Methods for automated analysis of macular OCT data

by

Andrew R. Lang

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

August, 2016

Abstract

Optical coherence tomography (OCT) is fast becoming one of the most important modalities for imaging the eye. It provides high resolution, cross-sectional images of the retina in three dimensions, distinctly showing its many layers. These layers are critical for normal eye function, and vision loss may occur when they are altered by disease. Specifically, the thickness of individual layers can change over time, thereby making the ability to accurately measure these thicknesses an important part of learning about how different diseases affect the eye.

Since manual segmentation of the layers in OCT data is time consuming and tedious, automated methods are necessary to extract layer thicknesses. While a standard set of tools exist on the scanners to automatically segment the retina, the output is often limited, providing measurements restricted to only a few layers. Analysis of longitudinal data is also limited, with scans from the same subject often processed independently and registered using only a single landmark at the fovea. Quantification of other changes in the retina, including the accumulation of fluid, are also generally unavailable using the built-in software.

ABSTRACT

In this thesis, we present four contributions for automatically processing OCT data, specifically for data acquired from the macular region of the retina. First, we present a layer segmentation algorithm to robustly segment the eight visible layers of the retina. Our approach combines the use of a random forest (RF) classifier, which produces boundary probabilities, with a boundary refinement algorithm to find surfaces maximizing the RF probabilities. Second, we present a pair of methods for processing longitudinal data from individual subjects: one combining registration and motion correction, and one for simultaneously segmenting the layers across all scans. Third, we develop a method for segmentation of microcystic macular edema, which appear as small, fluid-filled, cystoid spaces within the retina. Our approach again uses an RF classifier to produce a robust segmentation. Finally, we present the development of macular flatspace (MFS), a computational domain used to put data from different subjects in a common coordinate system where each layer appears flat, thereby simplifying any automated processing. We present two applications of MFS: inhomogeneity correction to normalize the intensities within each layer, and layer segmentation by adapting and simplifying a graph formulation used previously.

Dissertation committee:

Dr. Jerry L. Prince (First Reader)

Dr. Jin U. Kang (Second Reader)

Dr. Gerard G. L. Meyer

Acknowledgments

Pursuing a Ph.D. degree is not something to be taken lightly; I have seen many students come and go without finishing, which is not to say that they failed, but only that it wasn't the right path for them. For me, it was a long journey filled with many struggles along the way, but in the end, I am confident that it was the right decision. As a result, I am to incredibly thankful to my previous mentors who pushed me down this path. Without their guidance, I would not be where I am today, so thank you to Dr. Purang Abolmaesumi, Dr. Parvin Mousavi, and Dr. Gabor Fichtinger, all of whom advised me during my Master's program at Queen's University, and Dr. Ameet Jain, my mentor during an internship at Philips Research.

During my time at Johns Hopkins, I have been lead by two terrific professors, Dr. Jerry Prince and Dr. Bruno Jedynak. Before arriving at JHU, I was given a lot of advice that Dr. Prince is one of the best professors to work with, and I will happily give that same advice to anyone considering working with him. His guidance and support have been unwavering, looking out for my best interests from day one. Overall, he taught me how to be a better researcher, thinker, writer, and person, and

I am forever grateful for that. Dr. Jedynak showed me what it means to be passionate about a project, and from the day he took me in as a part of his team, I was excited to be a part of his vision. Thank you for all of your help, you were truly influential in my development. I also have to thank Aaron Carass, who kept all aspects of the lab in working order. Without his help, this journey would not have been nearly as smooth as it was. I am also thankful for his meticulous reading of the many papers I wrote in my time here. His advice consistently forced me to think critically, and absolutely improved the quality of my research.

There is a large number of people who have helped me in my research along the way, beginning with our clinical collaborators, who took time out of their busy schedules to help in any way they could. They also guided many of the research directions I went in, and provided all of data that I used in my experiments. So thank you to Dr. Calabresi, Dr. Saidha, Dr. Ying, and Dr. Solomon, as well as all of the people affiliated with their labs, including Elias, Anna, Omar, Pavan, and Julia, who helped us with data acquisition, gathered demographic data, and gave us feedback on the algorithms that we developed.

I am also thankful to others involved with the OCT project that I worked on including Dhruv, Min, Bhavna, and Aaron; your insightful conversations and feedback were instrumental to the progress of my work. I am also enormously grateful to all of the undergraduate help I have had, including Jennifer, Matt, Yexin, Emily, Albert, and Yeyi. Your countless hours of dedication to various aspects of the project, including

the parts nobody enjoys doing, like manual delineation, are very much appreciated. I feel like I got as much out of working with you all as I hope you did with me.

There are many others at JHU that helped me along the way, whom I am also thankful for. Thank you to both Laura and Debbie, who have been amazing in their administrative duties. I also want to thank Dr. Kang and Dr. Meyer for being on my thesis committee and providing helpful feedback on my work.

I have had the pleasure to work with a lot of great people in Dr. Prince's lab, all of whom I consider friends and I look forward to following your careers going forward. Roy, John, Min, Zhen, Chuyang, Fangxu, Nathanael, Amod, Murat, Gunnar, Jeff, Sanjukta, Can, Bhavna, David, and Blake, thanks for being such good people and making my time in the lab a special one.

Of course, without a strong foundation of life outside of the lab, I am not sure I would have made it as far as I did. I've met a lot of people during my time here, and made a lot of friendships that will last a lifetime. Thank you to Keith, Chris, John, Kevin, and Mike for taking me under your wing as a young graduate student and for ultimately being great friends. Dhruv, Tommy, and Yuriy, thanks for the good times while you were here. A big thank you goes out to everyone in the IACL, including John, Min, Aaron, Bhavna, and David, who helped drive activities away from the lab. And finally to David M. and the rest of my small group, thank you for all of your support and friendship, you have helped me to grow in so many ways.

Finally, I need to thank my family, who have been by my side through this entire journey despite never getting a satisfactory answer to the question “So when are you going to be done with school?”, to which I proudly have an answer now. To my parents, Anne and David, your unwavering support and love has encouraged me at every step of the way. To my brother Scott, his wife, Rebecca, and their three amazing children, Audrey, Landon, and Charlotte, I could always count on my visits to your house to be the perfect getaway from my busy academic life. Thank you all so much.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The field of ophthalmic imaging has evolved significantly over the last two centuries. With the development of the ophthalmoscope by Helmholtz in 1851, doctors first gained the ability to see inside of the eye [1]. This device gave clinicians an illuminated look at the back of the eye by directing light into the eye using precisely positioned mirrors. Despite this discovery, early methods for capturing these retinal images were limited, with only low quality photographs possible until the technology began to mature in the mid 20th century [2]. At this time, the emergence of fundus photography enabled highly detailed images of the *fundus* of the eye, or the interior region of the back of the eye. Fundus imaging provides a two-dimensional (2D) picture of the surface of the *retina*, the thin layer of tissue lining the interior of the eye. While many abnormalities can be detected in fundus images through changes in color and texture, the projective nature of photography yields only limited information. Later, the development of stereo fundus imaging incorporated depth to the images, informing the clinician about changes to the shape of the retina [3]. We could not, however, observe the full three-dimensional (3D) structure of the retina *in vivo* until the invention of optical coherence tomography (OCT) in the early 1990s [4]. OCT provides an unprecedented look into the eye, producing cross-sectional images of the retina, which can be used to see how the retinal structure changes with disease. Since the thickness of the retina is on the order of 200–300 micrometers (µm), it cannot be imaged at the same level of detail using other modalities like ultrasound and magnetic resonance imaging (MRI). As a result, the advent of OCT revolutionized the field of ophthalmology similar to how the ophthalmoscope did long ago.

The development of OCT has led to a need for practical and efficient ways to analyze the data. While measurements, like the thickness of retina, can be made at single points by hand, it is not feasible to manually measure the entire imaged area, which can be 12 mm or more with modern wide-field imaging (a single pixel is on order of 0.01 mm). The OCT scanner manufacturers, being the first point-of-contact to the data, provide their own software to run automated algorithms for extracting measurements of the retina. However, the variety of measurements available are often limited, leading researchers to seek other methods to do a more advanced analysis. Thus, there is a need to develop improved methods, providing more comprehensive, accurate, and efficient measurements when analyzing retinal OCT data.

In this thesis, we describe several methods developed to analyze retinal OCT data. In the remainder of this chapter, we present a preliminary introduction to the retina, OCT, and the current state-of-the-art in automated processing of OCT data. Additionally, since we frequently refer to data acquired on patients having multiple sclerosis (MS) throughout the thesis, we will briefly overview this disease and its effect on the eye. Finally, we conclude the chapter with the specific contributions of the thesis as well as an outline of the remainder of the thesis.

1.1 The human retina

The retina is a critical component in the human visual system, responsible for converting light entering the eye into an electrical signal interpretable by the brain. Before light is processed by the retina, it passes through many different structures within the eye [5], as shown in Fig. 1.1. Light first enters the eye through the *cornea*, where it is focused to the retina. It then passes through the *pupil*, which acts as a camera aperture to restrict the amount of light entering the eye. The size of the pupil is controlled by the *iris*, expanding or contracting to allow more or less light in. Next, the light travels through the *lens* as a second stage of focusing. Through accomodation, the lens changes its shape to focus light from objects at different distances. Within the large cavity of the eye is the *vitreous* (sometimes called the vitreous humor). This transparent, water-based gel fills the eye, helping to maintain the structural stability of the retina while acting as a medium for light to travel through.

The retina is the innermost of three layers that make up the outer tissue structure of the eye. These layers, from the exterior to the interior of the eye, are the sclera, the choroid, and the retina [5]. The region of the retina directly opposite the lens, along the optical axis of the eye, is called the *macula*, which is about 5 mm in diameter. At the center of the macula is the *fovea*, covering a diameter of about 1.5 mm. The fovea is perhaps the

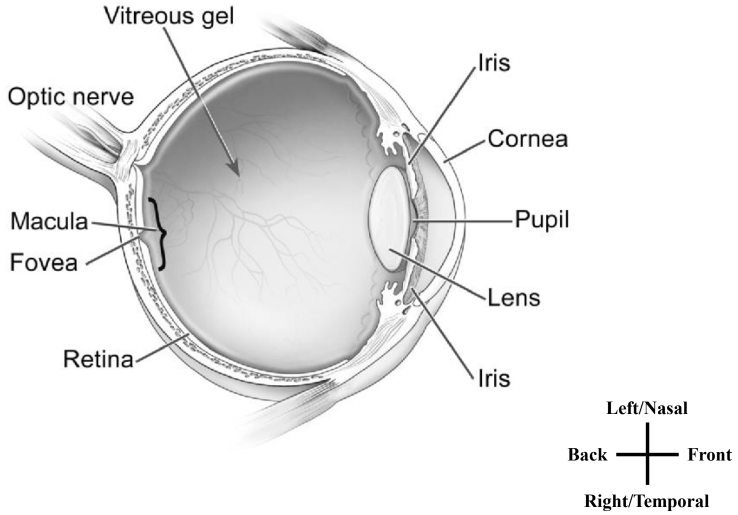


Figure 1.1: A schematic drawing showing a cross-sectional image of the human eye and its labeled components. Assuming a top down cross section, this image is of a right eye with the nasal side is to the top of the image. Image adapted from the National Eye Institute, National Institutes of Health (NEI/NIH).

most important area for vision, responsible for our central vision where visual acuity is highest. About 4 mm in the nasal direction from the fovea is the *optic nerve head* (ONH), or optic disc. The ONH is where the optic nerve, emanating from brain, connects to the eye, with its axons spreading throughout the retinal surface.

Structurally, the retina consists of three layers of neurons [6]. Within these neuronal layers are retinal layers representing different partitions of the neurons. This layered structure is shown in Fig. 1.2, with the neuronal cell composition shown on the right (with cell size exaggerated for illustration), and the corresponding retinal layers on the left (appearing more to scale with how the retina is organized). Specifically, we see the nuclei of the neuronal layers in the ganglion cell layer (GCL), the inner nuclear layer (INL), and the outer nuclear layer (ONL). Between the nuclei are the inner and outer plexiform layers (INL and ONL),

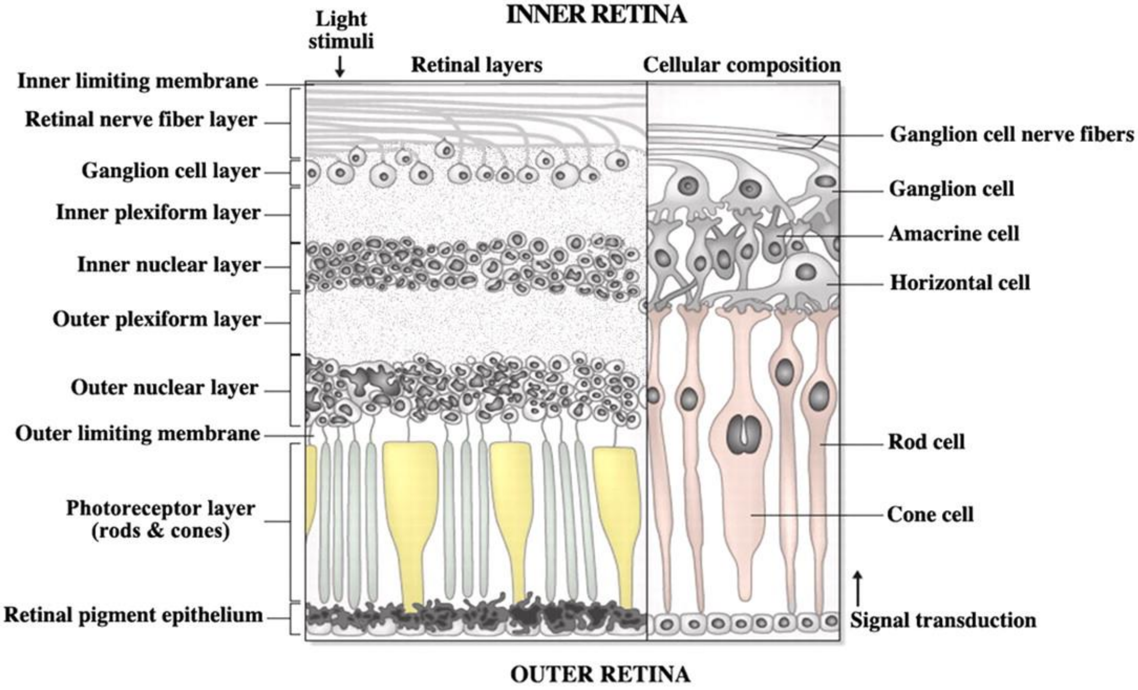


Figure 1.2: An illustration of the cellular layers of the retina. Light entering the eye goes through the retina from top to bottom, with the transduced signal traversing back through the retina from bottom to top. (Figure reproduced from [7] by permission of Oxford University Press.)

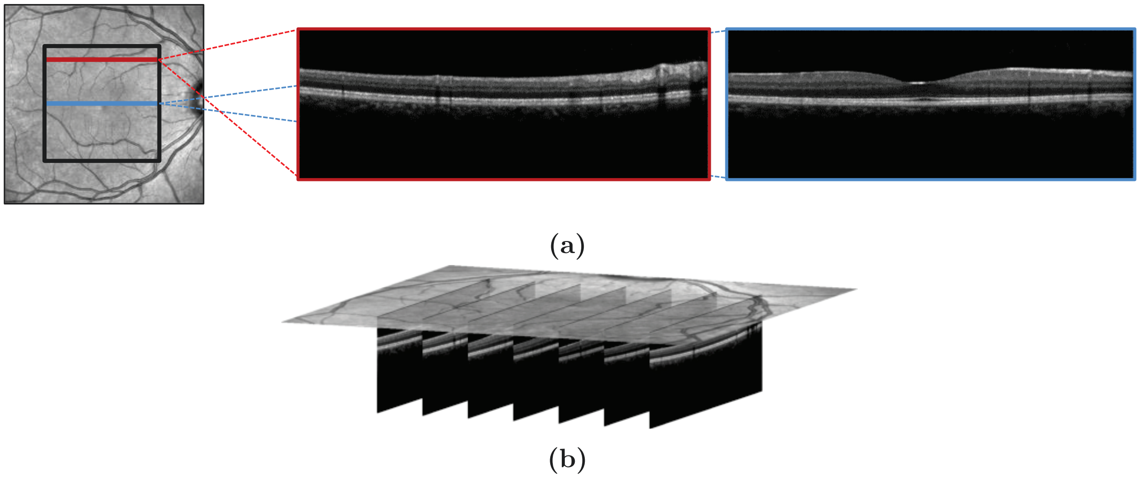
containing the interfaces between the neurons. At the surface of the retina is the retinal nerve fiber layer (RNFL) where the axons of the ganglion cell neurons continue along the surface of the retina to the brain. In the outer retina are the rods and cones, or the photoreceptor layer, as well as the retinal pigment epithelium (RPE). The thickness of these layers depends on the density of the cells, which varies across the retina. The ONL, for example, is about 90 µm at the fovea, and decreases to about 45 µm away from the fovea. The RNFL can have a thickness of over 100 µm near the ONH, where all of the axons exit the retina, and only about 20–30 µm temporally. Many of the remaining layers are typically in the 20–50 µm range.

Functionally, the retina can be divided into two parts: a sensory part responsible for phototransduction of light, done by the rods and cones in the outer retina, and a connective neural part, consisting of the neurons that carry the transduced signal from the retina to the brain [6]. Light coming into the eye travels through inner retinal layers to the photoreceptor layers. The converted electrical signal then travels from the outer retina to the inner retina and ultimately to the brain via the RNFL.

1.2 Optical coherence tomography

In recent years, OCT has become a widely used modality for imaging the retina. It produces high-resolution cross-sectional images of the retina, enabling a clear visualization of the retinal structure and its layers. Additionally, OCT's ease of use and rapid acquisition time, along with being safe to use, with minimal patient discomfort, have helped OCT to become a popular tool in ophthalmology. An example pair of OCT images are shown in Fig. 1.3(a) along with a fundus image indicating the location from where the OCT images were acquired. The fundus image provides a picture of the retinal surface at the back of the eye. In Fig. 1.3(b), we show the cross-sectional nature of OCT and indicate how a 3D volume is acquired as a sequence of B-scan images.

OCT works by measuring the optical reflectivity of tissues; this is analogous to ultrasound imaging where acoustic reflectivity is measured. Specifically, OCT uses low-coherence interferometry to very precisely measure the depth at which an input light source is reflected back to the scanner [8]. This light source has a near-infrared wavelength of around 850 nm. Imaging is done one *A-scan*, or column in the image, at a time. To construct a 2D OCT image (called a *B-scan* image), the light beam is mechanically moved across the



(a)

(b)

Figure 1.3: (a) The fundus image on the left shows the square area where an OCT scan is acquired. The colored lines indicate the position of the OCT images on the right. (b) A 3D OCT volume represents a series of cross-sectional slices through the retina.

retina, one A-scan at a time. A full 3D volume can be acquired in only a few seconds by imaging over an entire region of interest.

Many of the layers shown in Fig. 1.2 are visible in OCT, and a labeled B-scan image is provided in Fig. 1.4. While most labels are shared between the two figures, the IS and OS in Fig. 1.4 refer to the inner and outer segments, which create the photoreceptor layers shown in Fig. 1.2. The ILM, ELM, and BrM refer to the inner limiting, external limiting, and Bruch's membranes, respectively, with the ILM and BrM representing the inner and outer boundaries of the retina. After combining the GCL and IPL, which are not often visibly separable, we have eight total layers that can be seen clearly in an OCT image. Also indicated in Fig. 1.4 is the fovea, which appears as a depression of the retina in a healthy eye, where several layers merge together.

