coordinates. Optionally additional cores are activated here as this is the most intensive of the procedures.

Again, as with the single node implementation the rigid initialization stage is fast to the extent that overall execution time would not benefit from GPU acceleration of node-level parallelization.

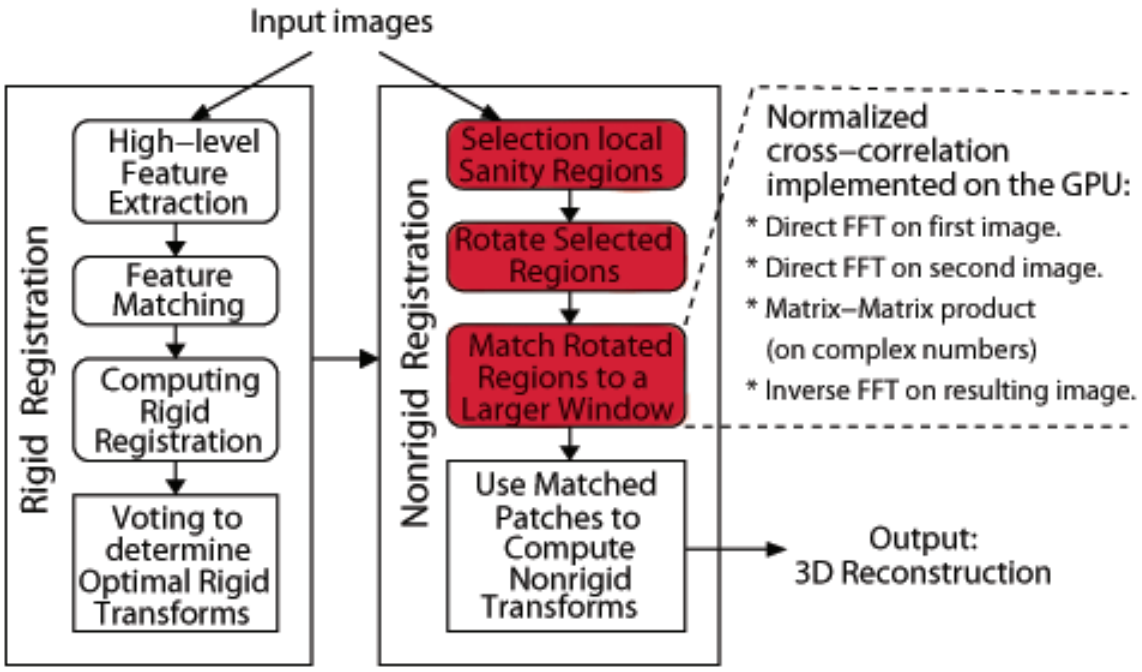


Figure 4.3: The workflow for the two stage algorithm as implemented on a cluster of GPU-equipped nodes. The most computationally demanding phase is selected to run on the GPU for a much faster execution. The highlighted operations are carried out in parallel at the node level.

4.4 Experimental results

The multiple node implementation of the two stage algorithm was applied to the benchmark dataset of mouse placenta and mouse mammary images described in Table 3.5. The

experiments of Table 3.4 were performed on the BALE cluster using the visualization nodes as computational resources and a enough number of additional frontend nodes playing an active I/O role so as not to introduce a bottleneck. From this point forward, all references to execution time are the total time without file I/O or conversion from RGB to grayscale. As before the average of several runs for each experiment is reported.

4.4.1 Workload

The execution time for the registration algorithm is sensitive to three parameters:

1. The image contents. The more features found on an image, the higher computational time. Figure 3.8 shows this variation between 10% and 30% for the small window size on the placenta data set.

2. The window size for feature search. The medium size is very unstable and by far the most demanding one in terms of computation (see Figure 4.4). The large window is more representative and stable, and with a time slightly higher than the small case, it will be the one chosen for parallel analysis on different numbers of BALE nodes.

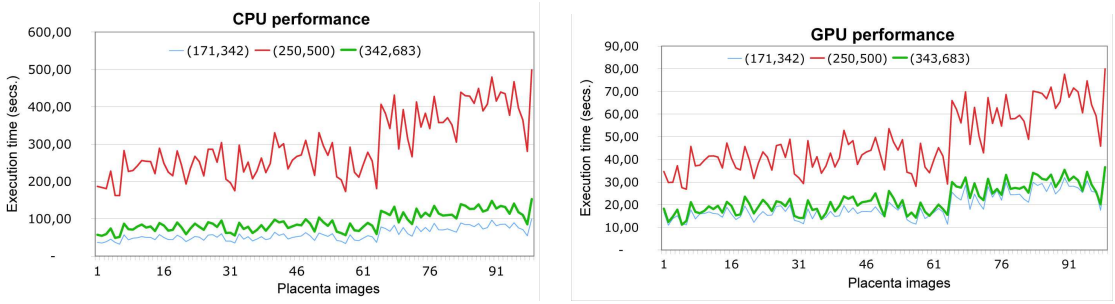
3. The input data set. Mammary images are around six times larger (see Table 3.5), but they contain approximately half of the feature density than the placenta images. This way, the computational time is expected to be around 4-5 times higher on mammary images.

Taking as departure point a fully serial C++ implementation, the placenta images each take approximately 110 seconds for execution. Moving to the high-end Opteron CPU on the BALE cluster and the mammary data set, the most computationally demanding case consumes 1304 seconds (see the longest bar in Figure 4.5). For an average of 500 images

such as required during a regular 3D reconstruction, this represents 181 hours of processing time, more than an entire week.

4.4.2 Single node analysis

Figure 4.4 compares the computation time on a single CPU node with a single GPU enabled to execute the convolution part of the registration algorithm for the placenta data set. The average speedup for the GPU-enabled versions are 3.00x on a small window size, 6.16x on the medium size and 4.11x on the large size.



(a) Execution times on the CPU. (b) Execution times on the GPU.

Figure 4.4: Execution times for the three window sizes depending on the input image on the placenta data set. (a) on the CPU and (b) for the combined CPU-GPU execution.

Figure 4.5 extends this analysis to an assorted set of configurations for the third image of the mammary data set. In general, the GPU gains on the mammary image set are more modest: 2.00x on a small window size and 2.59 for the large case (first chart column divided by fifth). Within the mammary image set, multiple CPUa become more effective and contribute to higher gains: Enabling a second CPU core provides a 1.82x factor improvement (pthreads version) with an additional 25% improvement when allocating the cores on

different sockets using MPI, and an almost optimal 1.93x factor when growing from two to four cores within a single node.

Note that four CPU cores are faster than the combination of two CPU cores and two GPU chips. This is caused by the communication time required to feed the GPU with data from its CPU partner via PCI-express. This cost is hidden in the four CPU case with parallel I/O reads from file and/or dual channel DRAM memory modules. This fact is also affecting parallel performance on the next section, where those configurations with double the number of CPUs than GPUs are studied. These configurations also maintain better workload balance considering that the convolution phase assigned to the GPU represents around 60% of the computational time (see Figure 3.5), and that the GPU exceeds by far the GFLOPS peak capacity of the CPU (see Table 4.1).

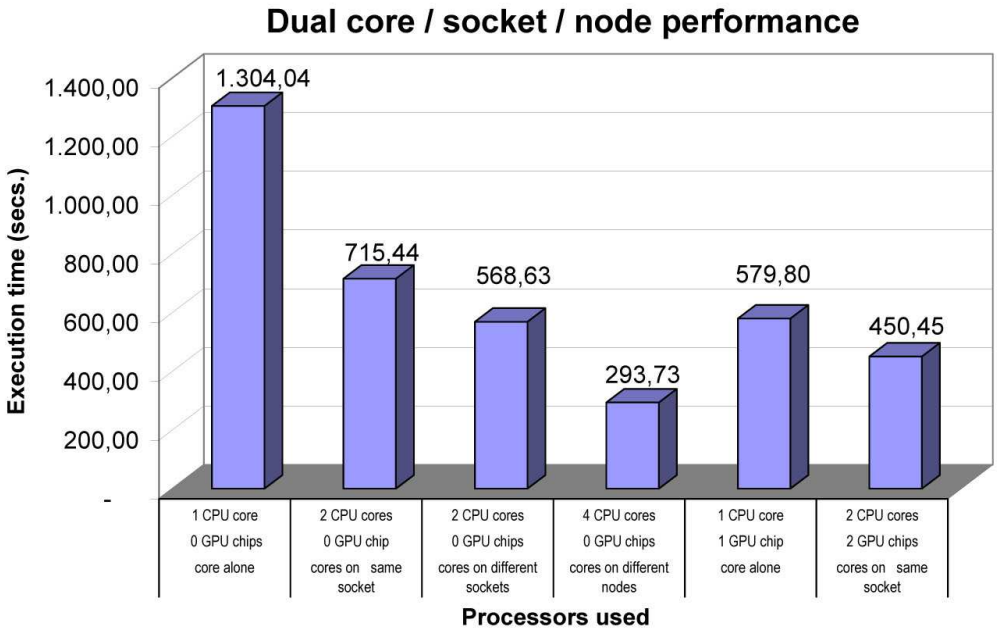


Figure 4.5: Single node performance under different node configurations for the largest image in the mammary data set and the large window size.

4.4.3 Parallel performance

Figure 4.6 shows the scalability of the algorithm on the CPU side, which is fairly consistent for all images belonging to the placenta data set when running on different number of CPUs. CPU executions are slower than GPU-assisted ones, but they are expected to be more scalable on a large number of nodes because computation can be performed more independently across multiple nodes than with the communication bindings of combined CPU-GPU executions. This analysis is ratified in Figure 4.7, where the analysis extends to the mammary data set for an assorted combination of CPUs and GPUs. A superlinear speedup case is even observed when moving from 2 to 4 CPUs (that is, from a dual-core to a dual-socket dual-core execution). This was anticipated by the results in Figure 4.5, where multi-socket parallelism was found to be more rewarding than the multi-core counterpart.

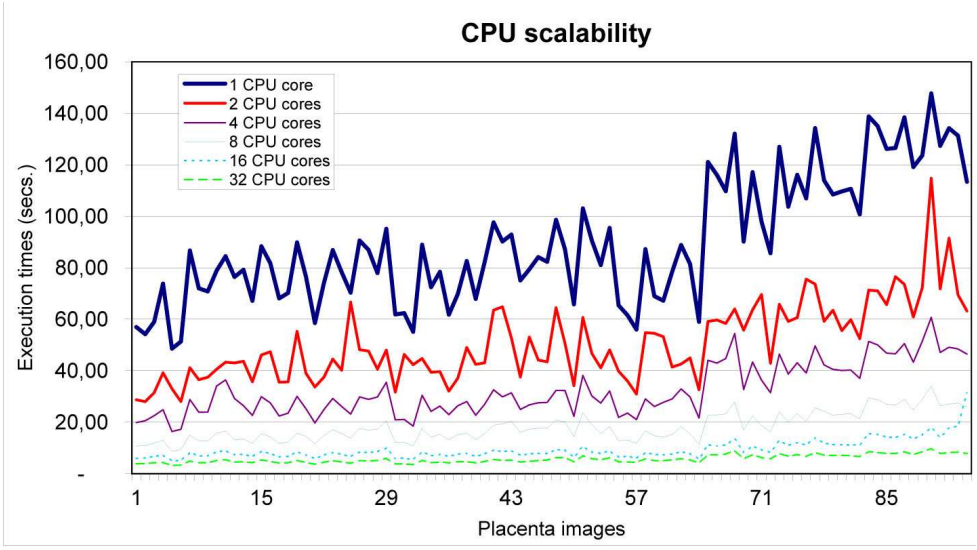
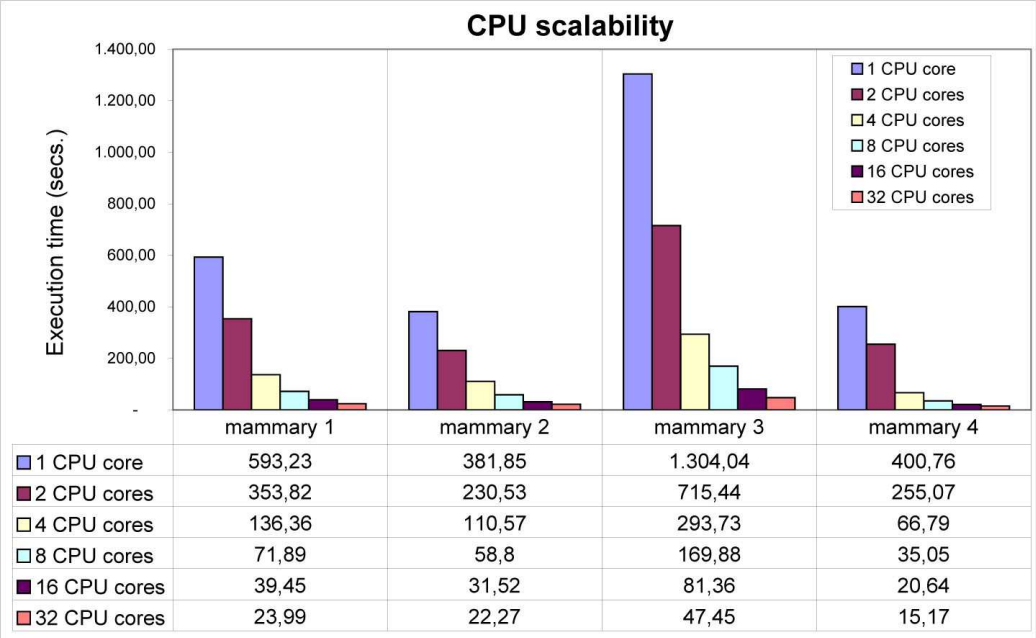
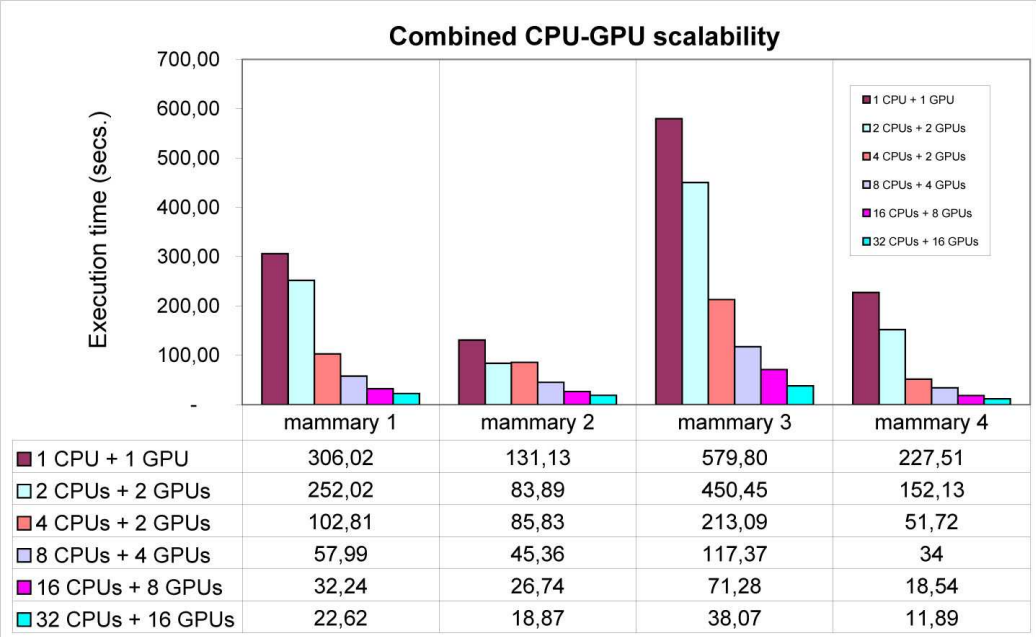


Figure 4.6: The scalability of the algorithm on CPUs for the placenta data set.



	mammary 1	mammary 2	mammary 3	mammary 4
1 CPU core	593,23	381,85	1.304,04	400,76
2 CPU cores	353,82	230,53	715,44	255,07
4 CPU cores	136,36	110,57	293,73	66,79
8 CPU cores	71,89	58,8	169,88	35,05
16 CPU cores	39,45	31,52	81,36	20,64
32 CPU cores	23,99	22,27	47,45	15,17

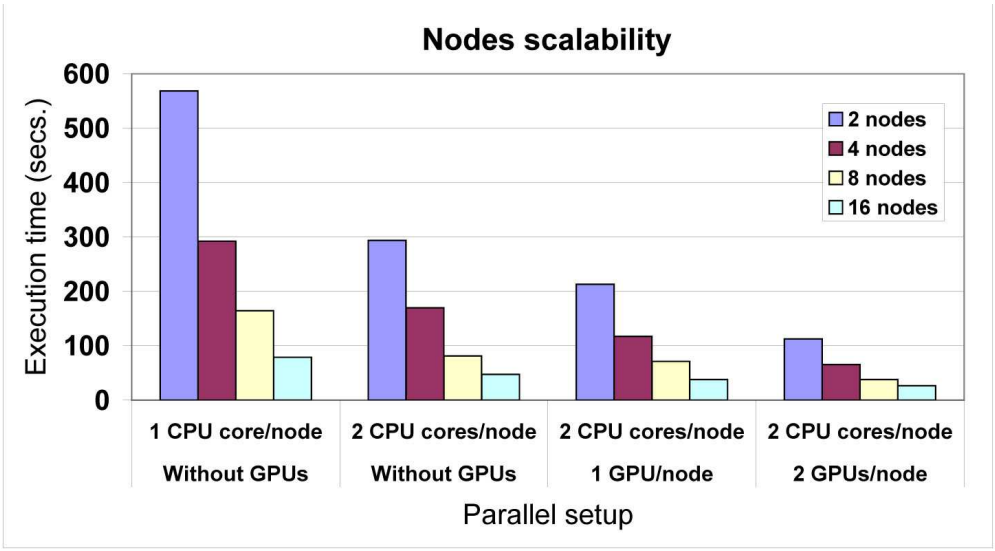
(a)



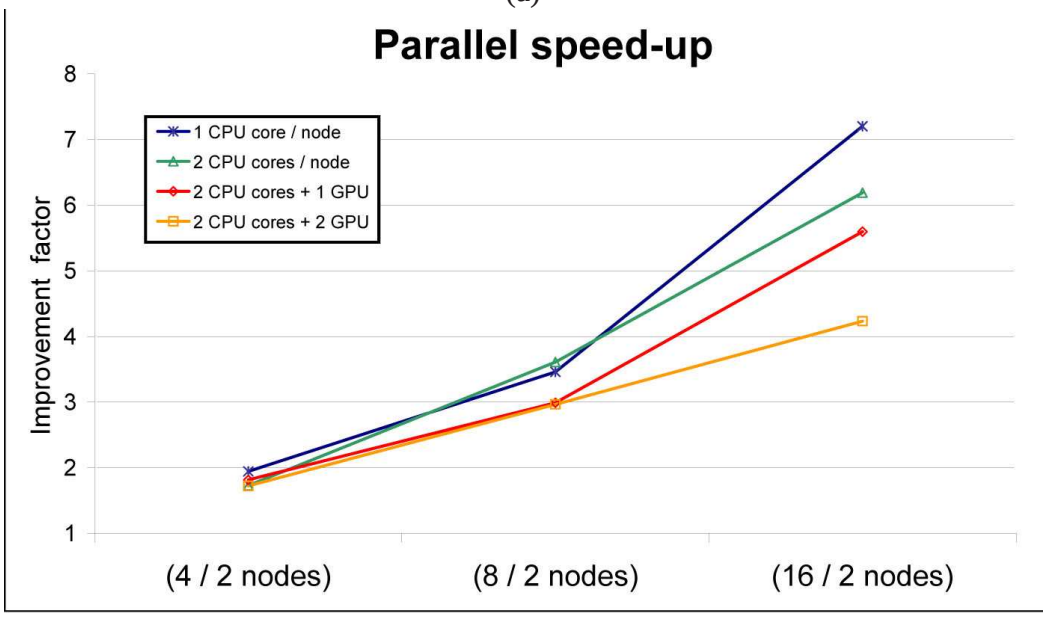
	mammary 1	mammary 2	mammary 3	mammary 4
1 CPU + 1 GPU	306,02	131,13	579,80	227,51
2 CPUs + 2 GPUs	252,02	83,89	450,45	152,13
4 CPUs + 2 GPUs	102,81	85,83	213,09	51,72
8 CPUs + 4 GPUs	57,99	45,36	117,37	34
16 CPUs + 8 GPUs	32,24	26,74	71,28	18,54
32 CPUs + 16 GPUs	22,62	18,87	38,07	11,89

(b)

Figure 4.7: Scalability on the mammary data set using the large window size (a) for the CPU executions, and (b) for the combined CPU-GPU execution.



(a)



(b)

Figure 4.8: (a) Scalability and (b) speedup on different number of nodes for the third image in the mammary data set.

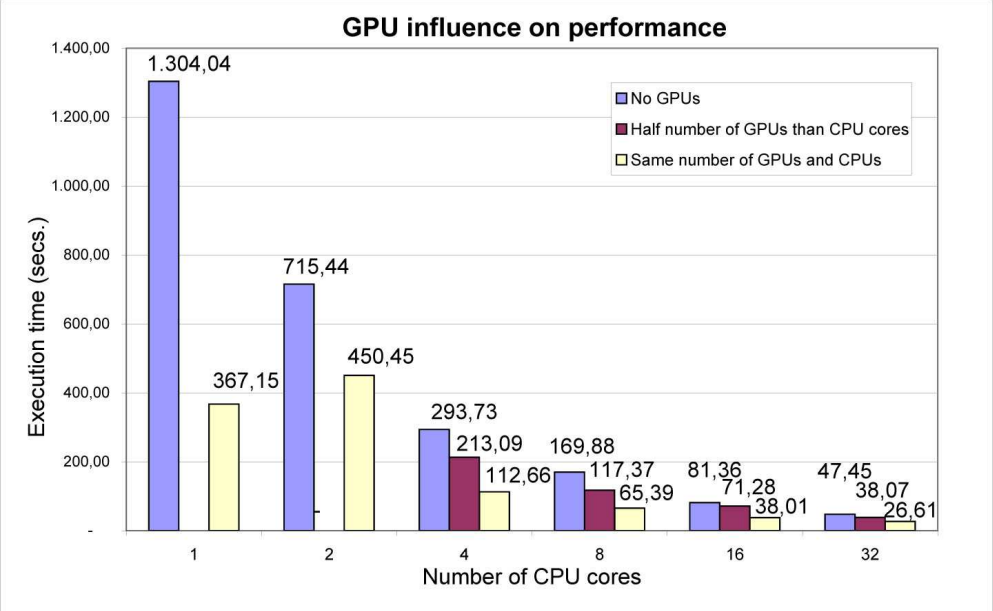


Figure 4.9: GPU influence on algorithm performance when using the largest image in the mammary data set and the large window size. The bars in the middle of the ”1” and ”2” cases are empty because they correspond to impractical cases.

For an increasing number of nodes, Figure 4.8 shows on the left the progressive reduction in the execution time for the particular case of the third mammary image. On the right, parallel speedup is more representative, telling us that the more aggressive a configuration becomes at the intra-node layer, the less effective results in inter-node parallelism. In other words, the fastest single-node configurations reduce their effectiveness on a large number of nodes, as a consequence of higher internal node communications.

Similarly, Figure 4.9 reports that GPUs are more effective on a small number of nodes for accelerating a CPU code, showing us a small tradeoff in performance on massively parallel computing.

Overall, from the departure point of 181 hours on a single Opteron CPU for a set of 500 images in the mammary data set, the parallel implementation on the 16 visualization

nodes of the BALE cluster was able to reduce the time to 3.7 hours (26.61 seconds for a single slide), achieving a total speedup of 49x when all 32 CPUs and GPUs participate in the cooperative environment.

4.5 Discussion and Conclusions

With the advances in imaging hardware, applications like the registration of large gigapixel images are increasingly popular, evolving towards computationally demanding algorithms for which parallel and scalable solutions become essential. Within this scope the contribution of this work is twofold: First, a parallelizable method is provided which has been successfully applied to biomedical studies for reconstructing the 3-D structures of biological specimens with micron resolution. Second, a solid heterogeneous and cooperative multiprocessor platform has been established where the best features of CPUs and GPUs meet for applying higher degree of parallelism at a variety of levels: (1) Multi-task, for simultaneous executions on CPU and GPU codes, (2) multi-node, using MPI for data partitioning across nodes, (3) SMP (Symmetric MultiProcessing) for multisocket CPUs and multicard GPUs using pthreads, (4) multi-cores, either using MPI or pthreads, and (5) SIMD (Simple Instruction Multiple Data), for the 128 stream processors of the GPU using CUDA.

For a mammary sample composed of 500 slides, it takes more than 181 hours to accomplish the registration process on a single Opteron CPU. This was reduced to 50 hours when enabling the GPU as co-processor, and minimized to 3.7 hours for a total speedup of 49x when all 32 CPUs and GPUs participate in our multiprocessor cooperative environment. While GPU-assisted versions were more effective at an intra-node layer, the CPU showed

higher gains on inter-node parallelism, suggesting that they may complement each other on hybrid supercomputers.

Overall, this study provides an illustrative example on how emerging architectures like multicore CPUs and GPUs meet and combine their power to assist non-computer scientists for efficiently adapting grand-challenge applications and providing almost real-time response to pathologists when working on the analysis of large scale biomedical images.

CHAPTER 5

REGISTERING HIGH RESOLUTION MICROSCOPIC IMAGES WITH DIFFERENT HISTOCHEMICAL STAININGS - A TOOL FOR MAPPING GENE EXPRESSION WITH CELLULAR STRUCTURES

The use of normalized cross correlation to identify precise correspondences from intensity information has several advantages, including efficient calculation and intuitive parameter selection. As Chapter 2 demonstrates, the resulting correspondences are accurate and can be used to produce genuine three dimensional reconstructions from sequences of microscopic images.

In some scenarios it is necessary to register two images with different stains to map molecular information to structure. Registering an immunohistochemically stained image to a hematoxylin and eosin stained image enables the visualization of the spatial distribution of proteins in microscopic structures at cellular resolution and beyond. The variation in color and morphological appearance between images with different stains creates a challenge for the task of identifying precise correspondences. How does normalized cross correlation perform when comparing content between differently stained slides?

In this chapter I investigate the issue of identifying correspondences between differently stained images using intensity information. Maximum normalized cross correlation is demonstrated to be ineffective as a classifier for correspondence accuracy, and a new

measure based on the topographical features of normalized cross correlation is proposed. The fast calculation of the proposed measure provides an advantage over the state-of-the-art in multi-modal similarity measures. A study mapping PTEN stain to hemotoxylin and eosin stain for breast cancer research is used to demonstrate the effectiveness of this new measure.

5.1 Introduction

One of the key problems in the post genomic era is to understand the regulation of gene expressions in organisms. Proteomics techniques such as genechips (microarray) and mass spectroscopy have provided a tremendous amount of information on gene expression patterns, however in most experiments these techniques are applied to biological samples that contain a diverse population of cells and therefore reflect unlocalized expression profiles. In contrast, gene expression profiles in different types of cells can be drastically different and studies show that even the same type of cell in the same tissue environment can exhibit heterogeneity in the expression levels of key proteins [79]. Therefore, the capability to map gene expression to individual cells is essential to explore gene regulation within tissue environments at the cellular level.

Microscopic imaging is an essential tool for investigating localized gene expression since it can capture both cellular distribution as well as gene expression information. However, the integration of cellular and molecular distribution information is a difficult task, since this information is usually obtained using different staining techniques on two or more different histological sections. This is a particularly challenging problem if a computational approach is taken due to the large size of microscopic images (usually in the

size of several gigabytes per image). In order to address this problem, this chapter develops a novel workflow for the precise nonrigid registration of microscopic images with different stain types. The workflow contains three stages: rigid registration, nonrigid registration, and multiresolution refinement. The "sharpness" of maxima in the normalized cross-correlation function is demonstrated as a similarity measure capable of identifying correspondences between an image pair. The use of correlation avoids the high computational cost of computing other measures used for multi-modal registration such as mutual information. The correspondences are used as control points to compute a nonrigid transformation between the two images. In order to improve the matching accuracy, a multiple resolution approach is adopted for accurately matching key regions of interests.

The proposed workflow was tested using mouse mammary gland images with a focus on the mammary duct regions that are the potential sites for tumorigenesis. Serial section images were obtained in pairs: one section stained to identify cellular structure using a specific immunohistochemical stain and the other section stained to show expression of an important tumor suppressor gene *PTEN*. By registering these two section images PTEN expression was mapped to structures of interest such as fibroblasts and epithelial cells. The results show that the proposed algorithm is highly accurate and applicable to large scale gene expression mapping studies for breast tumor microenvironment.

Three challenges need to be addressed to accomplish registration at the precision necessary for expression mapping: comparison of content between images with different stains, nonrigid deformation and natural morphological differences between sections, and the large size of high magnification histological images. These challenges are addressed with the following approaches:

1. **A new similarity measure for intensity feature matching.** The goal of image registration is to determine a transformation that maximizes the similarity between two images. Mutual information (MI) and normalized cross correlation (NCC) are commonly used as similarity measures for registration, however, it was observed that thresholding these raw similarity measures is not adequate to discriminate good matches from bad. A new similarity measure is proposed based on the "sharpness" of maxima in the cross correlation function.

2. **Adoption of a multiple resolution approach for nonrigid transformation.** In order to register the images as precisely as possible a large number of spatial correspondences are required to compute an accurate mapping. Due to the elasticity and heterogeneity of the tissues a local transformation cannot be extrapolated globally. To address this challenge a multiple resolution matching approach was implemented to align local regions of interest in a piecewise linear manner.

3. **Scalable workflow.** Microscopic images can be very large. Using an Aperio slide scanner to scan a $1.5cm \times 3cm$ section at 20X objective length generates an image at the resolution $0.5\mu m$/pixel that is $30,000 \times 60,000$ pixels and 6.5 GB in uncompressed form. These large sizes require algorithms that are efficient, scalable, and parallelizable. The proposed workflow is a slight variation of the two stage algorithm and so uses the same efficient operations and workflow demonstrated as efficient and parallelizable in Chapters 3 and 4.

This chapter is organized as follows: The biological application is discussed in Section 5.2. Related works are discussed in Section 5.3. The workflow is presented in Section 5.4. The novel correlation sharpness similarity measure is presented in Section 5.5. Results are offered in Section 5.6. Discussion and conclusion are contained in Section 5.7.

5.2 Biological Application

The application in this chapter uses on a transgenic mouse model: the PTEN gene knocked out in mammary gland fibroblast cells.[1] PTEN, also known as phosphatase and tensin homolog, is a well known tumor suppressor gene. Inactivation of PTEN is associated with several diseases including cancer [80]. It has been observed that this strain of mice inevitably develop epithelial breast tumor after the knockout. The biological question is how the inactivation of the tumor suppressor gene PTEN in fibroblasts leads to tumor development in epithelial cells. Answering this question will provide insight into cell interactions and tumorigenesis in the tumor microenvironment. A critical part of this inquiry is the expression mapping of key genes in different cell types.

The PTEN mapping is demonstrated using serial mammary tissue sections obtained from the transgenic mice with hematoxylin and eosin staining (H+E) and PTEN staining applied alternately to produce a sequence of sections with interleaved stain types. In this paper the focus is on producing a visualization for one pair of H+E and PTEN images, but the interleaved staining approach could also be used to produce a three-dimensional reconstruction that contains both the structural information from H+E and the expression information from PTEN. The work in [81] presents such a reconstruction for cervical tissue using an H+E/p16(INK4a)/CD3 interleaved staining.

5.3 Related Work

There are many works on observing the expression map of a specific gene in cells. The most direct approach is to use confocal microscopic imaging to visualize the co-expression of the gene product and the cell specific markers, however this approach requires extensive

[1]This is a tissue-specific knockout animal model.

molecular and genetic manipulation on the model animal system. In [82], a tissue microarray (TMA) approach was developed where small samples of retina (0.5mm$\times$0.5mm) were fixed in plastic and sectioned at 250nm interval. Each section was stained for a special molecule of interest using immunochemical staining. Therefore for a section of sample of $5\mu m$ thick, the products of twenty different genes can be determined. A limit of this approach is that it is difficult to extend this technique to larger samples in the multiple-millimeter scale. Another approach to obtain gene expression profile at high spatial resolution is to use laser capture microdissection (LCM) to carve out small piece of tissue in each section and conduct microarray analysis on the carved samples. This way the entire profile of gene expression can be mapped to a spatial resolution of tens of microns. Other approaches to obtain the gene expression information for multiple genes include multiple spectral imaging [83], multicolor staining [84] and multiwash technique, however, these techniques all require special experimental facilities and equipment.

Work on the automatic registration of images with different stains is limited. The authors of [85] propose a segmentation-based method for the nonrigid registration of images with different stains. This approach requires producing a consistent segmentation between the image pair by re-ordering and merging class labels prior to registering the class-label images.

5.4 Image Registration Workflow

The registration of two images with different stains essentially follows the same workflow as the two stage algorithm presented in Chapter 2. The proposed workflow in Figure 5.1 introduces two new elements to the standard approach: correlation sharpness similarity measure and multi-resolution refinement.

The images are first aligned with an approximate rigid registration and then this initialization is refined using precise comparisons of intensity information. Salient anatomical structures such as blood vessels or ducts are matched between images based on properties such as size and shape. These pairs are filtered based on geometric constraints to produce an estimate of the rigid registration parameters.

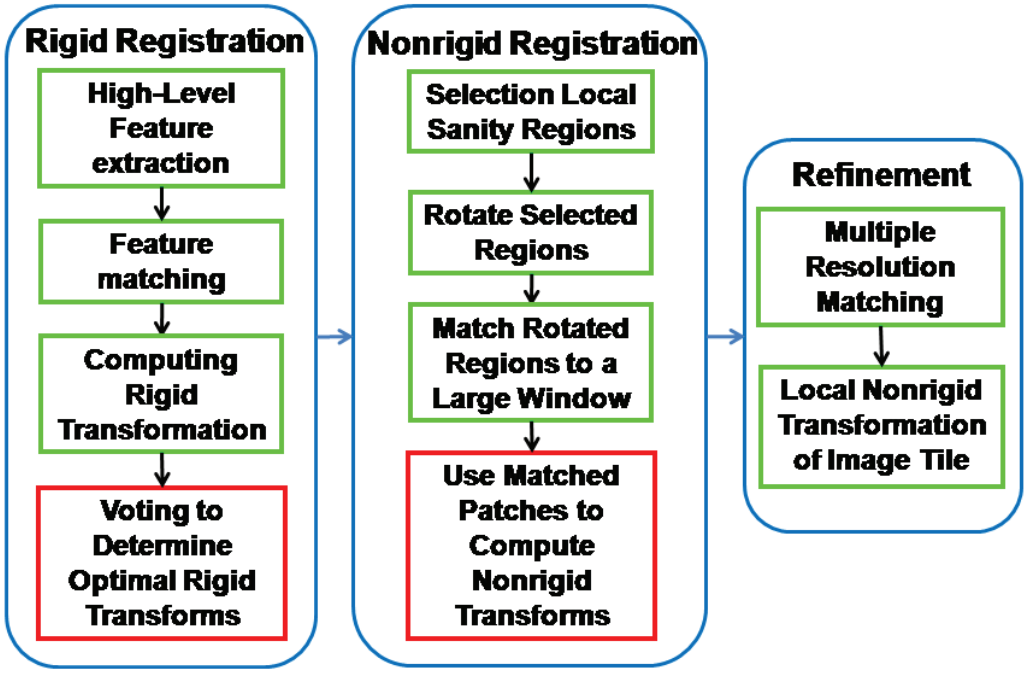


Figure 5.1: Image registration workflow. The algorithm consists of three stages: rigid registration, nonrigid registration, and refinement. The green blocks are independent local operations that can be straightforwardly carried out in parallel.