A typical OCT scan of the macula covers an area of 6×6 mm and a depth of 2 mm, which are the dimensions as shown in Fig. 1.3(a). The optical resolution of OCT in the

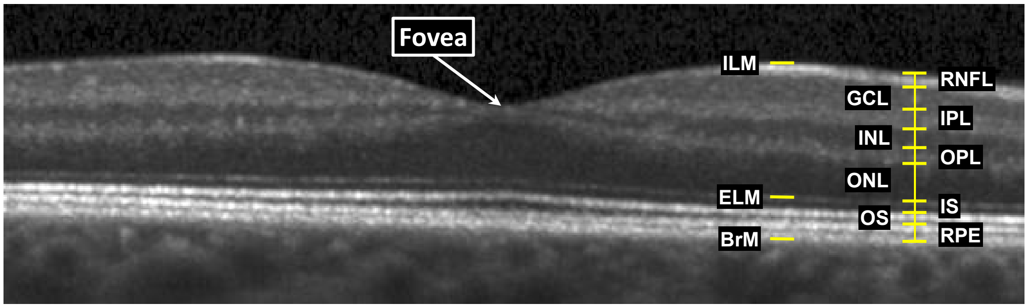


Figure 1.4: A B-scan image with demarcations indicating the layers and their boundaries.

depth direction is around 5–10 μm, with a transverse resolution of about 15 μm. Scans are generally sampled at a higher digital resolution, for example, data from a Zeiss Cirrus scanner has dimensions of 1024 × 512 × 128 pixels in the depth, transverse, and inter-B-scan directions resulting of pixels dimensions of 2, 12, and 47 μm, respectively. Thus, the issue of anisotropy should be addressed when developing algorithms to process OCT data.

One of the primary difficulties in working with OCT data is that it contains speckle noise [9]. Speckle is produced from the interference pattern of reflected light as it is scattered by the tissue. By averaging many images acquired at the same location, the speckle noise is reduced. Example OCT images acquired without and with averaging are shown in Fig. 1.5.

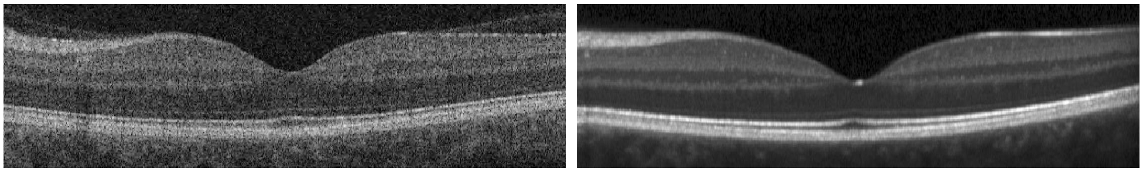


Figure 1.5: OCT images acquired (left) without and (right) with multi-frame averaging. Note that these images were acquired from different subjects using different scanners.

1.3 Multiple sclerosis and the eye

MS is a disease of the central nervous system affecting an estimated 2.3 million people around the world [10]. It primarily affects white matter areas in the brain, disrupting the protective myelin sheath surrounding the axons of the neurons [11]. As a result, a variety of symptoms are possible in MS, depending on the affected region. These include fatigue, walking difficulties, blindness, and paralysis [10]. The causes of the disease are not well understood, nor is the time course of the disease, with symptoms appearing and disappearing in a cyclic manner.

Up to 50% of MS patients also exhibit vision problems, which sometimes manifests as inflammatory attacks of the optic nerve called optic neuritis [12]. Many studies have therefore looked at the relationship between the eye and MS. Since OCT provides a detailed image of the retinal structure, it has increasingly been used to identify how the eye changes over the course of disease [13]. Initial work focused on using OCT to explore the thickness of the retina around the ONH, where the thickness of the RNFL was found to be thinner in MS [14]. More recent studies have focused on the macular region of the retina, where thinning was also shown in the GCL [15].

Despite the work done in recent years to analyze and identify the changes in the retina over the course of MS, a lot is still unknown about the disease and its impact on the eye. For example, different phenotypes of patients appear despite having the same sub-type of the disease. Saidha et al. [16] showed that some MS patients have no thinning, some have thinning in the GCL but not in other nuclear layers, while a third type have thinning in the nuclear layers but not in the GCL. While these findings are not well understood, they point to interesting areas of research going forward. As another example, a small proportion of MS patients develop small fluid-filled cystic areas, called *pseudocysts*, in the INL. This phenomenon has been termed microcystic macular edema (MME) in the literature [17]. Very little is known about why MME occurs or what it means for disease progression and prognosis. As a final example, the disease's effect on the outer layers of the retina is still largely unknown. It is hypothesized that there should be changes in these layers, but there is little evidence as of yet to support this idea. One possibility is that finding these changes will require a sub-micrometer level of accuracy.

1.4 Automated processing of OCT data

1.4.1 Layer segmentation

One measurement frequently made on OCT data is to compute the thickness of different layers, which requires delineation of the layer boundaries. An example of a fully delineated OCT image is shown in Fig. 1.6. Since thickness changes in the retina result from structural changes, these measurements help to inform clinicians about the health of the eye. Additionally, looking at thickness changes over time in the same patient allows us to see disease progression. Since manual methods of measuring layer thickness are not feasible, the development of automated methods to measure thicknesses is critical.

In the literature, automated segmentation of the layers in OCT has been explored extensively [18–30]. Previous work on retinal layer segmentation has used a variety of methods to explore retinal OCT data including analysis of intensity variation and adaptive thresholding [19], intensity-based Markov boundary models [18], and texture and shape

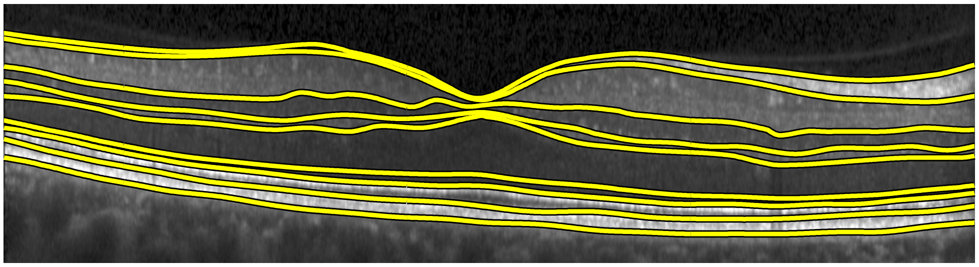


Figure 1.6: An OCT image overlaid with manually delineated layer boundaries.

analysis [23]. More recent works have had at their core more complex imaging segmentation tools. In particular, graph-theoretic approaches have been increasingly used, with the work of Chiu et al. [25] using a dynamic programming version of the classic shortest path algorithm, with similarities to Yang et al. [24]. The simultaneous multi-surface segmentation by a minimum cut graph approach of Garvin et al. [21, 22], with its extensions incorporating texture based features [28] and soft constraints [29, 30], are more examples of the many graph-based segmentation methods reported in the literature. Other more general methods have used active contour segmentation models [20, 26]. Concurrent to these works has been the development of machine learning approaches, including the use of support vector machine (SVM) [27] and random forest [28] classifiers, with features based on image intensities and gradients.

1.4.2 Fluid segmentation

Another measure important to clinicians is the quantification of fluid appearing in the retina, which occurs in diseases like age-related macular degeneration, diabetic retinopathy, detached retina, and macular edema [31]. This fluid may appear within specific layers or sub-retinally, and quantifying both the amount of fluid and its location are critical

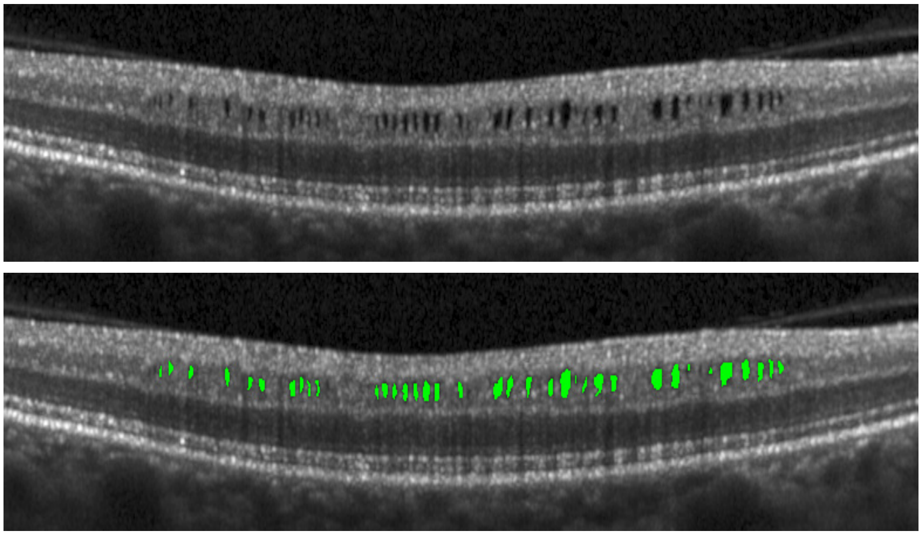


Figure 1.7: An OCT image of a subject with MME with a manual segmentation of the pseudocysts overlaid in the bottom image.

for measuring disease severity and progression. An example segmentation of an MS patient with MME, with fluid-filled pseudocysts appearing in the INL, is shown in Fig. 1.7.

A variety of methods for the segmentation of cystic changes in the retina have been reported. In particular, methods have been developed for segmentation in diabetic macular edema, retinal detachment, and age-related macular degeneration [32–37]. These methods use a variety of techniques to segment the cystoid areas. Two semi-automatic algorithms were described, one which uses a deformable model [32] and the other using a split Bregman segmentation algorithm to generate candidate fluid spaces [33]. Several fully automatic algorithms have been presented; one using a k-nearest neighbor (k-NN) classification plus a graph cut segmentation [34], and another using using a dynamic programming algorithm on polar transformed data of pilot estimates of the lesions [35]. In [36], a bilateral filter is used followed by thresholding, while in [37], k-means clustering and k-NN are used. No methods have previously been developed for the segmentation of MME, making it an unexplored area of research.

1.4.3 Registration

While the field of image registration is extremely active for imaging modalities like MRI and CT [38], it is not a well studied problem for use with OCT data. Registration, or alignment of data, can be carried out either on scans from the same subject (intra-subject), or between scans from different subjects (inter-subject), with the goal of aligning similar structures in each scan. Intra-subject registration is generally used to align longitudinal data, or data acquired from the same subject at different points in time. After alignment, changes in retinal thickness can be observed and potentially attributed to disease progression. Inter-subject registration is useful to align and compare structures across different populations.

For alignment of longitudinal data, the registration problem is often reduced to the 2D problem of aligning blood vessel patterns in the fundus view of the data. These vessels can be thought of as fixed landmarks, unchanged over time. For the specific purpose of longitudinal analysis, there have only been a small number of papers on the alignment of OCT data [39–41]. The two works of Niemeijer et al. [39,40] provide longitudinal registration as a motivating example, but do not include experiments to show improved longitudinal stability or accuracy. In Wu et al. [41], blood vessel points are extracted from an OCT fundus projection image and registered between scans using the coherent point drift algorithm [42]. This work was later used to evalute the change in thickness over time in patients with macular edema [43].

Registration has also been used to enable averaging of repeated OCT scans, improving the signal-to-noise ratio of the images [44,45], as well as for mosaicing to create wide-field images [39, 46, 47]. Very little work has been done to look at registration of OCT data for the inter-subject problem. Such a problem requires a deformable registration model since translation and rotation are insufficient. Only a few deformable registration methods have been developed for OCT data, including Chen et al. [48] which uses radial basis functions to register individual A-scans together, and Zheng et al. [49], incorporating the SyN registration method into a layer segmentation framework. In Lee et al. [50], extracted surfaces were deformably registered between subjects.

1.5 Contributions

In this thesis, we have developed four main contributions to the area of automated analysis of macular OCT data:

1. **A robust framework for layer segmentation utilizing random forest classification**

   Our first contribution is the development of a robust machine learning framework for layer segmentation of macular OCT data. Our framework is centered around the use of a random forest (RF) classifier [51]. By using manually labeled training data, along with a set of carefully handcrafted features, the algorithm learns individual layer boundary positions, with accurate performance shown for healthy and MS subjects, as well as in patients with retinitis pigmentosa, where the photoreceptor layers are disrupted. The RF output is converted into a final segmentation using either a simple boundary tracking method, or a more complex graph-based structure, with the second proving to be more accurate. Previous methods using a similar graph structure required careful tuning of the boundary cost function for different datasets/applications, while the RF proves to be more flexible, learning the profile of each layer from training data.

These contributions have been previously published in [52], [53], and [54].

2. **Methods for registration and segmentation of longitudinal OCT data**

   While methods for producing longitudinally consistent results have been developed for data from other modalities, there is little work on this for OCT data. We have developed two tools to improve the analysis of longitudinal data—one for registration and one for segmentation. To register two scans from the same subject, we use segmented blood vessel points to align the data, under the assumption that blood vessel patterns do not change over time. By registering data in this way, more accurate thickness measurements can be made, potentially leading to more accurate characterization of disease progression. Additionally, we have developed a method for simultaneously segmenting longitudinal data by extending our previously developed machine learning, graph-based layer segmentation algorithm. Registration of the data is done prior to segmentation, which ensures that similar structures are segmented in a consistent manner between images.

   These contributions have been previously published in [55] and [56].

3. **Segmentation of microcystic macular edema**

   In this contribution, a method for the segmentation of MME is proposed. MME is an ill-understood development in MS and as such, no automated methods have previously been developed to analyze the occurrence of MME in OCT data. Additionally, the small size of the MME pseudocysts makes the problem more difficult than might be expected. Previous work on finding fluid regions in OCT images in other diseases was tuned to finding much larger cysts than those in MME, and are thus inappropriate for finding the smaller pseudocysts. To target the specific characteristics of MME in OCT images, we again utilize a machine learning approach, using manually labeled data and an RF classifier to learn the variability in the size and shape of the pseudocysts in the retina. A separate set of features to those designed for the layer segmentation approach was used, where each feature was tuned specifically to capture the appearance characteristics of MME.

These contributions were jointly developed between the thesis author and a summer undergraduate student, Emily Swingle, who produced a conference paper from this work [57]. The work was subsequently extended into journal form with algorithm improvements and extensive validation by the thesis author in [58].

4. **Development of a macular flatspace for improved automated processing of OCT data**

   Finally, a computational domain for processing macular data, denoted macular flatspace (MFS), was developed to place images in a common coordinate space for further processing. By placing images in MFS, we are standardizing the appearance of the data, reducing the variability seen from eyes having different shapes and sizes. Additionally, MFS transforms the appearance of each layer to be flat, thereby allowing automated methods to perform better since this flat structure allows us to simplify any features and constraints used. We applied MFS as a pre-processing step for two applications: intensity normalization and layer segmentation. Intensity normalization is important for many automated methods since the intensity of OCT data can vary significantly between subjects and scanners. To correct the intensities found in each image, we adapted the N3 method [59], originally used for inhomogeneity correction in MRI data, by running the algorithm on data after conversion to MFS. Without this transformation, N3 is unable to adapt to the retinal structure, producing inferior results. As a second application, we show that by using MFS, the graph structure used in our previous layer segmentation work can be simplified, leading to improved efficiency and accuracy.

These contributions have been published previously in [60], [61], and [62].

1.6 Thesis organization

The organization of the thesis follows the order of the contributions described in Section 1.5, with one chapter dedicated to each contribution. Prior to this, Chapter 2 provides background material detailing two foundational methods necessary for understanding several of the methods developed in this thesis. Chapters 3 to 6 detail each of the four contributions, providing both methods and experimental results in each chapter. Finally, we finish in Chapter 7 with conclusions and future work.

Chapter 2

Background

In this chapter, we provide a technical background on two methods used throughout the work developed in this thesis. First, we review the random forest (RF) classifier originally described by Breiman [51]. The objective in classification is to estimate a label for a given set of data based on the values of different features computed on the data. These labels are predefined and learned through a training process using manually labeled data. Ultimately, we will be interested in identifying different components of the retina, like the layers, where we estimate a label for every pixel in an OCT image. Second, we review a graph-theoretical surface segmentation method, which was originally described by Li et al. [63] and later extended by others for use in OCT [21,22,29,30]. By modeling an image or volume as a graph with a specific structure, different surfaces (e.g. retinal layer surfaces) can be segmented by finding a cut in the graph minimizing a cost function. Such a minimization can be done exactly, producing a globally optimal solution.

2.1 Random forest classifier

Many algorithms have been developed for supervised classification [64], including support vector machines [65], RFs [51], boosting [66], and more recently, deep networks [67]. RF is one of the more appealing methods for many reasons including that it has only a small number of parameters to tune, it can accurately learn complex nonlinear relationships, its performance is comparable or better than other state-of-the-art classifiers, it can handle multiclass problems, it is computationally efficient, and it generates a probability for each label. This last benefit is especially important since the RF output can act as a soft classification, which is further refined to produce a better segmentation, for example, using the graph framework later described in Sec. 2.2.

2.1.1 Decision trees

At the core of the RF algorithm is the *decision tree*, which is a type of classifier in itself [68]. A decision tree divides a feature space into different regions through a recursive partitioning of the data. Figure 2.1 shows an example decision tree detailing its structure. Given an n dimensional feature vector $\mathbf{f} = [f_1, f_2, \ldots, f_n]$, we wish to classify the vector as belonging to a class $l \in \{1, 2, \ldots, k\}$. By starting at the top of the tree, a series of binary decisions are made depending on the values in $\mathbf{f}$. At each node in the tree (represented by circles), a single feature is examined and a decision is made to move along the left or right output of the node based on a simple threshold of that feature (e.g. if $f_i > t$, move left, and if $f_i \leq t$, move right).[1] The data then moves through the tree, using a different feature and

[1] Multiple features can be examined at each node, with a hyperplane fit to divide the data [69], but the added computational cost makes this method less common.

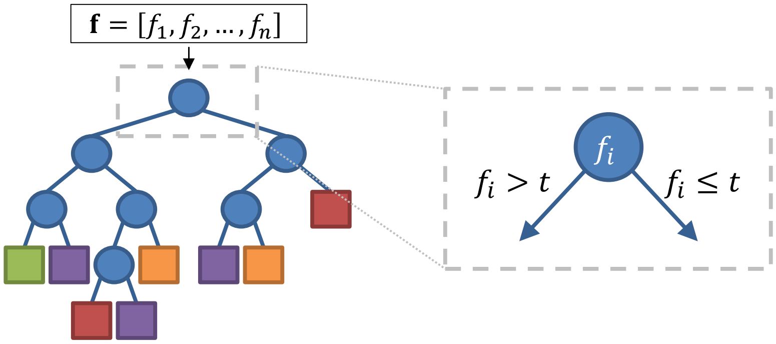


Figure 2.1: Diagram of a decision tree: a vector of feature values is input into the tree and labeled based on the label of the leaf node it ends up at.

threshold at each node, until the leaf node is reached (denoted by a square). Each leaf node has a class associated with it (represented by the different colors), and therefore, $\mathbf{f}$ is labeled based on the class of the leaf node it ends in.