The proposed refinement stage differs slightly from the standard two stage algorithm. Correlation sharpness replaces maximum normalized cross correlation as a similarity measure for intensity feature matches. Due to the heterogeneity of tissue in the biological samples there is also a focused refinement in regions of interest. In mouse mammary gland,

the adipose tissue and the extracellular matrix around mammary gland ducts have drastically different mechanical properties in terms of elasticity and rigidity. The difference in mechanical properties between these tissues leads to variations in the extent of local deformations in the histological sections. A global nonrigid transformation such as a polynomial transformation is not sufficient to compensate for more drastic local deformations and morphological changes. Other methods such as thin-plate spline and locally weighted basis functions require a large number of matched control points in this scenario, which is not computationally feasible. To accommodate this behavior special regions of interest are identified and a more precise matching is conducted iteratively in these regions at multiple resolutions to refine the correspondence accuracy.

After the initial matching of intensity features, the center points for the template regions and their corresponding matches are used as control points to generate global nonrigid transformation such as polynomial or piecewise affine transformation. In regions of interest such as mammary gland ducts and breast tumor stroma these matches are refined at higher resolution to achieve better matching precision. The selected regions are effectively split and rematched: the 500×500-pixel template patch surrounding the region is divided into four 250×250-pixel patches that are each re-matched. A piecewise affine transformation is then computed and applied to these regions based on the new local matches.

5.5 Sharpness of Normalized Cross Correlation Function as a Similarity measure

In many approaches to registration, including the two stage algorithm, correspondences are identified by comparing local regions of intensity using a similarity measure such as mutual information or correlation. The similarity measure is calculated over a 2D grid to

identify the most similar alignment, and the similarity at this best position is thresholded to determine if the match is satisfactory. Mutual information is often used in cases for comparing images with different modalities or staining, however the computation of MI requires the time-consuming calculation of a 2-D histogram at every point on the grid. In contrast normalized cross correlation can be computed very quickly over multiple grid locations using fast Fourier transform.

An extensive manual experiment was conducted to test normalized cross correlation and mutual information for effectiveness in match discrimination. One mouse mammary H+E/PTEN image pair was chosen and 320 template regions of 500×500 pixels each were manually selected throughout the H+E image from areas with ductal content. The corresponding regions from the PTEN image were also manually identified and a search window of 1000×1000 pixels was designated for each template region. Both mutual information and NCC functions were calculated between the region pairs to determine if the maximal similarity alignments were satisfactory. Figure 5.2 shows an example of the NCC from one of the pairs used for testing. *Interestingly, in most cases (291 out of 320), the peak location of NCC corresponds to a satisfactory match.* This important observation motivated a further investigation into how to use NCC for matching regions with different stain types due to its low computational cost.

Based on these manual classifications the distribution of maximal NCC values for the region pairs was examined to determine if simple thresholding could be applied to discriminate satisfactory matches. As shown in Figure 5.3, the ranges of NCC values for satisfactory and unsatisfactory matches overlaps significantly, indicating that maximal NCC value is not a good candidate for match classification. This is also demonstrated with the pair of

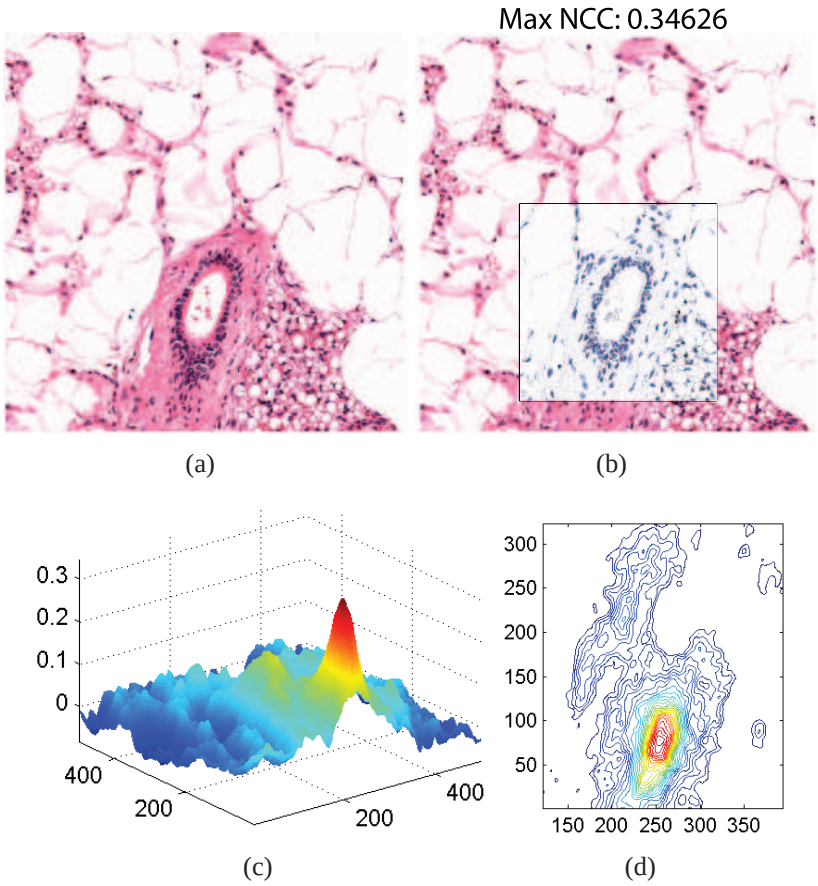


Figure 5.2: Example of satisfactory match. (a) H+E duct region. (b) PTEN overlaid on H+E corresponding to correlation peak. (c) 3-D surface view of the NCC function shows a peak in NCC value. (d) Isocontour of the normalized cross-correlation (NCC) function for the duct region between two images with respect to x- and y- translations.

regions shown in Figure 5.4. Although the two pairs have similar maximal NCC values, one of the match results is unsatisfactory.

5.5.1 Sharpness of the NCC function peak as a similarity measure

It was observed in the correlations between PTEN and H+E staining that although maximal values are not reliable for classifying matches, a distinguishably sharp peak is present in most cases where the matching is satisfactory. For this reason a sharpness measure for

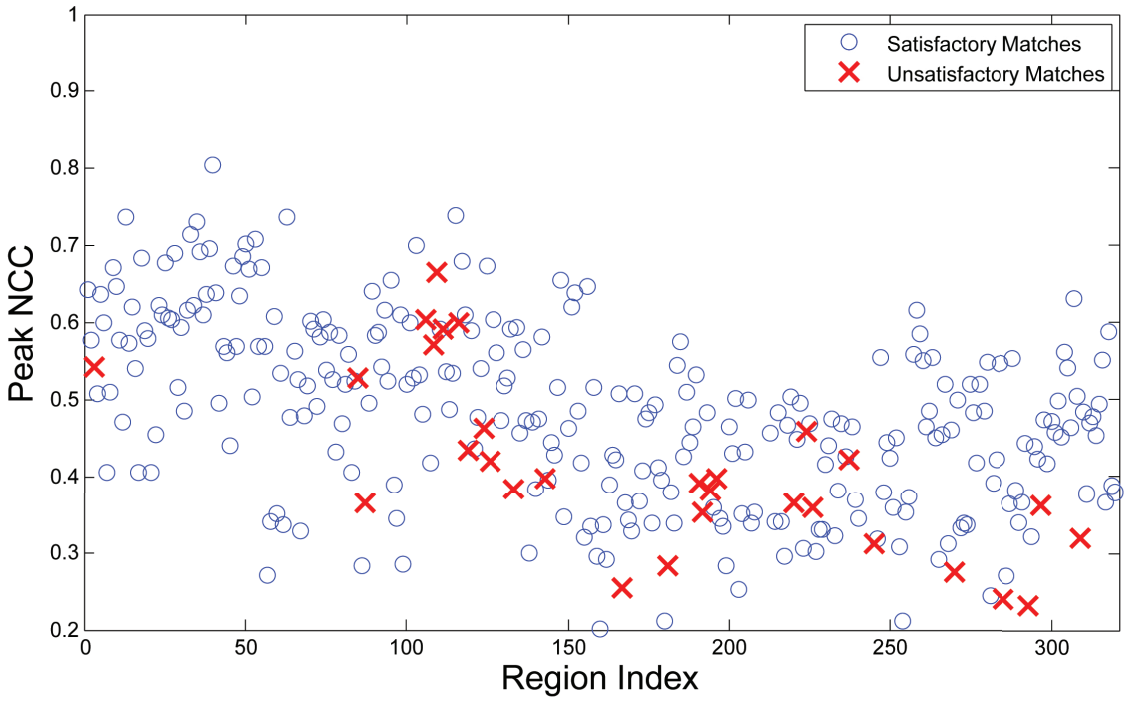


Figure 5.3: Peak NCC values for the 320 regions tested.

the correlation function was defined based on the cross sectional area of the peak S at different depths h (Figure 5.5). Specifically the measure R is defined as

$$R = h/\sqrt{S}. \tag{5.1}$$

For fixed depth h, the smaller the cross-section area S, the larger the sharpness measure R is. As shown in Figure 5.6, thresholding R at 0.0025 can discard most unsatisfactory matches with the cost of discarding some satisfactory matches as well.

5.5.2 Computation of NCC sharpness

The approximate calculation of the sharpness measure R can be achieved using a simple procedure. For a correlation surface $\rho(x, y)$ with a single peak, the area S can be

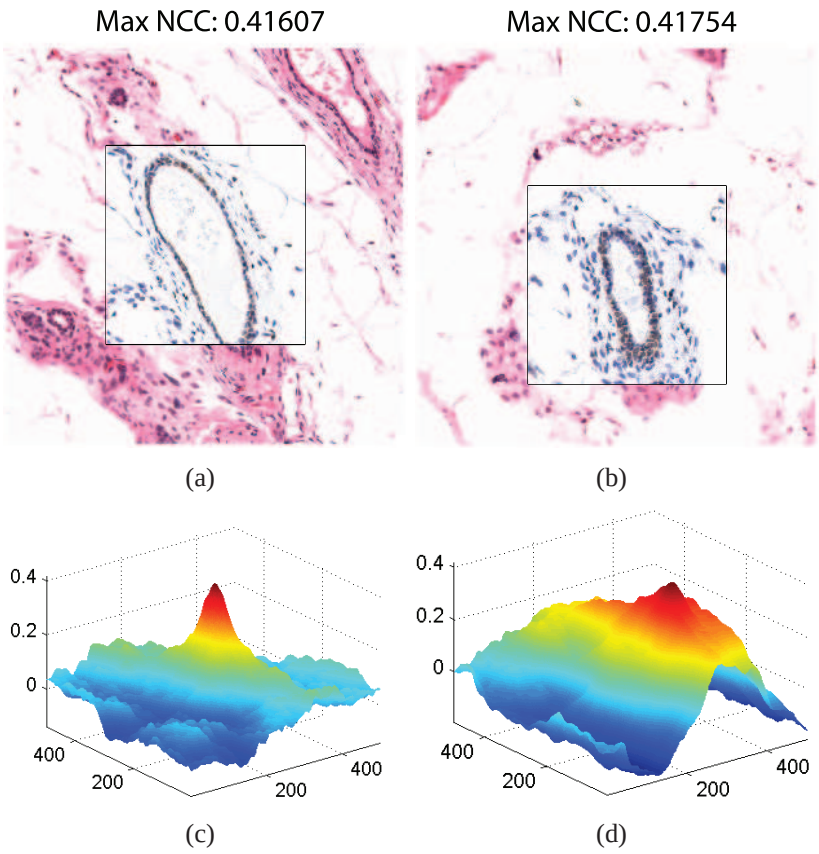


Figure 5.4: Peak sharpness is an indicator of match specificity. (a) Satisfactory alignment. (b) Unsatisfactory alignment. (c) The maxima of the correlation surface for the satisfactory alignment lies atop a prominent peak. (d) The peak for the unsatisfactory alignment is broad and gradual.

computed for small h by simply counting over the whole grid the number of correlation values $\rho(x, y) \geq \rho_{max} - h$. This assumes that all correlation values greater than $\rho_{max} - h$ lie under peak in question. In practice this assumption is met since most correlation surfaces contain only a single peak, be they sharp or broad. If the single-peak assumption does not hold then a more sophisticated approach can be used, identifying distinct regions where $\rho(x, y) \geq \rho_{max} - h$, and only counting the area of the region containing ρ_{max}. The single-peak assumption was used for the experiments presented in this chapter.

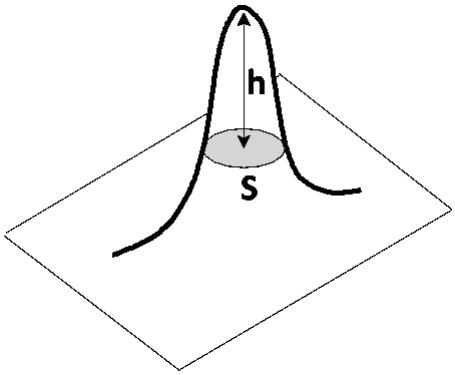


Figure 5.5: Illustration of a peak in which h defines the height of the level set and S defines the area of cross-section at height h.

5.6 Validation and Results

A pair of images as described in Section 5.1 was used to test the proposed algorithm. Since no ground truth is available the results were visually inspected and assessed.

5.6.1 The power of using the sharpness measure R as a similarity measure

The ROC curves were computed for thresholding on R and the peak NCC value respectively. As shown in Figure 5.7, the choice of threshold on R has a fairly large range without incurring any false positives (between 0.0026 to 0.0039 with at least 100 true positive but no false positives). In practice this is a desirable characteristic since the unsatisfactory matches can influence the quality of the final mapping results.

5.6.2 Multiple resolution matching

The goal of the multiple resolution matching is to improve the matching accuracy in key areas of interest. These areas can be either manually selected or automatically chosen based

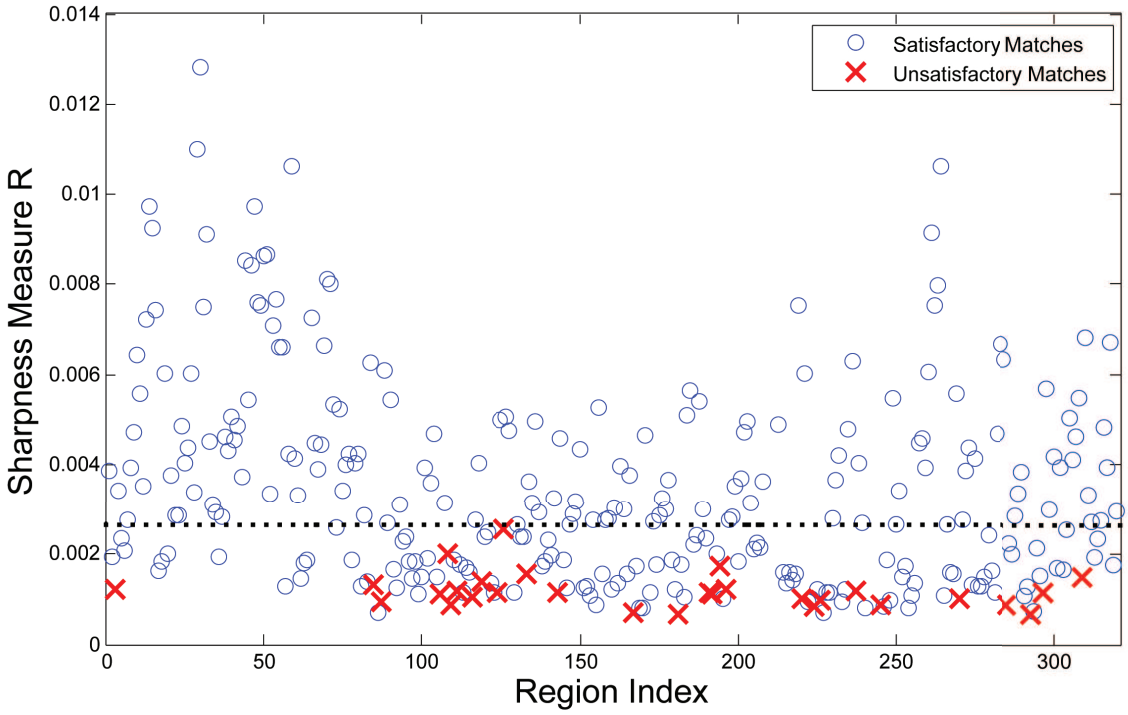


Figure 5.6: The distribution of R for 320 regions. The dashed line indicates that 0.0025 is a reasonable threshold for discarding unsatisfactory matches while preserving a significant number of satisfactory matches.

on biological criteria such as cell density or the existence of certain structures. For validation 80 regions were manually selected. Seventy-nine (98.75%) regions show improvement in matching accuracy in terms of continuity and smoothness of the structures. In order to visualize the results of the mapping the images were converted to gray scale with the H+E image as the red color channel and the PTEN image as the green channel. Overlapping regions of significant intensity appear as yellow. Two examples are shown in Figure 5.8 and Figure 5.9. Not only are large structures such as mammary gland ducts mapped well, microstructures such as cell nuclei and cell membrane are also closely aligned.

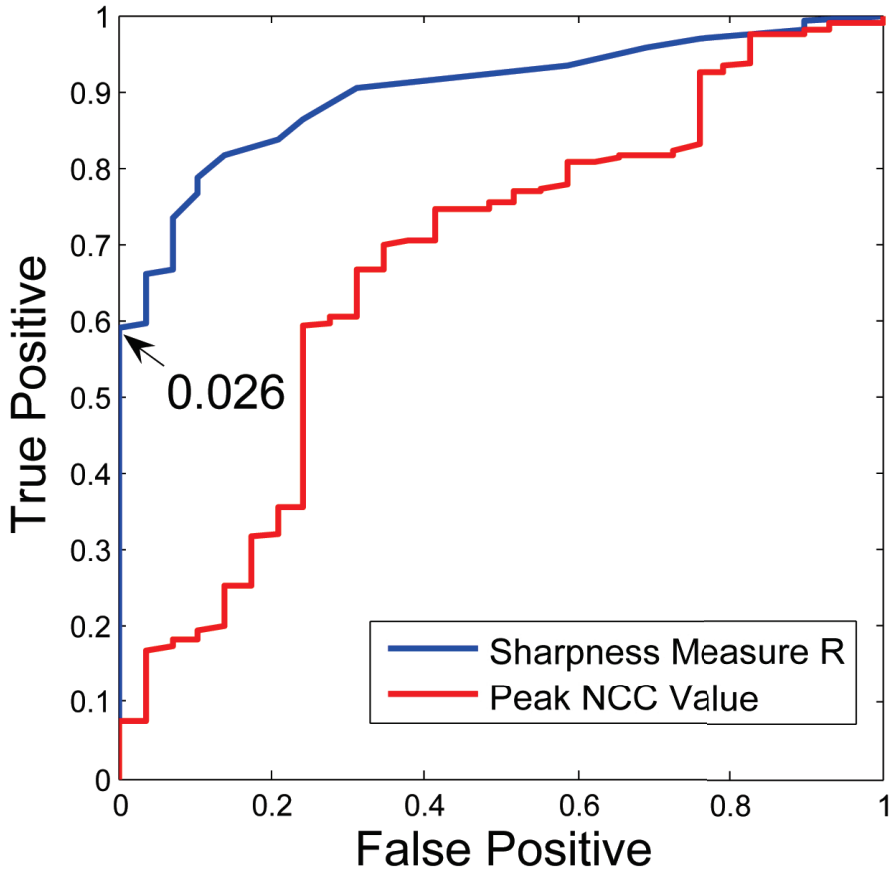
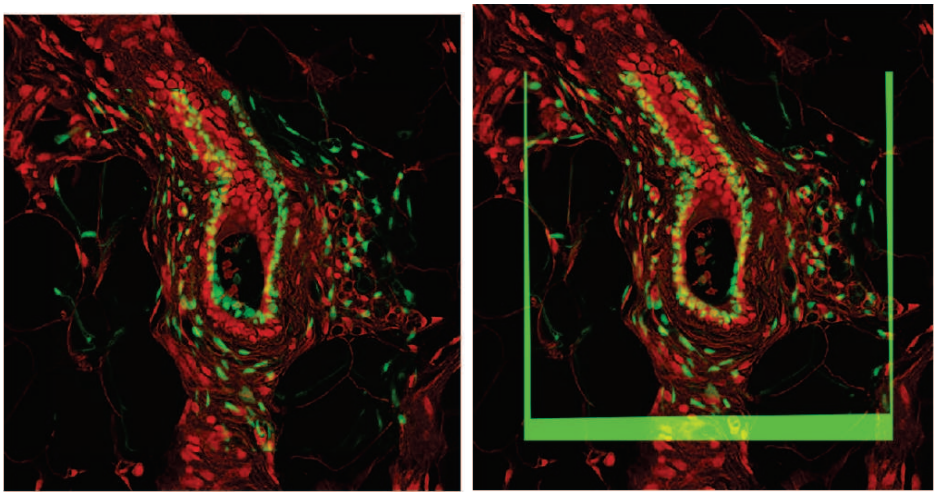


Figure 5.7: Comparison of ROC curves for thresholding on the sharpness measure R and on the peak NCC value.

5.6.3 Matching of mammary gland ducts

Mammary gland ducts are lined by a layer of epithelial cells which are thought to be the primary sites for breast epithelial tumor initiation. It is critical to have accurate matching for these cells. In most cases the overall mammary gland duct linings are accurate with the layer of epithelial cells tightly overlapped. The individual cell nuclei are not always matched, partially due to the fact that the gap between the two slides is $5\mu m$ and the nuclei in one section may not appear in the adjacent section. In general the mapping is accurate



(a) (b)

Figure 5.8: Visualization of multiresolution effect for stain mapping for regions of interest. The mapped images are converted to gray scale and the H+E is embedded in the red color channel and the PTEN in the green color channel. (a) Mapping before multiple resolution matching. (b) Mapping after multiple resolution matching.

within the inspected regions. The results show that the epithelial cells have normal PTEN expression while the fibroblasts that produce extracellular matrix in the periphery of the ducts are PTEN deprived.

As shown in Figure 5.10, there are usually red regions around the ducts. These regions are mainly composed of fibroblasts and the extracellular matrix (with collagen produced by fibroblasts). These regions are only stained in the H+E image but not the PTEN image since the PTEN gene is deactivated in the fibroblasts, however, the epithelial cells which form the lining of the ducts are stained in both sections as shown by the yellow color in the overlaid images, implying that PTEN expression is normal in the epithelial cells.

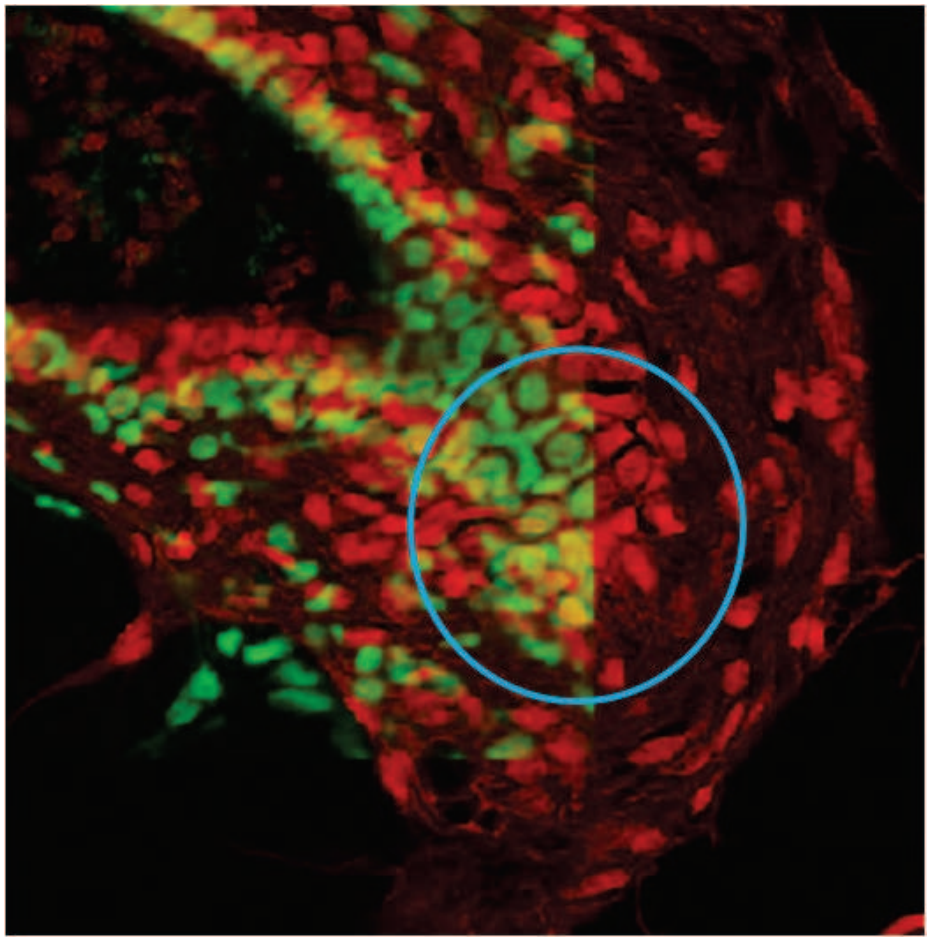


Figure 5.9: Zoomed mapping results. The matching of cell nuclei can be seen in the blue circle. However, in most cases this precise overlapping is not observed due to the natural morphological difference between the two images.

5.7 Discussion and Conclusions

In this chapter a new image registration framework is proposed for overlaying microscopic images with different stain types. In order to accurately register microscopic images, it was first established that the sharpness of the normalized cross-correlation function can be used as a similarity measure for comparing intensity information between the two images. This helps avoid the high computational cost of more sophisticated approaches, which is

critical for processing images at this scale. In order to improve the matching accuracy, a multiple resolution approach was adopted for key regions of interests. The algorithm has been tested using real histological images of mouse mammary gland sample in a breast tumor microenvironment study. The results show that the algorithm is highly accurate. This work lays the foundation for large scale gene expression mapping of mouse breast tumor microenvironment in where the plan is to map expression levels for 50-100 genes over four stages of tumor progression.

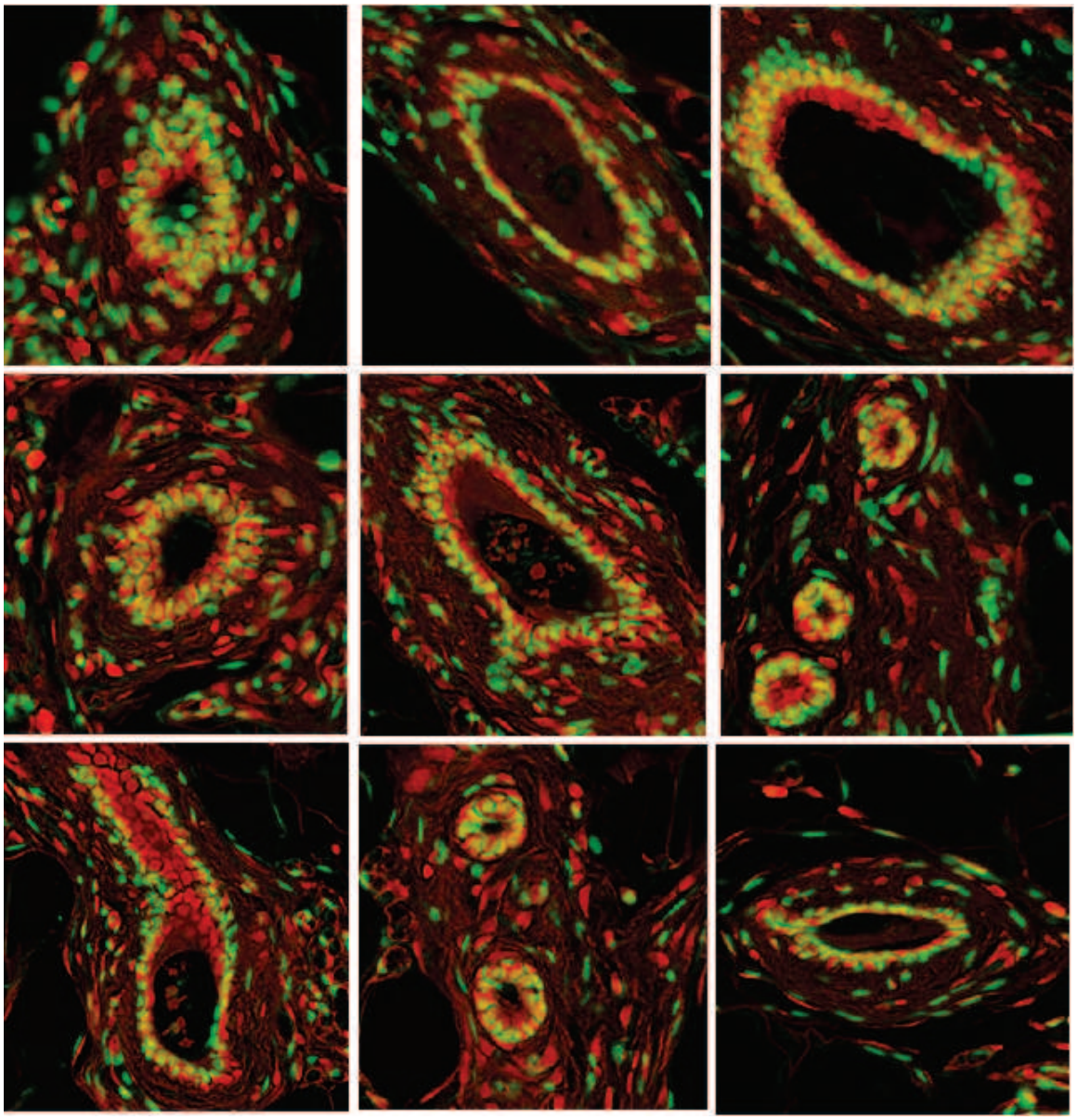


Figure 5.10: Examples of mapped mammary gland duct regions.

CHAPTER 6

FEATURE-BASED REGISTRATION OF HISTOPATHOLOGY IMAGES WITH DIFFERENT STAINS: AN APPLICATION FOR COMPUTERIZED FOLLICULAR LYMPHOMA PROGNOSIS

Correlation sharpness provides a means for registering microscopic images with different stains. The ability of correlation sharpness to make discriminating comparisons between intensity content provides the precise correspondences needed for nonrigid registration. In some cases, however, the content is either too dissimilar or lacks the saliency needed to generate accurate correspondences.

In this chapter I address this problem using a novel method for nonrigid registration based on the matching of groups of *high level features* that represent small but conspicuous anatomical structures through geometric constraints. This choice of feature provides a rich matching environment, but also one that is fraught with a high mismatch probability. Building upon the work of the fast rigid registration algorithm, this method increases matching confidence by using geometric constraints to establish local groups of coherent features. The proposed method is validated with a statistical analysis demonstrating that given a proper feature set the accuracy of the automatic nonrigid registration is comparable to a manual nonrigid registration.

This work is motivated by an application in the pathological grading of *Follicular Lymphoma* (FL). FL is the second most common type of non-Hodgkin's lymphoma. Manual

histological grading of FL is subject to remarkable inter- and intra-reader variations. A promising approach to grading is the development of a computer-assisted system that improves consistency and precision. Correlating information from adjacent slides with different stain types requires establishing spatial correspondences between the digitized section pair through a precise nonrigid image registration. However, the dissimilar appearances of the different stain types challenges existing registration methods.

6.1 Introduction

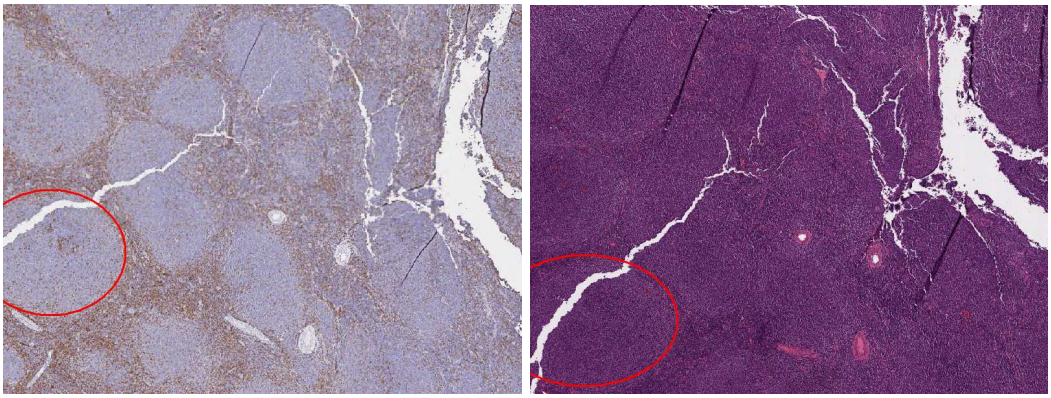
Histopathological examination is a crucial step in cancer prognosis. Pathological analysis of biopsy samples is necessary to characterize the tumor for treatment planning. Cancer prognosis that relies on this qualitative visual examination may have significant inter- and intra-reader variability due to due to several factors, such as experience or fatigue at the time of examination [86, 87]. Poor reproducibility of histological grading may lead to inappropriate clinical decisions on the timing and type of therapy, and may result in under- or over-treatment of patients with serious clinical consequences. A computer system capable of extracting quantitative, and thereby more precise and objective prognostic clues, may provide more accurate and consistent evaluations. For this reason a computer-assisted grading system is being developed for one particular cancer type, *Follicular Lymphoma* (FL) [88, 89].

FL is the second most common type of non-Hodgkin's lymphoma that consists of a group of cancers developing from the lymphatic system. The word of "follicular" is derived from round-shaped biological structures, namely "follicles", which are visible under microscope. In current clinical practice, the risk stratification and subsequent choice of

therapy for FL mainly depends on the histological grading process that involves computing the average number of centroblasts (CBs), i.e., malignant follicle center cells, as recommended by World Health Organization [90–92]. Due to the large number of follicles usually exhibited in biopsy samples, only ten follicle regions equivalent to a microscopic high power field (HPF) of $0.159mm^2$ are randomly sampled to make this process feasible in practice. Performing CB count over a limited number of follicles can introduce a considerable sampling bias as the selected follicles may not be representative of other sample regions, especially in heterogeneous tumors [87].

With sampling regions identified, centroblasts are then manually counted in HPFs of the selected follicle regions. FL cases are classified into three histological grades based on the centroblast average count: grade I (0-5 CB/HPF), grade II (6-15 CB/HPF) and grade III ($>$15 CB/HPF) [90]. Grade I is usually associated with indolent disease and not treated, while Grade III is associated with aggressive disease and treated aggressively. A multi-site study reported only $61\% \sim 73\%$ grading agreement across expert pathologists [86]. In addition to this inter-observer variation, the manual counting of centroblasts is very time-consuming, especially when a large number of biological samples need to be examined.

In the current follicle grading processes, pathologists usually resort to using pairs of adjacent slides dyed with different stains to enhance visual contrasts. For example, immunohistochemical (IHC) stains, e.g., CD3 and CD20, provide a clear visual contrast for the follicle structures at low magnifications, e.g. $2\times$, $4\times$ and $8\times$. By comparison, Hematoxylin and Eosin (H&E) stain enhance the contrast of the cytological components, and provide better cellular-level detail at higher magnifications, e.g. $20\times$ and $40\times$. Two representative sample image regions from IHC and H&E stained images captured at $2\times$ magnification are shown in Fig. 6.1, where follicle boundaries are clearly visible in the IHC (CD3



(a) (b)

Figure 6.1: Sample image regions from CD3 and H&E stained FL slides captured at 2× magnification. (a) and (b) correspond to adjacent sections from the same specimen and demonstrate local and global deformations and the difficulty of identifying follicles from H&E-stained slides. Sample regions corresponding to the same follicle are highlighted in red.

stain in this specific example) stained image, but are not clearly discernible in the H&E stained counterpart. The proposed computer-assisted system mimics the manual grading procedure, working jointly with pairs of images with IHC and H&E stains. The flowchart of this hybrid FL grading system is presented in Fig. 6.2.

One of the key steps in this system is to map the spatial coordinates of the detected follicle positions from the IHC stained image to the H&E counterpart image where the centroblast detection will occur. In order for the IHC image analysis to be able to interact with the H&E analysis process, an image registration algorithm is required that allows the output of IHC follicle detection to be fed into the H&E centroblast detection stage. In this chapter, such a methodology and its implementation on clinical cases are reported.

Image registration for biological applications has been studied extensively [9, 10, 13, 14, 22, 23, 26, 27, 32, 33, 35, 36]. Registration can be considered as an optimization problem,

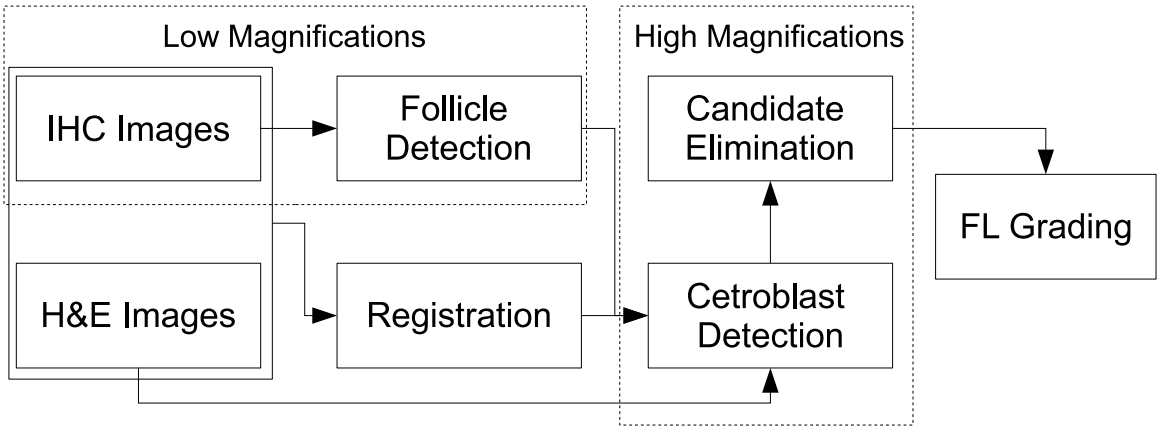


Figure 6.2: Flowchart of the computer-aided FL grading system.

posed as finding the optimal transformation $\mathcal{T}$ between two images I_1 and I_2 to maximize a defined similarity measure such as mutual information [38]. Registration may also be formulated as a problem of feature matching: finding correspondence between sets of representative features using descriptors and spatial relations [62]. The space of transformations includes rigid, that deals with only rotation and translation, and nonrigid, that compensates for deformations such as bending, stretching, shearing and warping [27, 39, 40]. Like most optimization processes, a good initialization is critical for a global optimum outcome. In many cases, a good rigid registration serves as an ideal initialization for non-rigid registration [26]. For large images with conspicuous deformations, hierarchical multi-resolution registration methods have also been widely used in medical imaging applications [44, 45].

The key challenge for the registration of sectioned histopathological images is to compensate for distortion introduced by slide preparation. The input slide pairs are cut with a 5 μm thickness from adjacent locations so that the morphological structures vary minimally between image pairs. However, there are discernible global and local deformations between these neighboring tissue sections due to the slide preparation procedure (i.e., sectioning,

fixation, embedding, and staining). The preparation process can introduce a variety of nonrigid deformations including bending, shearing, stretching, and tearing. At micron resolutions, even minor deformations become conspicuous and may prove problematic when accuracy is critical to the end application. In order to compensate for such deformations, a nonrigid registration is essential and success depends on establishing a large number of precise spatial correspondences throughout the extent of the image.

An additional challenge for the registration of histopathological images exists when the images to be registered are stained with different stain types, and consequently have dissimilar appearances. An approach based on intensity values requires the ability to resolve similarity between intensity signals using a measure such as mutual information. Such similarity is not necessarily guaranteed for combinations of stain pairs, since for some stain combinations only complex high-order perceptual qualities will be consistent. If the images do exhibit a significant visual similarity, then an approach exists that uses correlation sharpness as a means for classifying local similarity between intensity information [61]. However, in the case of follicular lymphoma images with H&E and IHC staining, content at local scales appears as a uniform texture of cellular components, certainly not an ideal condition for intensity comparison between distinct sections. Another approach exists that uses a segmentation of tissue types as input to a registration process [26]. The registration reconciles differences in the segmentation by calculating a displacement field that is used for nonrigid registration. Again, this approach is not reasonable in the case of follicular lymphoma, where the content is textural and segmentation is the original problem that a registration is intended to aid.

To address these challenges, this chapter proposes a registration approach based on the matching of small salient anatomical features. Small features such as blood vessels appear

universally in most tissues and have a common appearance in many stains, making their extraction and matching feasible. These features are used to establish spatial correspondences and register the images in two stages: first rigidly, to roughly align the images, then nonrigidly, to correct for elastic distortions introduced by preparation. The first stage uses a previously established mismatch-tolerant voting procedure [33]. With the rough alignment of the images calculated, the second stage establishes coherent local networks of matched features between the images to enhance the confidence of matching and reduce the probability of mismatch and provide a set of spatial correspondences that is satisfactory for nonrigid registration.

The outline of the remaining chapter is organized as follows. Section 6.2 describes the proposed algorithm for registering multi-stained consecutive histopathological FL images. Two components, including the feature extraction and the actual transformation, are presented. In Section 6.3, extensive experimental results and the validation processes are presented. Conclusions are presented in Section 6.4.

6.2 Methods

To address the challenges of comparing content from consecutive slides stained with different stain types, nonrigid distortion, and feature-rich content, a two stage algorithm is proposed that consists of rigid initialization followed by nonrigid refinement. Both stages operate by matching *high level features*, image regions that correspond to distinct and anatomically significant features such as blood vessels, other ductal structures, or small voids within the tissue area. These matches serve as the control points for calculating spatial transformations to register the image pair. Rigid initialization estimates the rigid alignment of the image pair from the loose consensus of correspondences between anatomical

features, following the method presented in [33]. The nonrigid stage refines the initialization, by establishing a more accurate set of feature correspondences at a local scale. Initialization reduces the search for matching in the refinement stage, resulting in a lower likelihood of erroneous matches and less computation.

6.2.1 Data

The input images of FL tissue slides are digitized using a Scope XT digitizer (Aperio, San Diego, CA) at $40\times$ magnification. Tissue slides are collected from the Department of Pathology, The Ohio State University in accordance with an IRB (Institutional Review Board) approved protocol. Slides are prepared by slicing the biopsy specimen in 5 micrometer sections. Adjacent sections are stained pairwise, one of each pair with CD3 and the other with H&E. In this study five pairs of whole-slide biopsy specimens associated with multiple FL patients having different grades of the disease were used.

6.2.2 Measure for Evaluating Image Registration

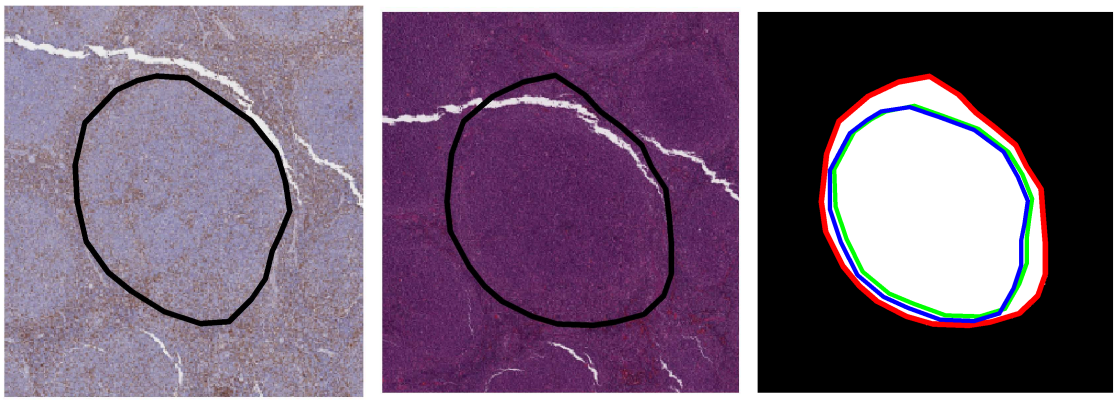
For images with the same stain type, an ideal registration would be expected to match the areas of corresponding follicles with perfect overlap, natural morphological differences aside. However, this expectation does not apply to the scenario of images with different stain types, as the difference in appearance of corresponding follicles in each stain type results in significantly different follicle boundaries. In general, the follicles in CD3-stained images appear smaller than their H&E counterparts due to the preparation process (the tissue is boiled or microwaved), and so when correctly registered the CD3 follicles only cover the interior "kernel" regions of those follicle regoins in the H&E images. As illustrated in Fig. 6.3, this fact implies a possible ambiguity in evaluating registration accuracy from a ground truth perspective in that a decision cannot be made on which result is more optimal.

However, since the aim is to identify regions of interest in the H&E image, this ambiguity will not compromise accuracy evaluation from the perspective of follicular lymphoma grading. Therefore, a performance measure is proposed as the ratio between the overlap area of the registered CD3 and H&E follicles and the area of the CD3 follicle as follows:

$$r = \frac{Area(\mathcal{T}(S_{CD3}) \cap S_{H\&E})}{Area(S_{CD3})}, \tag{6.1}$$

where S_{CD3} and $S_{H\&E}$ are follicle regions detected in the CD3 and H&E images and $\mathcal{T}$ is the transformation between the two images.

This quantity is measured for multiple manually marked follicles in each image as described in Section 6.3.



(a) (b) (c)

Figure 6.3: Overlap ratio score. The corresponding boundaries of a follicle from the CD3 image (a) and it H&E counterpart (b). As shown in (c), different registration results can produce a perfect overlap ratio score due to the differences in follicle appearance between the CD3 and H&E stains. In (c) The red line indicates the H&E follicle boundary, and the green and blue lines indicate different manual registrations of the CD3 follicle boundary to the H&E.

6.2.3 Feature Extraction

Extraction of high level features is a simple process as for most types of stains these features correspond to large contiguous regions of pixels with a common color characteristic. For each stain type, a particular color segmentation followed by morphological operations for cleanup usually suffices. Morphological opening is performed to reduce small noisy features resulting from the color segmentation, and morphological closing follows to fill in small gaps. The computational cost of these operations can be significantly reduced by performing the extraction on down-sampled versions of the original images without compromising the quality of the final nonrigid result. Fig. 6.4 demonstrates sample input and output of the extraction process.

Given the base image B, and float image F, their respective feature sets $\mathcal{B} = \{b_i\}$ and $\mathcal{F} = \{f_j\}$ are extracted according to the process described above. Each feature has associated with it a set of *descriptors* used for the matching processes, $\boldsymbol{b_i} = (\vec{\boldsymbol{x}}_i^b, s_i^b, e_i^b, \phi_i^b)$ and $\boldsymbol{f_j} = (\vec{\boldsymbol{x}}_j^f, s_j^f, e_j^f, \phi_j^f)$, where $\vec{\boldsymbol{x}} = (x, y)$ is the feature centroid, s the feature area in pixels, e the feature eccentricity, and ϕ the feature semimajor axis orientation.

6.2.4 Feature Matching

Both the initialization and refinement stages use feature matching schemes to establish correspondences between the base and float images. The following describes the conventions used for feature matching in both stages. Matches between individual features are referred to as *match candidates* if their size and eccentricity descriptors are *consistent*. That is, given the feature sets $\mathcal{B}, \mathcal{F}$, a match candidate (b_i, f_j) is established if the descriptors of size s_i^b, s_j^f and eccentricity e_i^b, e_j^f are consistent within given percent difference thresholds ϵ_s, ϵ_e

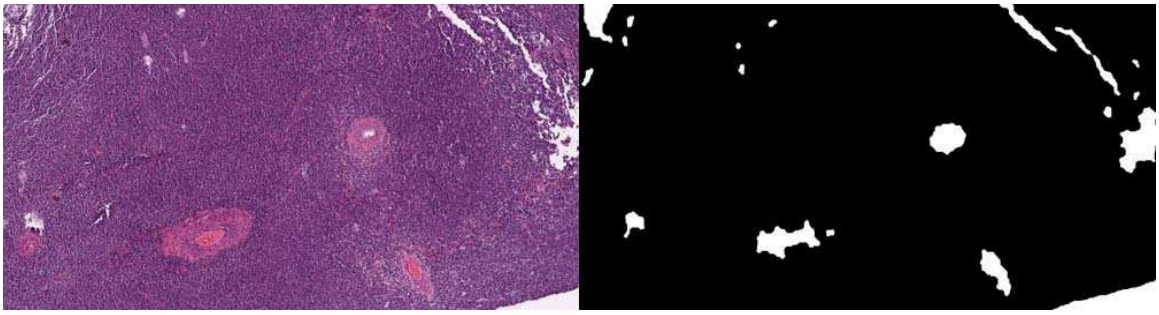


Figure 6.4: Feature extraction. This figure contains high-level feature extraction results from a typical H&E image (left). Extracted features, shown in a binary image(right), represent regions such as blood vessels recognized by the use of a combination of color segmentation and morphological operations. Descriptions of centroid location, size, eccentricity, and major-axis orientation are calculated for each feature.

$$(b_i, f_j) \Leftrightarrow \begin{cases} \frac{|s_i^b - s_j^f|}{min(s_i^b, s_j^f)} \leq \epsilon_s \\ \frac{|e_i^b - e_j^f|}{min(e_i^b, e_j^f)} \leq \epsilon_e \end{cases} . \tag{6.2}$$

If the base and float images are already roughly aligned then ϕ-consistency may also be enforced in the identification of match candidates.

Both stages also use feature matches to generate model rigid transformations $(\tilde{\theta}, \tilde{T}_x, \tilde{T}_y)$ as part of their matching schemes. Generating a model rigid transformation requires, at minimum, a pair of match candidates. To identify models originating from coherent pairs of match candidates, geometric consistency criteria are used to ensure consistent intra-image distances between feature centroids and also consistent feature orientations. For a pair of match candidates to form a *candidate pair*, $\{(b_i, f_j), (b_k, f_l)\}$, the intra-image centroid-to-centroid distances between features b_i, b_k and f_j, f_l are required to be consistent within the percent difference threshold $\epsilon_{\vec{x}}$. Additionally, for the initialization stage, the orientations of the feature semimajor axes must be consistent with the model transformation angle $\tilde{\theta}$

$$\{(b_i, f_j), (b_k, f_l)\} \Leftrightarrow \begin{cases} \frac{|\|\vec{x}_i^b - \vec{x}_k^b\|_2 - \|\vec{x}_j^f - \vec{x}_l^f\|_2|}{min(\|\vec{x}_i^b - \vec{x}_k^b\|_2, \|\vec{x}_j^f - \vec{x}_l^f\|_2)} \leq \epsilon_{\vec{x}} \\ |\phi_i^b - \phi_j^f - \tilde{\theta}| < \epsilon_\phi \\ |\phi_k^b - \phi_l^f - \tilde{\theta}| < \epsilon_\phi \end{cases} . \quad (6.3)$$

The model transformation $(\tilde{\theta}, \tilde{T}_x, \tilde{T}_y)$ for the candidate pair $\{(b_i, f_j), (b_k, f_l)\}$ is calculated by first solving for the angle $\tilde{\theta} = tan^{-1}((y_i^f - y_k^f)/(x_i^f - x_k^f)) - tan^{-1}((y_j^b - y_l^b)/(x_j^b - x_l^b))$, corrected to the interval $[-\pi, \pi]$. The translation components $\tilde{T}_x, \tilde{T}_y$ are calculated using $\tilde{\theta}$ and least squares.

The match candidate and candidate pair concepts are illustrated in in Fig. 6.5.

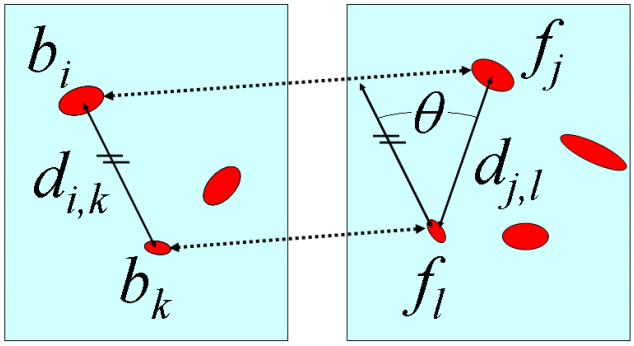


Figure 6.5: Rigid feature matching. Features are matched between the base and float images based on size and eccentricity to form *match candidates* $(b_i, f_j), (b_k, f_l)$. Intra-image distance between pairs of match candidates are compared to identify *candidate pairs*. A model rigid transformation, $(\tilde{\theta}, \tilde{T}_x, \tilde{T}_y)$, is defined for candidate pairs with consistent distances.

6.2.5 Rigid Initialization

The rigid initialization procedure is described in detail in Chapter 2.2.

Determining an estimate for rigid registration from a set of feature matches requires a method that is robust to erroneous matchings. This is especially true in microscope images

where many features are indistinguishable, and a substantial amount of mismatches are inevitable. The fundamental idea of the method presented in [33] is the recognition that any candidate pair $\{(b_i, f_j), (b_k, f_l)\}$ defines a model rigid transformation $(\tilde{\theta}, \tilde{T_x}, \tilde{T_y})$, and for carefully chosen candidate matches and candidate pairs, a large portion of the concomitant model transformations will concentrate around the desired parameters in the Euclidean transformation space. Careful choice of matches and match pairs is achieved with a set of consistency criteria enforced at two levels: between feature descriptors for matches between individual base and float features, and geometrically between pairs of such matches. With a set of model transformations identified from consistent candidate pairs, a histogram voting scheme is used to estimate the initialization parameters (θ, T_x, T_y).

Sample voting results from a follicular lymphoma image pair are presented in Fig. 6.6. The associated parameter values are presented in Table 6.1.

6.2.6 Nonrigid Refinement

The challenge in nonrigid registration is the sensitivity of computed nonrigid transformations to errors in matching, a consequence of the freedom of such transformations to accommodate distortion. In computing a relatively constrained transformation such as a rigid transformation, the effect of mismatches can be mitigated through the constraints of the transformation and least squares. For most common nonrigid transformation types the effect of a mismatched feature is certainly strong locally, and depending upon the number of matches used may also affect the registration quality globally.

For this reason the standard for establishing matches to compute a nonrigid transformation must be strict to achieve a low probability of mismatch. In the rigid stage, feature comparisons are made globally to accommodate the possibly gross misalignment of the

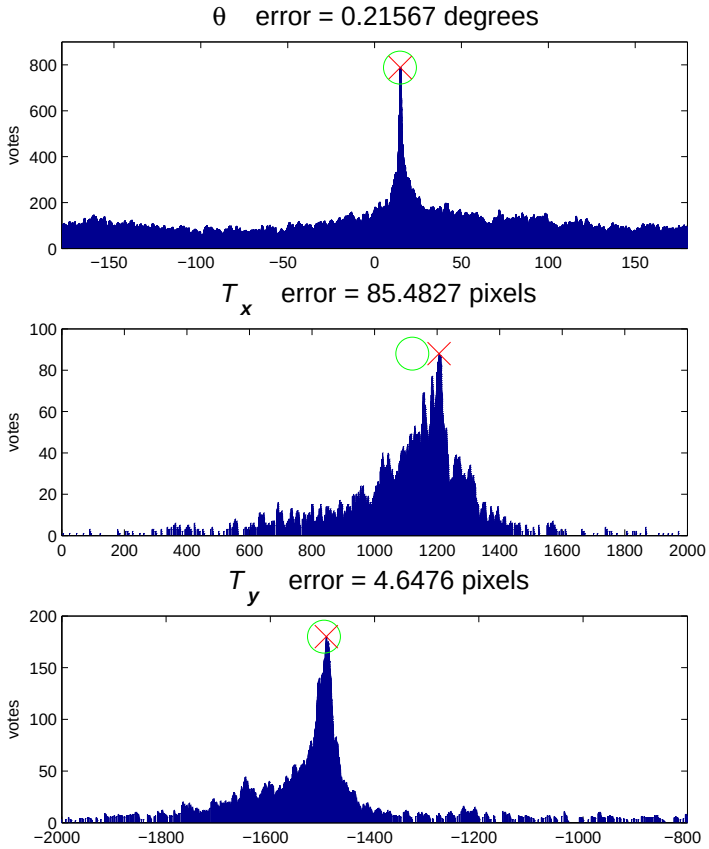


Figure 6.6: Sample histogram voting result for rigid initialization of follicular lymphoma image pair. Manual parameter results are shown in red and automatic results in green.

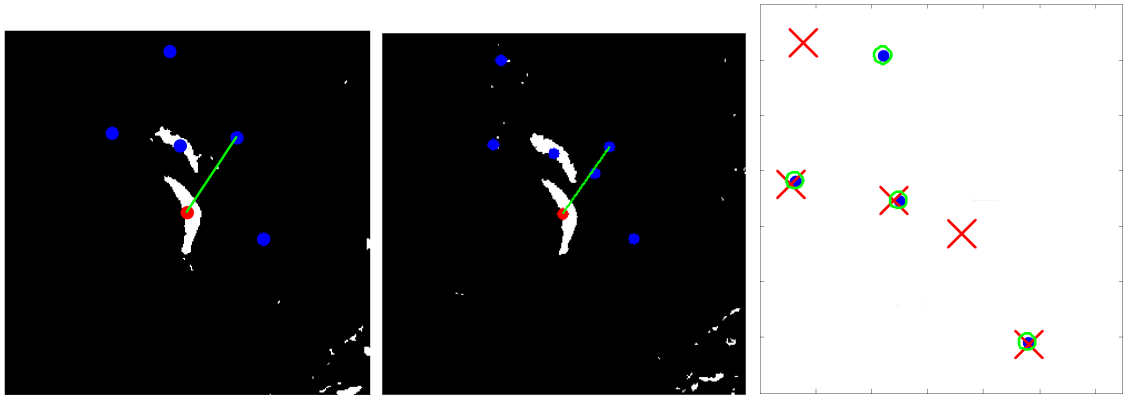
image pair. The rigid transformation is inferred from the modes of the collection of model transformations resulting from the set of all possible candidate pairs (which inevitably includes a large proportion of mismatches). Due to the presence of mismatches from model transformations surrounding these modes these candidate pairs are not appropriate input for computing the transformation of the nonrigid stage. However, the rigid initialization

provides a starting point that can reduce the search area for a stricter feature matching procedure that can reduce the likelihood of mismatching and also computation.

Given the rigid initialization, the problem of matching individual features with high confidence can be formulated as a pattern matching problem. Instead of comparing individual features solely via their descriptors, the spatial patterns formed by the collection of features within their neighborhood can be compared to increase the matching confidence. Features that match with a high degree of confidence will have similar spatial patterns of neighboring features with consistent descriptors. Since these neighborhood comparisons are made at a local scale nonrigid distortion is usually mild and local rigidity can be assumed.

Procedurally, the nonrigid matching scheme is as follows: Given feature sets $\mathcal{B}$ and $\mathcal{F}$, for each base feature b_i, the surrounding features in the R^b-neighborhood are identified. Match candidates for b_i are located in the float image within the $\mathcal{S}$-neighborhood centered at $\vec{x}_i^b$, and are matched to b_i based on size s_j^f, eccentricity e_j^f, and orientation ϕ_j^f (orientation can be used as criteria now that the images are rigidly aligned). For each match candidate f_j, the surrounding features are identified within the R^f-neighborhood of $\vec{x}_j^f$, and match candidates other than (b_i, f_j) are identified. From these other match candidates, candidate pairs are formed with (b_i, f_j), and pairs with model rotation angle $|\tilde{\theta}| > \tau$ are eliminated. The model for each of the remaining candidate pairs is used to transform the two neighborhoods, and the number of base features in R^b that fall within δ of an s, e, ϕ-consistent float feature are counted. A match (b_i, f_j) is established if the maximum count exceeds the pattern match threshold ν and $|\mathcal{R}_i^b|/2$. This process is illustrated in Fig. 6.7 and summarized in Algorithm Table 4.

Parameters for the nonrigid matching procedure have clear interpretations and can be selected by examining the features for a particular dataset. Neighborhood size R^b is chosen to capture small local networks of features, and depends on the density of features and scan magnification. The match candidate search neighborhood, S, is selected to account for error in the rigid alignment. The match neighborhood size, δ, is chosen to account for physical distortion and noise due to feature extraction including natural morphological differences. Parameter values for the dataset used in this chapter are presented in Section 6.3.



(a) (b) (c)

Figure 6.7: Nonrigid feature matching. **(a)** Locations of feature b_i (red) and surrounding features in R_i^b-neighborhood (blue). **(b)** Match candidate f_j (red) and surrounding features in the R_j^f-neighborhood (blue). Green lines in (a) and (b) indicate the pairings that generate a model local rigid transformation. **(c)** The float features of $\mathcal{R}_j^f$ (red x's) are transformed onto $\mathcal{R}_i^b$ features (blue dots). In this case, the number of base features with a consistent transformed float feature within its δ-neighborhood (green circle) is three.

Algorithm 4 Nonrigid Feature Matching

input: Feature sets $\mathcal{B}$ and $\mathcal{F}$, neighborhood sizes R^b, R^f, S, and δ, angle tolerance τ, and vote minimum ν
initialize matches $\mathcal{N} = \{\}$
apply rigid transform (θ, T) to float features and correct orientations
for each $b_i \in \mathcal{B}$
  identify $\mathcal{R}_i^b = \{b_j : \|\vec{x}_i^b - \vec{x}_j^b\|_2 \leq R^b\} \setminus b_i$
  identify $\mathcal{S}_i = \{f_j : \|\vec{x}_i^b - \vec{x}_j^f\|_2 \leq S\}$
  initialize match candidates $\mathcal{M} = \{\}$
  for each $f_j \in \mathcal{S}_i$
    compare s_i^b, s_j^f, e_i^b, e_j^f, and ϕ_i^b, ϕ_j^f
    if (b_i, f_j) s, e, ϕ-consistent **then** $\mathcal{M} = \mathcal{M} \cup \{(b_i, f_j)\}$
  end
  for each $(b_i, f_j) \in \mathcal{M}$
    identify $\mathcal{R}_j^f = \{f_k : \|\vec{x}_j^f - \vec{x}_k^f\|_2 \leq R^f\} \setminus f_j$
    identify match candidates $\mathcal{X}$ between $\mathcal{R}_i^b$, $\mathcal{R}_j^f$
    identify match pairs $\mathcal{P}$ between (b_i, f_j), $\mathcal{X}$
    for each $\{(b_i, f_j), (b_k, f_l)\} \in \mathcal{P}$
      compute model transformation $(\tilde{\theta}, \tilde{T}_x, \tilde{T}_y)$
      if $|\tilde{\theta}| \leq \tau$ **then**
        apply rigid transform $(\tilde{\theta}, \tilde{T}_x, \tilde{T}_y)$ to $\mathcal{R}_j^f$
        count $b_m \in \mathcal{R}_i^b$ within δ of consistent $f_n \in (\tilde{\theta}, \tilde{T})$-transformed $\mathcal{R}_j^f$
    end
    $c(j) = \max$ count
  end
  if $\max c \geq \nu$ **AND** $\max c \geq |\mathcal{R}_i^b|/2$ **then**
    $match = \arg\max_j c(j)$
    $\mathcal{N} = \mathcal{N} \cup (b_i, f_{match})$
end
output: $\mathcal{N}$

6.2.7 The polynomial transformation

The collection of point correspondences generated by nonrigid matching provides the information needed to form a mapping that transforms the float image into conformation with the base. A variety of nonrigid mappings are used in practice, differing in computational burden, robustness to erroneous correspondences, and existence of inverse form [27, 39, 40].

The desired transformation qualities include not only the capability to correct nonrigid distortions, but also robustness to match errors, closed inverse form, and computationally reasonable calculation and application. Of the commonly used nonrigid mapping types such as thin-plate spline, local weighted mean, affine, polynomial, and piece-wise variations, polynomial offers a good compromise between warp complexity and the aforementioned qualities. Thin plate spline provides a minimum energy solution which is appealing for problems involving physical deformation, however perfect conformity at correspondence locations can potentially cause large distortion in other areas and excess error if an erroneous correspondence exists. The lack of an explicit inverse form means the transformed image is calculated in a forward direction, likely leaving holes in the transformed result. Methods such as gradient search can be used to overcome the inverse problem, but at the cost of added computation, which can become astronomical when applied to each pixel in a gigapixel image. Kernel-based methods such as local weighted mean require a uniform distribution of correspondences. Given the heterogeneity of tissue features this distribution cannot always be guaranteed.

Polynomial warping admits an inverse form, is fast in application, and is capable of satisfactorily correcting the mild distortion encountered in sectioned images. Polynomial warping parameters can be calculated using least squares or its variants which can mitigate

the effect of matching errors. Affine mapping offers similar benefits but is more limited in the complexity of the warping it can represent.

Second degree polynomials are used for the results in this chapter. Specifically, for a point (x, y) in the base image, the coordinate (x', y') of its correspondence in the float image is

$$\begin{cases} x' = a_1x^2 + b_1xy + c_1y^2 + d_1x + e_1y + f_1, \\ y' = a_2x^2 + b_2xy + c_2y^2 + d_2x + e_2y + f_2, \end{cases} \tag{6.4}$$

Since each pair of matched correspondences provides two equations, at least six pairs of correspondences are needed to solve for the coefficients in (6.4).

6.2.8 Experimental Procedures

To demonstrate the effectiveness of the automatic nonrigid registration method, the feature extraction and registration algorithms were applied to the five image pairs described in Section 6.2.1. Magnification was reduced from 40x to 4x using Aperio's ImageScope software, resulting in images roughly $10,000 \times 7500$ pixels in size. For feature extraction, the same parameters for color segmentation and morphological operations were used for all image pairs. The automatic registration parameters, presented in Table 6.1, were also identical for all image pairs. For comparison, manual rigid and manual nonrigid registrations were also performed to the five image pairs, using eight manually selected control point pairs per image pair. A simple Euclidean transformation was used for the rigid registrations. A second degree polynomial transformation was used for the nonrigid registrations.

All computations were carried out on a dual core 2.6 GHz AMD Opteron system with 8 Gigabytes of RAM. Software was developed using a combination of Matlab, and Matlab's C/C++ interface MEX. With the RGB images loaded into memory, the entire process

Table 6.1: Summary of parameter values used in the tests and validation.

Rigid		Nonrigid	
Parameter Description	Value	Parameter Description	Value
Size similarity (ϵ_s)	0.1	Base neighborhood (R^b)	1000
Eccentricity tolerance (ϵ_e)	0.1	Float neighborhood (R^f)	1100
Distance tolerance ($\epsilon_{\vec{x}}$)	0.1	Search neighborhood (S)	250
Orientation tolerance (ϵ_ϕ)	5°	Match neighborhood (δ)	30
Voting interval for θ (ω_θ)	0.5°	$\tilde{\theta}$ angle tolerance (τ)	5°
Voting interval for T (ω_T)	30	Pattern match minimum (ν)	4

executes in two minutes for a single image pair. Less than one second of that is devoted to the nonrigid matching procedure.

Visual inspection of the feature extraction results revealed that features in two of the five image pairs are not uniformly distributed, being concentrated almost entirely in one half of the tissue area in each case. In regions where features are sparse, nonrigid refinement matches are hard to establish since it is difficult to identify coherent networks of features at a local scale. This can result in spatially clustered control points, and depending on the severity of distortion between the slides, a transformation that is significantly biased to the feature-rich areas of the tissue. The validation analysis that follows is carried out separately on these challenging image pairs and the feature regular image pairs, to illustrate the importance of feature input and the expected outcome if a sufficient feature set can be identified.

6.2.9 Validation

The procedure for registration validation was motivated by the application of automated FL grading. The goal in this application is to correctly register follicle regions so that follicle segmentations from the CD3 image can be used to direct grading analysis in

the counterpart H&E image. To evaluate registration performance in the context of this application, the overlap of manually identified follicle regions was compared for different registration methods.

For each H&E/CD3 image pair, five corresponding test follicle pairs were selected. The boundaries of each of these test follicle pairs were then marked by five different observers. The same test follicle pairs were marked by each observer, generating a total of 25 follicle pair markings per observer. The overlap ratio demonstrated in Figure 6.3 was then computed for every follicle test pair marking using the manual rigid, manual nonrigid, and automatic nonrigid registrations for each image pair. These overlap ratios for observer i, image pair j, and follicle test pair k are denoted as $\boldsymbol{Rigid}_i(j,k)$, $\boldsymbol{Manual}_i(j,k)$, and $\boldsymbol{Auto}_i(j,k)$ respectively. The feature regular image pairs are the set $j \in \{1,2,3\}$ and the challenge image pairs are the set $j \in \{4,5\}$.