When constructing a decision tree, we must decide both which feature to use at each node of the tree and a threshold value for that feature. To make these decisions, we use a set of training data, $D = \{(\mathbf{f}_j, l_j) : j \in 1, ..., N\}$, where N is the number of samples, $\mathbf{f}_j$ and l_j are the feature vector and label for sample j, respectively. Using this training data, for a given feature f_i, we choose a threshold by finding the value which maximizes the change in a specified impurity function, indicating how separated the classes are. There are many choices for the impurity function, with more common examples being Gini impurity [68] and information gain using entropy [70]. Gini impurity, used in the RF classifier, specifies the misclassification error rate when the labels of the data are randomly selected with a probability based on their frequency [71].

For a given split decision at a node, the dataset D is divided into two sets, D_l^t and D_r^t, along the left and right paths, respectively. The threshold t for splitting the data is chosen such that the impurity is maximally reduced, with the reduction defined as

$$r(D,t) = g(D) - \frac{N_l}{N} g(D_l^t) - \frac{N_r}{N} g(D_r^t), \tag{2.1}$$

where $g(D)$ is the impurity function, and N_l^t and N_r^t are the number of samples in the left and right split, respectively.

The other aspect of training a decision tree is choosing the feature used for thresholding at each node. Since the number of possible decision trees grows rapidly with an increasing number of features, it is computationally impractical to find a globally optimal tree with respect to some measure, like the misclassification error. Instead, trees are typically built up one layer at a time, choosing the feature at each node that maximizes $r(D,t)$ over all features. This type of greedy approach is most common [68, 70], but other methods exist, for example, using evolutionary learning to try to find a global optimum [72].

The final consideration for the creation of a decision tree is a stopping rule for growing the tree. One approach is to grow the tree fully, with leaf nodes having only samples from one class. Growing a tree in this way leads to overfitting of the data, meaning it models the training data and its noise accurately, but does not generalize well to unseen data. Pruning can be used to solve this problem by removing deeper nodes of the tree [68]. Other approaches for growing a decision tree include stopping when the number of samples in a node gets below a specified number (e.g. 10 samples), or when the tree reaches a specified depth (e.g. 5 levels) [73].

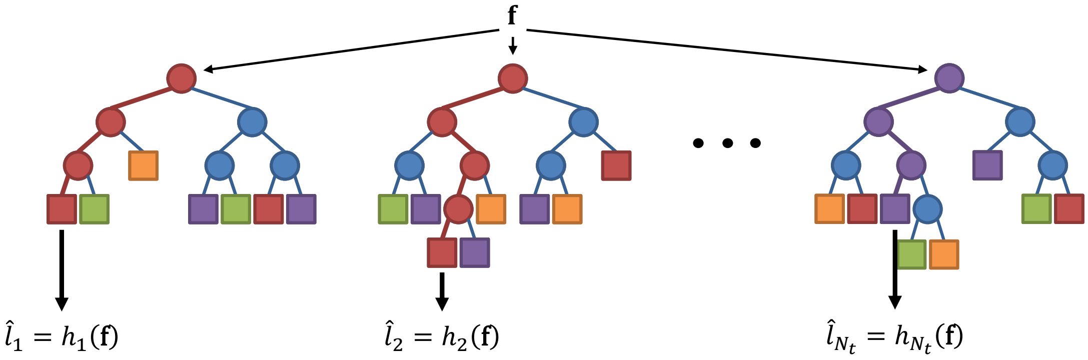

f
$\hat{l}_1 = h_1(\mathbf{f})$
$\hat{l}_2 = h_2(\mathbf{f})$
$\hat{l}_{N_t} = h_{N_t}(\mathbf{f})$

Figure 2.2: Example of a random forest, which consists of a set of N_t decision trees, each having a different structure and producing a separate output/label for the same input feature vector. A probability can be obtained by computing the frequency of each output label over all trees.

2.1.2 Random forests

The random forest method was created to solve the problem of overfitting when using only a single decision tree. It works by building an ensemble of N_t decision trees, $\{h_i(\mathbf{f}) : i \in 1, ..., N_t\}$, with each tree $h_i(\mathbf{f})$ acting to label the same feature vector $\mathbf{f}$ [51], see Fig. 2.2 for an example. Since using multiple decision trees is not useful when they all have the same structure, randomness is introduced to various aspects of building the trees, therefore discouraging correlation between them.

In Breiman's original work [51], he showed that the generalization error (GE), or the error in data unseen by the training process of the RF classifier, depends on two factors: the strength of the trees and the correlation between the trees. The strength is defined as the expected value of a margin function applied to each tree, where the margin function computes the difference between the probability of correctly classifying an input and the probability of classifying with the next most probable class. The correlation is computed as the average correlation of this margin function between each pair of trees [74]. Specifically, Breiman showed that the GE decreases (improves) as the strength of each tree increases, and it increases (worsens) as the correlation between trees increases. The intuition behind this second result is that combining the results from multiple classifiers cannot improve the accuracy of each classifier if the structure of each tree is highly correlated (thereby producing similar results).

Randomness is introduced into the decision tree training process in two ways. First, the training data is randomly sampled, *uniformly* and *with replacement*, to create a different training set for each tree. An average of 63% of the original data will be used in such a sampling scheme as $N \to \infty$, with the remaining samples being duplicates (see [71], p. 251 for a derivation). Such a procedure is called *bagging* or bootstrap aggregating. The second source of randomness is in limiting the number of features examined at each node of the tree. Specifically, for each node, only m randomly selected features out of the total of k features are included when looking to maximize the impurity criteria used for finding a splitting rule. As a result, the "best" feature will not be chosen at a given node if it is not randomly included. This randomness helps to reduce the correlation between trees since the greedy approach described previously will always choose the best feature for splitting.

Given a feature vector $\mathbf{f}$, every tree provides a label estimate from the set of all possible labels, $\hat{l}_i = h_i(\mathbf{f})$. By looking at the predicted label for each tree, a posterior probability for each class can be computed as $p(l = k|\mathbf{f}) = \frac{1}{N_t} \sum_{i=1}^{N_t} \mathbf{I}_k \left(\hat{l}_i\right)$, where $\mathbf{I}_k(\cdot)$ is an indicator function for class k [69]. In many classification applications, a final label estimate is taken as the label voted by the relative majority of the trees, $\hat{l} = \arg\max_k p(l = k|\mathbf{f})$.

When using RFs, there are only a few parameters which must be considered. These parameters include the number of trees to use, N_t, the number of features randomly selected when training each node, m, and the depth of each tree. Each of these parameters are application dependent, and should be tuned to maximize performance accordingly. The accuracy of the RF approaches a limit as the number of trees increases, which can be estimated by looking at the out-of-bag error, or the misclassification error of the samples not included in training each tree [51]. For the value of m, a recommended starting point is $m = \sqrt{k}$ [75], which provides a balance between using too few features producing a low accuracy, and too many features, which increases correlation between trees. For the depth of each tree, Breiman [51] recommends using fully grown trees, since averaging results over many trees counteracts the overfitting of any single tree. In practice, some authors have shown that fully grown RFs may still overfit [76]. Another practical reason for not using fully grown trees is to reduce both the memory footprint of trees trained with a large amount of data and the evaluation time to produce a result.

2.2 Graph theoretical surface segmentation

The use of graph theory to model and solve problems in image processing, including segmentation [77], denoising [78], and stereo matching [79], has been popular in recent years due to the development of efficient methods for solving these problems. In addition, these models are often rooted in probability theory—Markov random fields, for example—providing a strong theoretical basis for their use [78]. The idea of using graphical models for imaging data requires us to associate pixels or voxels with vertices in a graph, with edges used to connect the vertices to model the problem of interest.

One important development of using graph theory in computer vision is the graph cuts method, originally described by Greig et al. [78], and made popular by Boykov et al. in their widely cited papers [77, 80, 81]. The goal of graph cuts is to produce a labeling (segmentation) of an image by minimizing an energy function computed over the image domain. While the problem of labeling an image can be solved using a classifier like RF described in Sec. 2.1, graph cuts does not use training data to produce a segmentation. One important contribution from Greig's original work was to show that the graph cut energy minimization problem can be solved *exactly* when there are two labels (binary) by finding the minimum cost cut on an associated graph [78]. Through their formulation, the binary segmentation problem is equivalent to the problem of finding the maximum flow through a network (often referred to as min-cut/max-flow problems due to their equivalence), with algorithms for solving this type of problem going back several decades to the work of Ford and Fulkerson [82]. Boykov's work improved on prior work by providing both an efficient algorithm for solving the min-cut/max-flow problem [81], and an approximate method to solve the multilabel problem by optimizing for each label cyclically (α-expansion) [80].

While graph cuts can segment objects of arbitrary shape, related problems have been proposed by modifying the graph structure to restrict the geometry of the resulting segmentation, e.g. finding an ordered set of surfaces [63, 83], or enforcing a nested structure for adjacent labels [84]. The application of finding ordered surfaces is ideal for the problem of layer segmentation in retinal OCT since the retinal layers have this structure. Indeed, the work of Garvin et al. [21, 22] used this idea for retinal layer segmentation. The flexibility of the graph structure has lead to a lot of follow up work in different applications [28, 34, 47, 83, 85], as well as in adapting the graph structure for improved robustness [29, 30].

In the remainder of this section, we will cover an introduction to graph cut energy minimization problem, followed by a detailed description of solving the optimal surface segmentation problem in 2D, its extensions to 3D and multiple surfaces, and the different smoothness models that can be incorporated. Finally, we will discuss briefly about cost function design for the problem of OCT layer segmentation.

2.2.1 Graph cut segmentation

The goal of the graph cut segmentation problem is to label each pixel in an image as belonging to an object or background class [77,86]. Let $G = \langle V, E \rangle$ be a graph representation of an image I, consisting of a set of nodes or vertices V and a set of directed edges E connecting each vertex. Given a set of pixels $\mathcal{P}$ from I, every pixel $p \in \mathcal{P}$ corresponds to a vertex $v \in V$. Edges connecting vertices can be structured differently depending on the problem, but a common configuration is to connect a vertex with its 4-connected neighbors (in 2D). Also associated with every node and edge is a *non-negative* weight, or cost. For a node p, this weight represents the cost of labeling the associated pixel as either background or object, denoted as $R_p(0)$ and $R_p(1)$, respectively. Edge weights between nodes p and q are denoted $B_{p,q}$, where $B_{q,p}$ can either be the same for an undirected edge, or different for a directed edge. An optimal labeling A is one which minimizes the total cost [77]

$$E(A) = \sum_{p \in \mathcal{P}} R_p(A_p) + \lambda \sum_{p,q \in E} B_{p,q} \cdot \delta_{A_p \neq A_q}, \tag{2.2}$$

where $\delta_{A_p \neq A_q}$ is 1 if $A_p \neq A_q$ and 0 otherwise. The parameter λ balances between labeling based on pixel costs versus boundary costs. Often times this boundary cost will regulate the smoothness of the resulting segmentation by penalizing differences in the labels of adjacent pixels.

To solve the graph cut problem, the graph G is augmented with two additional nodes, called the source s and sink t. Fig. 2.3(a) shows an example graph constructed with these two added nodes. An edge is added from every node $v \in V$ to both t and s (unless the weight of the added edge is 0). The weight of an edge connecting v_p to t is set as $R_p(0)$ and the weight of an edge from s to v_p is set to $R_p(1)$. Note that in this new graph, all of the node weights R_p have been moved from the nodes themselves to these new edges connecting to s and t. In this configuration, finding a cut in the graph which minimizes the total cost of all cut edges produces an optimal labeling minimizing Eq. 2.2. Such a cut is shown in Fig. 2.3(b) with the resulting segmentation in Fig. 2.3(c). This type of cut is called an s-t cut and as mentioned previously, this problem is of the min-cut/max-flow form, which can be solved in polynomial time by variety of methods [81].

2.2.2 The 2D single boundary problem

The problem of finding a minimum cost boundary/curve in a 2D image or a surface in a 3D volume can be described by finding a single pixel along each column of an image, or a single voxel along a specified dimension of a volume, see Fig. 2.4(a). While this definition is specific, a more general formulation can be constructed by modifying the underlying structure of the data (e.g. by converting to polar/spherical coordinates to find closed surfaces) [63,83]. The minimum cost boundary problem is different than that of finding a minimum cost

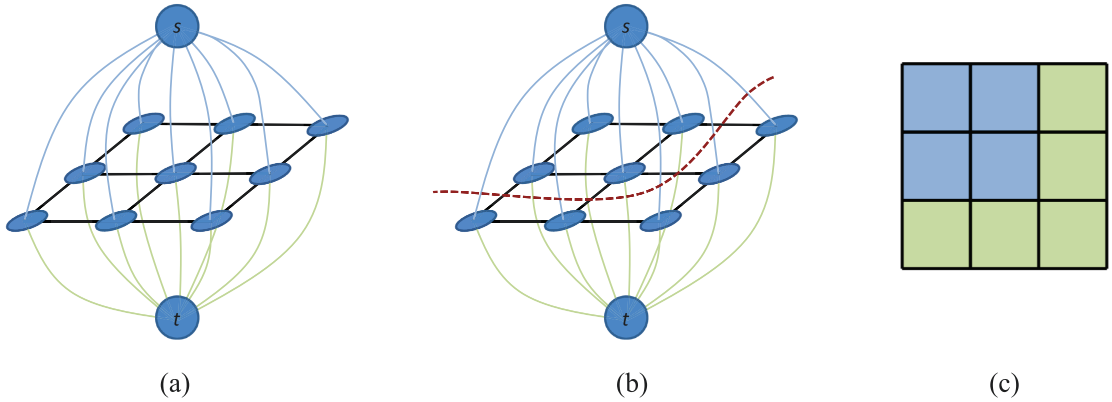


Figure 2.3: (a) An example graph constructed on a 3×3 image with (b) an example cut through the graph and (c) the resulting segmentation.

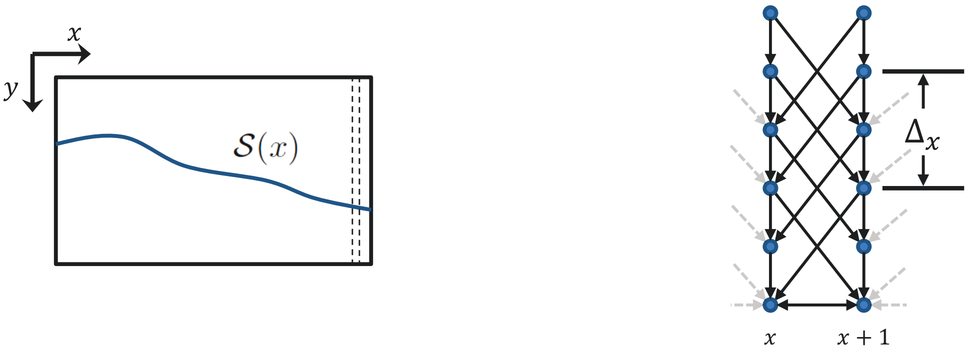
labeling by graph cut segmentation and requires a different graph construction [63]. The final result, however, is obtained in an identical way, by finding a minimum cost s-t cut in the resulting graph, thus producing a globally optimal solution.

We again consider a graph structure $G = \langle V, E \rangle$ associated with an image I as before, where I has a fixed size of $X \times Y$ pixels. Associated with every pixel is the cost, $c(x, y)$, of belonging to the boundary $y = \mathcal{S}(x)$. When constructing the graph, we assign a weight to every vertex as

$$w(x,y) = \begin{cases} c(x,y-1) - c(x,y) & \text{if } y \neq Y, \\ c(x,y) & \text{otherwise,} \end{cases} \tag{2.3}$$

which transforms the cost so it can be minimized using the graph structure described next.

Let $v(x, y)$ be the node at image coordinate (x, y), edges are added to the graph such that $v(x, y)$ is connected to $v(x, y + 1)$ for all $y < Y$. This produces a series of directed edges from a node to the node below it within a column of the data (an intra-column arc).



(a)

(b)

Figure 2.4: (a) A feasible boundary of the type which can be solved by a graph theoretical model. (b) An example of the graph structure constructed for every pair of adjacent columns of the image, like those shown as dashed lines in (a), to solve the minimum boundary problem. Since edge weights are set to infinity, a cut may not have any edges pointing out of the closed set formed below the cut. Thus, Δ_x is the limit to the amount of change in the resulting surface between columns.

By giving these edges a weight of infinity (meaning they cannot be cut), they act to enforce the single boundary pixel per column constraint. Edges are also added between adjacent columns of the image such that $v(x, y)$ connects to $v(x - 1, y + \Delta_x)$ and to $v(x + 1, y + \Delta_x)$. No connection is necessary if $y + \Delta_x > Y$, except between $v(x, Y)$ and $v(x \pm 1, Y)$. Again, by giving these edges a weight of infinity, these inter-column arcs cannot be cut, acting as a geometric constraint to limit the amount of change in the boundary between columns. An example of this graph structure is shown in Fig. 2.4(b).

For a given feasible boundary $y = \mathcal{S}(x)$, $x = 1, \ldots, X$, it can be shown that the set of vertices on or below the boundary form a *closed set* in G, meaning there are no edges directed out of the set [63]. Additionally, finding the minimum cost closed set corresponds to finding the minimum cost boundary. To ensure that the minimum cost closed set is not empty (which has zero cost), the total cost of the base set of vertices $\{v(x, Y) : x = 1, \ldots, X\}$, corresponding to the boundary $\mathcal{S}(x) = Y$ covering the pixels on the bottom row of the image, is offset so that it takes an arbitrarily negative value (therefore having a smaller cost than the empty set). Since all other closed sets contain this set, a nonempty set, or feasible boundary, is guaranteed to be found. To see why a minimum closure corresponds to a minimum cost boundary, we note that the graph costs were transformed in Eq. 2.3, allowing the total cost of all nodes in the closure to equal the total cost of the boundary (with a constant offset to ensure a nonempty set).

Finally, solving for the minimum closure in a graph is once again a known problem [87,88], which is done by finding the minimum s-t cut in the graph augmented with s and t nodes in the same way as was done in Sec. 2.2.1. This formulation allows min-cut/max-flow algorithms to be used and solved in an efficient manner [81].