This validation aims to illustrate two points: 1. that nonrigid registration is beneficial in terms of follicle overlap and 2. that the automatic nonrigid registration is comparable to a reasonable manual nonrigid registration. These points are addressed with three statistical analyses: the boxplot graphical analysis, significance testing by paired t-test, and the Bland-Altman graphical analysis.

The boxplot is a graphical analysis that presents the distributions of the overlap ratios for feature image pairs, separated by both registration method and observer. The median, inner-quartile range, and outliers are plotted for each observer-method set, $\{\boldsymbol{Method}_i(j,k)\}$, $\forall (j,k) \in \{1,2,3\} \times \{1,\ldots,5\}$, for some i.

To demonstrate the similarities of manual nonrigid registrations, significance testing was performed on these observer-method sets using the *paired t-test*. For each observer i, the overlap ratios were paired by method for all follicles in the feature regular image

pairs, $\{(\boldsymbol{Manual}_i(j,k), \boldsymbol{Auto}_i(j,k))\}\forall(j,k) \in \{1,2,3\} \times \{1,\ldots,5\}$. The t-statistic was calculated for these method-pair sets,

$$t_i = \overline{D}_i\sqrt{\frac{15}{\sigma_i^2}}, \tag{6.5}$$

where $\overline{D}_i$ and σ_i^2 are the mean and variance

$$\overline{D}_i = \sum_{j=1}^{3}\sum_{k=1}^{5}(\boldsymbol{Auto}_i(j,k) - \boldsymbol{Manual}_i(j,k)),$$

$$\sigma_i^2 = \frac{1}{15-1}\sum_{j=1}^{3}\sum_{k=1}^{5}(\boldsymbol{Auto}_i(j,k) - \boldsymbol{Manual}_i(j,k))^2.$$

The t-statistic t_i was compared against the Student's t distribution to compute the p-value p_i.

To further illustrate the similarities between automatic and manual nonrigid registrations, a Bland-Altman graphical analysis was performed. The Bland-Altman analysis is commonly used in biostatistics to examine the extent of agreement between to distinct measurement methods [93, 94]. It is included here because it illustrates the performance of the automatic and manual methods well. It is noted, however, that comparing the overall performance of two registration methods is fundamentally different from the assessment of the agreement of measurement methods. In the case of measurement assessment, agreement between individual samples is critical, since the measurements intended to provide the same information about some underlying physical state. In registration, follicle overlaps may disagree individually between methods, but the collection of overlaps may still indicate comparable performance.

For each observer i, the difference $d_{j,k}$ and mean $\mu_{j,k}$ were computed

$$\mu_{j,k} = \frac{\boldsymbol{Auto}_i(j,k) + \boldsymbol{Manual}_i(j,k)}{2} \tag{6.6}$$

$$d_{j,k} = \boldsymbol{Auto}_i(j,k) - \boldsymbol{Manual}_i(j,k), \tag{6.7}$$

and the mean and difference tuples $(\mu_{j,k}, d_{j,k})$ were plotted for all follicles in the feature regular image pairs. Along with the mean and difference tuples, the average-difference and 95% confidence intervals are plotted to provide information on the mean performance of the methods and their range of agreement.

Finally, a simple analysis is performed to demonstrate the spatial variation of registration quality in the challenge image pairs. For each follicle k, the overlap ratios $\boldsymbol{Auto}_i(j,k)$, $j \in \{4,5\}$ are averaged over observer i.

6.3 Results

The boxplot is presented in Figure 6.8. The corresponding means and standard deviations of the observer-method sets are presented in Table 6.2. Comparing manual rigid and manual nonrigid registrations, the nonrigid registration improves the mean overlap ratio for all markings except those of observer two, demonstrating the benefit of correcting nonrigid distortion. Mean overlap ratios for automatic nonrigid registration are comparable to manual nonrigid, with slight improvements noted for the markings of three observers.

Table 6.2: Mean overlap ratios and standard deviations for observer-method sets of feature regular image pairs.

Observer i	$\boldsymbol{Rigid}_i$ mean $\pm$ s.d.	$\boldsymbol{Manual}_i$ mean $\pm$ s.d.	$\boldsymbol{Auto}_i$ mean $\pm$ s.d.
1	0.8943 ± 0.0930	0.9373 ± 0.0889	0.9306 ± 0.1152
2	0.9223 ± 0.0667	0.9190 ± 0.0950	0.9213 ± 0.0718
3	0.9428 ± 0.0838	0.9520 ± 0.0617	0.9562 ± 0.0477
4	0.9167 ± 0.0850	0.9278 ± 0.0969	0.9316 ± 0.0727
5	0.9247 ± 0.0732	0.9384 ± 0.0691	0.9351 ± 0.0614

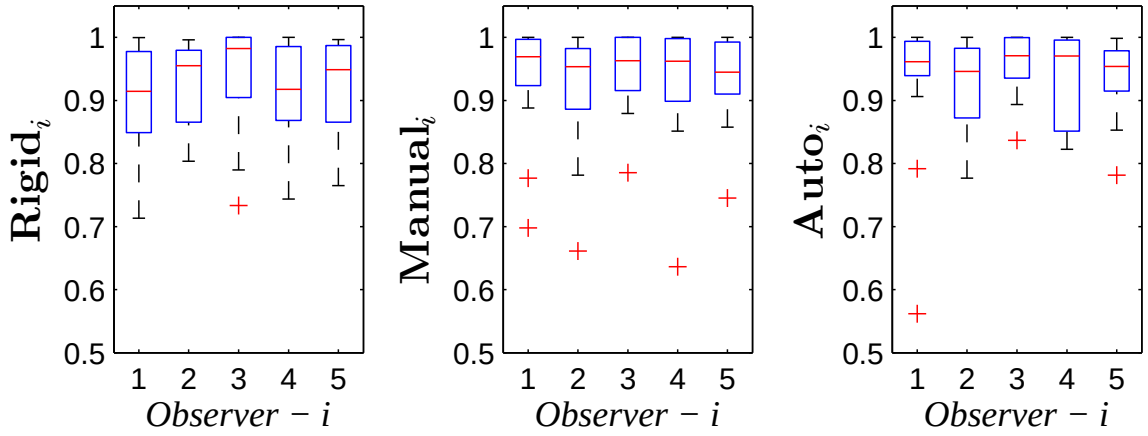


Figure 6.8: Boxplots of overlap ratios for observer-methods sets from feature regular image pairs. Outlier overlap ratios from poorly registered follicles are indicated by red cross markers. Mean performance is comparable between manual nonrigid and automatic nonrigid registrations.

The p-values for the t-statistics of the method-pair sets are presented in Table 6.3. These p-values range from 0.79 to 0.93 indicating no statistically significant difference between the manual and automatic methods.

Table 6.3: Significance values of paired t-tests for method-pair sets from feature regular images $\{(\boldsymbol{Manual}_i(j,k), \boldsymbol{Auto}_i(j,k))\}$. The p-values indicate no statistically significant difference between the overlaps for manual and automatic nonrigid registration methods.

Observer i	1	2	3	4	5
p_i	0.7981	0.9301	0.8194	0.8901	0.8905

The Bland-Altman plot is presented in Figure 6.9. Tuples plotted above zero indicate better performance for the automatic method. The average-difference is nearly zero for all observers. Most tuples are clustered tightly in the center right of their plot, indicating a

high average overlap and small difference for the manual and automatic methods. Each observer has at least one outlier tuple with a difference beyond the 95% confidence limits. For each outlier tuple indicating superior performance for the manual registration, there is a complementary tuple indicating superior performance for the automatic method.

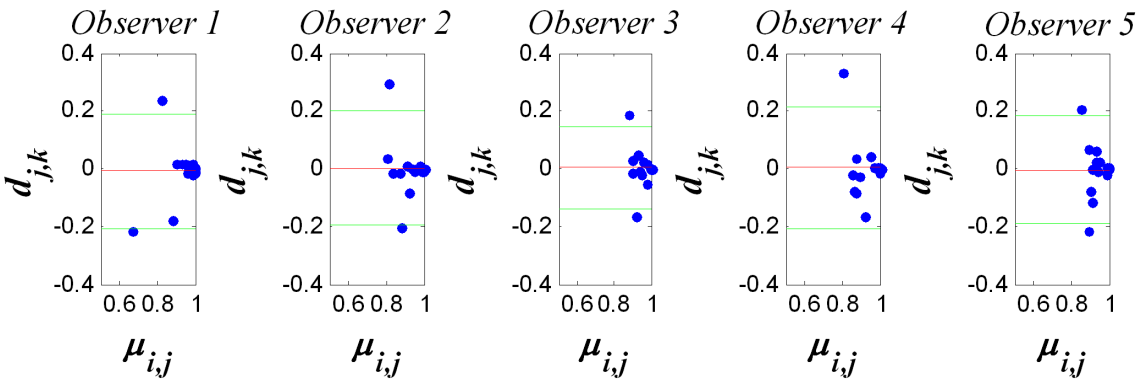


Figure 6.9: Bland-Altman analysis of manual and automatic nonrigid registrations. Average difference is indicated in red. The 95% confidence limits are indicated in green.

The overlap results from the challenge image pairs illustrate the impact of feature input to the automatic nonrigid registration. Where the test follicle pairs were chosen uniformly throughout the extent of the tissue, the features in the challenge image pairs were not uniformly distributed, resulting in a transformation that is biased to feature-rich areas. The overlap ratios of Table 6.4 demonstrate this point, where test follicles located in feature rich regions show comparable quality and others apparently suffer from a lack of proximal feature matches.

Table 6.4: Challenge image pair overlap ratios $\boldsymbol{Auto}_i(j,k)$, separated by follicle k, and averaged over observers i.

Follicle k	1	2	3	4	5
Image Pair $j=4$	0.8482	0.2601	0.0505	0.8377	0.9366
Image Pair $j=5$	0.9189	0.4540	0.9187	0.8862	0.9886

6.4 Discussion and Conclusions

This chapter presents a method for the nonrigid registration of distinctly stained follicular lymphoma section images. As a key step for fusing the information extracted from images of two different stains, i.e., IHC and H&E, computerized registration serves as a bridge that allows for the combination of valuable information otherwise unique in each resource in a meaningful way. In this particular study, the registration step makes it possible to recognize salient features from both stained images and map the follicle boundaries detected in IHC images to appropriate locations in H&E images. As a consequence, further grading analysis can proceed with H&E counterparts where cellular level analysis is favorable. In the end, by providing accurate follicle boundaries on the H&E images, the registration contributes to more precise CB count, the essential step in the FL grading process.

The automatic matching method presented in this chapter offers a solution for applications such as microscopy imaging, where a large number of nondescript features are to be matched with high-fidelity. Matching such features individually is a high probability-of-error endeavor, and matching errors can result in poor conformation between the registered image pair due to the freedom of nonrigid transformations. Here, confidence in matches

between individual features is enhanced by verifying the existence of coherent networks of features in the surrounding areas.

In terms of registration accuracy, the quality of transformations derived from automatic matching depends on the ability to extract features throughout the extent of the tissue area. When excluding the image pairs where extracted features are sparse and highly spatially clustered, the registrations based on automatic matching are indistinguishable from those based on the manual nonrigid method. This suggests that the registration framework could benefit from a more sophisticated feature extraction process. However, in practice, poorly registered follicles located in feature sparse areas could possibly be avoided by analyzing the spatial distribution of feature matches and their proximities to each follicle.

CHAPTER 7

REGISTRATION VS. RECONSTRUCTION: INCORPORATING STRUCTURAL CONSTRAINT IN BUILDING 3-D MODELS FROM 2-D MICROSCOPY IMAGES

The methods of Chapters 2, 5, and 6 enable the nonrigid registration of large microscopic images in a variety of scenarios. By establishing correspondences, either through intensity feature or high-level feature matching, the nonrigid distortions in section images can be corrected and the tissue reconstructed. In some cases though the freedom that nonrigid transformation provides has the unintended consequence of distorting the 3D structure of the biological specimen. This is similar to the data modeling problem of overfitting: by forcing features to conform perfectly the low-frequency trends in the 3D tissue reconstruction can be obscured.

In this chapter I demonstrate the overfitting phenomenon and present a method for the reconstruction of tissues that preserves 3D structure. The proposed method is entirely novel as the overfitting problem has not yet been demonstrated for reconstructions from sequences of sectioned images. The special case of tissues containing duct-like structures is addressed. By automatically tracking duct trajectories through an image sequence a structural constraint is created that permits nonrigid reconstruction without structural distortion. The structurally constrained reconstruction process is fully automatic and is demonstrated on a set of 160 mouse mammary images.

7.1 Introduction

Given a sequence (e.g., 200) of microscopy images taken as consecutive sections from a mouse mammary gland, the goal of reconstruction is to infer the 3D structure of the tissue, in this case specifically to study the microanatomy of the ductal structures in the mammary gland. Due to the prevalence of soft tissues in the sample, section images typically contain various distortions (e.g., bending, shearing, and tearing) and thus a pairwise nonrigid registration of the sequence is used as the traditional reconstruction approach. A common issue encountered with reconstruction is the evaluation/validation of the reconstructed tissue since there is no ground truth available. Does the reconstructed tissue meet reasonable expectations given the newly visible 3D anatomy? This question implies a fundamental problem with the traditional approach: the lack of structural constraint in the reconstruction process. As the registration is performed pairwise over the image sequence, only the consistency between any two images is considered, with the 3D anatomical structures usually serving as evaluation criteria rather than as a constraint to the reconstruction process. One consequence of this sequential approach is illustrated in Figure 7.1 in which two ductal structures are reconstructed as straight columns through perfect pairwise registration. All of the trajectory in the x-direction and y-direction (within the image plane) is lost. In other words, the traditional approach to reconstruction is more a *registration-for-registration* than a *registration-for-reconstruction*.

In this chapter a different approach is taken to the reconstruction problem by incorporating structural constraints into the processing pipeline. The incorporation of structural constraints implies that prior domain-specific (biological) knowledge is required. This chapter uses the example of one of the most commonly accepted structural constraints: the smoothness of ductal structures. In tissues, ductal structures such as blood vessels, lymph space,

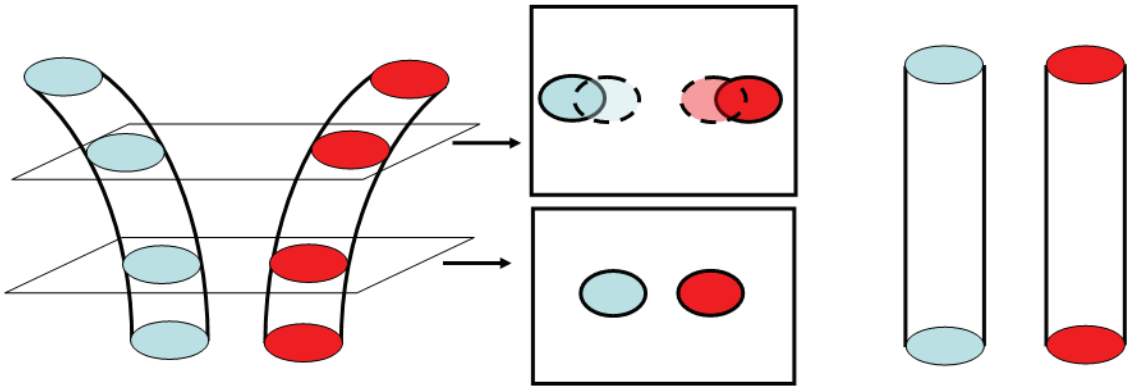


Figure 7.1: Left: original ductal structures sliced at different positions. Middle: the images for the ducts. Right: after the registration, the reconstructed ducts are column-like structures.

and mammary gland ducts are commonly encountered. It is generally assumed that these structures traverse smoothly throughout the tissues, neither jagged or perfectly straight.

An advantage of focusing on these ductal structures is that they are typically easily identifiable within an image and can be easily extracted. These types of features easily qualify as high-level features, and as demonstrated in Chapter 6 can be used to avoid expensive and sometimes error-prone comparisons of intensity information between images. This is especially useful in fast registration for large microscopy images datasets. The only assumption is that the nonrigid distortion of the sample is mild and so once an image pair is rigidly registered subsequent operations can be performed in a limited locale.

7.2 The Reconstruction Pipeline

The reconstruction process is composed of three main stages:

1. **Fast Rigid Registration.** Rigid registration can be achieved using either optimization based approach such as MMI [95] or high-level feature based approach such

as those in Chapter 2 or [96, 97]. In addition, specific methods can be applied to provide good initial estimate on registration. In the mammary gland example, the tissue samples have an elongated shape and principal component analysis (PCA) can be used to determine the principal direction of the tissue which is used to estimate the rotation angles between images. The purpose of this stage is to find the rigid transformations between the images which facilitates the matching of corresponding high-level features by narrowing down the search area.

2. **Duct Tracking.** In the example of this chapter the high-level features correspond to ductal structures. In most cases, these features can be easily segmentated via color space segmentation (see Figure 7.2). For instance, blood vessels usually have distinct red color and mammary gland ducts are distinct dark structures embedded in the light-colored adipose tissues. After the ductal structures are segmented and the images rigidly aligned, correspondences for each duct are located via search by normalized cross correlation. The centroids of the duct regions are then linked together between each image pair to form a trajectory. Due to the nonrigid distortion these trajectories tend to be jagged.

3. **Trajectory Smoothing and Transformation.** The trajectory of each duct is then smoothed using a smoothing filter. These smoothed trajectories then serve as the structural constraint. Nonlinear transformations such the thin-plate spline are then applied to each image to move the ducts to the locations of their respective smoothed trajectories. Thus instead of sequentially registering each duct to its neighbor, the ducts are registered to the desired structural configuration that incorporates information from more distant neighbors (see Figure 7.3).

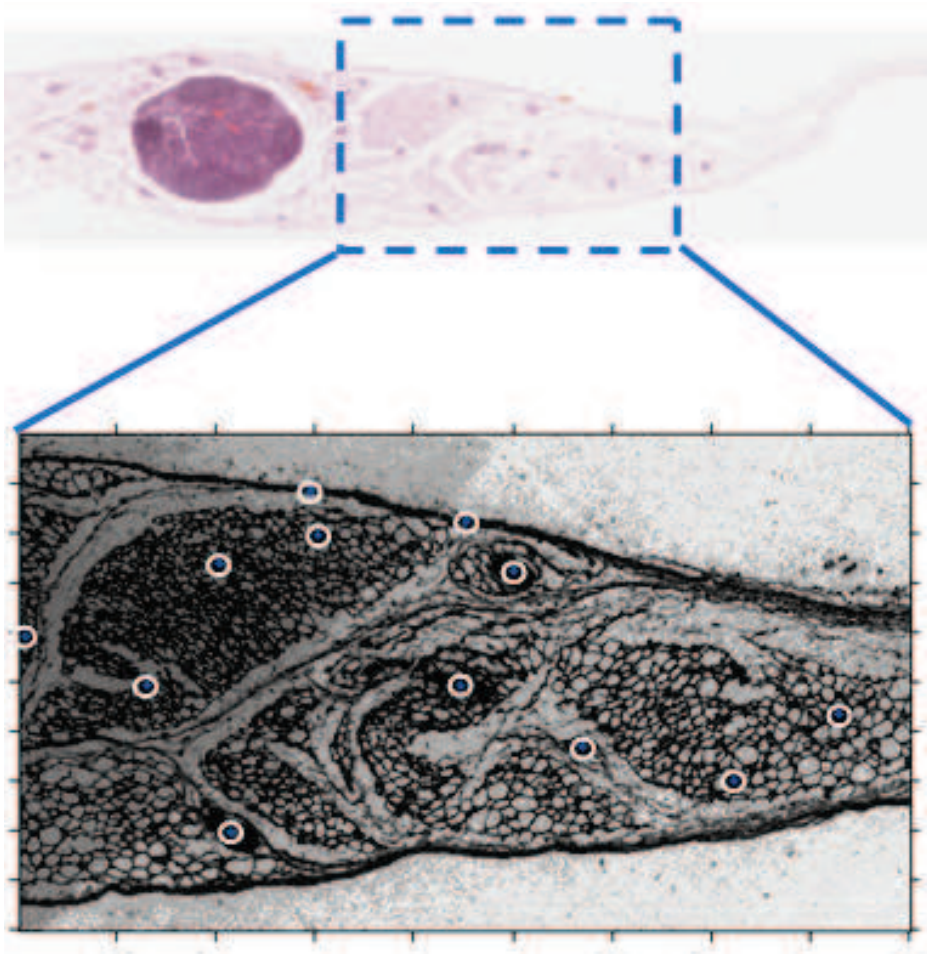


Figure 7.2: Registration of ducts to smoothed trajectories. The trajectories for each duct are tracked through the sequence of rigidly registered images. The resulting trajectories are smoothed, and the duct centroids are then nonrigidly registered to the smoothed trajectories.

7.2.1 Duct Tracking

Given a sequence of M rigidly registered images $i = 1, 2, \ldots, M$, the duct centroids for duct d_j^i are denoted as $\vec{\boldsymbol{x}}_j^i = (x_j^i, y_j^i)$. Starting with the ducts from image 1, a $T \times T$ template is taken from image 1 surrounding each duct d_j^1 with center $\vec{\boldsymbol{x}}_j^1$. A corresponding $S \times S$ search window is taken from image 2, with $S > T$ (typically $S = 1.5T$), also centered at $\vec{\boldsymbol{x}}_j^1$. The normalized cross correlation is computed between the T, S as in Equation 2.6.

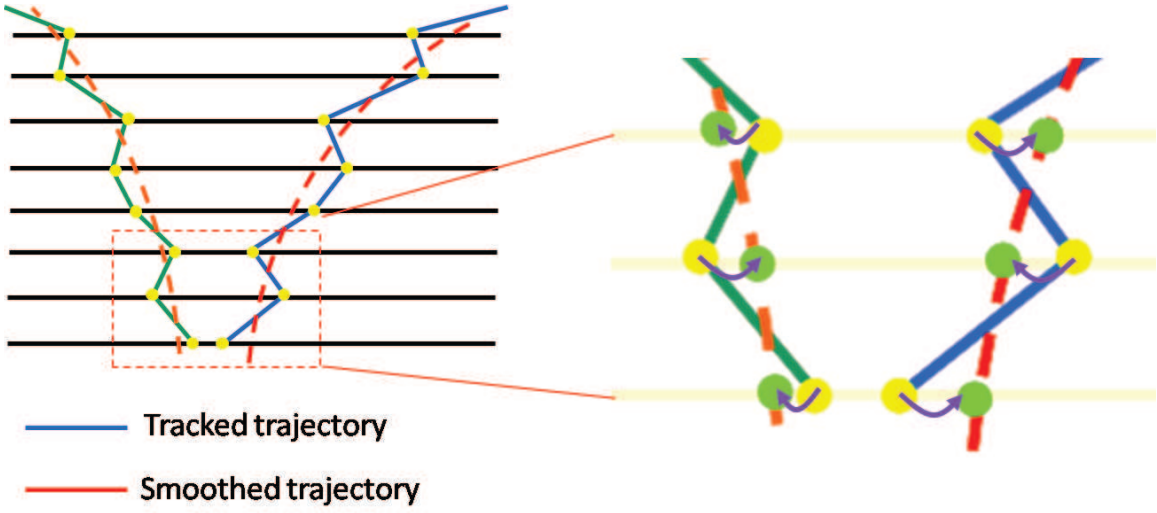


Figure 7.3: Registration of ducts to smoothed trajectories. The trajectories for each duct are tracked through the sequence of rigidly registered images. The resulting trajectories are smoothed, and the duct centroids are then nonrigidly registered to the smoothed trajectories.

The location of the maximum correlation is

$$(m, n) = \arg\max_{u,v} \rho(u, v).$$

If $\rho(m, n)$ exceeds a given threshold τ (typically 0.8), then the duct d_j^1 is linked to the duct d_k^2 with centroid nearest to the maximum correlation

$$k = \arg\min_{k}((x_j^1 + \frac{T-S}{2} - x_k^2)^2 + (y_j^1 + \frac{T-S}{2} - y_k^2)^2), \tag{7.1}$$

$$\rho(m, n) \geq \tau \Rightarrow d_j^1 \leftrightarrow d_k^2. \tag{7.2}$$

If a match satisfying the threshold cannot be identified, then the trajectory is terminated. If a collision occurs, that is, if two ducts in image 1 both match to the same duct in image 2 then the trajectory of the duct with lower maximum correlation is terminated. This process repeats for each image pair $i, i + 1$, extending the linkage for each duct as far as possible. Each unmatched duct in image $i + 1$ marks the start of a new trajectory at iteration $i + 1$.

The result is a sequence of linked ducts, each resembling $d_j^i \leftrightarrow d_j^{i+1} \leftrightarrow \ldots \leftrightarrow d_j^{i+D_j-1}$ for some D_j. Each linked duct has associated with it a 3D trajectory of the duct centroids which form the sequences

$$\vec{\boldsymbol{X}}_j[z] = \vec{\boldsymbol{x}}_j^z, \quad X_j[z] = x_j^z, \quad Y_j[z] = y_j^z, \qquad z \in \{i, i+1, \ldots, i+D_j-1\}. \tag{7.3}$$

7.2.2 Trajectory Smoothing and Transformation

Any number of techniques can be applied to smooth the trajectory sequences. The simplest approach is to apply a low pass filter to $X_j[z]$ and $Y_j[z]$ independently to form the smoothed trajectories

$$\bar{X}_j[z] = a_0 X_j[z] + a_1 X_j[z-1] + \ldots + a_N X_j[z-N], \tag{7.4}$$

$$\bar{Y}_j[z] = a_0 Y_j[z] + a_1 Y_j[z-1] + \ldots + a_N Y_j[z-N]. \tag{7.5}$$

The drawback of this smoothing approach is that the coupling between the X and Y directions is not taken into account. A more sophisticated approach using spline fitting could simultaneously incorporate information in both the X and Y directions.

The smoothed trajectories serve as the structural constraint for the 3D reconstruction. For any value of z the duct trajectory $\vec{\boldsymbol{X}}_j[z]$ incorporates not only the information from one neighbor (as it would with a pairwise scheme), but information from several neighbors. In the case of the traditional pairwise registration scheme, the information is flows in only one direction, duct d_j^i+1 is fixed to the same location of duct d_j^i, so that all subsequent ducts are fixed to the location of the first. In the structure preserving scheme the smoothing procedure can use an *acausal filter* to incorporate information from both directions, backwards and

also forwards in z

$$\bar{X}_j[z] = b_0 X_j[z + \frac{N}{2}] + \ldots + b_{\frac{N}{2}} X_j[z] + \ldots + b_N X_j[z - \frac{N}{2} - 1], \tag{7.6}$$

$$\bar{Y}_j[z] = b_0 Y_j[z + \frac{N}{2}] + \ldots + b_{\frac{N}{2}} X_j[z] + \ldots + b_N Y_j[z - \frac{N}{2} - 1]. \tag{7.7}$$

This way smoothing not only looks to where the duct has been, but also to where the duct is going.

With the smoothed trajectories computed, what remains is to nonrigidly register the duct centroids to the smoothed locations. For each image i the structural constraint as it lies within the same image plane is used as an atlas for registration. The centroid of each duct $(X_j[i], Y_j[i])$ is assigned to the smoothed location $(\bar{X}_j[i], \bar{Y}_j[i])$ to form a control point. The control points are then used to calculate a transformation based on the *thin plate spline* which guarantees perfect conformity of the centroids to the designated smoothed locations [98]. This transformation is calculated for each image and then applied to map the image to its structural constraint.

7.3 Results

The structural constraint registration pipeline was implemented in Matlab and applied to a set of 160 mouse mammary gland images (600×7500 pixels) as shown in Figure 7.4. Rigid registration was performed using PCA and MMI. The images were converted to grayscale and the ducts were identified using a segmentation via thresholding combined with morphological erosion to remove cell membranes of the adipose tissue. Each duct is tracked with the resulting trajectories shown in Figure 7.5. The trajectories were smoothed using a fifth order acausal low pass filter. The entire segmentation, tracking, and smoothing process took several minutes. Each image was transformed using the thin-plate spline method.

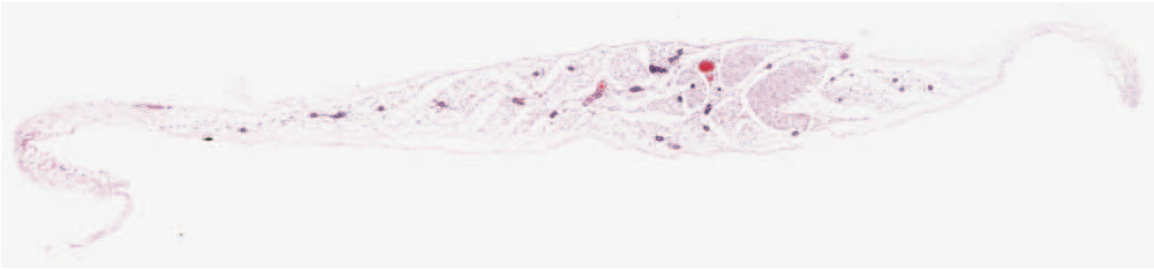
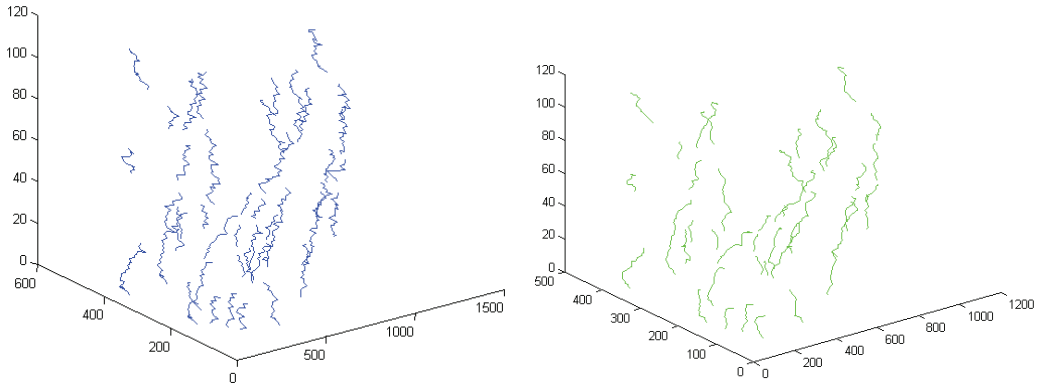


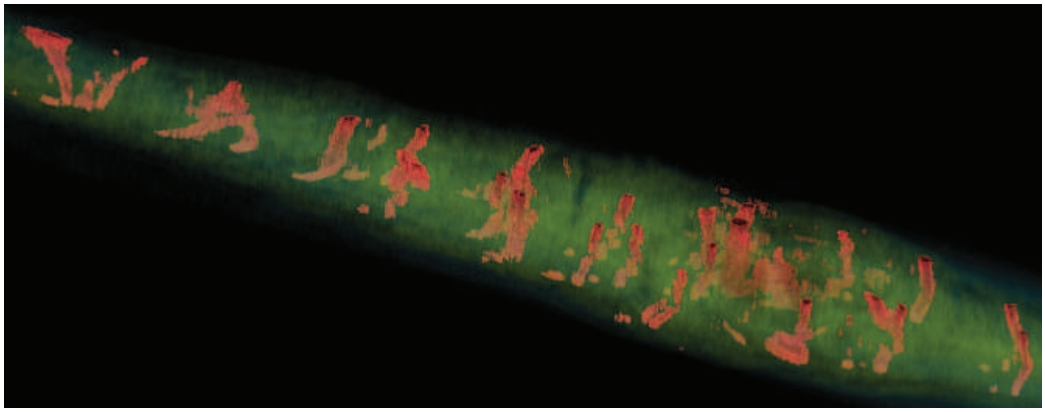
Figure 7.4: Sample mouse mammary gland image.



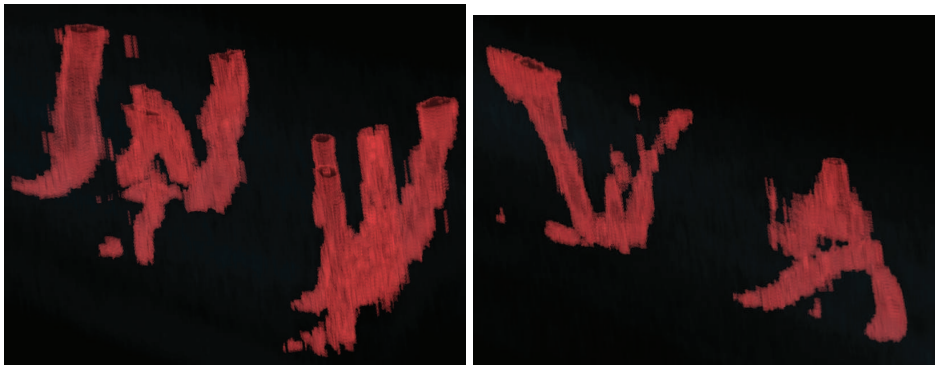
(a) (b)

Figure 7.5: Duct trajectories. (a) Unsmoothed trajectories. (b) Smoothed trajectories.

Figure 7.6 shows several views of the reconstructed ducts in 3-D space. The volumetric rendering is generated using VolSuite, a volumetric rendering software developed at the Ohio Supercomputing Center. From the detail views of the individual ducts it is apparent that the trajectory components lying within in the image xy-plane are not destroyed. Duct bifurcations are also visible. Compare this to the reconstruction without structural constraint shown in Figure 7.7. The traditional pairwise approach that sequentially stacks the duct centroids destroys the xy-components of the duct trajectories.



(a)



(b) (c)

Figure 7.6: Mouse mammary reconstruction with structural constraint. (a) Rendering of the reconstructed mouse mammary gland ducts. (b)-(c) Detailed views of the individual ducts.