2.2.3 Extension to 3D and multiple surfaces

The 2D problem described in Sec. 2.2.2 is easily extended for 3D data, where we now look to find a surface of the form $z = \mathcal{S}(x, y)$. The volume $I(x, y, z)$ now has size $X \times Y \times Z$ and a column of the data is formed by fixing the x and y dimensions and varying along z. To extend the method to 3D, we assume that we have a set of Y slices/images, each of size $X \times Z$ with an identical 2D graph structure described previously. The only change made is to add edges between adjacent slices. For a given node $v(x, y, z)$, we add two new edges, in addition to the two added previously, to $v(x, y \pm 1, z + \Delta_y)$. Again, no connection is necessary if $z + \Delta_y > Z$. Figure 2.5(a) shows an example of the 3D graph structure. (Note that the distinction between superscript l and u will be made apparent in the next section.) Given vertex weights in 3D, $w(x, y, z)$, there is no change to solving for

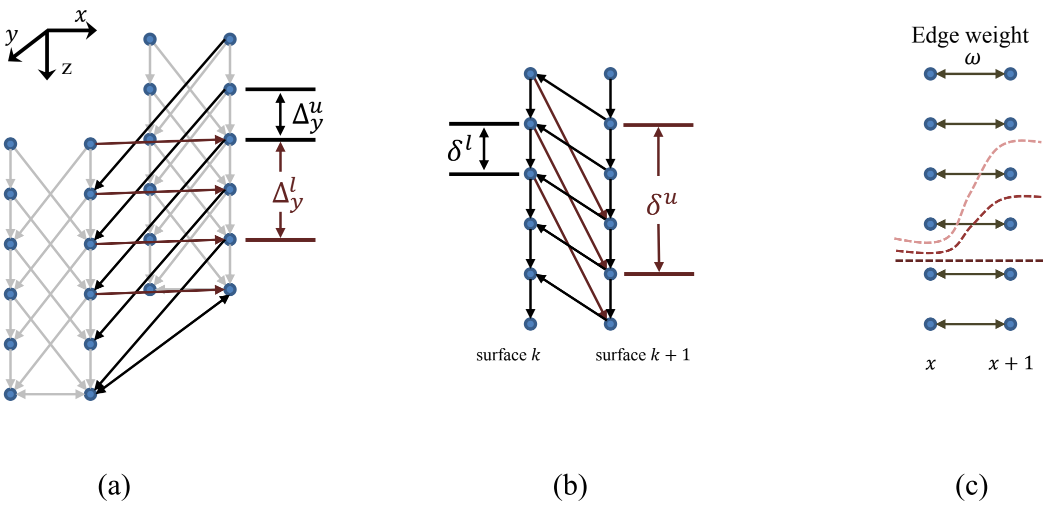


Figure 2.5: (a) Connections between 2D graphs to form a 3D graph. (b) Connections between graphs of adjacent surfaces enables constraints on the minimum and maximum distance between them. (c) By including finite, non-zero weighted edges, a smoothness penalty is incurred when edges are cut. Shown are three cuts, with penalties of 0, ω, and 2ω, depending on the number of edges cut.

the minimum surface as in 2D, since a minimum closure will again find the minimum cost surface.

One powerful aspect of the minimum surface graph-based formulation is that multiple surfaces can be found simultaneously with a fixed ordering (which is necessary for retinal data) [63]. The same graph framework is again used as before, with each surface having the same structure. Segmentation of multiple surfaces can be thought of as extending the 3D method into 4D in the same way we extended the 2D problem to 3D, where now we have weights $w_k(x, y, z)$, with $k \in \{1, \ldots, K\}$ indexing different surfaces. Additional edges are added to connect the 3D graphs for adjacent surfaces. Specifically, the vertex $v_k(x, y, z)$ is connected to both $v_{k-1}(x, y, z - \delta^l)$ and $v_{k+1}(x, y, z + \delta^u)$ for $1 < k < K$. Once again, edges have a weight of infinity, and therefore δ^l and δ^u can be interpreted as the minimum and maximum distance between surfaces in the final cut/segmentation. Such a connection is outlined in Fig. 2.5(b).

2.2.4 Smoothness models

In Sec. 2.2.2 and 2.2.3, the shape constraints Δ_x, Δ_y, δ^l, and δ^u were all globally fixed, constraining the surface shape and separation everywhere. While this model works for retinal layer segmentation in OCT, as done by Garvin et al. [21], some regions of the retina change more than in other regions, e.g., at the fovea. Therefore, in their follow up work, Garvin et al. [22] proposed incorporating spatially varying shape constraints into the graph structure.

Looking first at Δ_x, the previous constraint limited the absolute value of the change, or $-\Delta_x \leq \mathcal{S}(x+1,y) - \mathcal{S}(x,y) \leq \Delta_x$, for all x and y. In [22], the constraint was adapted to separately limit the maximum and minimum change, as well as to make it spatially varying. The modified constraints, $\Delta_x^l(x,y)$ and $\Delta_x^u(x,y)$, where superscript l and u represent the minimum and maximum change, respectively, now restrict the surfaces as $-\Delta_x^l(x,y) \leq \mathcal{S}(x+1,y) - \mathcal{S}(x,y) \leq \Delta_x^u(x,y)$. A similar modification was made to the constraints in the y direction, as well as to the thickness constraints, producing $\Delta_y^l(x,y)$, $\Delta_y^u(x,y)$, $\delta^l(x,y)$, and $\delta^u(x,y)$. Each of these constraints can additionally be different for every boundary (for Δ) and every layer (for δ). Since these constraints are applied spatially on the data, each scan must be roughly aligned before constructing the graph (i.e. the (x,y) coordinates should refer to approximately the same spatial region of the retina). Since the fovea is one of the few landmarks consistent between patient data in the macular region, it is generally used as the single point of alignment (thus producing a fovea centered coordinate system).

These constraints are learned from manually segmented data. One method to learn the constraints is to use the mean value of the population ± 2.6 standard deviations, which accounts for more than 99% of expected values [22, 29]. Modifications are likely to be necessary to apply the constraints for a diseased population or outlier data.

The smoothness constraints described above represent hard geometric constraints, setting upper limits on the allowed changes in the shape of the retina. Since these constraints allow for oscillations between adjacent pixels, an unsmooth surface can be produced. An additional constraint is needed to encourage adjacent pixels to have similar values, truly enforcing a smooth surface. Such a constraint was previously described, first by Song et al. [30] in their extension of Garvin's work [22], and alternatively by Dufour et al. [29] using a slightly different model.

To encourage smoothness in each surface, additional edges are added between vertices in the graph, each with a finite, non-zero weight. This is a similar idea to the $B_{p,q}$ weights from Sec. 2.2.1 for graph cut segmentation. If, for example, we added horizontal edges to the surface segmentation graph, a perfectly horizontal surface would incur no penalty since none of these added edges would be cut. The more that the surface deviates from being horizontal, the larger penalty the cut incurs will be. Specifically, the penalty at a given point would be equal to $\omega \left|\mathcal{S}(x) - \mathcal{S}(x+1)\right|$ in the 2D case, with ω being the edge weight. Figure 2.5(c) provides an example. If using a penalty function other than the absolute value is desired, Ishikawa's work [89] showed that any convex function can be used by adding multiple edges in specific way. This idea was used by Song et al. [30] to encourage smoothness in the shape of the retina by using a squared error penalty instead of absolute error. Edges can also be added beyond adjacent pixels to relax the penalty, since only integer valued changes can be produced by the graph [29]. The disadvantage of using these smoothness models is that they increase the size of the graph by adding edges, increasing the memory requirements of the model, which is already large for retinal OCT data.

2.2.5 Cost functions

The final component to the surface segmentation problem using the graph-theoretical approach is the design of the cost function to be minimized. Since the segmentation targets are boundaries, simply using gradient information can perform well [22, 29], with specific boundaries identifiable through bright-to-dark or dark-to-bright transitions [22]. Due to the noisy nature of OCT images, smoothing is necessary, which was previously done by anisotropic diffusion [90]. Since using only gradient features may not be robust to different types of data, machine learning approaches have also been used [28]. In the work of [28], a bank of filters at various scales and orientations was used by a classifier to find each layer. The RF classifier described in Section 2.1 is particularly useful for this problem, since it provides a probabilistic output which can be used in the cost function directly.

Chapter 3

Layer segmentation using random forest boundary classification

In this chapter, we present a framework for segmentation of retinal layers in macular OCT data. While the problem of layer segmentation is not new, with many methods described in the literature tackling this problem over the last 10 years, there are several motivations for developing a new method. First and foremost, little prior work has taken advantage of new developments in machine learning. Machine learning allows one to accurately and efficiently model the intensity, shape, and texture of imaging data, leading to robust estimation of where various structures of interest are within an image. Such an approach also allows us to satisfy other motivations, which include making the method robust to data acquired from different scanners, having different image quality, and with different pathologies. While other methods may require various adaptations to handle these differences, machine learning enables us to use a consistent framework by using a different set of training data.

This chapter is divided into two main parts. First, in Sec. 3.1, we describe our method and provide results on both healthy and MS subjects. Second, in Sec. 3.2, we detail the adaptation of our method for handling data from subjects with retinitis pigmentosa, a disease in which the outer retinal layers are severely degraded.

3.1 Layer segmentation using random forests—methods and validation

Our layer segmentation algorithm is centered around the use of an RF classifier [51] to find the retinal boundaries. The layers we are interested in finding are detailed in Sec. 1.2—with nine boundaries for eight layers. As described previously, one advantage of the RF approach is in its ability to generate probability maps for the boundaries, providing a "soft" classification of their positions. After computing these these probability maps, they are input into a second stage of our algorithm, where the probabilities are refined into a final segmentation, producing contours separating the retinal layers in each OCT image. Two approaches are explored for generating the final boundaries from the RF outputs: a simple boundary tracking approach and the graph-theoretic approach described in Sec. 2.2.

Similar machine learning based methods of layer segmentation have been previously developed, with Vermeer et al. [27] using an SVM classifier to label pixels within each layer. While SVM does not produce probabilities (like RF), it can produce continuous valued scores for each label, but these are not utilized in [27]. SVM is also an inherently binary classifier, and use for multiple classes requires a separate classifier to be trained for each label. One final difference to the work of [27] is in the refinement of the SVM output. To produce a final segmentation from the SVM output, a level set approach was used, which is known to be inefficient and produce suboptimal results.

Concurrent to our own work in using an RF classifier, Antony et al. [28] also used RF classification to aid in segmentation of OCT data, similarly followed by refinement using a graph-based approach [22]. One key difference with our own work is in the features used by the methods. Our method includes contextual features incorporating spatial information, while [28] includes only intensity-based features. Another important difference is that we classify boundaries, while this work classifies layers, which tends to be a more difficult problem due to the homogeneity within layers.

The rest of the section is organized as follows. Section 3.1.1 explains our method thoroughly, including preprocessing steps. Section 3.1.2 describes the experiments and results, including a comparison to a human rater on 35 data sets, and Section 3.1.3 has a discussion of the results, including their relation to other work and potential future impact.

3.1.1 Methods

Our algorithm is depicted in Fig. 3.1 and can be described in three steps. The first step consists of preprocessing the data using intensity and spatial normalization. In the second step, a set of 27 features is calculated on the preprocessed data and input into a trained RF classifier to produce boundary probabilities at every pixel. The classifier is trained from ground truth labels created by a human rater. In the last step, the final retina segmentation is generated from the boundary probabilities using a boundary refinement algorithm. We explore the use of two such algorithms, a simple boundary tracking method

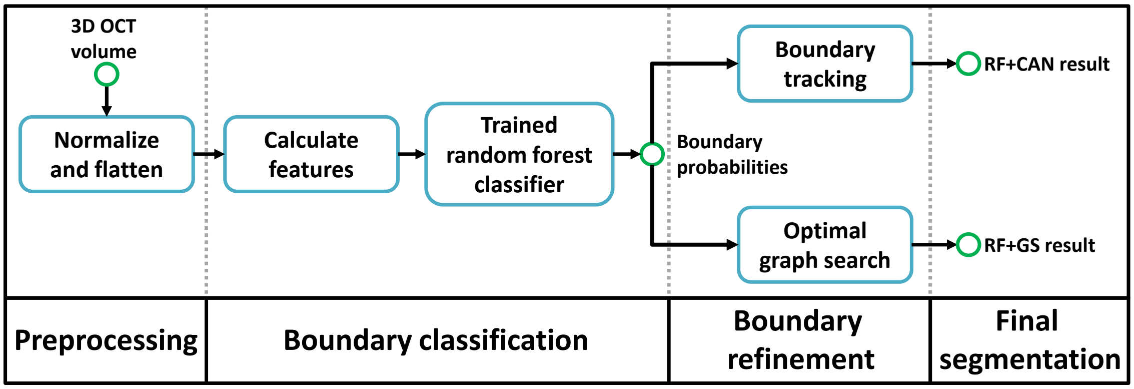


Figure 3.1: Flowchart of our algorithm. The RF+CAN result refers to the segmentation using the random forest (RF) boundary classification output with a Canny-inspired boundary tracking algorithm, while RF+GS refers to the result of using the RF output with an optimal graph search (GS) algorithm.

and an optimal graph search method [63].

Preprocessing

As with many segmentation algorithms, several preprocessing steps are used to transform the data to reduce variability between scans; this involves intensity normalization and a simple spatial alignment of the data called retinal boundary flattening.

Intensity normalization For any segmentation algorithm, it is important that the intensity ranges of the data are consistent. That is, the intensity values observed in a particular tissue type should be approximately the same within an image and across populations of images. Such consistency allows for better training in machine learning paradigms using features such as intensity and gradient profiles for each layer and boundary. The images used in our experiments showed considerable inconsistency, with two B-scans in the same volume often having very different intensity ranges, as exemplified in Fig. 3.2(a),

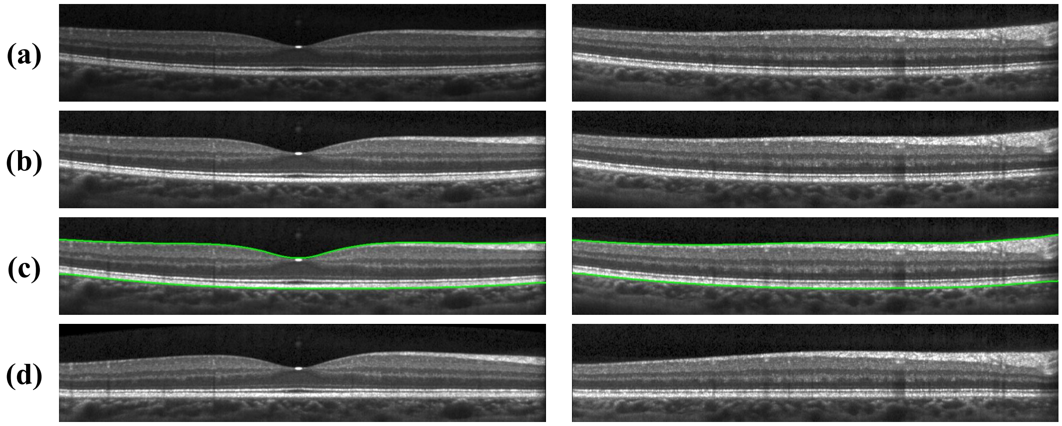


Figure 3.2: Row-wise: Shows two B-scans from within the same volume (a) with the original intensities, (b) after intensity normalization, (c) with the detected retinal boundaries, and (d) after flattening.

where the left and right images are different B-scans from the same volume. Two possible causes of these differences are (1) the automatic intensity rescaling performed by the scanner being adversely affected by high intensity reflection artifacts and (2) the automatic real-time averaging performed by the scanner, meaning one B-scan could undergo more averaging, causing differences in its dynamic range. These issues are scanner dependent and may not affect other scanners in the same way as in our experiments.

To address the intensity inconsistency issue, we carry out a contrast rescaling on each B-scan. Specifically, intensity values in the range $[0, I_m]$ are linearly rescaled to $[0, 1]$ while intensities larger than I_m are set to unity. The value I_m is interpreted as a robust maximum of the data, which is found by first median filtering each individual A-scan within the same B-scan using a kernel size of 15 pixels (58 µm). Then, I_m is set to the value that is 5% larger than the maximum intensity of the entire median-filtered image. This rescaling removes hyperintense reflections found at the surface of the retina while maintaining the overall intensity values in the B-scan. A result of this normalization step is shown in Fig 3.2(b).

Retinal boundary detection and flattening The second step in preprocessing is to estimate the retinal boundaries and flatten the image to the bottom boundary of the retina. This serves to give more meaning to the spatial coordinates of pixels for use in the RF classifier, to help to constrain the search area for the final segmentation, and to reduce the algorithm sensitivity to retinal curvature and orientation. The top and bottom boundaries of the retina are defined by the ILM and the BrM, respectively. Flattening is a common preprocessing step performed by many retina segmentation algorithms [22, 25, 91] and refers to translating all of the A-scans in each B-scan such that a chosen boundary in the image (the BrM, in this case) is flat. We note that these initial boundary estimates are later improved by our segmentation algorithm.

To find the top and bottom boundaries of the retina, our algorithm starts by applying a Gaussian smoothing filter ($\sigma = 3$ pixels isotropic or $\sigma_{(x,z)} = (17, 12)$ µm) on each B-scan separately. Then, it computes a vertical image derivative along each A-scan using a Sobel kernel [92]. Looking at each A-scan, we find an initial estimate of either the ILM or the IS-OS boundary from the two pixels with the largest positive gradient values more than 25 pixels (97 µm) apart, since both of these boundaries have a similar gradient profile. To find an estimate of the BrM, we take the pixel with the largest negative gradient below that of the IS-OS, but no more than 30 pixels (116 µm) from it. These two collections of largest positive and negative gradients are taken to be the ILM and BrM, respectively. Of course, using only the maximum gradient values leads to spurious points along each surface. Correction of these errors is accomplished by comparing the estimated boundaries to the boundary given by median filtering the two collections. The algorithm replaces outlying points through interpolation, with outlying points defined as those more than 15 pixels (58 µm) from the median filtered surfaces. The final retina boundary surfaces are then found after applying Gaussian smoothing to the position values of each surface ($\sigma_{(x,z)} = (10, 0.75)$ pixels or $(56, 91)$ µm for the ILM and $\sigma_{(x,z)} = (20, 2)$ pixels or $(111, 244)$ µm for the BrM). This final smoothing step acts to smooth out smaller outlying boundary points, often caused by variations in the choroid intensity. Examples of the detected boundaries and the flattened image are shown in Figs. 3.2(c) and 3.2(d), respectively.

Boundary classification

We find the retinal layers in an OCT image using an RF classifier that is trained from manual delineations to find the boundaries between layers. Focusing on identifying the one pixel wide *boundaries* between layers rather than directly finding the *layers* themselves is different than previous work [27, 28]. Since pixels found in between boundaries are more difficult to classify due to a weaker feature response, we believe that our approach takes better advantage of the distinctive features that exist on and near boundaries and also permits better control of layer ordering. Also note that we will be converting these boundaries to layer segmentations by assigning each boundary pixel to the layer above it.

We train the RF classifier using 27 features (defined next) calculated at each pixel. During training, the classifier uses ground truth labels—created by a human rater—to learn the relationship between the high-dimensional feature space and the boundary labels. Once trained, the classifier will be applied to unlabeled data sets by computing these features and inputting them into the classifier to retrieve a set of boundary probabilities at each pixel.

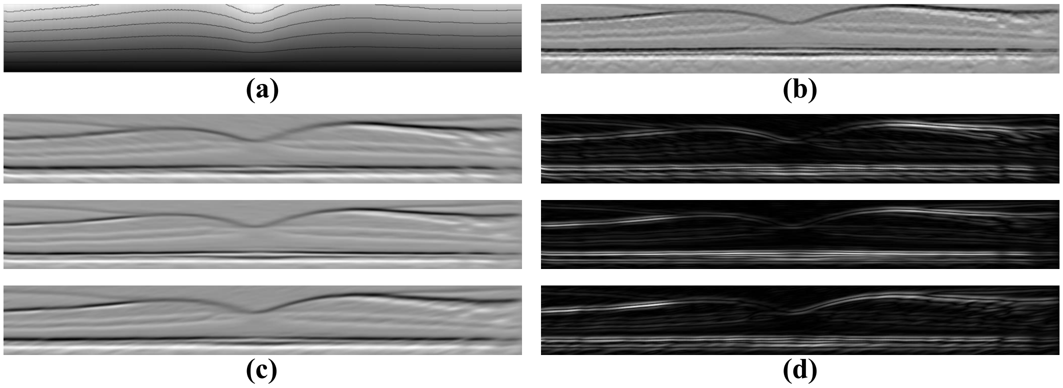


Figure 3.3: Example images of the different types of features used by the classifier: (a) the relative distance between the bottom and top boundary with contour lines overlaid, (b) the average gradient in a neighborhood below each pixel, and anisotropic Gaussian (c) first and (d) second derivatives oriented at −10 (top), 0 (center), and 10 (bottom) degrees from the horizontal.