7.4 Discussion and Conclusions

This chapter presents a novel approach for the 3D reconstruction of tissue from serial section images. The key contribution is the integration of a structural constraint into the reconstruction process. As opposed to the traditional pairwise sequential registration approach that infers structure from images one pair at a time, the proposed method uses information from multiple images to enforce a structural criteria. The motivating example

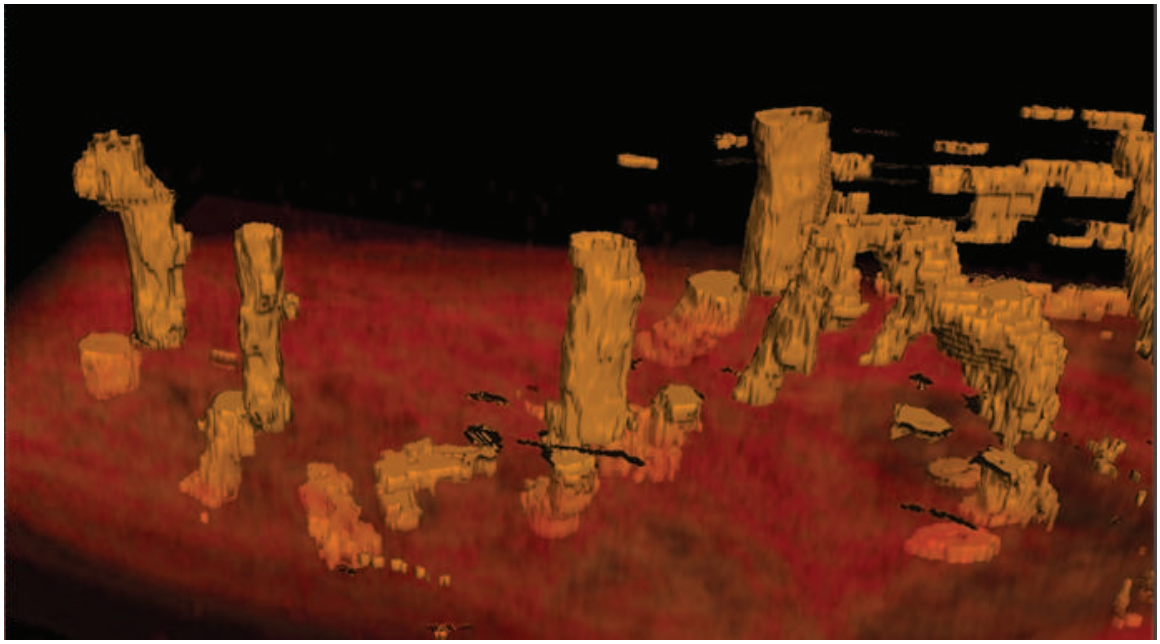


Figure 7.7: Mouse mammary reconstruction using traditional pairwise sequential registration. The ducts are reconstructed as straight columns void of any trajectory components within the image xy-plane.

of reconstructing mammary ducts provides a significant example of the benefits of this approach. By imposing a smoothness criteria the ducts can be registered naturally resulting in reconstructions with visible bifurcations. The use of an acausal smoothing filter enables the smoothing process to take into account not only where the duct has been but where it is heading. The entire process is fast, automatic, and produces credible representations of the morphology of structures of interest.

CHAPTER 8

TWO POINT CORRELATION FUNCTIONS

The value of the reconstructions presented in the previous chapters goes beyond visualization of tissues and microanatomy. As indicated in the introduction, reconstruction is only one element in the proposed image analysis pipeline. Since biological phenomenon are not contained to two-dimensional space, a complete picture of the tissue environment requires off-plane information, and so reconstructions serve as the starting point for many deeper quantitative analyses. Depending on the motivations any number of investigations can be performed on a reconstructed volume including morphological analysis of tissue layers, or an examination of the distributions and localization of different cell types. It is clear that in many cases identifying the tissue boundaries is a requirement for deeper quantitative analysis at the tissue or cellular levels of organization. This is known as the *tissue segmentation* problem, and will be the focus of the remaining chapters of this document.

The segmentation of tissues in histological images is a challenging problem due to both image content and size. The content of microscopic images is textural in nature, consisting of highly self-similar patterns of cellular and subcellular structures. The visual cues that distinguish one tissue from another are varied and include color, scale, and shape. Difference in these distinguishing characteristics from one tissue to another may be subtle even to a trained observer. In addition to challenging content, the size of histological images

tends to be very large, on the order of hundreds of millions or billions of pixels each. Altogether this creates a difficult scenario for the application of traditional image segmentation features. A segmentation scheme must be complex enough to incorporate the varied cues that distinguish tissues, but not so complex that it is computationally infeasible.

Fortunately these qualities also describe the content of images used for studies in a related discipline: the science of heterogeneous materials. The study of the physical properties of heterogeneous materials has many parallels with tissue analysis. At the microscopic resolution the microstructure of composites also often appears highly textural, consisting perhaps of "cells" of one or more substances of different sizes and shapes embedded within a another material.

In the pursuit of characterizing the physical properties of materials, a rich framework of *stochastic geometric* methods has been developed by the materials science community. This framework has been previously adapted for the segmentation of tissues in microscopic images. In particular, the *two point correlation functions* have been demonstrated as an effective feature for tissue segmentation. In this chapter I contribute several significant developments to the existing two point function segmentation methods. A fast and deterministic method for the calculation of two-point functions is presented. The two point functions are demonstrated to possess a peculiar low-dimensional structure in feature space that can be exploited for unsupervised segmentation. Furthermore it is shown that images can be segmented effectively using only a limited set of two point functions, the autocorrelation functions, resulting in a considerable reduction in computation. In light of these developments the effectiveness of the two point function as a feature for tissue segmentation is demonstrated on human follicular lymphoma and mouse placenta images.

This chapter is organized as follows: Section 8.1 casts the tissue segmentation problem in the light of heterogeneous materials and provides an overview of the tissue segmentation problem and the relevant research on heterogeneous materials and image segmentation. Section 8.2 describes the stochastic geometric tools with a focus on the two point correlation function. The segmentation algorithm based on two-point correlation function features is described in Section 8.3. Experimental results are provided in Section 8.4, including experiments performed on tissue and natural texture images. Section 8.5 contains a discussion of the results and conclusions.

8.1 Introduction

In the context of materials science, a *heterogenous material* is a substance composed of multiple materials, either a composite of distinct materials, or the same material in different physical phases. Examples include porous single materials (where the constituents are solid phase or void), soils, concrete, fluid suspensions, and biological tissues. A comprehensive overview of heterogeneous materials is available in [99]. Scientists have pursued descriptions of the macroscopic properties of heterogeneous materials through examination of their microscopic structure for more than one hundred years. Macroscopic properties like electrical conductivity, magnetic permeability, fluid transport properties such as trapping time, and physical properties such as elasticity all have roots in the microstructural characteristics of materials. A large collection of publications now exist that develop a rigorous and generalizable analytical framework for predicting macroscopic properties from knowledge of material microstructure [100–103].

In this chapter the analytical tools for heterogeneous materials are borrowed for the purpose of segmenting tissues in histological images. The methods of stochastic geometry provide the material scientist with measurements of statistics on the shape, size, and spatial arrangements of components in a heterogeneous material. Treating tissue as a heterogeneous material, a composite of biologically meaningful elements such as nuclei, cytoplasm, or cells of different types, these methods can provide similar statistics for the shape, size, and arrangement of these meaningful elements. With the understanding that the qualities of these elements vary from one tissue to another, the aim is to employ the stochastic geometric framework to derive robust features that are capable of distinguishing tissues.

8.1.1 Background

Image segmentation is one of the fundamental problems in image processing and computer vision and has been studied now for decades. Techniques include thresholding [104], region growing [105], histogram [106], edge detection [107], graph based [108], model based [109], multi-resolution [110], and level set methods [111]. Many of the works on texture image segmentation [112–114] and medical image segmentation [115, 116] make use of co-occurrence based methods that are closely related to the two point correlation function, as described in Section 8.2.

Most of the general approaches to image segmentation are represented in the works on microscopic image segmentation. The segmentation of subcellular structures such as nuclei and individual cells has been demonstrated using watershed [117], graph-based [118], level sets [119], and markov random field [120] approaches. Similar approaches are used for the segmentation of clusters of cells []. While there are an abundance of works on segmenting sub-cellular structures, individual cells, and cell clusters, there are relatively few works on

the segmentation of tissues. Most of the existing approaches are focused on specialized cases rather than offering generalizable methods. Blood vessels are segmented using a neural network classifier with color information in [121]. A graph-based method for identifying the interior boundaries of ducts in mammary tissue images is presented in [122]. A level sets method with fast-marching initialization is also used to identify mammary ducts in [123]. A more generalizable color histogram based method using a Bayesian classifier with color histogram features was proposed in [16]. The authors in [124] develop an object-based approach to segmentation that was demonstrated for segmenting cancerous regions in colon biopsy images. This method follows the example of the earlier methods described below in treating the tissue as a collection of discrete and biologically meaningful elements.

The N-point correlation functions were first proposed for the segmentation of tissues in [125], where a high-order SVD classifier was used for supervised segmentation of mouse placenta tissue layers. This provided promising results however the work was only validated using a single image. A more extensive validation was performed in [15] that incorporated more placenta images. This validation demonstrated that the NPCFs performed significantly better than both Haralick and Gabor features for the placenta tissue. An effort was made at reducing execution time of NPCF feature calculation using a multiscale approach in [126]. As in the previous works the same Monte Carlo approach for NPCF calculation was used.

8.2 Preliminaries

8.2.1 Phase Images

In the language of heterogeneous materials, the constituents of a composite material are referred to as *phases*. Where materials science is primarily concerned with physical

medium, the notion of a heterogeneous material is easily generalized to the two dimensional domain of images.

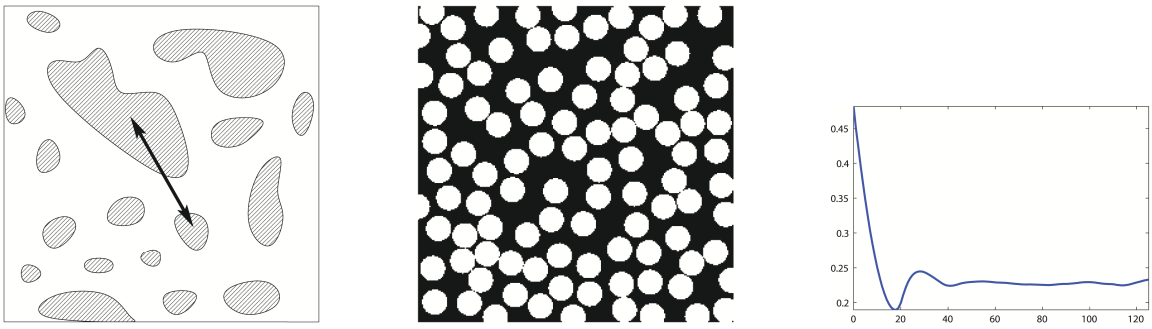
The term *phase image* is defined here to describe an image composed of discrete constituents. The phase image I with P phases is a 2D scalar field, partitioned into P complementary regions $\mathcal{V}_i$ that are both exhaustive $\mathcal{V}_1 \cup \cdots \cup \mathcal{V}_P = I$ and disjoint $\mathcal{V}_1 \cap \cdots \cap \mathcal{V}_P = \emptyset$. For the purposes of development, assume the phase image I is a random entity in sampling space Ω, and that $\omega \in \Omega$ is one realization from the ensemble. For each phase i, an indicator function is defined for $\boldsymbol{x} = (x, y) \in \mathbb{R}^2$ in I

$$\mathcal{I}^{(i)}(\boldsymbol{x}, \omega) = \begin{cases} 1, & \boldsymbol{x} \in \mathcal{V}_i(\omega) \\ 0, & else \end{cases} \quad . \tag{8.1}$$

The interpretation of phase is entirely specific to application. For example, in a biological specimen, the phases could correspond to biologically meaningful elements either subcellular components such as nuclei and cytoplasm, or cellular components such as different types of cells. The flexibility in defining the phases of an image is one of the strengths of the heterogeneous materials framework and will be discussed further in Section 8.3.

8.2.2 n-Point Correlation Functions

Statistical geometry appears in the study of physical phenomenon at both the microscopic and macroscopic scales. In the study of heterogeneous materials an extensive theory has been developed to characterize macroscopic physical properties using a statistical geometrical framework. In particular, the *n-point correlation functions* (NPCFs) have arisen in expressions related to transport phenomenon and electrical and mechanical properties [99]. The NPCFs are also used in a problem of much larger scale, for cosmological studies on dark energy and the distributions of galaxies [127, 128].



(a) (b) (c)

Figure 8.1: Two point correlation function. (a) By placing line segments of length r with random orientation on ω, the fraction of times the endpoints both land in phase i represents an estimate of $S_2(r)$.

Given the set of indicators $\mathcal{I}^{(i)}(\boldsymbol{x}, \omega)$, the n-point correlation function $S_n^{(i)}$ is defined as the probability of finding n points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ in phase i

$$S_n^{(i)}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) \equiv \mathrm{E}\{\mathcal{I}^{(i)}(\boldsymbol{x}_1)\mathcal{I}^{(i)}(\boldsymbol{x}_2)\ldots\mathcal{I}^{(i)}(\boldsymbol{x}_n)\} \tag{8.2}$$

$$= \Pr\{\mathcal{I}^{(i)}(\boldsymbol{x}_1) = 1, \mathcal{I}^{(i)}(\boldsymbol{x}_2) = 1, \ldots, \mathcal{I}^{(i)}(\boldsymbol{x}_n) = 1\}. \tag{8.3}$$

Of particular interest is the *two-point correlation function* (TPCF)

$$S_2^{(i)}(\boldsymbol{x}) \equiv \mathrm{E}\{\mathcal{I}^{(i)}(\boldsymbol{x}_1)\mathcal{I}^{(i)}(\boldsymbol{x}_2)\}. \tag{8.4}$$

which is the centerpiece of the tissue segmentation methodology. If I is statistically homogeneous, S_2 is invariant under translation and depends only on $\boldsymbol{x}_{1,2} = \boldsymbol{x}_1 - \boldsymbol{x}_2$ rather than absolute position. If I is also statistically isotropic then S_2 is rotationally invariant and depends only on distance $r = |\boldsymbol{x}_{12}|$. In this case the TPCF is denoted $S_2(r)$, and may be visualized as an experiment similar to the familiar Buffon's needle problem depicted in Figure 8.2.2.

The assumptions of a random field that is statistically homogeneous and isotropic are used here for the purposes of illustration. In practice digital images are used rather than random entities, and the TPCF is measured as a sample average within image boundaries. Although images are typically anisotropic, estimating S_2 under the isotropic assumption provides statistics that are insensitive to the orientation of content. This property is very desirable for descriptors of image content in a classification or segmentation application.

8.2.3 Relationship to Co-Occurrence Matrix

The TPCF represents the probability that phases are separated by a given distance, either directed or irrespective of orientation. Another popular measure used in the analysis of traditional texture images is the *co-occurrence matrix*. The co-occurrence matrix is based on a similar principal, namely the spatial distribution of image values [129], and is closely related to the TPCF as demonstrated below.

Given an intensity image G, the co-occurrence matrix $C_{\boldsymbol{x}}$ represents the frequencies that image values i, j are separated by $\boldsymbol{x}$

$$C_{\boldsymbol{x}}(i,j) = \sum_m \sum_n \begin{cases} 1, & G(m,n) = i, \\ & G(m+x, n+y) = j \\ 0, & else. \end{cases} \tag{8.5}$$

Here, (x, y) are assumed to take on integer values to measure co-occurrence between pixels. The diagonal frequencies of $C_{\boldsymbol{x}}$ are related to the sample TPCF of G through a normalization by the total comparisons in Equation 8.5

$$S_2^{(i)}(\boldsymbol{x}) = \frac{C_{\boldsymbol{x}}(i,i)}{(N-x)(M-y)}, \tag{8.6}$$

where N and M are the horizontal and vertical image dimensions.

Despite this relationship the application of TPCFs to image segmentation is fundamentally different from co-occurrence based approaches. The relationship between TPCFs and

co-occurrence matrices is further explored in Section 8.3 where the use of TPCF as a feature for segmentation is described, and in Section 8.4.1 where TPCF is compared to a commonly used co-occurrence based method, namely the Haralick features.

8.2.4 Sample TPCF Calculation

The TPCF can be determined analytically only in limited cases where the material is explicitly defined. When dealing with tangible media the TPCF is calculated by sampling a digitized representation of the heterogeneous material.

Given an $M \times N$ digital phase image I, to calculate $S_2^{(i)}$ the autocorrelation of indicator $\mathcal{I}^{(i)}(x, y)$ is calculated first

$$R^{(i)}(\Delta x, \Delta y) = \sum_m \sum_n \mathcal{I}^{(i)}(m, n)\mathcal{I}^{(i)}(m + \Delta x, n + \Delta y), \tag{8.7}$$

where $\Delta x, \Delta y \in \mathbb{Z}$. The correlation of indicators effectively counts the number of pixels of phase i that are separated by $(\Delta x, \Delta y)$, e.g. $(0, 0)$ represents a full-overlap of the indicators, and $R(0, 0)$ is the number of pixels of phase i in I. The values of R are normalized by the number of overlapping pixels to calculate probabilities

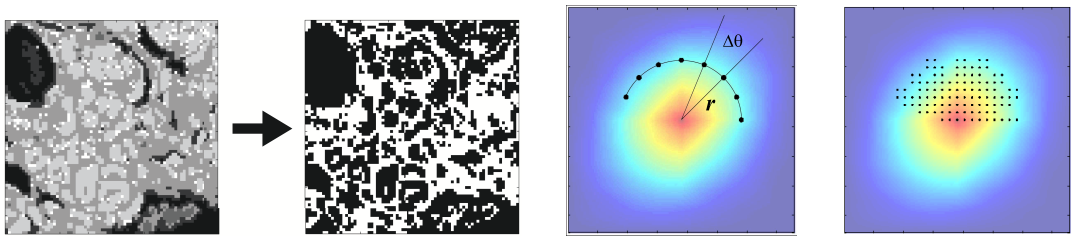
$$\hat{R}^{(i)} = R^{(i)}./(\mathbf{1}_{M\times N} * \mathbf{1}_{M\times N}), \tag{8.8}$$

where $\mathbf{1}_{M\times N}$ is an $M \times N$ matrix of ones, $./$ is element-wise division, and $*$ is convolution.

The normalized elements of $\hat{R}$ represent the anisotropic but homogeneous TPCF $S_2^{(i)}(\boldsymbol{x})$. To calculate the isotropic quantity $S_2^{(i)}(r)$ from $S_2^{(i)}(\boldsymbol{x})$, a process of *circumferential sampling* is used. Samples taken at a distance r from $\hat{R}^{(i)}(0, 0)$ are averaged over angle

$$S_2^{(i)}(r) = \frac{\Delta\theta}{\pi} \sum_{k=0}^{\frac{\pi}{\Delta\theta}-1} \hat{R}^{(i)}(r\cos(k\Delta\theta), r\sin(k\Delta\theta)), \tag{8.9}$$

where $\Delta\theta$ is the *angular interval*. This sampling procedure is depicted in Figure 8.2. Samples that do not fall on the discrete grid of $\hat{R}^{(i)}$ can be inferred using bilinear interpolation. Due to the symmetry of $\hat{R}^{(i)}$, the sampling angles can be restricted to $[0, \pi)$.



(a) (b) (c)

Figure 8.2: Sample TPCF calculation. (a) $\mathcal{I}^{(i)}$ is extracted from the phase image to calculate autocorrelation. (b) Circumferential samples are averaged at radius r from $\hat{R}^{(i)}(0,0)$ to calculate $S_2^{(i)}(r)$. (c) The pattern of on-grid samples required for interpolation is sparse. Here $\Delta\theta = \pi/8$ and r ranges from zero to $w/2$.

8.3 TPCF for Image Segmentation

The workflow for TPCF texture segmentation is presented in Figure 8.3. The process begins with the identification of phases from a color or grayscale image to generate a phase labeled image. Feature vectors containing the TPCFs of each phase are calculated from local regions throughout the phase image. The dimensionality of the feature vectors is reduced using principal component analysis, and the reduced dimension features are clustered in feature space. The clustering in the feature space is then mapped back to the image domain and refined if necessary to eliminate edge effects and aberrations. Each of these stages is described in further detail below.

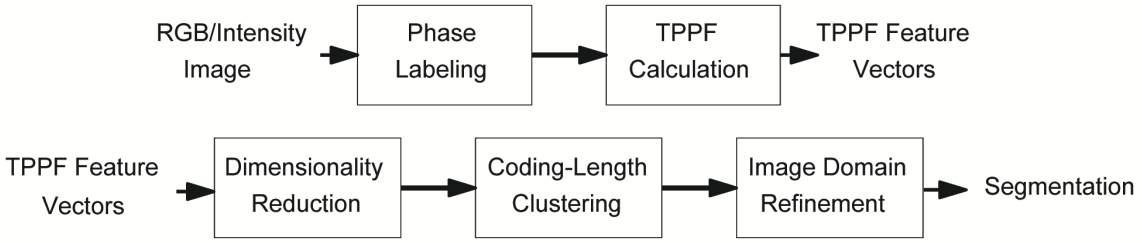


Figure 8.3: TPCF segmentation workflow.

8.3.1 Phase Labeling

Given a color or intensity image G with dimensions $M \times N$, the process of phase labeling assigns a label $i \in \{1, 2, \ldots, P\}$ to each pixel to generate the phase image I.

While phase has a very specific definition in the study of heterogeneous materials, in the imaging context phase is a flexible concept that provides a general approach to treating images as mixtures of constituents. These constituents can be identified by either low-level information such as intensity or color, or high-level information such as shape or size. In the case of low-level information, any number of mode-identifying segmentations such as mean shift [106] or K-means can be used to label constituents. If the distribution of color/intensity is more uniform than multi-modal then a simple quantization may be more effective. For high level information the determination of phase is certainly application specific since the phases likely represent meaningful units e.g. different types of cells in a tissue. A more complex knowledge-based approach may be required in this case, of which there are many specifically for microscopic images [130–134].

8.3.2 TPCF Feature Vectors

Define $\Phi(x, y)$ as the $w \times w$ region-of-interest with upper left corner $I(x, y)$. The anisotropic sample TPCF is computed inside $\Phi(x, y)$ for $r = 0$ to $w/2$ and for each phase $i \in \{1, 2, \ldots, P\}$ to form the $P(w/2 + 1)$-dimensional feature vector

$$v_{x,y} = [S_2^1(0), S_2^1(1), \ldots, S_2^1(\frac{w}{2}), S_2^2(0), S_2^2(1), \ldots, S_2^2(\frac{w}{2}), \ldots, S_2^P(0), S_2^P(1), \ldots, S_2^P(\frac{w}{2})]^{\mathrm{T}}. \tag{8.10}$$

This feature vector is computed over every position $(x, y) \in \{0, 1, \ldots N - w\} \times \{0, 1, \ldots M - w\}$ in the phase image I.

8.3.3 Dimensionality Reduction

Although the feature vectors $v_{x,y}$ reside in $P(w/2 + 1)$ space, their energy is typically concentrated in relatively few modes. Prior to segmentation the dimension of the feature vectors is reduced by projecting $v_{x,y}$ onto the first D primary two-point functions obtained through singular value decomposition.

8.3.4 Clustering

To achieve a segmentation of the image the reduced dimension feature vectors are clustered in the feature space and the clustering result is mapped back to the image space to form a segmentation map.

The choice of clustering algorithm depends on the application and the distribution of features in the feature space. As demonstrated in Section 8.4, the TPCF feature vectors tend to be either restricted to a smooth low-dimensional manifold or distributed among a mixture of low-dimensional linear structures. Several clustering methods are used in this chapter to exploit these feature space distributions and depending on feature distribution

and application constraints. K-means is used for a simple unsupervised clustering when the features do not follow the mixture linear distribution. K-nearest-neighbors is used for supervised clustering for problems where the feature distribution is not linear but is perhaps too complex for K-means. When the features adhere to multiple linear structures the lossy data coding method of [135] is used to achieve an unsupervised segmentation.

Lossy data coding

The method of coding-length segmentation applies principals of lossy data coding to achieve a robust segmentation of multivariate data by minimizing the coding length of the segmented data. The method of lossy coding requires only one parameter, the distortion ε, and is implemented using a simple iterative hierarchical procedure.

Given a set of vectors $V = (v_1, v_2, \ldots, v_m) \in \mathbb{R}^{n \times m}$ a lossy coding scheme maps the sequence to a binary representation up to an acceptable loss ε. If the vectors are assumed to be independent and identically distributed from a multivariate normal distribution then an approximation of the average coding rate is

$$R(V) \equiv \frac{1}{2} \log_2 \det \left(Id + \frac{n}{\varepsilon^2 m} V V^T \right) \tag{8.11}$$

where Id is the identity matrix. The overall coding length of the sequence $L(V)$ includes the coding length for the m vectors as well as the codebook length $nR(V)$

$$L(V) \equiv \frac{m+n}{2} \log_2 \det \left(Id + \frac{n}{\varepsilon^2 m} V V^T \right). \tag{8.12}$$

If the vectors are instead assumed to come from a mixture of normal distributions then it may be more effective to code the overall sequence $V = V_1 \cup V_2 \cup \cdots \cup V_k$ by coding each group V_i independently along with the group labels. In this case the coding length

Algorithm 5 Pairwise Steepest Descent for Lossy Coding Clustering

input: Data $V = (v_1, v_2, \ldots, v_m) \in \mathbb{R}^{n \times m}$ and distortion ε.
initialize clustering $\mathcal{S} = \{S_i = \{v_i\} | i = 1, 2, \ldots, m\}$
while $|\mathcal{S}| > 1$ **do**
    Given sets $S_i, S_j, i \neq j$ such that $L^s(S_i \cup S_j) - L^s(S_i, S_j)$ is minimal over all pairs
    if $L^s(S_i \cup S_j) - L^s(S_i, S_j) \geq 0$ **then** break.
    else $\mathcal{S} = (\mathcal{S} \setminus \{S_i, S_j\}) \cup \{S_i \cup S_j\}$
end
output: clustering $\mathcal{S}$

becomes

$$L^s(V_1, V_2, \ldots, V_k) = \sum_{i=1}^{k} L(V_i) - |V_i| \log_2 \left(\frac{|V_i|}{m} \right). \tag{8.13}$$

The two terms in the summand of Equation 8.13 represent the coding length for each group V_i and the (lossless) coding of group labels respectively.

This notion is the fundamental concept of lossy coding for clustering: an ideal clustering into groups V_i should correspond with an ideal coding length for the overall sequence. By identifying the partitioning which produces the best compression, the segmentation into clusters is obtained. In practice this can be achieved in a steepest-descent fashion using a hierarchical clustering with coding length gain as the measure of distance. This procedure is described in Algorithm 5. The optimality of this procedure is demonstrated in [135].

8.3.5 Segmentation Refinement

In some applications the results of the feature space segmentation may be unsatisfactory. The limited spatial resolution of local TPCF calculations can produce edge effects at texture boundaries, and the loss of spatial relationships between features in the image space can result in mild segmentation noise. Simple corrections can be applied directly to the segmentation result in the image domain to correct these problems. The approach

used depends on application requirements. Section 8.4.2 provides a refinement example for tissue segmentation.

8.3.6 Computation

8.3.7 FFT method for sample TPCF calculation

The most computationally demanding portion of the TPCF calculations are the correlations. These correlations may be computed efficiently using the Fast Fourier Transform (FFT), as in Chapter 2. The implementation of TPCF segmentation is treated in further detail in Chapter 9 where a more efficient method for calculating TPCF features is described.

8.4 Experiments and Results

Experiments were performed with both natural textures and microscopic tissue images using the procedure described in Section 8.3. For natural textures TPCF features were compared with both raw co-occurrence matrix features and traditional Haralick features for images taken from the Brodatz texture collection. For microscopic tissue images TPCF features were clustered with both supervised and unsupervised clustering methods to demonstrate the ability to identify tissue boundaries.

8.4.1 Natural Textures

Three 128×128 images were selected from the Brodatz collection and arranged as in Figure 8.4(a). The grayscale arrangement was quantized to two bits to produce a phase image with $P = 4$ phases.

Three sets of features were calculated from the phase image: raw co-occurrence, Haralick, and TPCF. Each set was independently reduced to $D = 10$ using PCA and clustered using K-means with $K = 3$. All feature sets were calculated in a sliding window of $w = 32$.

TPCF features were calculated at distances $r = 0, 1, \ldots, 16$ to generate 68-dimensional features. Raw Co-occurrence features are the unwrapping of $C_{\boldsymbol{x}}$ into a 16-dimensional vector, with $C_{\boldsymbol{x}}$ computed at $\boldsymbol{x} = (0, d), (\lceil\sqrt{2}d\rceil, \lceil\sqrt{2}d\rceil), (d, 0), (-\lceil\sqrt{2}d\rceil, \lceil\sqrt{2}d\rceil)$ for $d = 1, 2, \ldots, 16$, and then averaged over the four orientations to form 256-dimensional features. The Haralick features of contrast, correlation, energy, and homogeneity were calculated from the unaveraged co-occurrence matrices and then averaged for each distance to form 64-dimensional features.

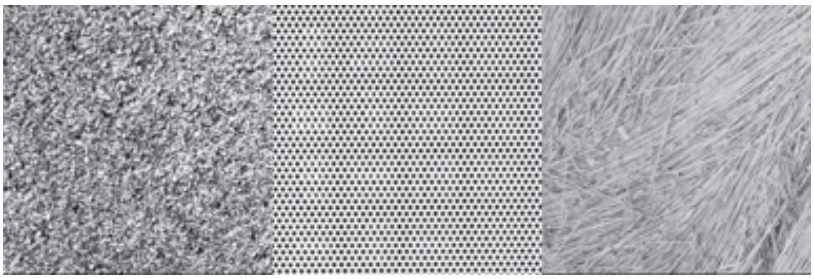
The singular values of the three feature sets are presented in Figure 8.4(b) (normalized for comparison). Each feature set clearly contains most of its energy in relatively few modes. The features were reduced to 10 dimensions prior to clustering with K-means with $K = 3$.

The segmentations are shown in Figure 8.4(c). The confusion matrices for these segmentations are contained in Table 8.1. For each feature type all segmentation errors occur within $w/4$ of the texture boundaries. The accuracy is comparable for each feature type at 94.1%, 97.3%, and 96.6% for Haralick, co-occurrence, and TPCF respectively.

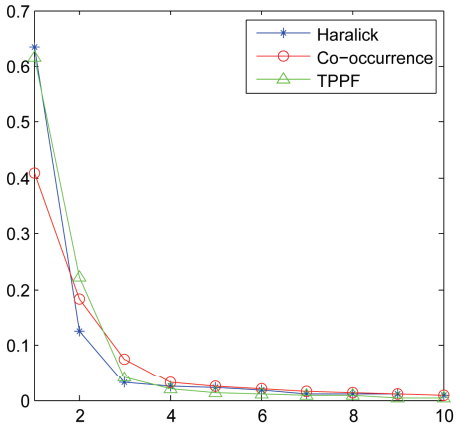
A three-dimensional visualization of the TPCF feature space was produced using PCA and is presented in Figure 8.4(d). The features conform to a smooth manifold-like structure. The low-dimensional characteristic of the TPCF features suggests that this is an accurate representation.

8.4.2 Tissue Segmentation

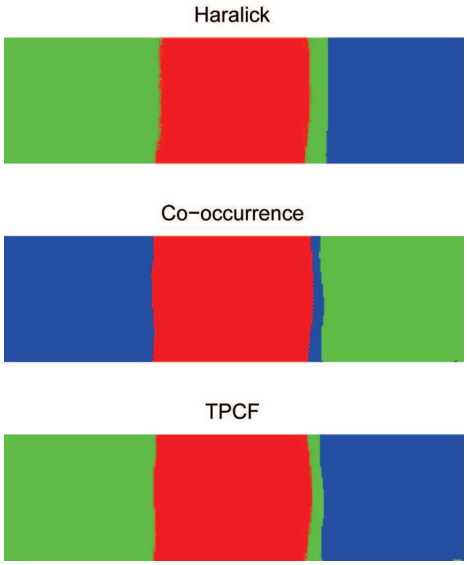
Two applications for tissue segmentation were explored using TPCF segmentation, using images and scenarios from previous chapters. The first is the identification of follicle



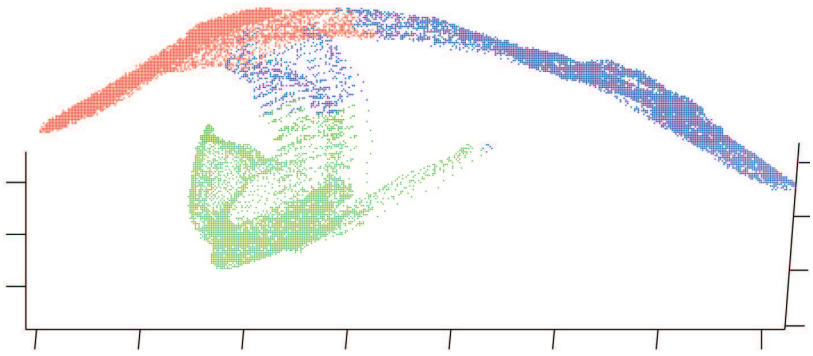
(a)



(b)



(c)



(d)

Figure 8.4: Natural texture segmentation using TPCF, Haralick, and raw co-occurrence matrices. (a) Brodatz textures grass, holes, straw, left to right. (b) Normalized singular values for each feature set. (c) K-means segmentations. (d) Three-dimensional visualization of TPCF features.

Table 8.1: Confusion matrices for natural texture segmentations.

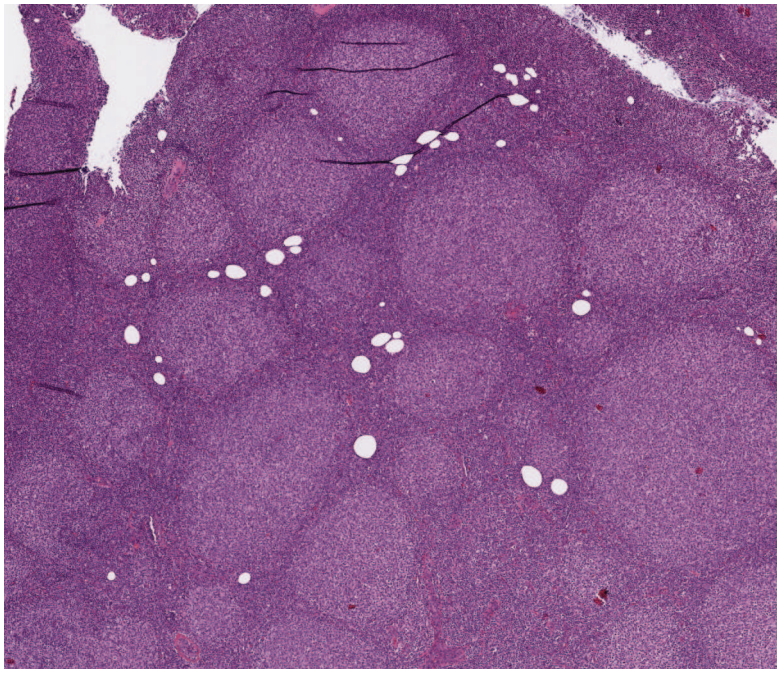
	Haralick			Co-oc			TPCF		
class	1	2	3	1	2	3	1	2	3
1	10961	0	0	10961	0	0	10961	0	0
2	1269	11147	0	606	11810	0	837	11579	0
3	750	0	10114	307	0	10557	335	0	10529

regions in human follicular lymphoma images from Chapter 6. The second is the distinction of spongiotrophoblast from labyrinth tissue in mouse placenta images from Chapter 2. In each application the phases were identified as cellular and subcellular components representing nuclei, cytoplasm, red blood cells, and background.