Features We use 27 features as inputs to our RF classifier. The first three features give spatial awareness, while the remaining are local, contextual features. The first spatial feature is the relative distance of each pixel (along the A-scan) between the initial retinal boundaries (see Fig. 3.3(a)). The second and third features are the signed distance to the center of the fovea in the x and y directions,[1] respectively, with the center of the fovea being taken as the thinnest position between the retinal boundaries near the center of the volume. Together, these three features help to localize retinal pixels within a generic retina using a coordinate system that is defined by the geometry of the subject-specific retina.

The first set of contextual features we use are the intensity values of the nine pixels in a 3 × 3 neighborhood around each pixel. Together, these nine values allow the classifier to learn local relationships between neighboring points without explicitly calculating any new features. It has previously been shown to be an effective feature when compared to other sets of filter banks [93].

[1] Note that the x and y directions are the same as those defined in constructing the 3D graph in Fig. 2.5.

Although the 3×3 neighborhood pixels are useful for providing context locally, larger neighborhoods are also beneficial. Therefore, we supplement these features with an added filter bank of vertical first and second derivatives taken after oriented anisotropic Gaussian smoothing at different orientations and scales [94] (see Figs. 3.3(c) and 3.3(d)). A similar type of filter bank has been used in the texture classification literature [95]. Twelve features per pixel are generated by using the signed value of the first and magnitude of the second derivatives at two scales and three orientations. The two scales for Gaussian smoothing are $\sigma_{(x,z)} = (5, 1)$ pixels ($(30, 4)$ µm) and $\sigma_{(x,z)} = (10, 2)$ pixels ($(61, 8)$ µm) at an orientation of 0 degrees. These kernels are then rotated by -10 and 10 degrees from the horizontal (-6.4 and 6.4 degrees when scaled to µm) for oriented filtering [94]. Since the data are previously flattened, these three orientations are sufficient for learning the central foveal shape. The final contextual features are the average vertical gradients in an 11×11 neighborhood located at 15, 25, and 35 pixels (58, 97, and 136 µm) below the current pixel, calculated using a Sobel kernel on the unsmoothed data (see Fig. 3.3(b)). These features help to determine whether or not other boundaries exist in the areas below the current pixel. For example, the OPL-ONL and IPL-INL boundaries can be differentiated since the IPL-INL has the positive gradient of the INL-OPL below it, while the OPL-ONL boundary does not have a similar response below it.

Random forest training Our full data set comprises 3D OCT volumes and manual delineations from 35 subjects. Each volume has 49 B-scans, and 3–8 of these are *foveal B-scans*, meaning that they include part of the foveal depression. Training the classifier with all of these data would take a long time, but because there is significant similarity across the volumes, it is sufficient to use a subset for training. We explore this experimentally below (Section 3.1.2) by optimizing over two training parameters: N_s, which is the number of subjects to use for training, and N_b, which is the number of B-scans to include per subject. To balance the training process, we use $N_b/2$ foveal and non-foveal B-scans each, all randomly selected from within each respective collection. For a given B-scan, we use all of the segmented boundary points for training—1024 points for each of the nine boundaries (since there are 1024 A-scans per B-scan in our data). Since the number of background pixels greatly outnumbers the boundary points, we balanced these training data by randomly choosing 1024 background pixels for training from each of the layers between boundaries and from the regions above and below the retina.

Boundary refinement

The output of the RF classifier is a set of boundary probabilities, as shown in Fig. 3.4. Although the boundaries are well-defined visually, the automatic identification of a set of one-pixel thick, properly-ordered boundaries is still challenging due to boundaries that have dropouts or are spread out vertically. We implemented and evaluated two methods to generate boundary curves from the boundary probabilities to compare the necessary complexity required to compute the final boundaries. The first, more simple method follows the spirit of the Canny edge detector [96] and is referred to as RF+CAN. The second, and current state-of-the-art method, uses an optimal graph-based search algorithm [63] and is referred to as RF+GS. The RF+CAN method can be classified as a 2D algorithm, operating on each B-scan independently, while RF+GS operates on the entire 3D volume. As noted in Sec. 2.2, the graph search optimization approach has been used previously for

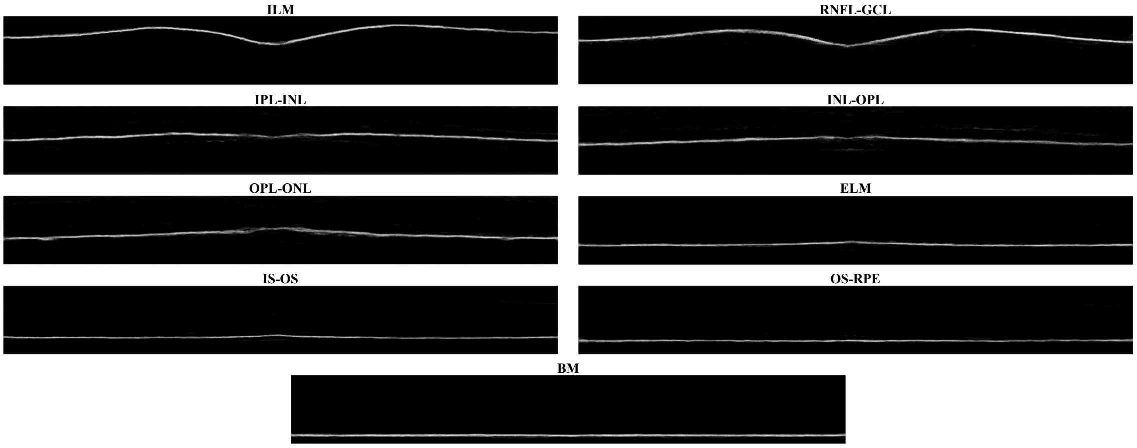


Figure 3.4: An example of the probabilities for each boundary generated as the output of the random forest classifier. The probabilities are shown for each boundary, starting from the top of the retina to the bottom, going across each row.

OCT [21, 22, 28, 29], though not with the costs we define. Also, we use only the basic graph algorithm in [63] and do not incorporate the spatially-varying smoothness, regional costs, and soft constraints that are used in more recent works, which can add computational complexity. The well-defined results of the RF also reduces their necessity. Our two approaches are described next.

RF+CAN approach The RF+CAN approach uses the non-maximal suppression and hysteresis thresholding steps that are found in Canny's seminal work on edge detection [96]. While Canny's work found edges by looking for image gradient peaks, our algorithm finds boundaries by looking for peaks in the probability images. Given a boundary probability map, we apply the following steps to find the final boundary:

1. Two-dimensional Gaussian smoothing with $\sigma = 2$ pixels isotropic $\big(\sigma_{(x,y)} = (12, 8)\,\mu\text{m}\big)$;
2. One-dimensional non-maximal suppression on each A-scan;
3. Hysteresis thresholding; and

4. Boundary tracking from left to right.

Gaussian filtering smooths out abrupt jumps within the data, thereby helping to reduce spurious results in the following steps. Non-maximal suppression [92] examines all three-pixel windows in each A-scan and zeros out the probabilities at the center pixel that are not maximal in the window. Remaining points are considered to be *strong boundary points* if their probabilities exceed 0.5 and *weak boundary points* if their probabilities are between 0.1 and 0.5. All other points are removed from consideration. Hysteresis thresholding [92] is applied next in order to remove all weak boundary points that are not connected to strong boundary points. All remaining points are considered to be highly likely to belong to the the final boundary.

Given the collection of probable boundary points determined in the first three steps, the fourth step defines the RF+CAN boundary by connecting boundary points across the entire B-scan image. First, the boundary point having the largest probability in the leftmost A-scan (which is by definition the one that is farthest away from the optic nerve, assuming a right eye) is identified. The boundary continues its path to the right by following the maximum point within three pixels above and below in the next A-scan. If there exists a second largest non-zero intensity pixel within the A-scan search window (taking note that the majority of the values are zero due to the non-maximal supression step), we also keep track of potential paths following from this point. In this way, if the main (primary) path has no points to move to, we check to see if any alternative (secondary) paths continue beyond where the primary path stops. If these secondary paths do continue beyond the primary path, it is now considered the primary path for tracking the boundary and we continue to track it accordingly. If there are no better secondary paths, we continue the boundary straight across. Therefore, in the absence of boundary evidence, this strategy favors flat boundaries, which is consistent with our initial flattening step.

This four-step process is repeated for each boundary, starting by finding the ILM and RNFL-GCL interfaces in a top to bottom order, and then finding the BrM through IPL-INL boundaries in a bottom-to-top order. To resolve any discrepancies in layer ordering, during the boundary tracking step we simply move any conflicting points one pixel away from the previously estimated boundary points. The direction of movement is down for the RNFL-GCL boundary and up for all other layers. (We find that there is never a discrepancy between the RNFL-GCL and IPL-INL boundaries where the two boundary detection processes meet.)

RF+GS approach The RF+GS approach defines a cost based on the estimated boundary probabilities in all B-scan images and finds the collection of boundary surfaces having the minimum cost over the whole 3D volume. The graph-theoretic algorithm described in Sec. 2.2 is used to find an optimal collection of layered structures. Accordingly, the RF+GS algorithm constructs graphs for each retinal surface boundary and then connects the graphs together such that inter-surface relationships are preserved. Multiple constraints are used to limit both the intra-surface distances between adjacent pixels in each direction (Δ_x and Δ_y, for the x and y directions, respectively) and the inter-surface distances (δ^l and δ^u, representing the minimum and maximum distance between surfaces). In our work, we use the values $\Delta_x = 1$, $\Delta_y = 10$, $\delta^l = 1$, and $\delta^u = 100$ pixels (with respective values of 4, 39, 4, and 388 μm). Also note that since this algorithm finds a *minimum nonnegative* cost solution, the cost is specified as 1 minus the boundary probabilities. Once the graph is constructed, a min-cut/max-flow algorithm is used to solve for the optimal collection of surfaces [81].

Note that solving for the optimal cut simultaneously for all nine boundaries requires an enormous amount of computer memory. To alleviate this problem, we separately estimate the final surfaces in three groups. These three groups are the ILM surface alone, the 2nd to 4th surfaces, and the 5th to 9th surfaces, with the boundary numbering going from top to bottom of the retina. Following this process, we did not find any problems with ordering between the groups. Similar schemes were used to solve for the different boundaries in [22] and [29].

3.1.2 Experiments and results

Data from the right eyes of 35 subjects were obtained using a Spectralis OCT system (Heidelberg Engineering, Heidelberg, Germany). The research protocol was approved by the local Institutional Review Board, and written informed consent was obtained from all participants. Of the 35 subjects, 21 were diagnosed with MS while the remaining 14 were healthy controls. All scans were screened and found to be free of MME. The subjects ranged in age from 20 to 56 years old with an average age of 39.

All scans were acquired using the Spectralis scanner's automatic real-time function in order to improve image quality by averaging at least 12 images of the same location. The resulting scans had signal-to-noise ratios of at least 20 dB. Macular raster scans ($20^\circ \times 20^\circ$) were acquired with 49 B-scans, each B-scan having 1024 A-scans with 496 pixels per A-scan. The B-scan resolution varied slightly between subjects and averaged 5.8 μm laterally and

3.9 μm axially. The through-plane distance (slice separation) averaged 123.6 μm between images, resulting in an imaging area of approximately 6 × 6 mm. The volume data was exported from the scanner using the vol file format. For all processing steps in our algorithm except for intensity normalization, we used the intensity data after transforming the original values by taking the fourth root.

The nine layer boundaries were manually delineated on all B-scans for all subjects by a single rater using an internally developed protocol and software tool. The manual delineations were performed by clicking on approximately 20–50 points along each layer border followed by interpolation between the points using a cubic B-spline. Visual feedback was used to move each point to ensure a curve that correctly identifies the boundary.

Parameter selection

The general properties of our RF classifier are specified by the number of trees N_t and the number of features m that are used at each node of each tree. The quality of training is dependent on the number of subjects N_s and number of B-scans N_b per subject. In selecting values for these parameters, we are interested in finding the set which provide a good segmentation accuracy without adding significant computational cost (as would be the case with more trees, more training data, etc.). We are not necessarily interested in finding the optimal set. To find suitable values for these parameters, we evaluated the performance of our RF classifier (using the RF+CAN algorithm) in a series of four experiments applied to 10 out of the 35 subjects. We did not use the entire dataset to carry out parameter selection because of the computational burden.

In each experiment, we swept through the values of one of the four parameters while keeping the other three fixed to a reasonable value (thus generating a new set of parameters). For example, to find appropriate values for each of N_t, m, and N_b, we used three training subjects ($N_s = 3$). To reduce the possibility of bias from training on particular subjects, a *cross-validation* strategy was used whereby a set of 10 classifiers were trained for *every* parameter set, each trained with N_s different randomly chosen subjects (from the pool of 10 subjects). For each trained classifier, we generated segmentations for the $10 - N_s$ test subjects not used in training. The overall error for each parameter set was calculated by averaging the absolute error between the segmentation and the manual segmentation across all test subjects evaluated with each of the 10 classifiers. Figure 3.5 provides an example error plot for each layer as the parameter N_s is varied from one to nine. The error bars represent the standard deviation of the error across the 10 trials. Similar experiments were performed for the other 3 parameters. Finally, we note that for the N_s experiments, a separate test set of 10 subjects was used to maintain a consistant number of test subjects as N_s was varied (it would not be fair to compare $N_s = 1$ with $N_s = 8$ by evaluating on $10 - N_s = 9$ and 2 test subjects, respectively).

Each of the parameters exhibited good stability and accuracy over a range of values. As a balance between performance and efficiency, we chose the *final set of parameters* (FSP) to be $\{N_t = 60,\ m = 10,\ N_s = 7,\ N_b = 8\}$. Using values larger than these show only a small performance improvement at a much larger computational burden. With $N_s = 7$ and $N_b = 8$, a total of 56 manual B-scan segmentations are needed for training. In an effort to reduce the amount of training data that are needed and to reduce the loading time, computation time, and memory requirements of the classifier, we also evaluated the algorithm using a *minimal*

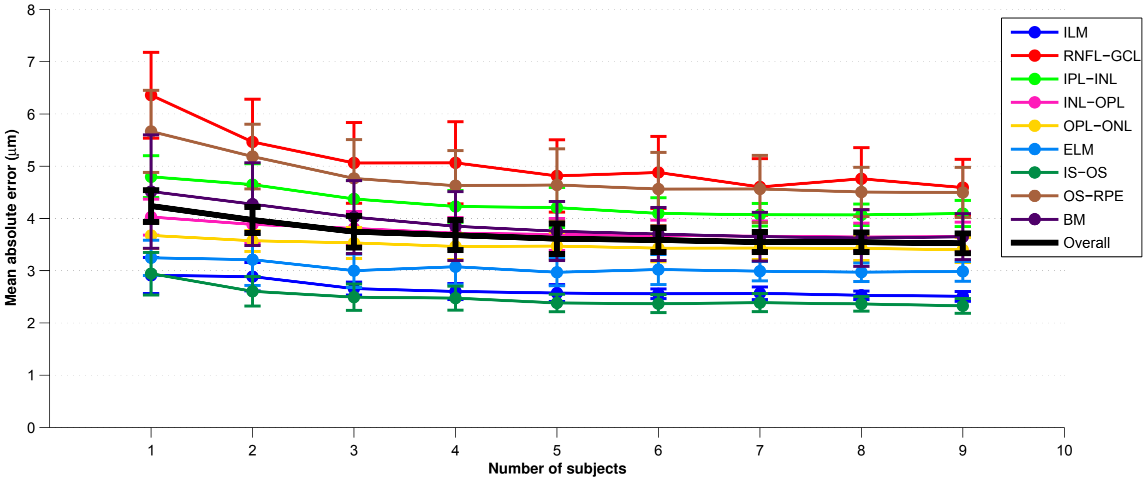


Figure 3.5: A plot of the mean absolute error across all boundary points versus the number of subjects, N_s, used in training the classifier. For each value of N_s, the experiment was repeated with a random set of subjects ten times. Averages are across these ten trials and error bars represent one standard deviation.

set of parameters (MSP), chosen to be $\{N_t = 20,\ m = 5,\ N_s = 2,\ N_b = 4\}$. In this case, with $N_s = 2$ and $N_b = 4$, only 8 manual B-segmentations are needed for training. We denote this set of parameters as minimal since we feel that using this set requires the minimum amount of training data necessary for the algorithm to perform acceptably well, in addition to being more efficient in the time required to compute the final segmentation. The memory footprint of the classifier is also significantly smaller, from 4 GB down to about 200 MB (a factor of 20), making it possible to run the algorithm on a wider variety of computational platforms.

Results

We evaluated our two segmentation methods, RF+CAN and RF+GS, on all 35 subjects using both the MSP and FSP. Since a cross-validation strategy was used in the parameter selection, there were 10 previously trained classifiers constructed using the FSP. We used these classifiers for the final evaluation of each algorithm. With the FSP, $N_s = 7$

randomly chosen subjects (out of the pool of 10 subjects) were used to train each of the 10 classifiers. Each classifier was evaluated by computing a segmentation on the $35 - 7 = 28$ remaining test subjects. To be clear, we are simply extending the results of the parameter selection experiments using the FSP to the entire set of subjects. Since the MSP was chosen in a more *ad-hoc* manner, this parameter set did not have a corresponding parameter selection experiment. Therefore, we trained 10 classifiers using the MSP with a random set of $N_s = 2$ subjects chosen from the same pool of 10 subjects used. In our first set of results, we compare RF+CAN and RF+GS using both parameter sets. We then show additional results using only the best algorithm and parameter set, which is RF+GS with the FSP.

To compare the results of our algorithm against the manual delineations, we calculated the absolute and signed boundary errors for every point on every surface. These errors were then averaged over all boundaries, subjects, and cross-validation runs. Table 3.1 shows the results for the two different algorithms with both parameter sets. The standard deviations were calculated assuming that every error value is separate (i.e. errors were not averaged for each subject before calculation). For both algorithms, we observe significantly better performance using the FSP over the MSP ($p < 0.001$). Significance was not found when comparing RF+CAN with RF+GS using the FSP, but was found using the MSP ($p < 0.01$). Significance was calculated on the average signed error over the whole volume across subjects using a one-sided paired Wilcoxon signed rank test. The MSP still performs well, however, having mean absolute errors about 0.60 μm larger than the FSP. For the same parameter set, the main difference between RF+CAN and RF+GS is that RF+GS has a lower standard deviation of error.

Table 3.1: A comparison of the two boundary refinement algorithms. Both mean signed and absolute errors with the minimal and final parameter sets are included. Units are in μm and standard deviations are in parentheses.