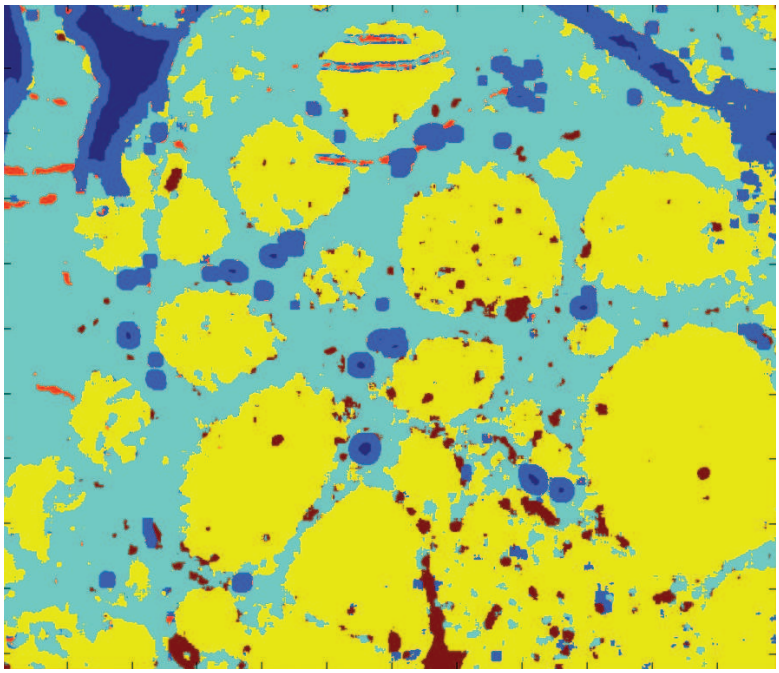
Follicular Lymphoma

The motivation for the registation method of Chapter 6 was the difficulty in segmenting follicle regions in H+E stained images. To grade follicular lymphoma tumors it is necessary to generate statistics on the concentration of centroblast cells within follicle boundaries. While centroblast cells are easily identifiable in an H+E stain, the follicle regions are not, prompting the use of nonrigid registration to map a follicle segmentation from an IHC stain to the H+E image.

Experiments were performed to demonstrate the capability of TPCF features to segment follicles directly in H+E images. Two follicular lymphoma images with H+E stain were selected from the dataset described in Section 6.2.1. One 1000×1000 region was selected from each image to contain a mixture of follicles and other tissues, as shown in Figures 8.5(a) and 8.7(a). The pixels of these regions were labeled using a nearest neighbor classification to generate a four-phase image. TPCF feature vectors were generated for both



(a)



(b)

Figure 8.5: Follicle segmentation example one. (a) H+E stained follicular lymphoma section. Follicles appear as large elliptical regions. (b) Unsupervised segmentation using lossy data coding clustering.

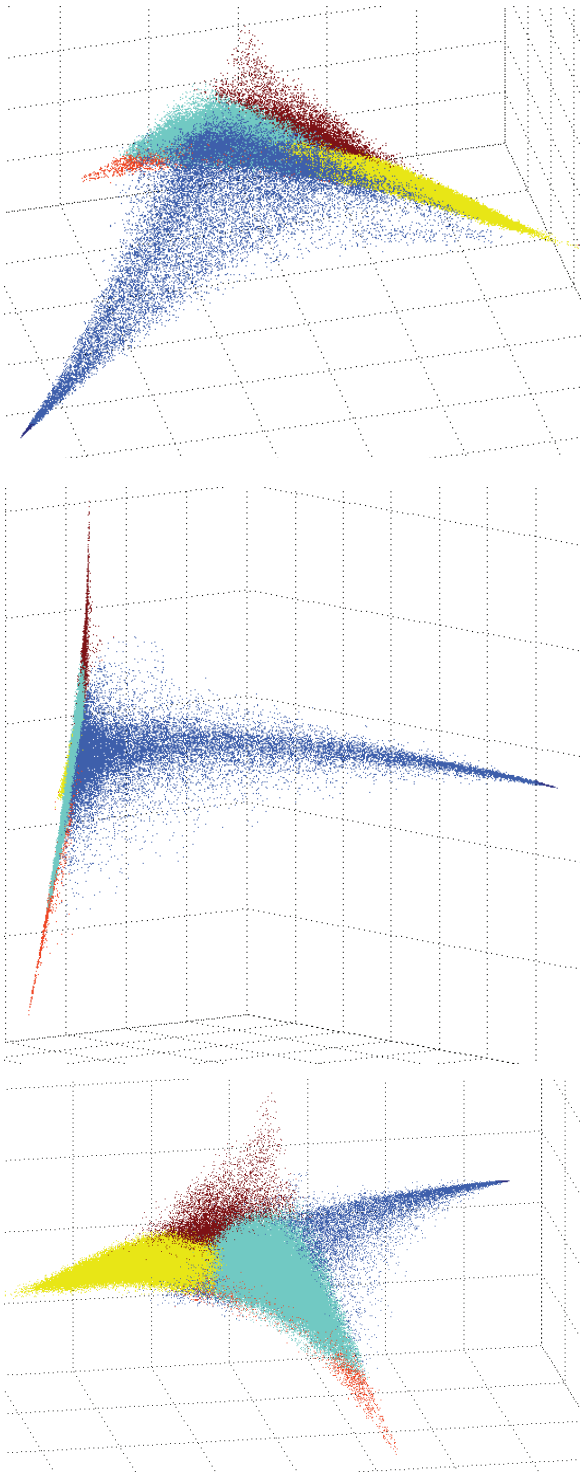
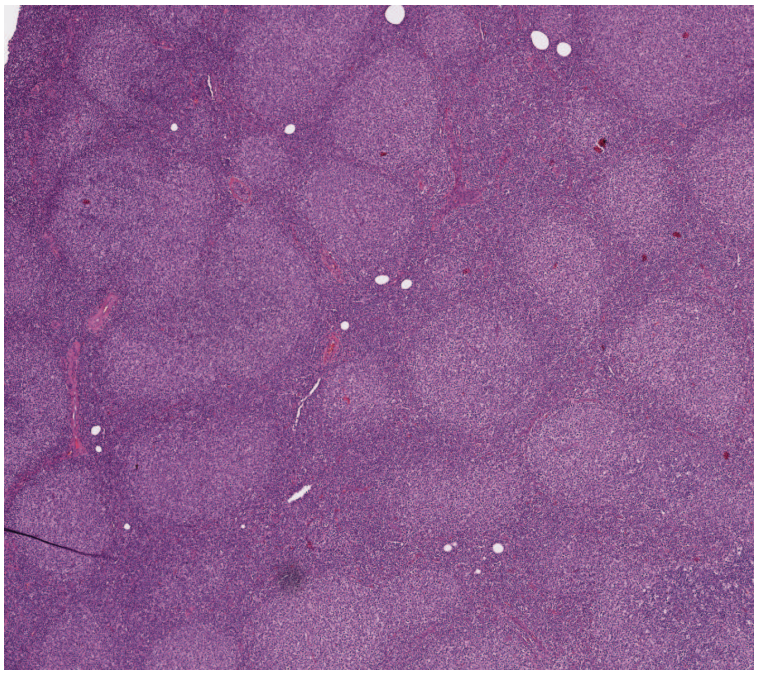
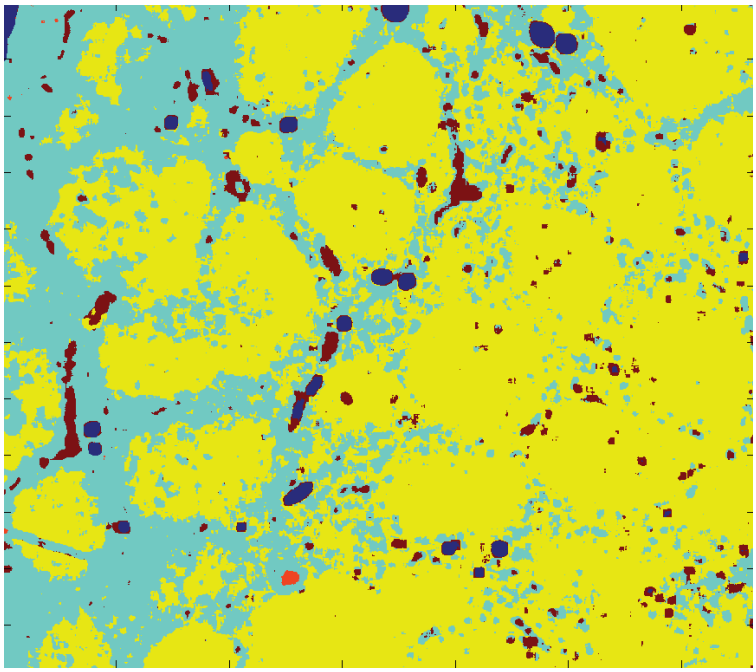


Figure 8.6: Visualization of TPCF features for follicular lymphoma example one. Clusters are color coded to correspond with Figure 8.5(b).



(a)



(b)

Figure 8.7: Follicle segmentation example two. (a) H+E stained follicular lymphoma section. Follicles appear as large elliptical regions. (b) Unsupervised segmentation using lossy data coding clustering.

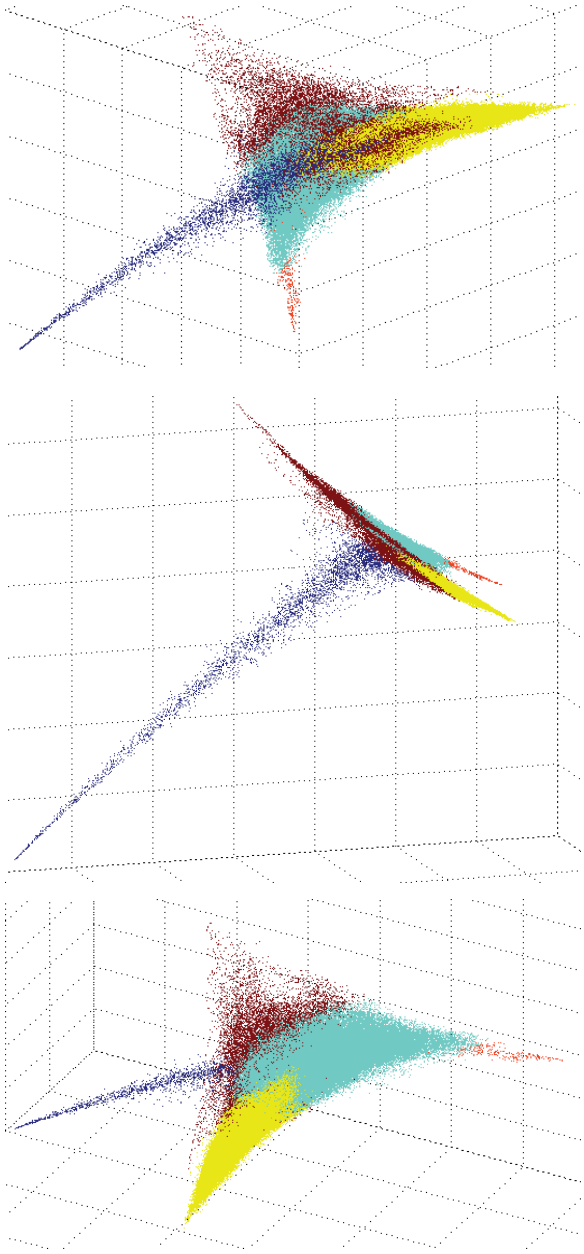


Figure 8.8: Visualization of TPCF features for follicular lymphoma example two. Clusters are color coded to correspond with Figure 8.7(b).

phase images using the parameters $w = 16$, $\Delta\theta = \pi/8$, and lengths $r = 0, 1, \ldots, 8$. The 36-dimensional feature sets were reduced to ten-dimensional space using singular value decomposition. Each reduced feature set was then clustered using the lossy coding length algorithm in a semi-supervised configuration: to reduce clustering time the feature vectors were sampled from the phase images with a horizontal and vertical stride of four pixels and subsequently clustered. The remaining unsampled features were assigned to the resulting clusters based on the most favorable coding length using Equation 8.13.

The results of the clustering are presented in Figures 8.5(b) and 8.7(b). In each case some follicles are clearly identified. The quality of segmentation suffers towards the physical center of each tissue section, which corresponds to the lower right corner of each image. From examining the RGB color images it is clear that there is a strong contrast gradient in the same direction, with follicles closer to the physical center appearing less conspicuous. This gradient may be due to either lack of stain penetration, nonuniform section thickness, or nonuniform illumination in the scanning process.

Visualizations of the clustered TPCF features are presented in Figures 8.6 and 8.8. These visualizations were obtained by using singular value decomposition to project the 36-dimensional features into three-dimensional space. For both examples the TPCF features follow a similar distribution. The features corresponding to the follicle and other tissue regions are restricted to a smooth planar surface. The features corresponding to the background and background-tissue transition regions form a conspicuous protrusion that is approximately orthogonal from the planar surface. The features of the planar surface originate from the tissue interior and so have very little energy in the TPCF feature vector

components corresponding to the background phase. Likewise, the features of the protrusion have very little energy in the components corresponding to the tissue phases. These facts together explain the peculiar distribution of the TPCF features in these examples.

Placenta

The mouse placenta images originate from a study on the role of the Rb gene in inducing morphological changes in mouse placenta [22]. The placenta contains several tissue layers namely labyrinth, spongiotrophoblast, trophoblast, and glycogen. The aim of the example segmentation application here is to distinguish the labyrinth layer from the spongiotrophoblast layer as they are the least distinctive pair of adjacent layers (see Figure 1.1).

A 1000×1000 pixel area was selected from each of 18 placenta images to contain approximately half labyrinth layer and half other-tissue layers. A maximum likelihood classifier was applied to these areas to classify the pixels into red blood cell, cytoplasm, nuclei, extracellular matrix and background as in [15]. These classifications serve as seven-phase images from which TPCF feature vectors are calculated. The parameters ROI size $w = 32$, length $r = 0, 1, \ldots, 16$, and angular interval $\Delta\theta = \pi/8$ produced 68-dimensional feature vectors that were then reduced to ten-dimensional space prior to clustering.

The labyrinth tissue layer was manually marked to generate a ground truth segmentation for validation. To generate training data for KNN clustering the TPCF feature vectors for a single image were spatially sampled both horizontally and vertically with a stride of $16 = w/2$ pixels. The remaining seventeen images were then clustered using this training feature set with $K = 50$. This was repeated using each of the eighteen images as training data to explore the sensitivity of training data selection. The clustering results were then mapped to the image domain and refined using morphological operations. Small objects and holes appearing due to clustering noise were removed with the understanding that the

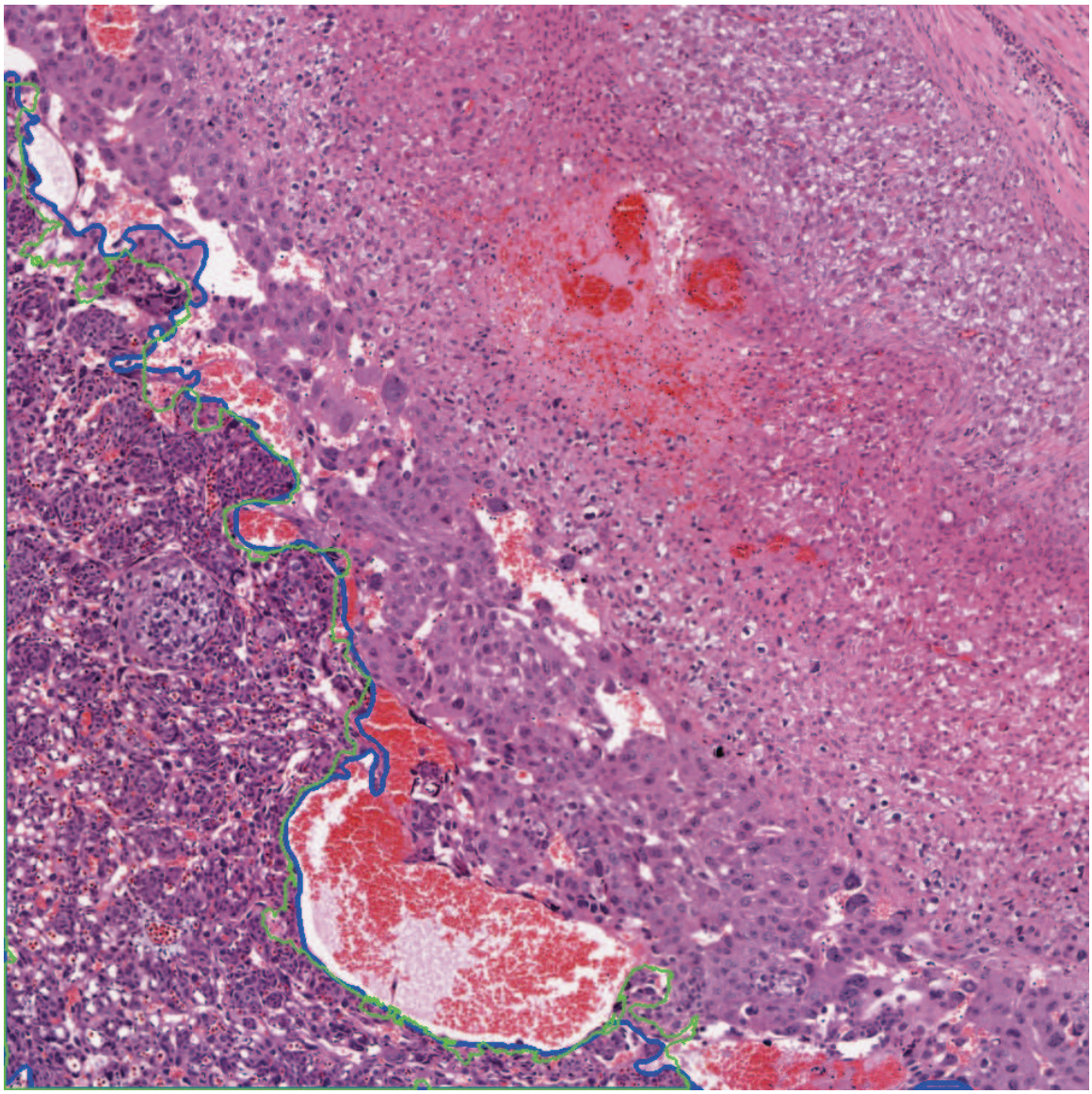


Figure 8.9: Placenta image 22. The blue line represents the manual segmentation. The green line indicates the segmentation with image 15 used for training.

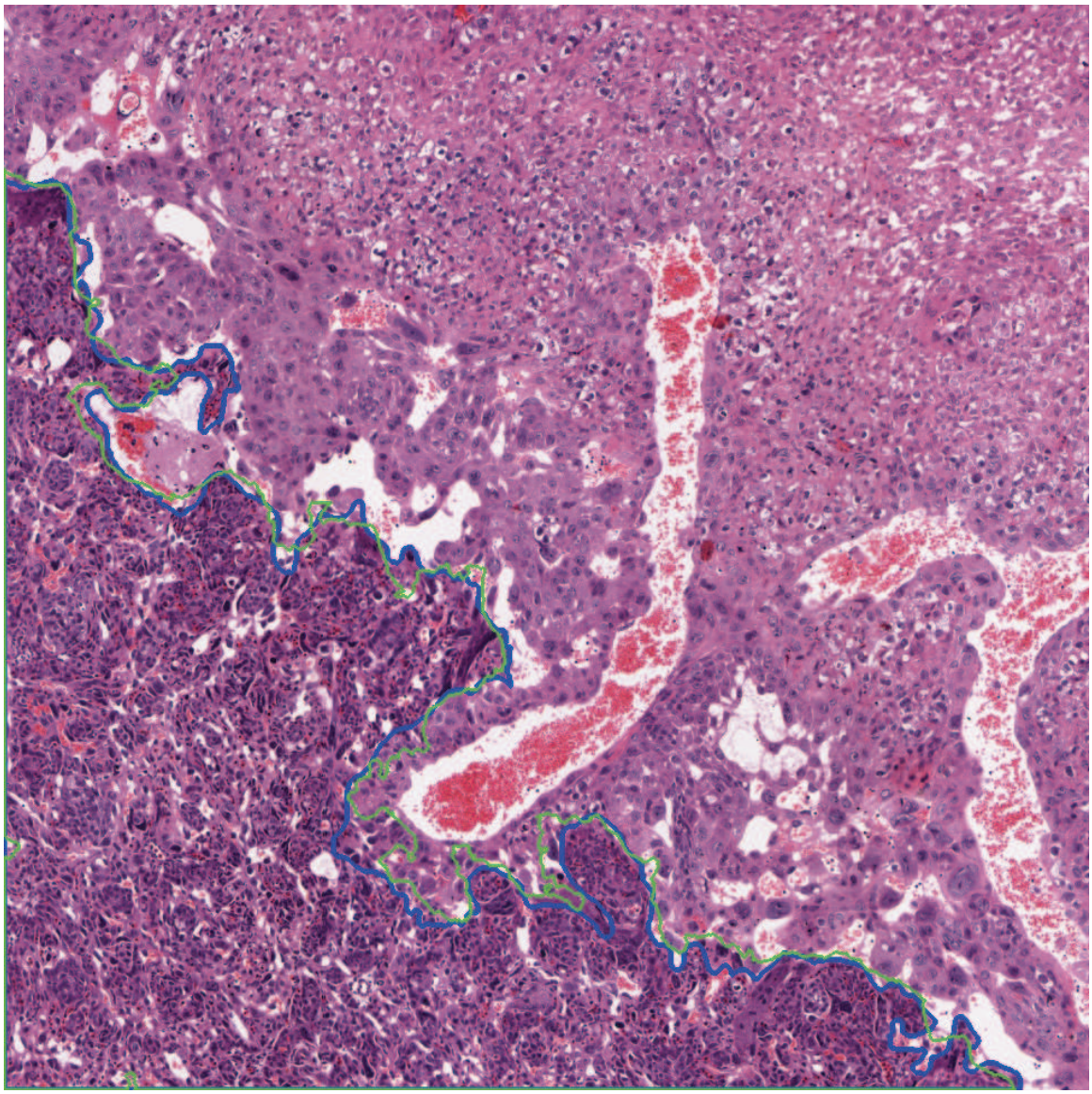


Figure 8.10: Placenta image 19. The blue line represents the manual segmentation. The green line indicates the segmentation with image 18 used for training.

labyrinth layer is contiguous. Accuracy was then measured on the refined segmentations as the total percentage of correctly segmented pixels from both labyrinth and other-tissue classes.

A summary of the segmentation accuracy is included in Appendix A, Table A.1. Each row contains the segmentation results for one training image. The row medians indicate the effectiveness of each image as training data, column medians indicate the quality of segmentations for each image from all training data. With the exception of image 18, each image was effectively segmented by at least one other. The images tend to form cliques that produce mutually effective segmentation. For example, image 1 effectively segments image 10 and visa versa, but poorly segments images 16 and 17. Likewise images 16 and 17 segment image 10 poorly. This indicates that there is variation in image content, either owing to natural biological differences and/or the algorithm used to generate phase images from color slide scans.

Two example segmentations are illustrated in Figures 8.9 and 8.10. In both examples the segmentation boundary conforms closely to the manually marked labyrinth tissue boundary. In the upper left corner of Figure 8.9, the TPCF segmentation actually corrected an error in the manual marking. The tissue between the segmentation boundary and manual boundary in this area is not labyrinth tissue but giant cells from the spongiotrophoblast layer. A mistake in the segmentation is apparent in the lower portion of the same image where the densely packed cluster of giant cell nuclei could not be distinguished from the labyrinth layer. The results of Figure 8.10 are similar. Some areas of densely packed giant cells from spongiotrophoblast tissue are again mistaken for labyrinth tissue. In the upper left corner of this image there is a small protrusion of labyrinth tissue that is lost in the segmentation due to the limitation in spatial resolution for the ROI size $w = 32$.

8.5 Discussion and Conclusions

By considering tissues as arrangements of discrete and biologically meaningful components, the problem of tissue segmentation can be cast into the heterogeneous materials framework. The two point correlation function, a stochastic geometric function, provides a means for acquiring statistics on the shape, size, and spatial distributions of these biological components to use as cues for the segmentation of different tissues.

TPCF features were compared to the raw co-occurrence matrix and the Haralick features for the problem of natural image segmentation. All results were comparable in terms of segmentation accuracy, indicating that there is redundancy in the extra information of both the raw co-occurrence and Haralick features. The raw co-occurrence and Haralick features are calculated over multiple distances and averaged over orientation, as are the TPCF features. Unlike the raw co-occurrence or Haralick features, the "off-diagonal" i-to-j phase comparisons are not used for TPCF. The co-occurrence matrices are often sparse, and so the raw frequencies are not used directly as features for segmentation or classification, rather measures such as the Haralick features are computed from C to extract features. Neither the off-diagonal frequencies or feature extraction were beneficial in the example.

The TPCF features were demonstrated to be effective for the ultimate aim of tissue segmentation. In the follicular lymphoma example the feature distributions (again low dimensional) permit an unsupervised segmentation using lossy coding clustering. The identification of follicles in the two examples provided was subject to the quality of inputs. The lack of uniformity in the color distributions of the raw images implies a more sophisticated preprocessing is required to produce consistent phase image for TPCF feature calculation. The spatial resolution of TPCF features also presents a problem for identifying the narrow channels that separate adjacent follicles. To have meaning the TPCF must be calculated

in some finite neighborhood which naturally implies limitations in spatial resolution. This neighborhood must be large enough to capture the statistics of the components that distinguish tissues, but not larger. In some scenarios the minimum neighborhood size may indeed result in obscuring delicate or complex tissue boundary regions.

The phenomenon of non uniformity in phase images was also observed in the supervised segmentation of placenta, where images were bound into cliques as indicated by effective mutual segmentations of one another. If natural biological variations are the root of the difference in the distributions of components in the phase image then training data must be chosen accordingly. If it is a matter of variation from slide preparation then phase images must be generated using more sophisticated algorithms that can adapt to the differences in content from one image to another. Regardless, the supervised segmentation of placenta achieved 95%+ accuracy in many cases. This level of accuracy is certainly adequate for application in many biological studies.

CHAPTER 9

COMPUTATION OF TPCF FEATURES WITH CORRELATION UPDATING, PARALLELIZATION, AND GPU

TPCF features demonstrate promising results in the segmentation of tissues in microscopic images, however they are accompanied by a significant computational burden. Consider the following example: computing TPCF features for a $16K \times 16K$ four-phase image with $w = 128$ implies the calculation of more than one billion correlations. Performing these correlations is a considerable task, with large image datasets pushing the correlation calculations into the trillions.

In this chapter I present three approaches to reduce the execution time for the computation of TPCF features. The first approach is a novel method called *correlation updating* that uses a derived relationship between TPCFs of neighboring regions-of-interest to update TPCF values rather than computing them from scratch. This innovation results in an extremely efficient TPCF calculation that does not waste computation on unused correlation values, and that eliminates the strong time-dependency on window size that exists for FFT-based correlation. The second approach is the parallelization of feature calculations on the multi-node and multi-socket levels. The third approach is the implementation of correlation updating on GPU, taking advantage of the fine-grained parallelism and fast on-chip memory to further optimize TPCF feature calculation.

9.1 Introduction

The computation of TPCF features is depicted in Figure 9.1. The computation consists of three main processes: correlation calculation, normalization, and sampling/interpolation.

Given an $M \times N$ digital phase image I, a $w \times w$ region-of-interest (ROI) $\Phi_{x,y}$ is defined with upper left corner $I(x, y)$. For each phase i in the ROI, the autocorrelation of the binary mask $\mathcal{I}_{x,y}^{(i)}$ is calculated

$$R^{(i)}(\Delta x, \Delta y) = \sum_m \sum_n \mathcal{I}_{x,y}^{(i)}(m, n)\mathcal{I}_{x,y}^{(i)}(m + \Delta x, n + \Delta y), \tag{9.1}$$

where $\Delta x, \Delta y \in \mathbb{Z}$. The values of R^i are normalized by the number of overlapping pixels to calculate probabilities

$$\hat{R}^{(i)} = R^{(i)}./(\mathbf{1}_{M\times N} * \mathbf{1}_{M\times N}), \tag{9.2}$$

where $\mathbf{1}_{M\times N}$ is an $M \times N$ matrix of ones, $./$ is element-wise division, and $*$ is convolution.

The normalized elements of $\hat{R}$ represent the homogeneous anisotropic TPCF $S_2^{(i)}(\boldsymbol{x})$. The isotropic quantity $S_2^{(i)}(r)$ is calculated using the process of circumferential sampling depicted in Figure 8.2. Samples taken at a distance r from $\hat{R}^{(i)}(0, 0)$ are averaged over angle

$$S_2^{(i)}(r) = \frac{\Delta\theta}{\pi} \sum_{k=0}^{\frac{\pi}{\Delta\theta}-1} \hat{R}^{(i)}(r\cos(k\Delta\theta), r\sin(k\Delta\theta)), \tag{9.3}$$

where $\Delta\theta$ is the *angular interval*. Samples that do not fall on the discrete grid of $\hat{R}^{(i)}$ can be inferred using bilinear interpolation. Due to the symmetry of $\hat{R}^{(i)}$, the sampling angles can be restricted to $[0, \pi)$.

This procedure is repeated for every phase i in the ROI $\Phi_{x,y}$ to calculate the feature vector $v_{x,y}$. The ROI is positioned at every complete location in the phase image $(x, y) \in \{0, 1, \ldots, N - w\} \times \{0, 1, \ldots, M - w\}$ to generate a set of $(M - w + 1)(N - w + 1)$ feature vectors.

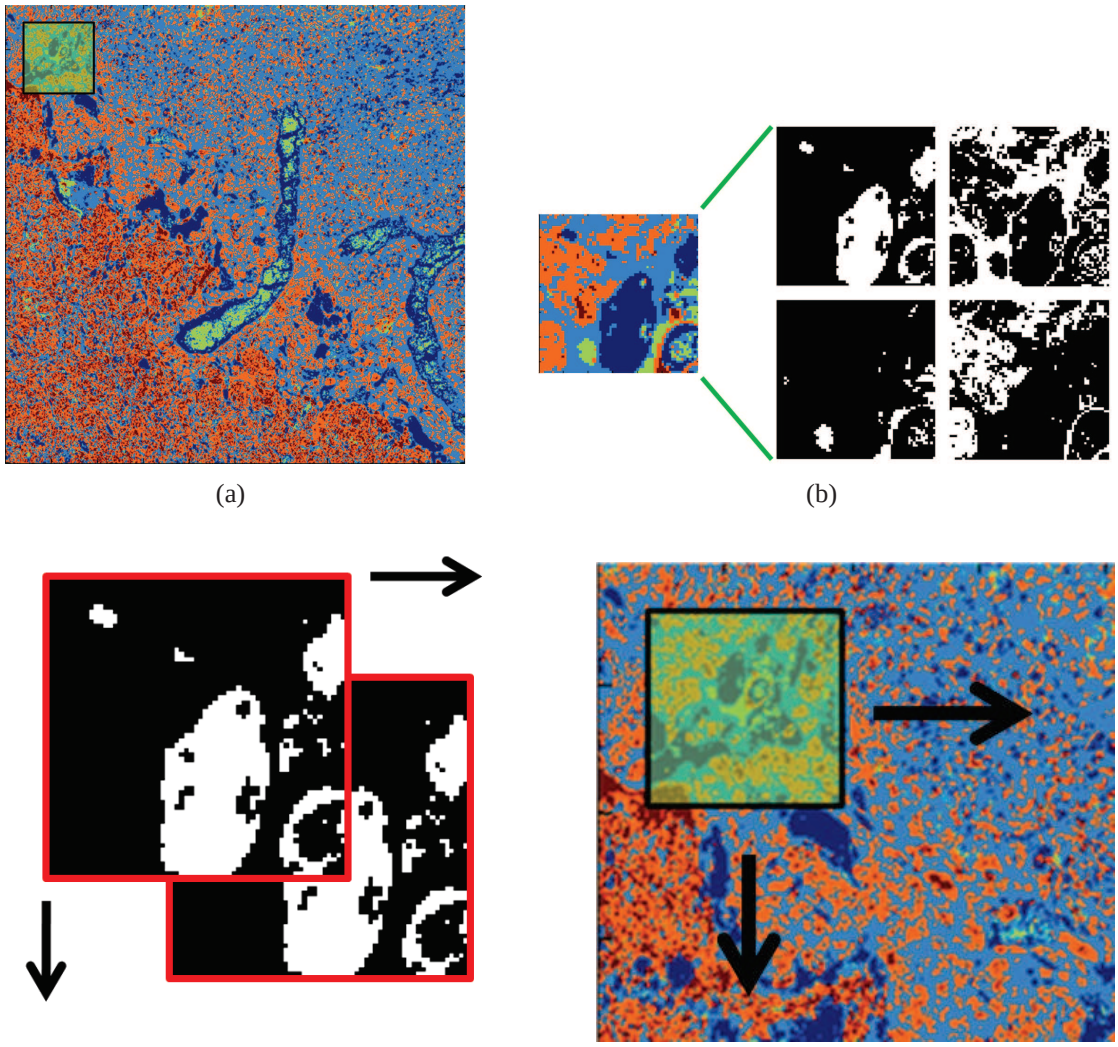


Figure 9.1: Computation of TPCF features. (a) A ROI $\Phi(x, y)$ is defined in the phase image. (b) A binary mask is generated for each phase of the ROI. (c) The autocorrelation $R^{(i)}$ is calculated for each mask and normalized and sampled to generate the TPCF $S_2^{(i)}(r)$. (d) The ROI is iterated throughout the entire image.

9.2 Direct FFT-based correlation

The most computationally demanding portion of the TPCF calculations are the correlations of Equation 9.1. These correlations may be computed efficiently using the Fast Fourier Transform (FFT), as in Chapter 2. The binary mask $\mathcal{I}_{x,y}^{(i)}$ is padded to the size $2w-1$

$$\mathcal{P}_{x,y}^{(i)} \equiv \begin{bmatrix} \mathcal{I}_{x,y}^{(i)} & \mathbf{0}_{w\times w-1} \\ \mathbf{0}_{w-1\times w} & \mathbf{0}_{w-1\times w-1} \end{bmatrix} \tag{9.4}$$

and transformed forward to the discrete frequency domain

$$\mathcal{F}[k,l] = \frac{1}{\sqrt{(}2w-1)} \sum_{n=0}^{2w-1} \sum_{m=0}^{2w-1} \mathcal{P}_{x,y}^{(i)}[m,n] e^{-2\pi j \frac{mk+nl}{2w-1}}. \tag{9.5}$$

The power spectrum is calculated by taking the magnitude of the complex elements $\mathcal{F}[k,l]$ and the inverse transformation is computed to obtain the autocorrelation R

$$R^{(i)} = \frac{1}{\sqrt{(}2w-1)} \sum_{l=0}^{2w-1} \sum_{k=0}^{2w-1} \mathcal{F}_{x,y}^{(i)}[k,l] e^{2\pi j \frac{mk+nl}{2w-1}}. \tag{9.6}$$

The dimension $2w-1$ is critical for the performance of the FFT calculations. The most widely used FFT library, FFTW [48], offers optimal performance for powers of two or small prime factors. The padding of Equation 9.4 may be manipulated to achieve these sizes, only by adding zeros to achieve the next most favorable size. A demonstration of the effects of transform size and padding is presented in Section 9.8.

9.2.1 Sparse sampling

The FFT calculates all $(2w-1)^2$ elements of the autocorrelation R, however only a small set of these are required for the circumferential sampling procedure of Equation 9.3. This is apparent in Figure 9.2, where only 10% elements of $R^{(i)}$ are used to interpolate $S_2^{(i)}(r)$. Although algorithms exist for computing subsets of FFT outputs [136–138], the

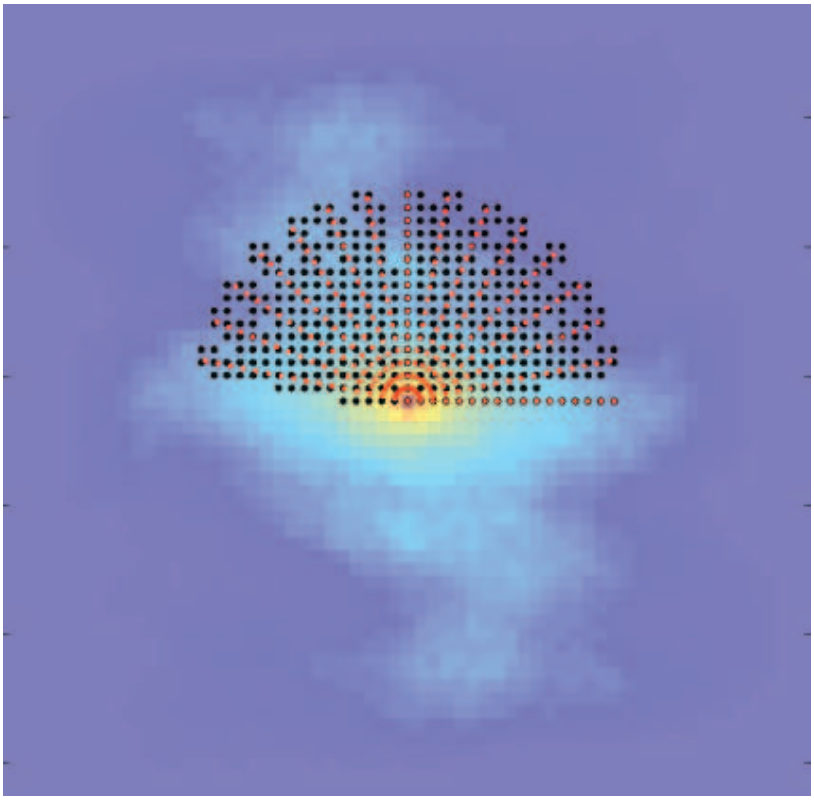


Figure 9.2: Sparsity of samples for autocorrelation circumferential sampling. The full autocorrelation matrix with the sampling pattern imposed is shown above. Here, $w = 32$ and $\Delta\theta = \pi/8$. Red points indicate the interpolation locations. Black points indicate the sampling points required for bilinear interpolation. In this case only 395 of the total 3969 elements of R are used for interpolation.

available implementations of ordinary full-output FFT are optimized to the extent that only a relatively large transform will benefit [48].

9.3 Correlation updating

In addition to the sampling sparsity, the shared content between neighboring ROIs also points to significant amounts of wasted computation. For example, although $\Phi_{x,y}, \Phi_{x+1,y}$

differ by only two w-length columns of pixels, a straight-forward FFT method calculates correlations from scratch for each.

The observations of sparsity and shared content may be simultaneously addressed using the linearity of correlation. Rather than computing $R^{(i)}$ from scratch for each ROI, the portions of neighboring ROIs, say $\Phi_{x,y}$ and $\Phi_{x+1,y}$, that are not shared may be used to update $R^{(i)}$ from $\Phi_{x,y}$ to $\Phi_{x+1,y}$ instead. Furthermore, if this updating is performed directly in the image domain then the locations used in sampling may be selectively updated, and the spatial dependency between the image and frequency domains can be avoided.

Given two horizontally adjacent $w \times w$ ROIs $\Phi_{x,y}, \Phi_{x+1,y}$ with corresponding indicators

$$\begin{aligned} \mathcal{I}^{(i)}_{x,y} &= [c_x, c_{x+1}, \ldots, c_{x+w-1}] \\ \mathcal{I}^{(i)}_{x+1,y} &= [c_{x+1}, c_{x+2}, \ldots, c_{x+w}], \end{aligned} \tag{9.7}$$

where c are w-length columns of pixels. The autocorrelation of $\mathcal{I}^{(i)}_{x,y}$ is denoted $R^{(i)}_{x,y}$. Given that $I^{(i)}_{x,y}, I^{(i)}_{x+1,y}$ are distinguished only by c_x, c_{x+w+1}, the autocorrelation $R^{(i)}_{x+1,y}$ can be calculated from $R^{(i)}_{x,y}$ by adding the contribution of c_{x+w+1} and removing the contribution of c_x.

Define the correlation sums between the columns and their respective regions

$$\begin{aligned} a^{-}_{\Delta x,\Delta y} &\equiv \sum_m \mathcal{I}^{(i)}_{x,y}(\Delta x, m) c_x(m + \Delta y) \\ a^{+}_{\Delta x,\Delta y} &\equiv \sum_m \mathcal{I}^{(i)}_{x+1,y}(\Delta x, m) c_{x+w}(m + \Delta y). \end{aligned} \tag{9.8}$$

The *update matrices* containing these correlation sums represent the contributions of c_x to $R^{(i)}_{x,y}$ and c_{x+w+1} to $R^{(i)}_{x+1,y}$

$$A^- \equiv \begin{bmatrix} a^-_{w-1,w-1} & \cdots & a^-_{0,w-1} & a^-_{1,1-w} & \cdots & a^-_{w-1,1-w} \\ a^-_{w-1,w-2} & \cdots & a^-_{0,w-2} & a^-_{1,2-w} & \cdots & a^-_{w-1,2-w} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a^-_{w-1,0} & \cdots & a^-_{0,0} & a^-_{1,0} & \cdots & a^-_{w-1,0} \\ a^-_{w-1,-1} & \cdots & a^-_{0,-1} & a^-_{1,1} & \cdots & a^-_{w-1,1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a^-_{w-1,1-w} & \cdots & a^-_{0,1-w} & a^-_{1,w-1} & \cdots & a^-_{w-1,w-1} \end{bmatrix},$$

$$A^+ \equiv \begin{bmatrix} a^+_{0,1-w} & \cdots & a^+_{w-1,1-w} & a^+_{w-2,w-1} & \cdots & a^+_{0,w-1} \\ a^+_{0,2-w} & \cdots & a^+_{w-1,2-w} & a^+_{w-2,w-2} & \cdots & a^+_{0,w-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a^+_{0,0} & \cdots & a^+_{w-1,0} & a^+_{w-2,0} & \cdots & a^+_{0,0} \\ a^+_{0,1} & \cdots & a^+_{w-1,1} & a^+_{w-2,-1} & \cdots & a^+_{0,-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a^+_{0,w-1} & \cdots & a^+_{w-1,w-1} & a^+_{w-2,1-w} & \cdots & a^+_{0,1-w} \end{bmatrix}. \tag{9.9}$$

The relationship between the autocorrelations for adjacent regions is then

$$R^{(i)}_{x+1,y} = R^{(i)}_{x,y} - A^- + A^+. \tag{9.10}$$

This updating procedure clearly applies to vertically adjacent ROIs as well.

Since only a subset of the elements of R are required for sampling, the corresponding update elements of A^+, A^- may be calculated individually for the sampling locations. Each sampling location will then require only $2w$ multiply-add operations for updating from one ROI to the next. Given the updating procedure, to calculate TPCF features over an entire phase image requires only P total FFTs to initialize $R^{(i)}_{0,0}, i = 1, \ldots, P$. With the initialization calculated the updating procedure is used to iterate the ROI both horizontally and vertically through the remaining positions.

The updating procedure does not compromise numerical accuracy in the calculation of autocorrelation. Since the elements of R represent counts of pixels with a given separation

there is no accumulation of error through repeated rounding operations. The updating procedure also provides flexibility in choosing the ROI size w since performance is not subject to the restrictions on FFT size.

Results comparing the time performance of updating against the ordinary FFT method are presented in Section 9.8.

9.4 Parallelization

The procedure for calculating TPCF feature vectors from a phase image is a simple data parallelism. The image may be divided among different nodes, sockets, or cores, with each computing TPCF features for its portion.

The implementation used for the experiments of this chapter assumes a head/worker organization. A single node loads the phase image, partitions it into horizontal strips and distributes the strips to processing elements (including itself) using asynchronous communication and double buffering to overlap communication with disk operation. Each node calculates the TPCF features for its portion of the image and returns the results to the head node. MPI is used for communication between sockets and nodes [77] to achieve multi-node and SMP parallelism.

9.5 GPU implementation

The process of calculating TPCF features contains both fine and coarse parallelisms: the computation of sample updates (fine) and computation of ROIs (coarse). The fine level of parallelism exists within a single ROI and is the computation of the update values from Equation 9.9, the normalization of updated locations, and the bilinear interpolation to calculate $S_2^{(i)}(\boldsymbol{x})$. There are no dependencies in any of these processes so each is easily

distributed. The coarse level of parallelism is the simultaneous calculation within multiple ROIs. Clearly the process of correlation updating is dependent though, as updating the autocorrelations for $\Phi_{x+1,y}$ requires the autocorrelations for $\Phi_{x,y}$ to be available.

Both of these levels of parallelism can be effectively exploited using CUDA on GPU using block and thread parallelisms. A complete review of the CUDA programming model is available in Chapter 3.2.

At the fine level, the computation of update values and bilinear interpolations can be divided among threads in a single block. Fine level details aside, at the course level sequences of dependent ROIs can be divided among blocks. The arrangement into dependent sequences of ROIs is essential since distinct blocks are unable to cooperate. A simple way to achieve this arrangement is for each block may process a horizontal strip of ROIs $\{\Phi_{0,y}, \Phi_{1,y}, \ldots, \Phi_{N-w,y}\}$. Each block may then perform the updating sequentially on its sequence of ROIs while other blocks do the same, achieving the coarse parallelism.

For the fine level parallelism the kernel runs in an iterative manner, starting with initialized values for $R^{(}_{0,y}i)$, $i = 1, 2, \ldots, P$ and lists of the sampling locations and interpolation coordinates. In the first step, portions of $\Phi_{0,y}$ are loaded into shared memory, and each thread calculates the update values for one sampling location until the list is exhausted. These sampling locations are then updated and normalized. In the next step, each thread then calculates one interpolation until the interpolation list is exhausted. The threads then reduce the interpolations, averaging to calculate S. The kernel repeats this process for each ROI in the dependent sequence and then expires.

9.5.1 Memory access patterns and shared memory

With each thread calculating a pair of update values, the memory accesses to Φ overlap significantly among threads. Each update value requires w multiply-add operations, and some elements of Φ may be accessed up to w times. For this reason Φ is stored in shared memory to avoid repeated reads to global memory. This decision is key since effective shared memory usage is one of the critical components of algorithm performance on GPU.

Due to limited shared memory sizes, the autocorrelation matrices are maintained in global memory. This presents a problem as the calculation of update values cannot be organized among the threads so that accesses to $R^{(i)}$ are coalesced. Storing R in texture memory would be beneficial for caching and hardware interpolation, however textures are read only and cannot be changed within the duration of a kernel.

The limitations on shared memory size prohibit a general implementation for different ROI sizes. For this reason, the implementation used in Section 9.8 focused on the case $w = 32, \Delta\theta = \pi/8$ that is useful for the analysis of 5X magnification images. Each block was assigned 128 threads, 2176 bytes of shared memory, and 8 registers/thread to achieve a 100% occupancy on a Quadro FX5600 card.

9.6 Related works

A description of works in the area of high performance computing for image processing is presented in Chapter 3.6.

For the proposed fast correlation updating algorithm, a similar idea is found in the work on fast median filtering [139]. In this algorithm the ROI filter response is calculated at every position in the image by updating a kernel histogram based on the incoming and exiting information as the ROI shifts.

9.7 Experimental Setup

Experiments were performed to examine the effects of correlation updating, parallelization, and GPU implementation. Four implementations for calculating TPCF features were produced:

1. **Serial direct FFT.** A fully serial implementation of the direct FFT-based method, written in C++ using the FFTW library [48].

2. **Serial correlation updating.** A fully serial implementation of the correlation updating method, written in C++.

3. **Parallel correlation updating.** A parallel SMP/multi-node implementation of the correlation updating method, written in C++ using MPI.

4. **GPU correlation updating.** A GPU implementation of correlation updating, using C++/CUDA. The implementation is specific for $w = 32, \Delta\theta = \pi/8$.

9.7.1 Hardware

The above implementations were tested on a GPU equipped cluster, the BALE system at the Ohio Supercomputer Center (see Figure 4.1). The BALE supercomputer is endowed with 55 workstation nodes based on a dual-core Athlon 64 X2 architecture with integrated graphics card and 16 visualization nodes enhanced with dual-socket x dual-core AMD Opteron 2218 CPUs and dual-card Nvidia Quadro FX 5600 GPUs. All of these nodes are interconnected with Infiniband, and include a 750 GB, 7200 RPM local SATA II hard disk with 16 MB cache.

All GPU experiments and comparisons were run on the sixteen visualization nodes, where each node has 8 GB of DDR2 DRAM running at 667 MHz on the CPU side and

2x1.5 GB of on-board GDDR3 DRAM running at 1600 MHz on the GPU side, for a total of 11GB available DRAM per node. The remaining experiments were performed on the workstation nodes.

9.7.2 Data

Two sets of data were used in testing the three implementation varieties. The first dataset is used to compare direct-FFT and correlation updating and to examine scalability for the parallel implementation. It consists of ten 1000×1000 five-phase images taken from the placenta dataset described in Chapter 8.4.2.

The GPU time performance experiments used randomly generated images of size 256×256, 512×512, and 1024×1024 with two, four, and eight-phase variations. The accuracy performance experiment used one of the follicular lymphoma images described in Chapter 8.4.2.

9.8 Results

9.8.1 Correlation Updating

To compare the performance of correlation updating with the direct-FFT method, TPCF features were calculated for the ten test images using the parameters of Table 9.1. Parameters were chosen to reflect typical choices for the segmentation $5\times$ and $20\times$ magnifications, and also favorable and unfavorable FFT sizes. In the power of two cases the $2w - 1$ DFT was padded to $2w$. The transforms for the non power of two cases were not padded. Justification for this choice is provided in Table 9.2 where it is clear that this padding would be detrimental in the $w = 130$ case, and would only help marginally in the $w = 34$ case.

The execution times for the serial direct-FFT and correlation updating calculations are presented in Figure 9.3. The average per-image execution times for direct-FFT are 1280,

Table 9.1: Correlation updating and direct-FFT comparison parameters.

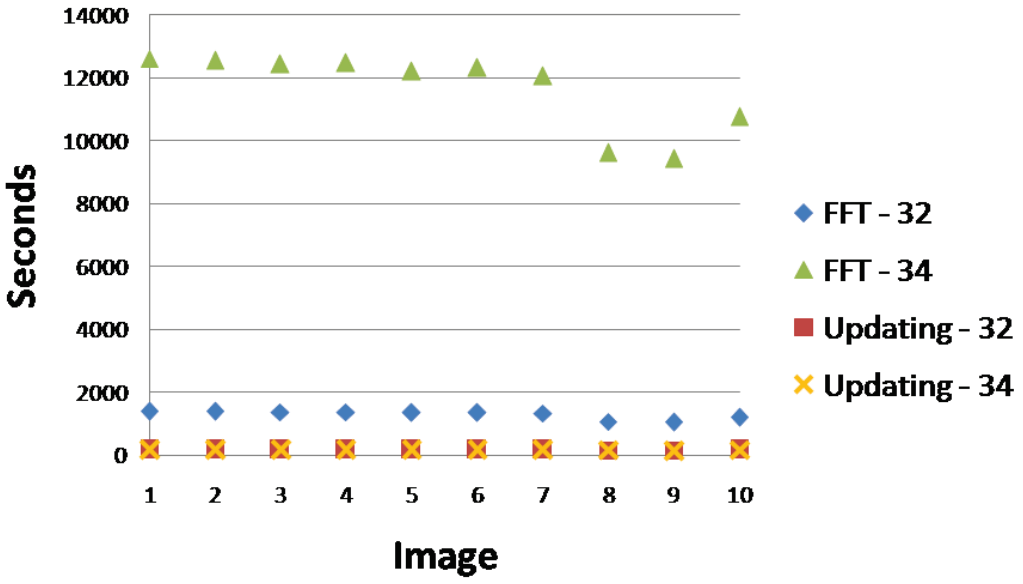
case	small-pow2	small	large-pow2	large
w	32	34	128	130
$\Delta\theta$	π/8	π/8	π/16	π/16

Table 9.2: Effect of padding on DFT transform time. Averaged over 100 transforms.

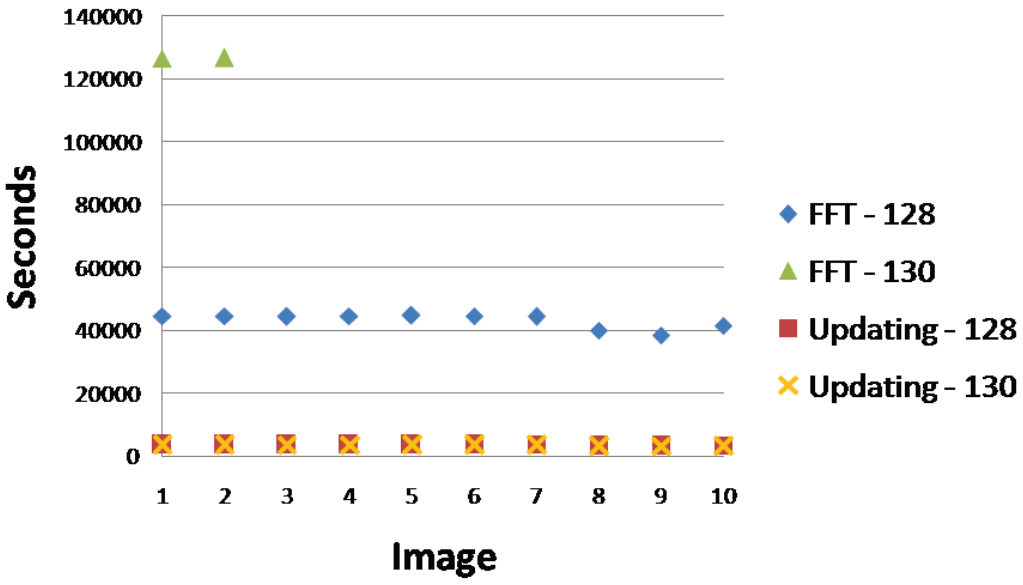
w	32	34	64	128	130	256
milliseconds	0.27	2.6	1.9	10	31	53

11637, 43129, and 126489 seconds for the w =32, 34, 128, and 130 sizes respectively. The corresponding average times for correlation updating are 162, 178, 3474, and 3557 seconds. Overall the increase in execution times from the small window cases to the large window cases are considerable. From $w = 32$ to 128, the increase for direct-FFT is 34x where the corresponding increase for correlation updating is only 21x. There is a strong penalty with the direct-FFT implementation for non power of two cases, roughly a 10x increase for the small window sizes and 3x for the large. The correlation updating implementation does not suffer the same penalties with commensurate increases limited to 1.1x for the small window case.

The average speedup factors for correlation updating are presented in Table 9.3. The speedup factors range between 8x and 67x depending on w. The larger speedup factors correspond to the non-power-of-two sizes due to the large penalty on FFT performance.



(a)



(b)

Figure 9.3: Execution times for serial direct-FFT and correlation updating. (a) Small w case. (b) Large w case.

Table 9.3: Average speedup for correlation updating.

	Small		Large	
w	32	34	128	130
speedup	7.9x	67.0x	12.4	x

9.8.2 Parallelization

To demonstrate parallelization scalability the TPCF features were calculated for the large power of two case using the parallel implementation of correlation updating on 2, 4, 8, 16, 32, and 64 processors on 1, 2, 4, 8, 16, and 32 nodes. The execution times for these configurations are presented in Figure 9.4. Table 9.4 contains the speedup factors for the parallel execution as compared to a fully serial single node implementation. These speedup factors are depicted graphically in Figure 9.5. There is a consistent reduction in execution time all the way through 16 processors, with more limited gains for the 32 and 64 processor cases, indicating that the unparallelized portions of execution account for a considerable portion of the total execution time. There is a relatively large amount of communication required for the worker nodes to report TPCF values to the head node, with each ROI generating $P(w/2+1)$ double-precision elements. In the case of a single 1000×1000 test image this corresponds to approximately 1.85 GBytes. Where increasing the number of nodes reduces the time spent in computation, the time spent in communication remains unchanged and the result on scalability is apparent.

9.8.3 GPU Implementation

To demonstrate execution time performance TPCF features were calculated using correlation updating implementations on both CPU and GPU for random images of size 256×256,

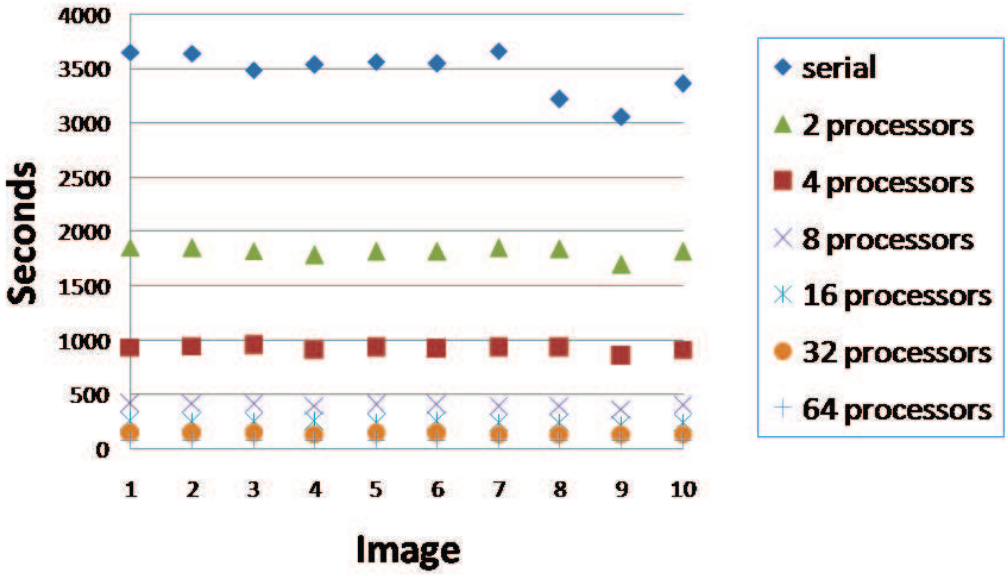


Figure 9.4: Execution times for parallel correlation updating, $w = 128$ case.

Table 9.4: Average speedup for parallel correlation updating, $w = 128$ case.

processors	2	4	8	16	32	64
speedup	1.9x	3.8x	8.5x	13.8x	24.5x	41.9x

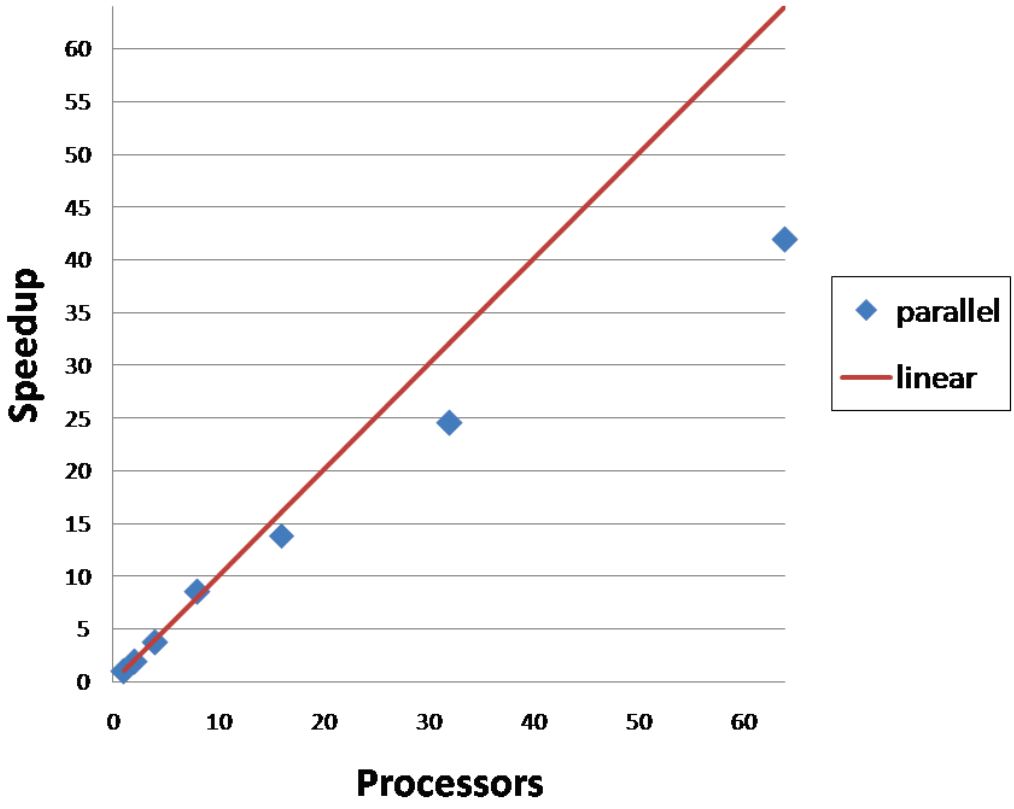


Figure 9.5: Scalability of parallel TPCF correlation updating implementation.

Table 9.5: Execution times for GPU correlation updating implementation.

	CPU			GPU		
phases	256×256	512×512	1024×1024	256×256	512×512	1024×1024
2	3.64	16.69	71.07	0.33	1.15	4.52
4	7.28	33.22	141.13	0.55	2.21	8.84
8	14.54	66.41	282.71	1.02	4.32	$\star$

$\star$ - watchdog timer intervention

512×512, and 1024×1024 with two, four, and eight phases. The results from this experiment are presented in Table 9.5. The corresponding speedup factors are presented in Table 9.6. All measures of execution time include communication and transfer of data between the CPU and GPU. For both CPU and GPU, execution time increases linearly with image size and the presence of additional phases, as expected. The speedup factor is greater for the more compute-intensive cases with larger image sizes and more phases, as the total amount of time spent in communication represents a smaller percentage of the total execution time. The kernel execution is interrupted by the CUDA watchdog timer in the case of 1024 $\times$ 1024 eight phase image. This is a feature of CUDA enabled to interrupt a kernel after a prescribed period to prevent a loss of graphics response for the user. The duration of the kernel depends on the sizes of the dependent sequences of ROIs, so to avoid watchdog timer interruptions the the sizes of these sequences must be limited based on the allowed kernel execution maximum.

Numerical Accuracy

The calculation of TPCF feature vectors is just one step in the segmentation procedure. After the features are calculated, they are subjected to dimensionality reduction prior to being clustered to form a segmentation. To demonstrate the effect on the end segmentation

Table 9.6: GPU/CPU speedup.

phases	256×256	512×512	1024×1024
2	11.1	15.6	15.7
4	13.2	15.0	16.0
8	14.2	15.4	⋆

⋆ - watchdog timer intervention

Table 9.7: Confusion matrix between single-precision GPU segmentation and double-precision CPU segmentation.

class	1	2	3	4	5	6
1	75692	0	0	0	0	0
2	0	136739	0	0	0	0
3	0	0	263018	0	0	0
4	0	0	0	253975	0	0
5	0	0	0	0	11532	0
6	0	0	0	0	0	1493

result, segmentations were generated for one of the Follicular Lymphoma examples (see Chapter 8.4.2) using both double-precision CPU calculated features, and single-precision GPU calculated features. The confusion matrix between the CPU and GPU generated segmentations is presented in Table 9.7. The segmentations are identical, indicating that the loss of precision has no impact on the outcome of the downstream analysis in this case.

9.9 Discussion and Conclusions

TPCF features provide a method for the segmentation of histological images, however, this capability is accompanied by a significant computational burden. The direct-FFT

method for deterministic TPCF calculation makes use of an efficient algorithmic staple, but execution time is strongly influenced by ROI size w as dictated by FFT transform size guidelines. The direct method also neglects the sparse autocorrelation sampling pattern and and close relationship between neighboring regions of interest resulting in significant amounts of wasted computation.

This chapter proposes a novel method of correlation updating that uses the derived relationship between the autocorrelations of neighboring ROIs to update TPCF values rather than computing them from scratch. This method simultaneously addresses the considerations of wasted computation and ROI size sensitivity without compromising accuracy. Using the linearity of correlation, the autocorrelation calculations can be updated from one ROI to the next, rather than computed from scratch. Furthermore, performing these updates directly in the image domain permits the sampling locations to be selectively updated, and frees the algorithm from the sensitivity to ROI size. The improvements of correlation updating result in a speedup from 8-67x over the direct-FFT method.

Both multi-node and GPU hardware solutions were pursued to further reduce execution time. The parallelization of feature calculations produces a scalability up to 42x on 64 processors, reducing the total execution time for the set of ten 1000×1000 test images from 9.6 hours to just 13 minutes. General purpose GPU implementation of correlation updating provides a further 16x improvement over CPU, without compromising accuracy in segmentation results. This gain is impressive considering it is more than equivalent to using 16 processors on eight nodes, and puts performance within reach of end users who do not have access to production computing clusters.

CHAPTER 10

CONCLUSION

Microscopic imaging will play a central role in addressing the emergent grand challenges in biology. In the post-genomic era the ability to localize molecular information in tissue will be critical in understanding the roles of genes and discovering the structures of the molecular networks that they regulate. A realistic picture of complex phenomenon like cancer requires more than just the molecular information averaged over a heterogeneous tissue that ordinary “omic” approaches such as microarray provide. Information with resolution at the scale of individual cells and beyond is needed to understand both intracellular regulation as well as the role of intercellular interactions.

The scale of the data involved in the emerging problems in bioimaging is daunting. High throughput microscopy techniques enable scientists to generate hundreds of gigabytes to terabytes of high-resolution imagery for a single study that is limited in scope to one gene or interaction. The manual analysis of this quantity of visual information is often beyond the capability of determined individuals, let alone the issues with regards to inter or intra-observer variabilities. Both the scale of data and the need for a more quantitative approach suggests that image processing technology will play a role in the next phase of biological discovery.

This dissertation presents solutions for two common problems in microscopic image analysis: *reconstruction* and *tissue segmentation*. The proposed algorithms fit into a framework that is intended to provide researchers in biology with the tools to explore and quantify large image datasets (see Figures 1.2, 1.3). The algorithms were developed to be generalizable with wide applicability to different tissues and stains. Emphasis is placed on addressing the challenges of content and image size that microscopic images pose to the state-of-the-art in image processing. For each algorithm an implementation was pursued that uses both theory and parallelization to reduce execution times. The emerging GPU architecture was especially useful in this regard.

Chapter 2 describes the *two-stage algorithm* for the reconstruction of tissues from sequences of serial section images. The algorithm is fast, scalable, and parallelizable and is capable of correcting the nonrigid distortions of sectioned microscope images. Rigid initialization follows a simply reasoned process of matching *high level features* using feature descriptions and geometric constraints. Nonrigid registration refines the rigid initialization by using the estimates of rigid initialization to precisely match intensity features using an FFT-based implementation of normalized cross-correlation.

Chapter 3 describes the implementation of the two stage algorithm using general purpose computing on graphics processors (GPU). A computational framework was been developed to expedite execution by parallelizing FFT computations using general purpose computing on GPU. A solid heterogeneous and cooperative multiprocessor platform was established using an AMD Opteron CPU and a pair of Nvidia Quadro GPUs, where the best features of each processor were fully exploited for applying higher degree of parallelism at a variety of levels: Multi-task for simultaneous executions of CPU and GPU codes, SMP (Symmetric MultiProcessing) for multicard GPUs using pthreads, and SIMD (Simple

Instruction Multiple Data) for the 128 stream processors of the GPU using CUDA. The features of GPUs combined with multi-socket programming achieved speed-up factors of up to 4.11x on a single GPU and 6.68x on a pair of GPUs using CUDA and pthreads versus a fully serial C++ CPU implementation. Execution results were shown for a benchmark composed of large-scale images derived from two different sources: mouse placenta (16K 16K pixels) and mouse mammary (23K 62K pixels). Using a fully serial C++ implementation it takes more than 12 hours to register a typical sample composed of 500 placenta slides. This time was reduced to less than 2 hours using two GPUs.

Chapter 4 extends the GPU implementation of the two stage algorithm to clusters of GPU-equipped computing nodes. The heterogeneous and cooperative multiprocessor system of Chapter 3 was augmented to include parallelisms at the multi-node level, using MPI for data partitioning across nodes, and the multi-core level, using either MPI or pthreads. For a mammary sample composed of 500 slides (23K 62K pixels each), it takes more than 181 hours to accomplish the registration process on a single Opteron CPU. This was reduced to 50 hours when enabling the GPU as co-processor, and minimized to 3.7 hours for a total speedup of 49x when all 32 CPUs and GPUs participate in our multiprocessor cooperative environment. While GPU-assisted versions were more effective at an intra-node layer, the CPU showed higher gains on inter-node parallelism, suggesting that they may complement each other on hybrid supercomputers.

The problem of registering images of tissues with different stains is addressed in Chapter 5, where a novel metric of correlation sharpness is proposed for comparing intensity signals. The sharpness of the normalized cross-correlation function was established as a similarity measure for comparing intensity information between two images with different stains. This helps avoid the high computational cost of more sophisticated approaches,

which is critical for processing images at this scale. In order to improve the matching accuracy, a multiple resolution approach was adopted for key regions of interests. The algorithm has been tested using real histological images of mouse mammary gland sample in a breast tumor microenvironment study. The results show that the algorithm is highly accurate. This work lays the foundation for large scale gene expression mapping of mouse breast tumor microenvironment where the plan is to map expression levels for 50-100 genes over four stages of tumor progression.

Chapter 6 also addresses the problem of different stain registration, but in the scenario where intensity information is not sufficient for accurate matching. An automatic matching method is presented that builds on the high-level feature matching procedure of rigid initialization. Since matching high-level features individually is a high probability-of-error endeavor, using these matches for nonrigid registration typically results in poor conformation between the registered images, due to the freedom of nonrigid transformations. Confidence in matches between individual features is increased by verifying the existence of coherent networks of features in the surrounding areas, allowing the matches to serve as control points for automatic nonrigid registration. Validation using a follicular lymphoma image dataset showed that the automatic nonrigid registrations were equivalent to manual nonrigid registrations when a sufficient feature set can be extracted.