	Minimal Parameter Set		**Final Parameter Set**	
Algorithm	**Absolute Error**	**Signed Error**	**Absolute Error**	**Signed Error**
RF+CAN	4.09 (6.41)	-0.60 (7.58)	3.40 (4.82)	-0.12 (5.90)
RF+GS	4.01 (5.70)	-0.56 (6.95)	3.38 (4.10)	-0.11 (5.31)

To learn how each algorithm performs in certain regions of the macular cube. We assume that the acquired macular volumes are in alignment across the population. Therefore, the means and standard deviations of boundary error on each A-mode scan can be displayed as a fundus image, as shown in Fig. 3.6. Only the FSP was used here. Each image is oriented with the superior and nasal directions to the top and right, in agreement with the fundus image in Fig. 1.3(a). Although the subjects are not spatially aligned before averaging, this figure provides a meaningful illustration as the volumes are all similarly orientated with the foveae at their center.

Figs. 3.6(a) and 3.6(b) shows that the mean errors in RF+CAN and RF+GS are almost identical. In fact, the errors show some structural bias, indicating that particular areas of the retina are equally difficult to accurately segment for each algorithm. The central fovea area is a consistent source of larger error; an expected result as this region is where several layers converge. Since the layers are converging, the localization of individual boundaries becomes more difficult not only for the algorithm, but also for a manual rater. We also see areas of larger error in the nasal (right) side of the RNFL-GCL boundary as well as in the outer area of the OS-RPE boundary. The errors in the RNFL-GCL boundary are most likely due to the presence and large size of blood vessels running through this

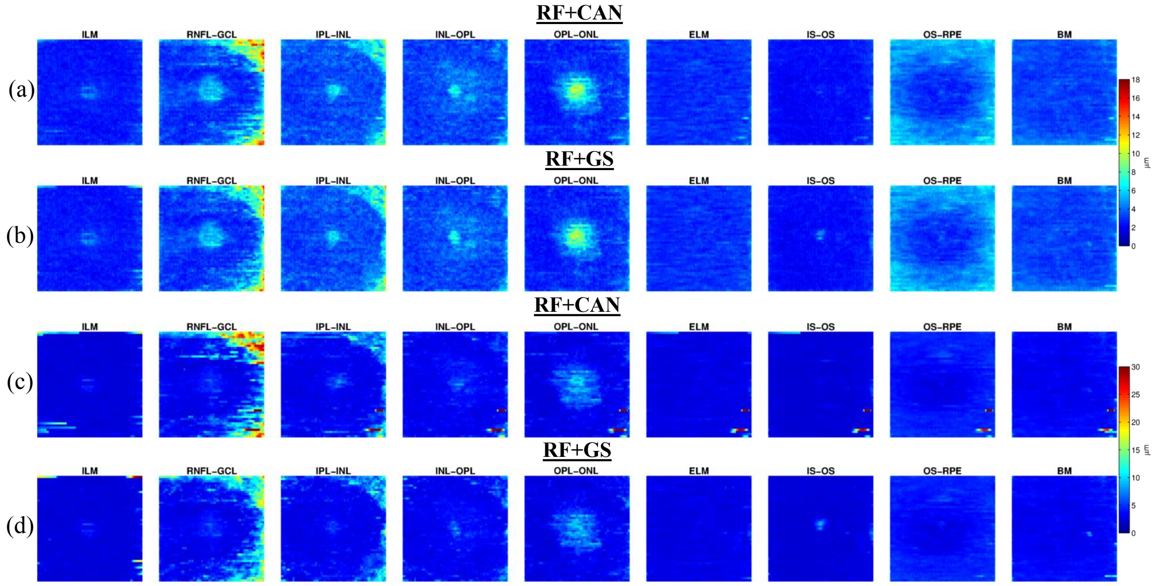


Figure 3.6: (a,b) Images of the mean absolute error (µm) of each boundary at each pixel for the RF+CAN and RF+GS algorithms, respectively, with (c,d) the corresponding standard deviation of the errors. Averages are taken over all subjects and all cross-validation runs (280 values).

region. We can attribute the errors in the OS-RPE boundary to the fact that this boundary is more difficult to see in these areas as there is a transition from mostly cones near the fovea to mostly rods in the outer macula. Looking at the images in Figs. 3.6(c) and 3.6(d), the patterns of standard deviation between the two algorithms are visually similar. RF+CAN shows larger overall standard deviations, particularly in the RNFL-GCL boundary and occasional very large outliers. Since the RF+GS algorithm shows more overall stability, all further experimental results are shown only for the RF+GS algorithm using the FSP.[2]

Boundary specific errors for RF+GS are given in Table 3.2, with an additional breakdown by population—all subjects, controls, and MS patients. The best performing boundaries, the ILM and IS-OS, are the boundaries with the largest gradient response,

[2] In fact, all further experiments described in this thesis will use the RF+GC algorithm using the FSP.

thereby making them readily identifiable. The average errors are quite consistent across all boundaries, with errors of less than 1 pixel in all but the the RNFL-GCL and OS-RPE boundaries. Interestingly, we see very few differences with boundary errors between control and MS subjects. The average boundary errors between the two groups all have differences of less than 1 micron, with no statistically significant differences between them.

Fig. 3.7 shows standard box plots of the boundary errors across all of the subjects. A total of 49 points were included for each subject, where each point represents the absolute error averaged across all boundaries and cross-validation repetitions of a single B-scan. Subjects are divided by disease diagnosis and ordered by age within each diagnostic group. This figure again shows that our algorithm yields similar results in both MS and control subjects, with no age-dependent error trends in either population. Outliers are few relative to the numbers of trials carried out and still mostly fall below 2 pixels in error. A detailed examination of these outliers shows the presence of blood vessel artifacts in these scans.

Fig. 3.8 shows estimated boundaries from two B-scans taken from two different subjects. Boundaries for each of 10 cross-validation trials are plotted on the same B-scan, using a different color for each boundary. The manually traced boundary is plotted using a black curve after all other boundaries are drawn. When only the black curve is apparent, this indicates that all estimated curves agree with the truth. In areas where colors are visible near the black curves, this is indicative of some or many boundary estimates disagreeing with the truth. We observe that larger errors tend to be located within the shadows of blood vessels.

So far, our focus has been on boundary accuracy; however, the more clinically

Table 3.2: Mean absolute and signed errors (and standard deviations) in µm for the RF+GS algorithm on all segmented boundaries for all the subjects and broken down by control and MS patients.

	Absolute Errors (µm)		
Boundary	All	Control	MS
ILM	2.60 (3.33)	2.62 (3.89)	2.59 (2.89)
RNFL-GCL	4.03 (6.34)	4.00 (6.11)	4.04 (6.48)
IPL-INL	3.87 (4.54)	3.78 (4.41)	3.94 (4.62)
INL-OPL	3.57 (3.75)	3.44 (3.61)	3.66 (3.84)
OPL-ONL	3.27 (4.06)	3.40 (4.24)	3.19 (3.93)
ELM	2.96 (2.84)	2.79 (2.68)	3.07 (2.93)
IS-OS	2.34 (2.56)	2.38 (2.49)	2.30 (2.61)
OS-RPE	4.32 (4.23)	4.16 (4.13)	4.43 (4.30)
BrM	3.50 (3.56)	3.87 (3.69)	3.24 (3.44)
Overall	3.38 (4.10)	3.38 (4.09)	3.39 (4.10)

	Signed Errors (µm)		
Boundary	All	Control	MS
ILM	-0.22 (4.22)	-0.04 (4.69)	-0.34 (3.86)
RNFL-GCL	-0.88 (7.45)	-0.78 (7.26)	-0.95 (7.58)
IPL-INL	-1.93 (5.65)	-1.66 (5.57)	-2.11 (5.69)
INL-OPL	0.79 (5.12)	0.36 (4.97)	1.08 (5.19)
OPL-ONL	0.23 (5.21)	0.37 (5.42)	0.14 (5.06)
ELM	-0.65 (4.05)	-1.04 (3.73)	-0.39 (4.23)
IS-OS	0.13 (3.47)	0.33 (3.43)	0.00 (3.48)
OS-RPE	0.79 (6.00)	1.51 (5.67)	0.31 (6.17)
BrM	0.74 (4.93)	1.63 (5.10)	0.14 (4.72)
Overall	-0.11 (5.31)	0.08 (5.31)	-0.23 (5.31)

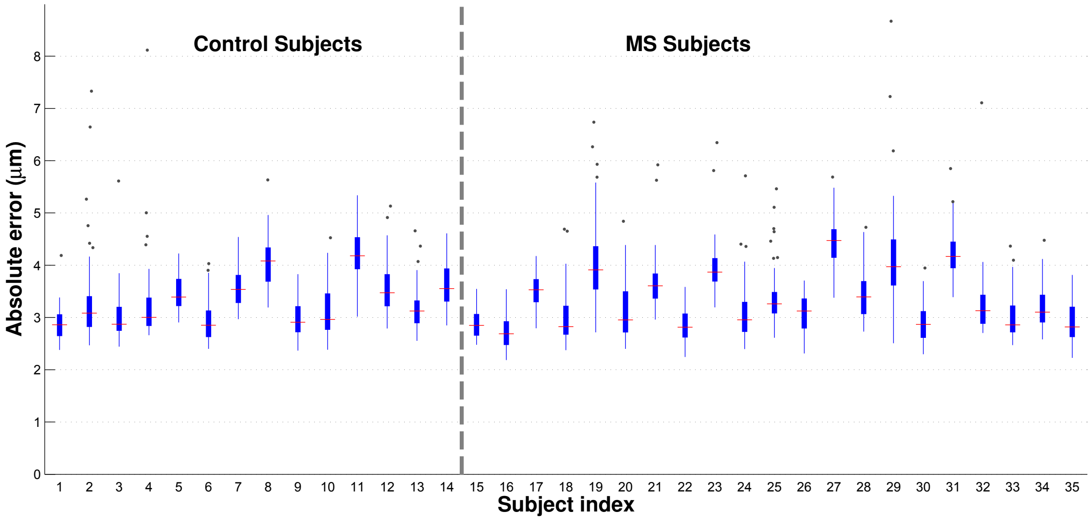


Figure 3.7: Box and whisker plots of the mean absolute errors for every subject used in this study. Subjects are ordered by diagnosis and then age (increasing from left to right within each diagnostic group). A total of 49 data points were used to generate each subject's plot, with each data point representing the error of a particular B-scan averaged across all cross-validation runs. For each subject, the red line represents the median absolute error and the edges of the box correspond to the 25th and 75th percentile of the error. All points lying outside of the whiskers are greater than 1.5 times the interquartile range.

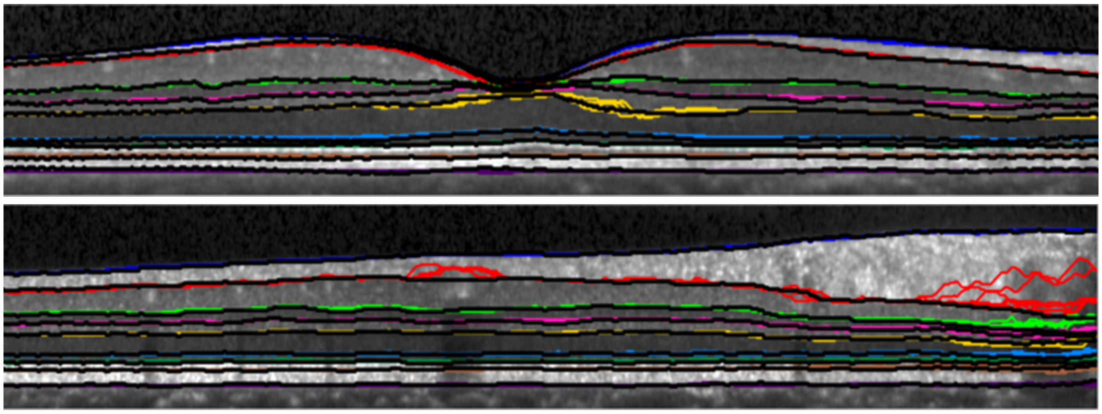


Figure 3.8: Two B-scan images from two different subjects are shown with the resulting boundaries from each of the 10 cross-validation runs overlaid. Each boundary is represented by a different color with the manual delineation shown atop the other boundaries in black.

important measure is considered to be layer thicknesses. These thicknesses are often reported as average values within different sectors of the macula surrounding the fovea [16, 97]. A standardized template is centered over the fovea and used to divide the macula into these sectors [98]. Fig. 3.9 shows this template over a retinal fundus image. The sectors are labeled with C1 representing the central 1 mm diameter area, S3, I3, N3, and T3 representing the superior, inferior, nasal, and temporal areas of the inner 3 mm diameter ring, and S6, I6, N6, and T6 representing the same areas in the outer 6 mm diameter ring. We also use M for the macular region within the dashed box (the imaged area). Table 3.3 lists the absolute errors of the average thickness for each layer within the nine sectors and the whole macula. The template was aligned to the center of each subject's volume, again assuming that all of the data is already in rough alignment with the fovea at the center. The OPL and ONL show the largest errors, especially in the central area where the layers converge. Many of the other sectors have errors around 4 μm with standard deviations less than 2 μm.

Computational performance

The algorithm was coded in MATLAB R2012b (The Mathworks, Inc., Natick, MA, USA) using external, openly available packages for the RF classification [99], the min-cut/max-flow algorithm [81], and calculation of anisotropic Gaussian features [94]. All other code used built-in MATLAB functions. Experiments were performed on a computer running the Fedora Linux operating system with a 3.07 GHz quad-core Intel Core i7 processor.

To assess the algorithms' computational behavior we calculated the average time taken to perform each step of the algorithm. The preprocessing stages (normalization and flattening) took an average of 17 seconds per volume and calculation of image features took

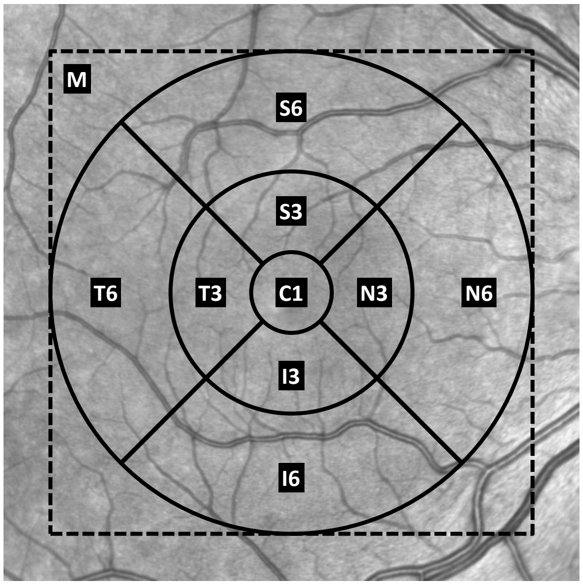


Figure 3.9: The template for the sectors of the macula overlaid on a fundus image. The dashed square surrounding the template represents the imaged area. The concentric circles are centered on the geometric center of the OCT volume and have diameters of 1 mm, 3 mm, and 6 mm.

an average of 33 seconds. The random forest classification averaged 114 seconds per volume using the FSP and 24 seconds with the MSP. Boundary tracking using the Canny approach took an average of 19 seconds per volume. Therefore, the RF+CAN algorithm took an average of 183 seconds per volume for the FSP and an average of 93 seconds per volume for the MSP. Boundary tracking using the graph segmentation approach took an average of 54 seconds per volume. Therefore, the RF+GS algorithm took an average of 218 seconds per volume for the FSP and 128 seconds for the MSP. Thus, the best performing algorithm (RF+GS using the FSP) takes less than four minutes to process a volumetric macular scan comprising 49 B-scans.

Training time is a bit more difficult to analyze, as manual delineation time is involved and is the most time consuming part of the process. Based on feedback from our

Table 3.3: Retinal layer thickness absolute errors (in μm, with standard deviation) calculated for different sectors of the macula (see Fig. 3.9 for sector positions). The 'Macula' column represents the absolute error of the average thickness of the entire imaged area. Errors are between the results of RF+GS in comparison to the manual rater.

Layer	C1	S3	I3	N3	T3
RNFL	2.88 (1.78)	1.73 (1.28)	0.97 (0.83)	1.50 (0.97)	1.75 (1.19)
GCIP	2.62 (1.84)	2.14 (1.26)	1.76 (1.23)	1.54 (1.21)	2.14 (1.14)
INL	2.62 (1.84)	3.35 (1.86)	2.73 (1.86)	3.86 (1.90)	2.88 (1.54)
OPL	3.46 (3.35)	3.16 (2.94)	2.71 (4.20)	2.86 (2.80)	2.10 (2.50)
ONL	4.35 (3.26)	3.02 (2.08)	3.36 (3.85)	2.83 (2.26)	2.51 (2.13)
IS	2.54 (1.98)	2.58 (1.94)	2.60 (1.90)	2.75 (1.80)	2.39 (1.77)
OS	2.48 (1.79)	2.31 (1.93)	1.97 (1.55)	2.33 (1.70)	2.04 (1.68)
RPE	2.08 (1.57)	2.80 (2.05)	2.81 (1.97)	2.55 (1.92)	2.65 (2.13)
Overall	2.88 (2.37)	2.64 (2.05)	2.37 (2.55)	2.53 (2.03)	2.31 (1.85)

Layer	S6	I6	N6	T6	Macula
RNFL	1.87 (2.00)	1.61 (1.44)	2.19 (2.39)	1.36 (1.03)	1.33 (1.29)
GCIP	1.51 (1.10)	1.49 (0.93)	1.69 (1.16)	2.03 (0.96)	1.24 (0.76)
INL	2.90 (1.74)	2.76 (1.74)	3.37 (1.98)	2.48 (1.60)	2.90 (1.56)
OPL	1.53 (1.27)	1.61 (1.12)	1.94 (1.56)	1.44 (1.08)	1.54 (1.21)
ONL	2.05 (1.40)	2.18 (1.44)	2.13 (1.48)	1.83 (1.24)	1.96 (1.26)
IS	2.72 (2.00)	2.65 (1.95)	2.87 (2.07)	2.40 (1.70)	2.48 (1.86)
OS	3.44 (3.07)	2.96 (2.97)	3.06 (2.73)	2.71 (2.23)	2.52 (2.35)
RPE	4.06 (3.16)	3.67 (3.24)	3.51 (2.70)	3.54 (2.60)	3.14 (2.36)
Overall	2.51 (2.27)	2.37 (2.14)	2.60 (2.18)	2.22 (1.78)	2.14 (1.80)

manual rater, we estimate that it takes about 10 minutes to manually delineate a single B-scan. Since there are 56 B-scans required to train using the FSP, this alone takes 560 minutes (9 hours and 20 minutes). Training the random forest takes only 17 minutes for these 56 delineations, which means that it takes just under 10 hours to train for the FSP. Since the minimal set requires only 8 B-scans and 25 seconds to train the classifier, the random forest can be trained—including time for manual delineation—in just one hour and 20 minutes for the MSP.

The algorithm can be sped up significantly by taking advantage of several potential optimizations. For example, we can use a faster programming language, such as C, and we can also speed up the classification part of the algorithm by parallelizing the random forest across multiple computer cores or utilizing a graphics processing unit [100].