The final topic on reconstruction is contained in Chapter 7 which proposes a method for the reconstruction of tissues under constraints on the structure of microanatomy. The key contribution is the integration of a structural constraint into the reconstruction process. As opposed to the traditional pairwise sequential registration approach that infers structure from images one pair at a time, the proposed method uses information from multiple images to enforce a structural criteria. The motivating example of reconstructing mammary ducts

provides a significant example of the benefits of this approach. By imposing a smoothness criteria the ducts can be registered naturally resulting in reconstructions with visible bifurcations. The use of an acausal smoothing filter enables the smoothing process to take into account not only where the duct has been but where it is heading. The entire process is fast, automatic, and produces credible representations of the morphology of structures of interest.

Chapter 8 introduces the problem of segmenting tissues and proposes the *two point correlation function* as a feature for tissue segmentation. By considering tissues as arrangements of discrete and biologically meaningful components, the problem of tissue segmentation can be cast into the heterogeneous materials framework. The TPCF, a stochastic geometric function, provides a means for acquiring statistics on the shape, size, and spatial distributions of these biological components to serve as cues for the segmentation of tissues. For both natural and tissue image examples, TPCF features were demonstrated to posses a simple but peculiar distribution in feature space, being confined to smooth manifold-like structures with relatively low dimension. In the follicular lymphoma example these distributions permit an unsupervised segmentation using lossy coding clustering, however, the lack of uniformity in the color distributions of the raw images implies that more sophisticated preprocessing is required to produce consistent phase images for TPCF feature calculation. The same phenomenon was also observed in the supervised segmentation of placenta where images were bound into cliques as indicated by effective mutual segmentations of one another. Regardless, many placenta segmentations were effective at 95% and beyond.

Chapter 9 presents methods for the acceleration of TPCF calculations, based on shortcuts derived from theory and hardware solutions. A novel method for the calculation of

TPCF features is derived, based on the linearity of correlation and the relationship between the autocorrelations of neighboring regions-of-interest with shared content. This method simultaneously addresses the considerations of wasted computation and ROI size sensitivity without compromising accuracy, resulting in improvements from 8-67x over a naive direct-FFT calculation method. The multi-node parallelization of feature calculations produces a scalability up to 42x on 64 processors, reducing the total execution time for the set of ten 1000×1000 test images from 9.6 hours to just 13 minutes. General purpose GPU implementation of correlation updating provides a further 16x improvement over CPU on a single node. This gain is impressive considering it is more than equivalent to using 16 processors on eight nodes, and puts performance within reach of end users who do not have access to production computing clusters.

APPENDIX A

SEGMENTATION RESULTS FOR MOUSE PLACENTA LABYRINTH

Table A.1: Segmentation accuracy (%).

Image	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	median
1	95.0	91.9	88.3	90.7	92.5	80.2	87.8	89.4	85.2	93.5	94.8	97.0	95.6	86.7	91.2	68.3	73.1	58.3	90.0
2	87.8	97.0	89.7	90.7	91.8	89.8	88.5	88.7	90.5	77.8	82.0	89.9	91.4	87.2	89.5	86.9	86.0	57.5	89.1
3	89.4	97.2	90.1	91.3	93.7	89.6	90.7	91.6	91.0	79.2	84.3	94.8	93.0	89.6	89.2	87.2	85.3	59.8	89.8
4	90.5	96.9	90.2	93.1	93.1	91.2	90.6	91.4	88.5	85.0	87.9	96.3	93.5	89.4	93.8	80.2	76.8	59.1	90.5
5	88.1	97.1	89.6	90.9	95.7	93.0	91.4	91.8	92.7	80.7	89.0	96.5	94.9	92.5	97.7	89.0	86.7	59.6	91.6
6	87.6	96.4	89.5	89.6	93.8	90.0	89.5	91.3	92.0	77.2	86.6	94.8	92.3	89.1	95.9	91.7	88.4	59.6	89.8
7	88.0	96.2	89.3	89.6	94.9	89.3	91.1	90.9	92.4	75.6	85.9	95.8	95.8	93.2	95.7	93.8	90.6	62.1	91.0
8	89.6	97.1	90.7	91.8	96.4	93.0	93.8	94.1	93.5	82.7	92.5	96.7	97.1	95.8	98.0	90.1	90.2	61.0	93.2
9	85.1	95.6	88.7	87.0	93.1	89.4	89.4	91.0	94.0	73.3	84.9	91.2	92.0	87.5	96.0	94.9	91.2	61.8	90.2
10	92.7	81.1	79.4	84.1	81.2	70.7	67.4	78.2	67.7	96.7	72.6	92.7	78.4	67.3	79.0	59.1	66.6	58.3	78.3
11	90.1	96.9	89.8	92.7	96.0	91.8	94.1	93.9	92.1	85.1	95.6	97.7	98.1	95.7	97.8	86.9	81.6	60.9	93.3
12	90.0	96.9	89.9	92.4	96.2	92.2	94.5	93.7	91.5	87.1	94.9	97.8	98.0	94.3	97.4	83.2	79.2	60.3	93.0
13	87.9	97.2	89.4	90.8	95.9	92.8	92.5	92.5	93.1	83.4	93.2	97.1	97.6	94.6	98.0	86.7	85.4	59.7	92.6
14	84.7	94.9	88.0	88.0	94.6	91.0	92.2	91.3	92.3	70.4	89.6	96.4	97.5	95.6	98.2	94.5	94.5	65.6	92.2
15	84.8	94.8	88.3	87.4	94.9	90.9	91.4	91.4	92.3	70.4	89.1	96.3	96.7	95.1	98.1	94.2	93.5	64.6	91.8
16	74.8	84.0	81.5	70.0	82.1	75.3	79.9	78.2	81.2	52.6	71.5	81.7	86.8	83.0	92.4	96.0	89.4	67.9	81.3
17	82.3	89.6	85.9	83.4	91.9	82.7	88.2	84.3	86.5	61.3	78.4	94.9	94.0	93.3	94.0	94.5	95.4	69.1	87.4
18	67.5	69.6	58.4	54.6	39.6	48.0	69.5	43.4	74.2	35.4	59.5	36.9	48.6	48.0	46.6	59.2	61.8	78.6	56.5
median	87.9	96.3	89.4	90.1	93.7	89.9	90.6	91.3	91.8	78.5	87.2	96.1	94.4	91.1	95.8	88.1	86.3	60.6	

BIBLIOGRAPHY

[1] Colin L. Smithpeter, Andrew K. Dunn, A. J. Welch, and Rebecca Richards-Kortum. Penetration depth limits of in vivo confocal reflectance imaging. *Applied Optics*, 37(13):2749–2754, 1998.

[2] Min Gu and Xiaosong Gan. Penetration depth in multiphoton fluorescence microscopy. In *Proceedings of the Second Asian and Pacific Rim Symposium on Biophotonics, 2004.*, pages 72–73, Dec. 2004.

[3] Rafael Yuste. Fluorescence microscopy today. *Nature Methods*, 2(12):902–904.

[4] Jeff Lichtman. Fluorescence microscopy. *Nature Methods*, 2(2):910–919.

[5] M Chalfie, Y Tu, G Euskirchen, WW Ward, and DC Prasher. Green fluorescent protein as a marker for gene expression. *Science*, 263(5148):802–805, 1994.

[6] Nathan C. Shaner, Paul A. Steinbach, and Roger Y. Tsien. A guide to choosing fluorescent proteins. *Nature Methods*, 2(12):905–909, November 2005.

[7] R. Heim, A. B. Cubitt, and R. Y. Tsien. Improved green fluorescence. *Nature*, 373(6516):663–664, February 1995.

[8] Xiaokun Shu, Antoine Royant, Michael Z. Lin, Todd A. Aguilera, Varda Lev-Ram, Paul A. Steinbach, and Roger Y. Tsien. Mammalian Expression of Infrared Fluorescent Proteins Engineered from a Bacterial Phytochrome. *Science*, 324(5928):804–807, 2009.

[9] C. Levinthal and R. Ware. Three-dimensional reconstruction from serial sections. *Nature*, 236:207–210, 1972.

[10] J. Capowski. Computer-aided reconstruction of neuron trees from several sections. *Computational Biomedical Research*, 10(6):617–629, 1977.

[11] E. Johnson and J. Capowski. A system for the three-dimensional reconstruction of biological structures. *Computational Biomedical Research*, 16(1):79–87, 1983.

[12] D. Huijismans, W. Lamers, J. Los, and J. Strackee. Toward computerized morphometric facilities: a review of 58 software packages for computer-aided three-dimensional reconstruction, quantification, and picture generation from parallel serial sections. *The Anatomical Record*, 216(4):449–470, 1986.

[13] V. Moss. The computation of 3-dimensional morphology from serial sections. *European Journal of Cell Biology*, 48:61–64, 1989.

[14] R. Brandt, T. Rohlfing, J. Rybak, S. Krofczik, A. Maye, M. Westerhoff, H.-C. Hege, and R. Menzel. A three-dimensional average-shape atlas of the honeybee brain and its applications. *The Journal of Comparative Neurology*, 492(1):1–19, 2005.

[15] Kishore Mosaliganti, Firdaus Janoos, Okan Irfanoglu, Randall Ridgeway, Raghu Machiraju, Kun Huang, Joel Saltz, Gustavo Lenoe, and Michael Ostrowski. Tensor classification of N-point correlation function features for histology tissue segmentation. *Medical Image Analysis*, 13(1):156–166, 2009.

[16] T. Pan and Kun Huang. Virtual mouse placenta: Tissue layer segmentation. In *International Conference of the Engineering in Medicine and Biology Society*, pages 3112–3116, Jan. 2005.

[17] K. Mosaliganti, R. Machiraju, and K. Huang. Geometry-driven visualization of microscopic structures in biology. In *IEEE International Symposium on Biomedical Imaging*, pages 828–831, May 2008.

[18] P. S. Umesh Adiga, B. B. Chaudhuri, and K. Rodenacker. Semi-automatic segmentation of tissue cells from confocal microscope images. In *Proceedings of the International Conference on Pattern Recognition*, page 494, Washington, DC, USA, 1996. IEEE Computer Society.

[19] R. Fernandez-Gonzalez, T. Deschamps, A. Idica, R. Malladi, and C. Ortiz de Solorzano. Automatic segmentation of histological structures in mammary gland tissue sections. *Journal of Biomedical Optics*, 9(3):444–453, 2004.

[20] Adam Karlsson, Kent Stråhlén, and Anders Heyden. A fast snake segmentation method applied to histopathological sections. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 261–274, 2003.

[21] A. Karlsson, K. Strahlen, and A. Heyden. Segmentation of histopathological sections using snakes. In *Proceedings of Scandinavian Conference on Image Analysis*, pages 595–602, 2003.

[22] P. Wenzel, L. Wu, R. Sharp, A. de Bruin, J. Chong, W. Chen, G. Dureska, E. Sites, T. Pan, A. Sharma, K. Huang, R. Ridgway, K. Mosaliganti, R. Machuraju, J. Saltz, H. Yamamoto, J. Cross, M. Robinson, and G. Leone. Rb is critical in a mammalian tissue stem cell population. *Genes & Development*, 21(1):85–97, 2007.

[23] J. Hajnal, H. Derek, and D. Hawkes. *Medical Image Registration*. CRC Press, Boca Raton, FL, 2001.

[24] A. Goshtasby. *2-D and 3-D Image Registration: For Medical, Remote Sensing, and Industrial Applications*. Wiley-Interscience, Hoboken, NJ, 2005.

[25] J. Streicher, D. Markus, S. Bernhard, R. Sporle, K. Schughart, and G. Muller. Computer-based three-dimensional visualization of developmental gene expression. *Nature Genetics*, 25:147–152, 2000.

[26] U. Braumann, J. Kuska, J. Einenkel, L. Horn, M. Luffler, and M. Huckel. Three-dimensional reconstruction and quantification of cervical carcinoma invasion fronts from histological serial sections. *IEEE Transactions on Medical Imaging*, 24(10):1286–1307, 2005.

[27] W. Crum, T. Hartkens, and D. Hill. Non-rigid image registration: Theory and practice. *The British Journal of Radiology*, 77(2):S140–S153, 2004.

[28] W. Hill and R. Baldock. The constrained distance transform: Interactive atlas registration with large deformations through constrained distance. In *Proceedings of the Workshop on Image Registration in Deformable Environments*, Edinburgh, UK, Sept. 8, 2006.

[29] T. Yoo. *Insight into Images: Principles and Practice for Segmentation, Registration, and Image Analysis*. AK Peters, Wellesley, MA, 2004.

[30] S. Sarma, J. Kerwin, L. Puelles, M. Scott, T. Strachan, G. Feng, J. Sharpe, D. Davidson, R. Baldock, and S. Lindsay. 3d modelling, gene expression mapping and post-mapping image analysis in the developing human brain. *Brain Research Bulletin*, 66(4-6):449–453, 2005.

[31] A. Jenett, J. Schindelin, and M. Heisenberg. The virtual insect brain protocol: creating and comparing standardized neuroanatomy. *BMC Bioinformatics*, 7:544, 2006.

[32] L. Cooper, K. Huang, A. Sharma, K. Mosaliganti, and T. Pan. Registration vs. reconstruction: Building 3-d models from 2-d microscopy images. In *Proceedings of the Workshop on Multiscale Biological Imaging, Data Mining and Informatics*, pages 57–58, Santa Barbara, CA, Sept. 7-8, 2006.

[33] K. Huang, L. Cooper, A. Sharma, and T. Pan. Fast automatic registration algorithm for large microscopy images. In *Proceedings of the IEEENLM Life Science Systems & Applications Workshop*, pages 1–2, Bethesda, MD, July 13-14 2006.

[34] P.A. Koshevoy, T. Tasdizen, and R.T. Whitaker. Implementation of an automatic slice-to-slice registration tool. SCI Institute Technical Report UUSCI-2006-018, University of Utah, 2006.

[35] J. Prescott, M. Clary, G. Wiet, T. Pan, and K. Huang. Automatic registration of large set of microscopic images using high-level features. In *Proceedings of the IEEE International Symposium on Medical Imaging*, pages 1284–1287, Arlington, VA, April 6-9, 2006.

[36] R. Mosaliganti, T. Pan, R. Sharp, R. Ridgway, S. Iyengar, A. Gulacy, P. Wenzel, A. de Bruin, R. Machiraju, , K. Huang, G. Leone, and J. Saltz. Registration and 3d visualization of large microscopy images. In *Proceedings of the SPIE Medical Imaging Meeting*, pages 6144:923–934, San Diego, CA, Feb. 11, 2006.

[37] O. Schmitt, J. Modersitzki, S. Heldmann, S. Wirtz, and B. Fischer. Image registration of sectioned brains. *International Journal of Computer Vision*, 73(1):5–39, 2007.

[38] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, 16(2):187–198, 1997.

[39] F. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.

[40] K. Rohr. *Landmark-Based Image Analysis: Using Geometric and Intensity Models*. Springer, New York, NY, 2007.

[41] S. Henn. A levenberg-marquardt scheme for nonlinear image registration. *BIT Numerical Mathematics*, 43(4):743–759, 2003.

[42] W. Woods., N. Galatsanos, and A. Katsaggelos. Em-based simultaneous registration, restoration, and interpolation of super resolved images. In *IEEE International Conference on Image Processing*, pages II: 303–306, 2003.

[43] A. Gueziec, X. Pennec, and N. Ayache. Medical image registration using geometric hashing. *IEEE Computational Science and Engineering*, 4(4):29–41, 1997.

[44] R. Bajcsy and S. Kovacic. Multiresolution elastic matching. *Computer Vision, Graphics, and Image Processing*, 46:1–21, 1989.

[45] M. Lefebure and L.D. Cohen. A multiresolution algorithm for signal and image registration. In *Proceedings of IEEE International Conference on Image Processing*, pages 3: 252–255, Washington, D.C., Oct. 26-29, 1997.

[46] A. Ruiz, M. Ujaldon, L. Cooper, and K. Huang. Non-rigid registration for large sets of microscopic images on graphics processors. *Journal of Signal Processing Systems*, Accepted.

[47] J. P. Lewis. Fast normalized cross-correlation. In *Vision Interface*, pages 120–123, Quebec City, CAN, June 15-19, 1995.

[48] M. Frigo and S.G. Johnson. Fftw library. Website, December 2007. `http://www.fftw.org`.

[49] Vijay S. Kumar, Benjamin Rutt, Tahsin M. Kurc, Umit V. Catalyurek, Joel H. Saltz, Sunny Chow, Stephan Lamont, and Maryann E. Martone. Imaging and visual analysis - large image correction and warping in a cluster environment. In *Proceedings of IEEE conference on Supercomputing*, page 79, 2006.

[50] S. Contassot Vivier and S. Miguet. A load-balanced algorithm for parallel digital image warping. *International Journal of Pattern Recognition and Artificial Intelligence*, 13(4):445, June 1999.

[51] Yan huang Jiang, Zhi ming Chang, and Xuejun Yang. A load-balanced parallel algorithm for 2d image warping. In *International Symposium on Parallel and Distributed Processing and Applications*, pages 735–745, 2004.

[52] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever. Mutual-information-based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, 22(8):986–1004, 2003.

[53] H. Wolfson and I. Rigoutsos. Geometric hashing: an overview. *IEEE Computational Science and Engineering*, 4(4):10–21, 1997.

[54] H. Peng, F. Long, M. Eisen, and E. Myers. Clustering gene expression patterns of fly embryos. In *Proceedings of IEEE International Symposium on Biomedical Imaging*, pages 1144–1147, Arlington, VA, April 6-9, 2006.

[55] M. Botnen and H. Ueland. The GPU as a computational resource in medical image processing. Technical report, Dept. of Computer and Information Science, Norwegian Univ. of Science and Technology, 2004.

[56] J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Kruger, A. E. Lefohn, and T. J. Purcell. A survey of general-purpose computation on graphics hardware. *Journal of Computer Graphics Forum*, 26(1):21–51, 2007.

[57] Nvidia. CUDA. Website, December 2007. `http://developer.nvidia.com/object/cuda.html`.

[58] GPGPU. A web site dedicated to the general-purpose processing on the GPU. Website, January 2008. `http://www.gpgpu.org`.

[59] Massimiliano Fatica, David P. Luebke, Ian A. Buck, John D. Owens, Mark J. Harris, John E. Stone, James C. Phillips, and Bernard Deschizeaux. Cuda tutorial at acm/ieee conference on supercomputing (november). December 2007.

[60] Nvidia. Cufft library. Website, December 2007. `http://developer.download.nvidia.com/compute/cuda/1_1/CUFFT_Library_1.1.pdf`.

[61] L. Cooper, S. Naidu, G. Leone, J. Saltz, and K. Huang. Registering high resolution microscopic images with different histochemical stainings - a tool for mapping gene expression with cellular structures. In *Proceedings of the Workshop on Microscopic Image Analysis with Applications in Biology*, Piscataway, NY, Sept. 21, 2007.

[62] T. Kim and Y.-J. Im. Automatic satellite image registration by combination of matching and random sample consensus. *IEEE Transactions on Geoscience and Remote Sensing*, 41(5):1111–1117, 2003.

[63] F. Ino, K. Ooyama, and K. Hagihara. A data distributed parallel algorithm for nonrigid image registration. *Parallel Computing*, 31(1):19–43, 2005.

[64] S. Warfield, F. Jolesz, and R. Kikinis. A high performance computing approach to the registration of medical imaging data. *Parallel Computing*, 24(9–10):1345–1368, 1998.

[65] T. Rohlfing and C. Maurer. Nonrigid image registration in shared-memory multiprocessor environments with applications to brains, breasts, and bees. *IEEE Transactions on Information Technology in Biomedicine*, 7(1):16–25, 1998.

[66] M. Ohara, H. Yeo, F. Savino, G. Iyengar, L. Gong, H. Inoue, H. Komatsu, V. Sheinin, S. Daijavad, and B. Erickson. Real time mutual information-based linear registration on the cell broadband engine processor. In *Proceedings of the IEEE International Symposium on Medical Imaging*, pages 33–36, Washington, DC, April 12-15, 2007.

[67] Z. Fan, F. Qiu, A. Kaufman, and S. Yoakum-Stover. GPU cluster for high performance computing. In *Proceedings of IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 47–53, Washington, DC, Nov. 11-17, 2006.

[68] W. Wu and P.A. Heng. A hybrid condensed finite element model with GPU acceleration for interactive 3d soft tissue cutting: Research articles. *Computer Animation and Virtual Worlds*, 15(3-4):219–227, 2004.

[69] Zhao, Y., Y. Han, Z. Fan, F. Qiu, Y. Kuo, A. Kaufman, and K. Mueller. Visual simulation of heat shimmering and mirage. *IEEE Transactions on Visualization and Computer Graphics*, 13(1):179–189, 2007.

[70] F. Ino, J. Gomita, Y. Kawasaki, and K. Hagihara. A gpgpu approach for accelerating 2-d/3-d rigid registration of medical images. In *Proceedings of International Symposium on Parallel and Distributed Processing and Applications*, pages 769–780, Sorrento, IT, Dec. 1-4, 2006.

[71] S. Guha, S. Krisnan, and S. Venkatasubramanian. Data visualization and mining using the GPU. In *Data Visualization and Mining Using the GPU, Tutorial at International Conference on Knowledge Discovery and Data Mining*, Chicago, IL, Aug. 21-24, 2005.

[72] M. Hadwiger, C. Langer, H. Scharsach, and K. Buhler. State of the art report on GPU-based segmentation. Technical Report TR-VRVIS-2004-17, VRVis Research Center, Vienna, Austria, 2004.

[73] K. Fatahalian, J. Sugerman, and P. Hanrahan. Understanding the efficiency of gpu algorithms for matrix-matrix multiplication. In *Proceedings of ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 133–137, New York, NY, USA, 2004. ACM.

[74] Kenneth Moreland and Edward Angel. The fft on a gpu. In *HWWS '03: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 112–119, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[75] Nvidia. High performance computing (HPC) - nvidia tesla many core parallel supercomputing. Website, January 2008. `http://www.nvidia.com/object/tesla_computing_solutions.html`.

[76] ATI. Gpu hardware solutions from amd/ati. Website, January 2008. `http://ati.amd.com/products/streamprocessor/specs.html`.

[77] MPI forum. MPI: A Message-Passing Interface Standard. Website, September 2008. `http://www.mpi-forum.org`.

[78] David R. Butenhof. *Programming with POSIX threads*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.

[79] H. Sugimoto, T. Mundel, M. Kieran, and R. Kalluri. Identification of fibroblast heterogeneity in the tumor microenvironment. *Cancer Biology & Therapy*, 5(12):1640–1646, 2006.

[80] C. Eng. Pten: one gene, many syndromes. *Human Mutation*, 22(3):183–198, 2003.

[81] Nicolas Wentzensen, Ulf-Dietrich Braumann, Jens Einenkel, Lars-Christian Horn, Magnus von Knebel Doeberitz, Markus Lffler, and Jens-Peer Kuska. Combined serial section-based 3d reconstruction of cervical carcinoma invasion using h&e/p16(ink4a)/cd3 alternate staining. *Cytometry Part A*, 71A:327–333, 2007.

[82] R. Marc and B. Jones. Molecular phenotyping of retinal ganglion cells. *Journal of Neuroscience*, 22(2):413–427, 2002.

[83] G. Bearman and R. Levenson. *Biological Imaging Spectroscopy*. CRC Press, 2003.

[84] R. van Vlierberghe, M. Sandel, F. Prins, L. van Iersel, C. van de Velde, R. Tollenaar, and P. Kuppen. Four-color staining combining fluorescence and brightfield microscopy for simultaneous immune cell phenotyping and localization in tumor tissue sections. *Microscopy Research and Technique*, 67(1):15–21, 2005.

[85] Ulf-Dietrich Braumann, Jens Einenkel, Lars-Christian Horn, Jens-Peer Kuska, Markus Lffler, Nico Scherf, and Nicolas Wentzensen. Registration of histologic colour images of different staining. In Heinz Handels, Jan Ehrhardt, Alexander Horsch, Hans-Peter Meinzer, and Thomas Tolxdorff, editors, *Bildverarbeitung fr die Medizin*, Informatik Aktuell, pages 231–235. Springer, 2006.

[86] F. Dick, VanLier S., and Banks P. Use of the working formulation for non-hodgkin's lymphoma in epidemiological studies: agreement between reported diagnoses and a panel of experienced pathologists. *Journal of National Cancer Institue*, 78:1137–1144, 1987.

[87] G. E. Metter, B. N. Nathwani, J. S. Burke, C. C. Winberg, R. B. Mann, M. Barcos, C. R. Kjeldsberg, C. C. Whitcomb, D. O. Dixon, T. P. Miller, and S. E. Jones. Morphological subclassification of follicular lymphoma: variability of diagnoses among hematopathologists, a collaborative study between the repository center and pathology panel for lymphoma clinical studies. *Journal of Clinical Oncology*, 3(1):25–38, 1985.

[88] O. Sertel, J. Kong, G. Lozanski, U. Catalyurek, J. H. Saltz, and M. Gurcan. Computerized microscopic image analysis of follicular lymphoma. In *Proc. of SPIE Medical Imaging*, 2008.

[89] O. Sertel, J. Kong, U. V. Catalyurek, G. Lozanski, J. H. Saltz, and M. Gurcan. Histopathological image analysis using model-based intermediate representation and color texture: Follicular lymphoma grading. *Journal of Signal Processing Systems for Signal, Image, and Video Technology*, Accepted, 2008.

[90] E. S. Jaffe, N. L. Harris, H. Stain, and J. W. Vardiman. *Tumors of haematopoietic and lymphoid tissues*. IARC Scientific, Lyon, France, 2001.

[91] W. Hiddemann, C. Buske, M. Dreyling, O. Weigert, G. Lenz, R. Forstpointner, C. Nickenig, and M. Unerhalt. Treatment strategies in follicular lymphomas: current status and future perspectives. *Journal of Clinical Oncology*, 23(26):6394–6399, 2005.

[92] R. B. Mann and C. W. Berard. Criteria for the cytologic classification of follicular lymphomas: a proposed alternative method. *Hematology Oncology*, 1:187–192, 1983.

[93] D. G. Altman and J. M. Bland. Measurement in medicine: The analysis of method comparison studies. *The Statistician*, 32(3):307–317, 1983.

[94] J. M. Bland and D. G. Altman. Statistical methods for assessing agreement between two methods of clinical measurement. *Lancet*, 1(8476):307–310, February 1986.

[95] R. Sharp, R. Ridgway, S. Iyengar, Alexandra Gulacy, P. Wenzel, A. de Bruin, Raghu Machiraju, Kun Huang, Gustavo Leone, and Joel H. Saltz. Registration and 3d visualization of large microscopy images. In *Proceedings of the SPIE Annual Medical Imaging Meetings*, pages 923–934, Dec 2006.

[96] Jeffrey Prescott, M. Clary, Gregory J. Wiet, and Kun Huang. Automatic registration of large set of microscopic images using high-level features. In *Proceedings of the IEEE International Symposium on Medical Imaging*, pages 1284–1287, Dec 2006.

[97] Kun Huang, L. Cooper, A. Sharma, and T. Pan. Fast automatic registration algorithm for large microscopy images. *Life Science Systems and Applications Workshop*, 0:1–2, 2006.

[98] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.

[99] Salvatore Torquato. *Random Heterogeneous Materials Microsctructure and Macroscopid Properties*. Springer-Verlag, New York, NY, 2002.

[100] Adam B. Hopkins, Frank H. Stillinger, and Salvatore Torquato. Dense sphere packings from optimized correlation functions. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 79(3):031123, 2009.

[101] Antonello Scardicchio, Chase E. Zachary, and Salvatore Torquato. Statistical properties of determinantal point processes in high-dimensional euclidean spaces. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 79(4):041108, 2009.

[102] Y. Jiao, F. H. Stillinger, and S. Torquato. Modeling heterogeneous materials via two-point correlation functions: Basic principles. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 76(3):031110, 2007.

[103] Y. Jiao, F. H. Stillinger, and S. Torquato. Modeling heterogeneous materials via two-point correlation functions. ii. algorithmic details and applications. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 77(3):031135, 2008.

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

[105] R.M. Haralick and L.G. Shapiro. Image segmentation techniques. *Computer Vision Graphics and Image Processing*, 29(1):100–132, January 1985.

[106] Dorin Comaniciu, Peter Meer, and Senior Member. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:603–619, 2002.

[107] R.o. duda, p.e. hart, and d.g. stork, pattern classification, new york: John wiley & sons, 2001, pp. xx + 654, isbn: 0-471-05669-3. *Journal of Classification*, 24(2):305–307, 2007.

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

[109] J. Rittscher, P.H. Tu, and N. Krahnstoever. Simultaneous estimation of segmentation and shape. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 486–493 vol. 2, June 2005.

[110] Luc Florack and Arjan Kuijper. The topological structure of scale-space images. *Journal of Mathematical Imaging and Vision*, 12(1):65–79, 2000.

[111] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, Feb 2001.

[112] L.S. Davis, M. Clearman, and J.K. Aggarwal. An empirical evaluation of generalized cooccurrence matrices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(2):214–221, March 1981.

[113] K. Valkealahti and E. Oja. Reduced multidimensional cooccurrence histograms in texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):90–94, January 1998.

[114] K. Hammouche, M. Diaf, and J. G. Postaire. A clustering method based on multidimensional texture analysis. *Pattern Recognition*, 39(7):1265–1277, 2006.

[115] Brent Woods, Bradley Clymer, Joel Saltz, and Tahsin Kurc. A parallel implementation of 4-dimensional haralick texture analysis for disk-resident image datasets. In *Proceedings of ACM/IEEE conference on Supercomputing*, page 48, Washington, DC, USA, 2004. IEEE Computer Society.

[116] R Sivaramakrishna, KA Powell, ML Lieber, WA Chilcote, and R Shekhar. Texture analysis of lesions in breast ultrasound images. *Computerized Medical Imaging and Graphics*, 26(5):303–7, 2002.

[117] P. S. Umesh Adiga and B. B. Chaudhuri. An efficient method based on watershed and rule-based merging for segmentation of 3-d histo-pathological images. *Pattern Recognition*, 34(7):1449–1458, 2001.

[118] Vinh-Thong Ta, Olivier Lézoray, Abderrahim Elmoataz, and Sophie Schüpp. Graph-based tools for microscopic cellular image segmentation. *Pattern Recognition*, 42(6):1113–1125, 2009.

[119] A. Hafiane, F. Bunyak, and K. Palaniappan. Clustering initiated multiphase active contours and robust separation of nuclei groups for tissue segmentation. pages 1–4, 2008.

[120] Vannary Meas-Yedid, Sorin Tilie, and Jean-Christophe Olivo-Marin. Color image segmentation based on markov random field clustering for histological image analysis. In *Proceedings of the 16th International Conference on Pattern Recognition Volume 1*, page 10796, Washington, DC, USA, 2002. IEEE Computer Society.

[121] Roberto Rodríguez, Teresa E. Alarcón, and Juan J. Abad. Blood vessel segmentation via neural network in histological images. *Journal of Intelligent and Robotic Systems*, 36(4):451–465, 2003.

[122] R. Fernandez-Gonzalez and C.O. de Solorzano. A tool for the quantitative spatial analysis of mammary gland epithelium. In *26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2004.*, volume 1, pages 1549–1552, Sept. 2004.

[123] R. Fernandez-Gonzalez, T. Deschamps, A. Idica, R. Malladi, and C. Ortiz de Solorzano. Automatic segementation of histological structures in normal and neoplastic mammary gland tissue sections. In J.-A. Conchello, C. J. Cogswell, and T. Wilson, editors, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 4964 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 123–134, July 2003.

[124] Akif Burak Tosun, Melih Kandemir, Cenk Sokmensuer, and Cigdem Gunduz-Demir. Object-oriented texture analysis for the unsupervised segmentation of biopsy images for cancer detection. *Pattern Recognition*, 42(6):1104–1112, 2009.

[125] R. Ridgway, O. Irfanoglu, Raghu Machiraju, and Kun Huang. Image segmentation with tensor-based classification of n-point correlation functions. *Proceedings of the Microscopic Image Analysis with Applications in Biology (MIAAB) Workshop in MICCAI*, Dec 2006.

[126] Firdaus Janoos, M. Okan Irfanoglu, Kishore Mosaliganti, Raghu Machiraju, Kun Huang, Pamela Wenzel, Alain de Bruin, and Gustavo Leone. Multi-resolution image segmentation using the 2-point correlation functions. In *Proceedings of IEEE International Symposium on Biomedical Imaging*, pages 300–303, 2007.

[127] Hang Zhang and Xiao Qing Li. Scaling argument for the amplitudes of clustering correlation functions. *The Astrophysical Journal*, 535(1):24–29, 2000.

[128] Donghai Zhao, Y. P. Jing, , and G. Borner. Pairwise velocity dispersion of galaxies at high redshift: Theoretical predictions. *The Astrophysical Journal*, 581(2):876–885, 2002.

[129] Robert M. Haralick, K. Shanmugam, and Its'hak Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, 3(6):610–621, 1973.

[130] Brian Lovell, Ross Walker, Ross F. Walker, Paul Jackway, and Paul Jackway. Cervical cell classification via co-occurrence and markov random field features. In *In Proceedings of DICTA-95, Digital Image Computing: Techniques and Applications*, pages 294–299. IEEE, 1995.

[131] Kai Huang, Meel Velliste, and Robert F. Murphy. Feature reduction for improved recognition of subcellular location patterns in fluorescence microscope images. In *Proceedings of the SPIE, 4962, in press*, pages 307–318, 2003.

[132] Meel Velliste and Robert F. Murphy. Automated determination of protein subcellular locations from 3d fluorescence microscope images. In *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pages 867–870, 2002.

[133] Jean-Philippe Thiran, Benot Macq, and Jacques Mairesse. Morphological classification of cancerous cells, 1994.

[134] Kishore Mosaliganti, Lee Cooper, Richard Sharp, Raghu Machiraju, Gustavo Leone, Kun Huang, and Joel H. Saltz. Reconstruction of cellular biological structures from optical microscopy data. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):863–876, 2008.

[135] Y. Ma, H. Derksen, W. Hong, and J. Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

[136] K.S. Knudsen and L.T. Bruton. Recursive pruning of the 2d dft with 3d signal processing applications. *IEEE Transactions on Signal Processing*, 41(3):1340–1356, Mar 1993.

[137] Zhong Hu and Honghui Wan. A novel generic fast fourier transform pruning technique and complexity analysis. *IEEE Transactions on Signal Processing*, 53(1):274–282, Jan. 2005.

[138] Franz Franchetti and Markus Puschel. Generating high performance pruned fft implementations. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009*, pages 549–552, April 2009.

[139] T.S. Huang, G.J. Yang, and G.Y. Tang. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27:13–18, February 1979.