3.1.3 Discussion and conclusion

The results for both of our algorithms show excellent performance with an absolute error of less than 3.5 μm averaged over all boundaries and less than 4.5 μm for any one specific boundary. Although the overall average errors are nearly identical between the two algorithms, the standard deviation of the RF+CAN algorithm is slightly larger due to the possibility that the boundary tracking algorithm fails in some areas. The hard constraints imposed by the graph search algorithm prevent these failures in the RF+GS algorithm. Looking at the thickness values calculated using the RF+GS algorithm, we see average errors of less than 3 μm in 81% of the sectors and standard deviation values less than 3 μm in 90% of the sectors indicating a high level of confidence in these measurements. When using the minimal training set, the errors are larger but the performance is still quite good, with errors only slightly larger than the axial pixel size. Therefore, training the RF from a new data source—i.e., for a new system or for new parameters on an existing system—could be carried out within only a few hours in order to achieve adequate performance when using the MSP.

When comparing our algorithm with other retinal segmentation methods found in the literature, we see comparable performance to the best algorithms [24, 25, 29], each of which shows average errors of between 3–4 μm. This comparison is inherently difficult, however, as the methods are evaluated on data acquired from different scanners using different numbers of manually delineated subjects (as few as 10 subjects). Due to the time consuming nature of manual segmentation, evaluations are often performed against only a subset of the full data (5–10 B-scans per subject). In our case, evaluations were carried out against entire volumes (all 49 B-scans per subject), which includes many poor quality images. Our manual segmentations are also generated as smooth spline curves from a small set of control points, which is different from other manual delineations and thus may introduce bias. Additionally, only a few methods provide results showing that they are able to accurately segment eight layers or more [23, 24, 26]. Although it may be possible to use other algorithms to segment all of the retinal layers, it is not clear how they will perform. In terms of computational performance, our algorithm runs significantly faster than the most similar method to ours [27], which uses machine learning for classification and regularization on only one layer at a time. We still lag behind faster algorithms including [24] and [29], the latter of which does a 3D segmentation of six layers in about 15 seconds. Complete characterization of the advantages and disadvantages of these algorithms will require a direct comparison on the same data using the same error criteria.

Looking to the future, although the algorithm performance was not degraded for MS subjects, we expect that modifications will be necessary to handle other pathologies, and show one such adaptation next in Sec. 3.2. The appearance of cysts would negatively impact the performance of our algorithm's ability to segment the layers due to the poor performance of the random forest classifier in these areas. The spatial features and spatial constraints on the graph may also need modification since these were trained on healthy or

MS subjects.

In the case of structurally normal eyes, there are still several potential areas of improvement—for example, allowing the RF to know if an A-scan has a blood vessel shadow could improve our results. Looking at Table 3.3, we know which regions and layers are more difficult to segment allowing us to focus on finding improvements for these areas. As far as feature selection goes, little effort was put into selecting the best set of features for retinal segmentation. Other groups have used a larger set of multi-scale features, including intensities, gradients and Gabor filters [27, 28]. It is also possible that a sampling of features from a large class of generalized features within the training phase of the algorithm [101], will help improve results. With regards to the graph-based segmentation in the second part of our work, learning boundary specific constraints at each surface position would improve the segmentation and further help to eliminate outliers in the final segmentation [22, 29].

3.2 Adaptation for segmentation of retinitis pigmentosa data

While many algorithms have been developed to segment the retinal layers in optical coherence tomography (OCT) images, pathological data with large scale changes due to edema, detachments, cystoid spaces, or atrophy, among other changes, often presents a challenge which cannot be handled by existing algorithms, including our own. To account for specific physiological changes that occur as a result of certain diseases, either these algorithms need to be adapted or new algorithms need to be developed. Examples of diseases requiring adaptation of previously developed methods include age related macular degeneration [35], diabetic macular edema [102], and retinitis pigmentosa (RP) [103]. Unfortunately, these

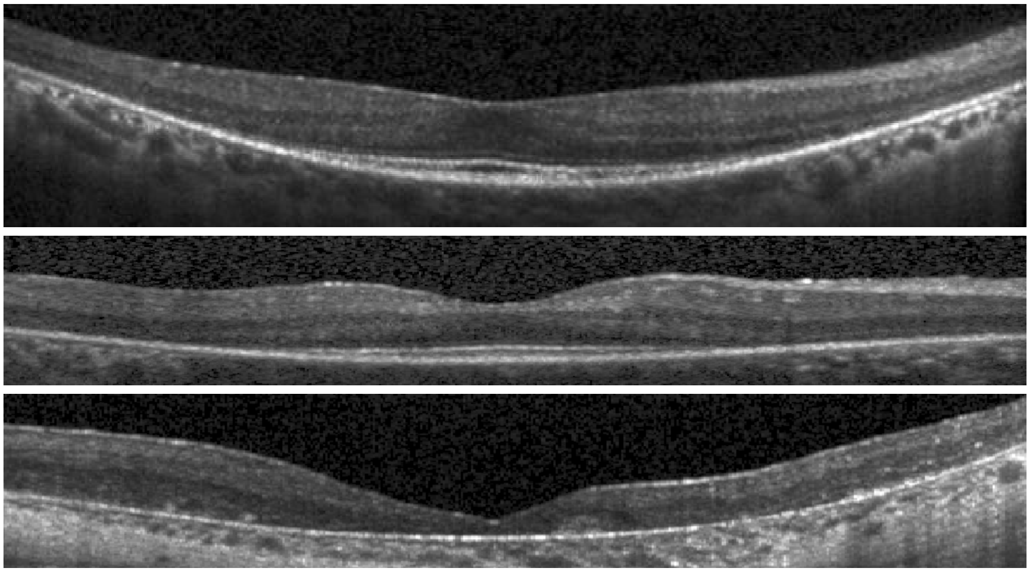


Figure 3.10: B-scan images from three RP subjects.

methods are often developed to only find a few specific layers within the retina.

RP is a hereditary degenerative disease which results in a loss of vision due to deterioration of the photoreceptor (PR) layers in the outer retina. Using OCT, the severity of RP can be quantified and tracked over time by measuring the thickness of the outer layers [104]. By looking at full two-dimensional fundus maps of these thicknesses, we can explore the spatial changes in specific areas of the retina. While the outer retina is the primary target for the disease, the inner retina has also been shown to undergo changes [105]. Thus, it is important to be able to accurately measure thicknesses for *all* of the layers in the retina to better understand the disease characteristics and progression. Example OCT images from three different RP subjects are shown in Fig. 3.10, where we see the amount of variability in this data.

While the major difficulty in the segmentation of RP data is due to deterioration in the outer layers, the images often have poor quality, making the inner boundaries difficult to distinguish as well. Only one method has been previously developed for automated RP segmentation [103]. This method used a dynamic programming approach to estimate four boundaries in the retina, finding only two layers along with the total retina thickness. Thus the problem of finding all of the retinal layers remains open to be explored. Here, we present an adaptation of our segmentation method in Sec. 3.1 for the segmentation of eight layers in macular RP data.

3.2.1 Methods

To segment the RP data, we use the RF+GC method described in Sec. 3.1, with training using the FSP. Some small changes were necessary to improve the performance and better capture the variability in the RP data. These changes are outlined below.

Retinal boundary estimation

The boundary estimation step was modified slightly with the BrM surface estimated after Gaussian filtering using a larger kernel size ($\sigma = 200$ µm) than originally used. Due to the reduced contrast between the BrM and the choroid, the BrM surface is not as accurately estimated in the RP subjects at this stage, and therefore, the flattening result is not as smooth. However, the RF used for boundary classification is insensitive to this inaccuracy, provided the result is not extremely poor.

Intensity normalization

For the RP data, a different intensity normalization strategy was used than previously described. The OCT scans from these subjects have a much larger variability in intensity due to both the loss of the PR layers and the overall low quality seen with RP data. The normalization is formulated to produce images where the top and bottom layers (the RNFL and RPE, respectively) have approximately uniform intensity distributions (with a value near 1). Therefore, we start by finding a bias field for each of these layers, $N_{\mathrm{RNFL}}(x, y)$ and $N_{\mathrm{RPE}}(x, y)$, which estimates a single scale factor for each A-scan to produce these uniform distributions. Note that this normalization strategy is in contrast to the method we later present in Sec. 6.1, which normalizes the intensities within all layers, as opposed to only two. The irregular structure of the RP data makes such a result more difficult to obtain.

For each of $N_{\mathrm{RNFL}}(x, y)$ and $N_{\mathrm{RPE}}(x, y)$, we decompose the bias field into *two parts*; one varying smoothly over the entire volume (in x and y), and one varying smoothly within each B-scan separately (only in x). The intensities in the respective layers before normalization are estimated from the maximum value within 80 μm of the top and bottom retinal boundaries in each A-scan. From this 2D map of intensities, the volumetric component of the bias field is estimated by fitting a bivariate tensor cubic smoothing spline to this map of intensity values for each of the RNFL and RPE. A fixed value of 0.9 was used for the regularization weight of the spline, which removes higher frequency variations. Finally, a per-B-scan component is estimated using a robust linear regression fit to the intensity profile of the A-scans in each B-scan. A bisquare weighting function was used in the robust fitting. Later, in Chapter 5, we will show this normalization process as applied to MME data, specifically in Fig. 5.2.

The final 3D bias field $N(x, y, z)$ is computed as the linear fit along each A-scan between the two bias fields as

$$m(x,y) = \frac{N_{\mathrm{RNFL}}(x,y) - N_{\mathrm{RPE}}(x,y)}{S_{\mathrm{ILM}}(x,y) - S_{\mathrm{BrM}}(x,y)}$$

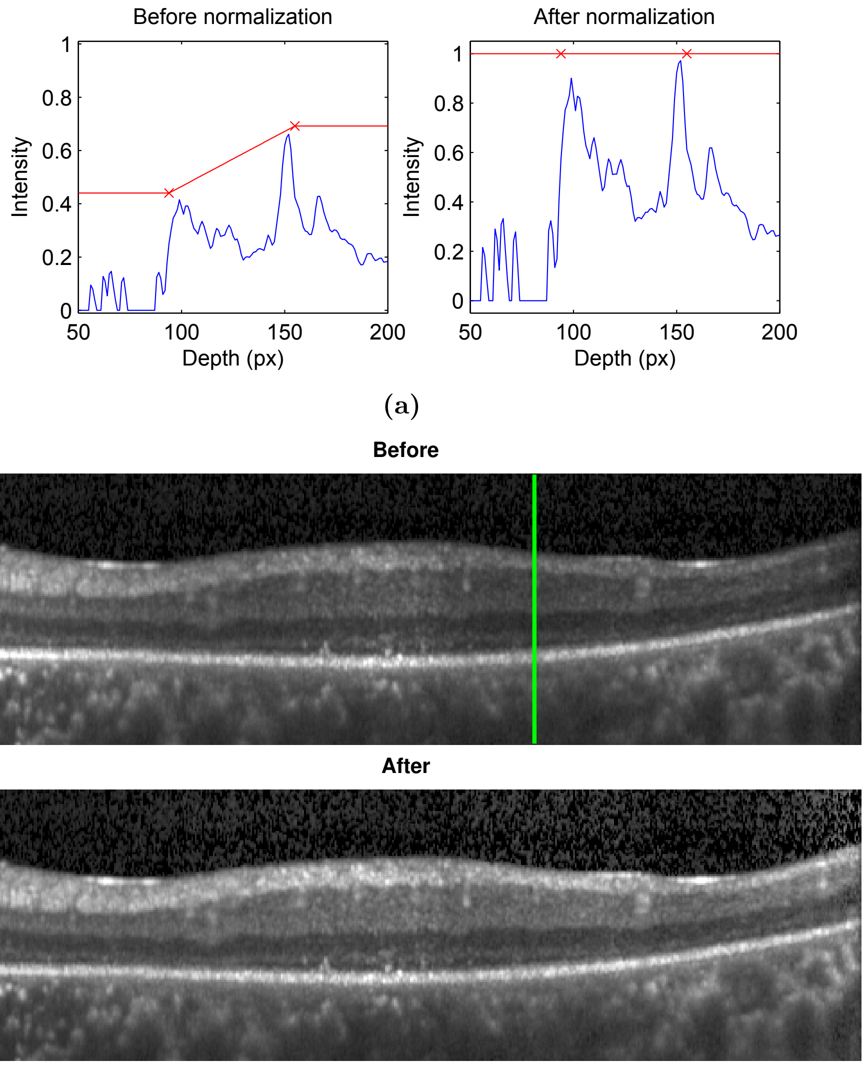
$$b(x,y) = N_{\mathrm{RNFL}}(x,y) - m(x,y) * N_{\mathrm{RPE}}(x,y)$$

$$N(x,y,z) = m(x,y) * z + b(x,y)$$

where $m(x,y)$ and $b(x,y)$ are the slope and intercept of the linear fit, and S_{ILM} and S_{BrM} are the estimated ILM and BrM boundary surfaces. The final normalized image is then computed as $\hat{I}(x,y,z) = I(x,y,z)/N(x,y,z)$. An example of how this bias field correction works for an individual A-scan is shown in Fig. 3.11(a) with the red line showing how the bias field changes linearly between the two retinal surfaces. An example B-scan image is then shown before and after normalization in Fig. 3.11(b).

Boundary learning

To better capture the wider variability in contrast of the RP data, we modified the features used for the RF classification. In total, a set of 44 features were used. The same three spatial features were used, consisting of the distance to the fovea in the x and y directions, as well as the relative vertical distance within the retina. Among intensity-based features, the 3×3 neighborhood around each pixel was used again. Instead of using *oriented* Gaussian first and second derivative features, we included *isotropic* Gaussian first and second derivative kernels at 6 scales: $\sigma = 1,\ 2,\ 5,\ 10,\ 15,$ and 20 μm. We also included the average intensity and gradient within a 20 μm area from -90 to 90 μm away from the each pixel in the vertical direction, in increments of 20 μm. The removal of the oriented filtering, as



(a)

(b)

Figure 3.11: (a) On the left and right are A-scan profiles before and after normalization, respectively. The red ×'s show the bias field correction values for the ILM and BrM. B-scan images before and after normalization are shown in (b), where the green line represents the A-scan shown in (a).

well as the use of more filter scales, appear to have the largest impact for improving the performance on RP data. By filtering at only three orientations before, we were restricting the shape of the boundaries, which have a much higher variability in RP due to the layer atrophy.

The RF was trained on manual segmentation data from eight subjects. Nine boundaries were delineated on seven B-scan images for each subject, where three of the B-scans selected near the fovea. All boundary points were used for training with half of the background pixels used, reducing the class imbalance and the computation time. Background pixels were sampled proportionally to the number of points within each layer.

Optimal graph segmentation

Previously, we used fixed constraints when constructing the graph used in the segmentation, which are described by Li et al. [63], but here, we include the spatially varying constraints $\Delta_x^l(x, y)$, $\Delta_x^u(x, y)$, $\Delta_y^l(x, y)$, $\Delta_y^u(x, y)$, $\delta^l(x, y)$, and $\delta^u(x, y)$ introduced by Garvin et al. [22] (also see Sec. 2.2.4). These constraints are learned from average values computed over a set of healthy subjects in a similar way to this prior work. Specifically, the minimum and maximum values were set to lie within 2.6 standard deviations of the mean value for a given parameter. The resulting parameters were subsequently smoothed using a 100 μm Gaussian kernel.

Since the constraints were learned on healthy data, and to account for differences in the retina due to RP, we modified the constraints in the following ways. The smoothness constraints are increased 3 times larger for the first five boundaries and 1.5 times more for the last four. These changes are designed to handle the increased variability added by the deterioration of the PR layers. Next, the maximum thickness of the first three layers is increased by 50% to account for larger thickness values sometimes found in these layers. Finally, the minimum thicknesses of the outer five layers were reduced to 25% of the learned values. Due to the discrete nature of the graph, this minimum went to zero in many cases after rounding.

The final change to the graph construction is that the first four layers are constrained to have zero thickness at the fovea. This constraint locks in the position of layers at the fovea where the RF sometimes produces a weaker response due to the disappearance of these layers. This constraint was included by changing the minimum and maximum thickness to zero within a 75 μm radius around the fovea. The location of the fovea was estimated using the total retina thickness by template matching to maximize the cross correlation with a template of the fovea generated from a healthy population.

3.2.2 Experiments and Results

Macular OCT scans from nine subjects with RP were acquired using a Heidelberg Spectralis scanner. Scans were acquired in a 6×6 mm area with 512 A-scans and a variable number of B-scans ranging from 19 to 49. The axial resolution was 3.9 μm with a depth of 1.9 mm. Manual segmentation was carried out on each of the scans. In some areas, the boundaries between layers could not be differentiated due to poor scan quality. These boundaries were traced based on the typical location found in other scans. In places where the PR layers showed significant degradation to the point where a layer could not be seen, the boundaries were placed arbitrarily close together, but not necessarily to have zero thickness.

A leave-one-out cross validation scheme was used to evaluate the segmentation results. The RF was trained on eight scans with the evaluation done on the ninth scan. The average boundary errors across all subjects are presented in Table 3.4 with average thickness errors in Table 3.5. Figure 3.12 shows example results on three different subjects. The boundaries with the two largest errors were the RNFL-GCL and OPL-ONL interfaces.

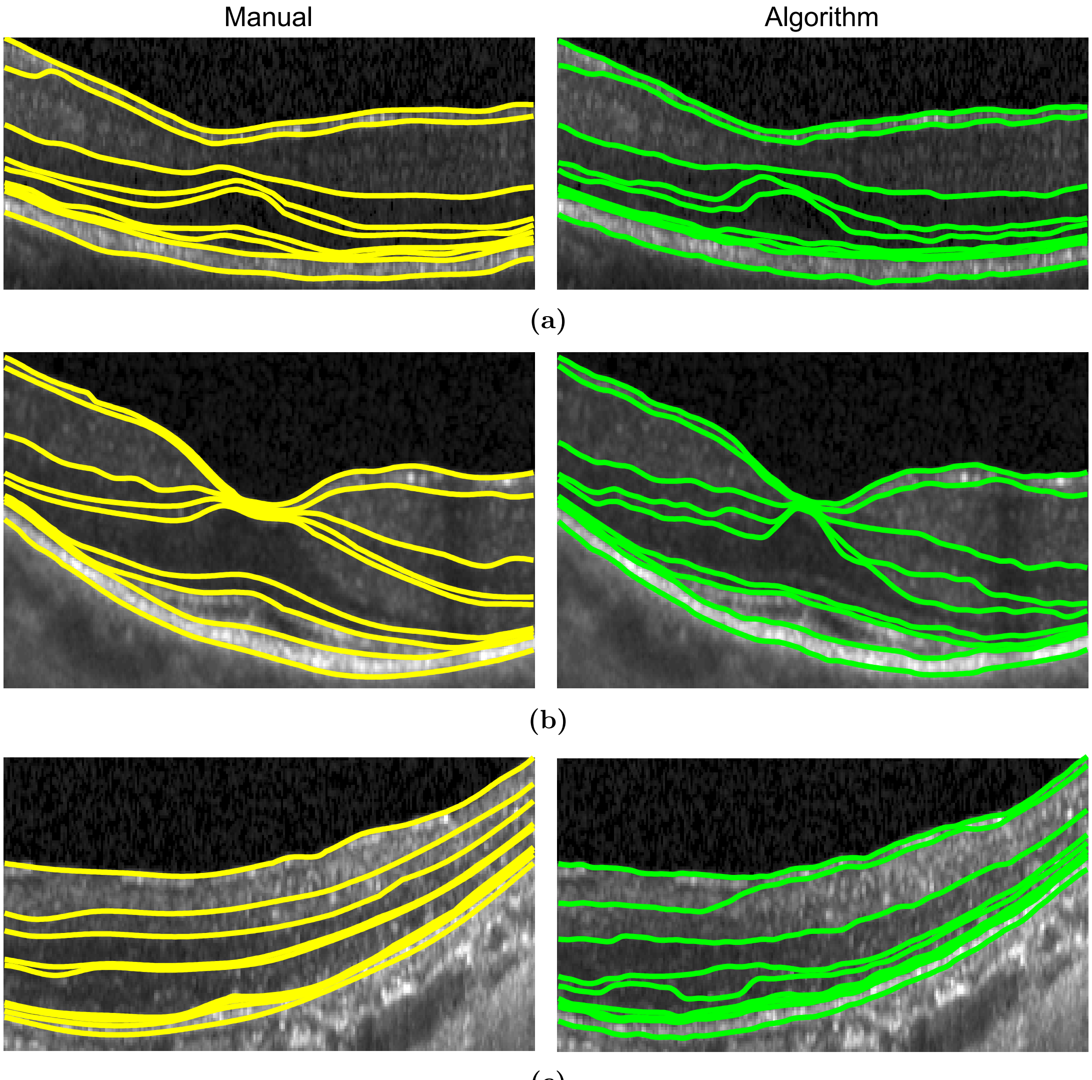
Table 3.4: Boundary errors averaged over all subjects (μm). Standard deviation values are in parentheses.

Boundary	Signed error	Absolute error
ILM	0.12 (1.69)	2.52 (0.69)
RNFL-GCL	1.82 (1.76)	6.25 (3.39)
IPL-INL	-2.70 (2.88)	4.83 (2.27)
INL-OPL	-1.65 (2.90)	5.47 (2.27)
OPL-ONL	-3.77 (3.19)	6.43 (2.86)
ELM	-1.19 (2.76)	4.39 (1.58)
IS-OS	-0.65 (1.78)	3.23 (1.33)
OS-RPE	0.02 (1.14)	2.68 (0.67)
BrM	-0.13 (0.79)	2.20 (0.58)
Mean	-0.90 (2.65)	4.22 (2.44)

In some of the scans, the RNFL and GCL are difficult to distinguish, partially due to scan quality and partially due to the lack of context provided by the missing layers. Figure 3.12(c) shows an example of this type of error. The errors at the OPL-ONL boundary were due to similar problems in addition to the disparity of visual appearance due to Henle's fiber layer (e.g. in Fig. 3.12(b)).

To show the importance of the intensity normalization and the spatial constraints in the graph, we ran the algorithm without these elements. The average absolute errors over all layers was 5.80 μm without the intensity normalization, 5.82 μm without the added spatial constraints, and 7.14 μm without either of the steps, which are all worse than the errors with the improvements (5.14 μm). While the setup without the spatial constraints was the same as our previous experiments in Sec. 3.1, the added variability in the RP data proved to be too large. Most of the improvement when the constraints were included was in the first four layers where the average errors decreased by more than 1 μm per layer.

One subject with particularly poor image quality, where the layers were difficult to



(a)

(b)

(c)

Figure 3.12: A comparison of the manual delineation and algorithm result on the left and right, respectively, for three subjects.

Table 3.5: Mean error of the computed thicknesses (μm). Standard deviations are in parentheses.

Layer name	Signed error	Absolute error
RNFL	1.70 (1.66)	6.59 (2.80)
GCL+IPL	-4.52 (3.84)	8.10 (3.66)
INL	1.05 (2.66)	6.04 (1.81)
OPL	-2.12 (1.97)	4.63 (1.17)
ONL	2.58 (4.51)	6.59 (2.37)
IS	0.54 (1.35)	2.78 (0.49)
OS	0.67 (1.94)	2.98 (1.09)
RPE	-0.15 (1.72)	3.39 (0.67)
Mean	-0.03 (3.33)	5.14 (2.69)

delineate, showed larger errors than the rest of the subjects (boundary error = 7.56 μm). Removing this subject, the mean of the average absolute errors decreases to 3.84 μm and 4.72 μm for the boundary and layer errors, respectively. An example delineated image from this subject is shown in Fig. 3.12(c).

3.2.3 Conclusions

In this work, we modified our RF+GS segmentation framework for use with RP data, a disease where the outer retinal layers deteriorate, especially in the peripheral regions. The graph search algorithm was used to segment the data with constraints modified to accommodate for the increased or decreased thickness of specific layers, as well as the increased variability in the surface smoothness. The performance of the algorithm was generally good for all of the layers with the average absolute boundary error of 4.22 μm comparing favorably with the average error of 3.38 μm reported on healthy data previously. Many of the changes to the algorithm, including modification of the RF features, intensity normalization, and the zero thickness constraint at the fovea, are expected to be beneficial if applied on the healthy data as well. Future work includes learning RP specific constraints to specifically model the changes of the disease, as well as exploration of the spatial changes in each layer to better understand the disease and how it progresses.

3.3 Summary

In this chapter we presented a method for layer segmentation of macular OCT data. To do this, we leveraged machine learning to produce a probability model of each layer boundary. Specifically, an RF classifier was used, which proved to be accurate, efficient, and robust, enabling the method to be adapted for different types of data, including those from healthy, MS, and RP subjects. While all of the data used in this chapter was from a Spectralis scanner, where the images are averaged to produce better quality, the method is easily adapted for use with data from other scanners. While no results are presented in this thesis, the algorithm has been used for segmentation of data from a Zeiss Cirrus scanner, which has considerably noisier data (see Fig. 6.2 for a comparison). In Bhargava et al. [106], a comparison was done between running the RF+GC algorithm on both Spectralis and Cirrus data, with good agreement found between the scanners in many of the layers. The algorithm was minimally adapted to work for the Cirrus data, using the same graph structure and RF features. This promising result indicates that our method can be applied to data from any scanner once trained for the particular scanner.

An important problem inherent to many layer segmentation methods is that they do not work well with pathological data “out-of-the-box”, meaning adaptations are necessary to handle changes in this type of data. Our method is no different in this respect, however, we have shown that the machine learning framework utilizing both RF and the graph-theoretical model are robust, being adapted to work for RP data with only a few modifications. We anticipate that these modifications will be specific to target different diseases, especially if the retinal structure changes significantly. For data having cystoid or fluid regions within the layers, we expect that a simultaneous segmentation of both the layers and the cysts will improve the performance of the method. In preliminary work, we have shown this idea for MME data, where we combine our layer and MME segmentation methods together (see Chapter 5 for details on the MME segmentation) [107]. Since the MME cysts are small, we still need to explore the problem of having larger cysts.

Chapter 4

Registration and segmentation of longitudinal OCT data

The retina can undergo many types of changes over the course of different diseases, including both gross pathologies, such as macular edema and fluid accumulation, and smaller changes, such as RNFL and GCL thinning. OCT, with its highly detailed picture of the retina, has been used to provide clinicians with a direct measure of these effects by looking at the thickness of the retina and its layers. Often, the very small retinal changes are observed in population studies where, for example, the average thickness of specific layers are significantly different in one population versus another. On an *individual* level, however, tracking small changes in a single patient over short time intervals is significantly more challenging. In MS, the thickness of the RNFL is believed to decrease by approximately 1.0 μm per year [108], with another study showing changes of only 0.2–0.4 μm per year when averaged over the macula [15]. The ability to make accurate measurements on this scale is limited by several factors including the fixed resolution of the scanner, the accuracy of segmentation algorithms, the variable positioning of the eye, and movement of the eye during a scan.

When thickness measurements are reported by the scanner, they are usually averaged over different regions of the retina, like in the grid shown previously in Fig. 3.9. This grid is aligned only using the position of the fovea, not accounting for any rotation or scale differences, even when measuring the same subject. Since the eye is not in exactly the same position when it is scanned at different visits, any longitudinal comparisons will be made at slightly different locations on the retina. Therefore, to increase the accuracy of these comparisons, the data must first be registered so measurements are in correspondence spatially.

In Sec. 4.1, we describe a method for the 2D registration of OCT data, aligning the data in the x-y fundus plane, which will lead to longitudinal measurements reported with higher accuracy. In addition, we simultaneously correct for motion artifacts in the data, further improving the accuracy of the results.

In Sec. 4.2, we present an alternative approach to improve longitudinal thickness measures. Here, the OCT data is aligned first in 3D and then used to simultaneously segment multiple longitudinal OCT volumes in a consistent manner. By doing this, differences in the segmentations due to noise or other artifacts will be reduced, providing more confidence in the result. Since the data is registered before segmentation, the results of the segmentation can be used directly for comparison of the data. This method extends our graph-based segmentation method described in Sec. 3.1.

4.1 Combined registration and motion correction of longitudinal retinal OCT data

In this section, we present our work on both registration and motion correction of longitudinal retinal OCT data. The problem of motion, which has not been previously addressed in this thesis, is an important one to consider, especially for scanners without eye tracking technology. While modern scanners have eye and pupil tracking to address the issue of alignment and eye motion, retrospective studies are often performed on data acquired from older scanners without these improvements. Despite many acquisition protocols requiring a rescan when eye motion is detected, subtle movements often go unnoticed, which affects any longitudinal comparisons made with this data.

Prior work on registration of OCT data was previously described in Sec. 1.4.3 and, while the problem is not a common one, previous approaches have generally relied on extracting blood vessel points to do the registration [40, 41]. To address the problem of motion correction, several algorithms have been developed requiring a pair of orthogonally acquired scans [109, 110], data which is not usually acquired. Another method, which does not require multiple scans, uses a particle filter to track different features between images [111]. In work by Montuoro et al. [112], eye motion was corrected using a single scan by estimating the lateral translation between successive B-scan images by maximizing the phase correlation. The variability of this method is quite large, however, due to images having slightly different features (e.g. blood vessels) despite being close together.

As an alternative to acquiring multiple scans (orthogonal or otherwise) at the *same* visit, we propose to use data from successive longitudinal visits to simultaneously register and motion correct the data, with no restriction on the orientation of the data. With such a framework, more accurate measurements can be made when only a single scan is available at each visit. Perhaps the closest related work to our own is that of Vogl et al. [43], where their previous work on motion correction was followed by a point-based registration to align data before doing a longitudinal analysis. Since we simultaneously do these two steps, we overcome any drawbacks found in doing the correction and registration steps separately by taking advantage of the complementary information in successive scans.

4.1.1 Methods

An overview of our method is presented in Fig. 4.1. We begin with two OCT volumes acquired from the same subject at different times. A fundus projection image (FPI) is created from each volume by projecting the intensities along each A-scan (vertically) to the x-y plane. The blood vessel patterns are shown clearly in the FPI since the vessels create a shadow below their location. A set of points representing the vessels are then extracted and correspondences between the two point sets are estimated. Finally, we iterate between a point-based registration (using a rigid + scale transformation) to align the data and a lasso regression to do motion correction. With the data in alignment, any measurements made on the two scans, for instance thickness values, will be in correspondence and thus more accurate than if the data was misaligned.

Fundus projection image generation

Alignment of the OCT data relies on accurate extraction of the blood vessel points. To extract the blood vessel locations from the 3D OCT scans, we need to project the data

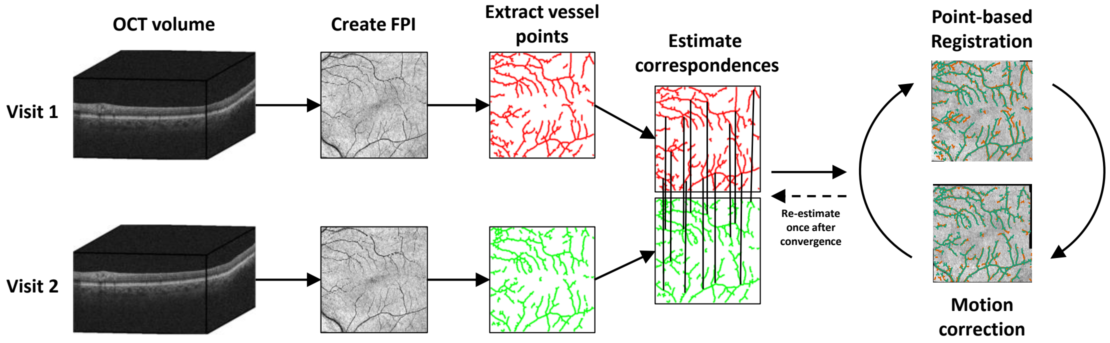


Figure 4.1: Overview of the algorithm for motion correction and registration of OCT data from two visits. The final iterations between point-based registration and motion correction are carried out until convergence. Note that correspondences are estimated for every point and only a reduced set are shown for display purposes.

to 2D, creating FPIs showing the blood vessels in the retina. This is done by averaging the intensities along each A-scan over different layers of the retina. We use the fact that the vessels produce a hyperintense area in the inner retina while their shadows produce a hypointense area in the outer retina.

A single FPI, $f(x, y)$, is generated as a linear combination of FPIs created separately on the inner and outer retina, $f_i(x, y)$ and $f_o(x, y)$, respectively, which is formed as $f(x, y) = f_o(x, y) + \alpha\,(1 - f_i(x, y))$. Specifically, f_o is the average intensity from the bottom boundary of the INL to the BrM. To create f_i, we take the pixel intensity in the 75th percentile of all values in the GCIP layer between 40% and 80% of the distance between the bottom boundary of the RNFL to the top of the INL. In this region, the blood vessels typically produce a hyper-intense reflection. Each FPI is then normalized to have intensities between 0 and 1. Note that the final FPI is created using the term $1 - f_i$, which changes the bright values of the inner retina to be compatible with the dark vessels in the outer retina. The layer boundaries used to compute each FPI are found as described in Sec. 3.1.

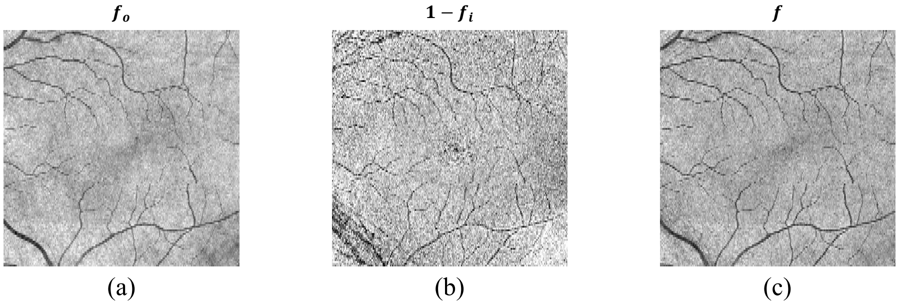


Figure 4.2: FPIs generated from (a) the outer retina, (b) the inner retina, and (c) their combined FPI.

Figure 4.2 shows an example of the FPI generation, showing f_o, $1 - f_i$, and the resulting combined FPI f. We see that while f_i has more noise, it has better contrast for some of the smaller vessels. We empirically chose a value of $\alpha = 0.5$, which provides a good balance between noise and vessel contrast.

Blood vessel segmentation and point extraction

Before registration of the data, we need to extract the blood vessel points from the FPIs, which is done from from a binary segmentation of the vessels. An overview of the process is shown in Fig. 4.3. We first resize each FPI to have a roughly isotropic size of 256×256 pixels. Next, we process the images by applying background subtraction to reduce inhomogeneity in the images, followed by using a Frangi filter to enhance the vessel structures and reduce the noise [113]. Background subtraction was done by subtracting the image with the grayscale morphological closure of the image using a disk structuring element with a radius of 7 pixels. The resulting processed image was rescaled to have intensity values between 0 and 1 and then thresholded at a value of 0.09 to create a binary image containing the vessels. Any connected components with an area of less than 15 pixels were removed to

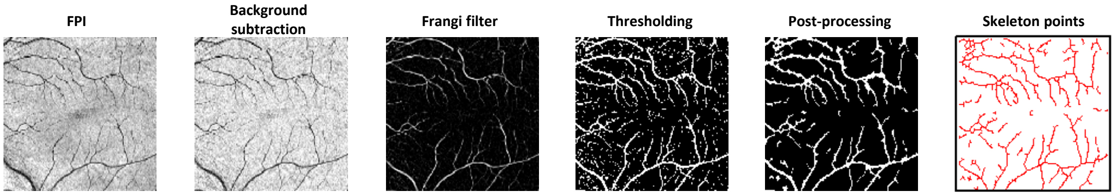


Figure 4.3: The processing pipeline for segmenting and extracting blood vessel points from the FPI.

again filter out noisy results. To create the final segmentation, we applied a morphological closing with a disk structuring element with a radius of 2 pixels to the binary image to connect small discontinuities in the vessels. Finally, the vessel points are extracted from the binary skeleton of the segmentation. These skeleton points are used instead of using the segmentation directly, which both reduces the number of points used for the registration and provides less ambiguity in finding correspondences.

Vessel registration and lasso regression

To solve the point-based registration problem, we define our two vessel point sets as $\{\mathbf{p}_i : i \in \{1, ..., n\}\}$ and $\{\mathbf{q}_i : i \in \{1, ..., n\}\}$, where n is the number of points. We require that point correspondences are known, meaning $\mathbf{p}_i$ and $\mathbf{q}_i$ correspond to the same vessel point for every i. Since the segmentation results in two sets of points with unknown correspondences, we must first estimate correspondences between them. A method commonly used to do this is the iterative closest point (ICP) algorithm, which iterates between estimating correspondences based on the closest points and a rigid point-based registration [114]. Since we expect our point sets to both have a lot of noise and have motion artifacts caused by eye movements (and thus non-rigid deformations), we instead use the coherent point drift (CPD) algorithm, a deformable point-based registration method, to generate an initial set of point correspondences [42]. CPD uses the EM algorithm to find correspondences and is quite robust to the outliers and noise we see in our data. Since these correspondences may still have errors, we run the full registration and motion correction algorithm twice, estimating correspondences using CPD at the start of both iterations.

While the CPD result aligns the data well, its non-rigid transformation violates the assumption that the transformation between OCT scans acquired from the same subject should be rigid, with an additional scale component to account for camera position. While the deformations introduced by eye motion are non-rigid, they are modeled in a separate way as described later. The rigid plus scale transformation allows rotation, translation, and scaling of the points. A point $\mathbf{p}_i = (p_{i,x},\ p_{i,y})^\intercal$ from one FPI is related to a point $\mathbf{q}_i = (q_{i,x},\ q_{i,y})^\intercal$ through the relation $\mathbf{p}_i = s\mathrm{R}\mathbf{q}_i + \mathbf{t}$ where s is the scale, R is a 2×2 rotation matrix parameterized by a single rotation angle, and $\mathbf{t}$ is the translation.

To model the motion correction problem, we assume eye motion results in the displacement of a B-scan's position relative to the previous one. This assumption is appropriate since images are acquired one at a time, in raster order. We denote the displacement of B-scan image j as $\boldsymbol{\gamma}_j$ for an image in the first OCT volume and $\boldsymbol{\beta}_j$ for an image in the second. Note that, without loss of generality, we assume the same number of B-scan images for each volume, and thus $j \in \{1, ..., n_B\}$ for both, where n_B is the number of B-scans. As we are concerned with motion in both the x and y plane (the axes of the FPI), $\boldsymbol{\gamma}_j$ and $\boldsymbol{\beta}_j$ each have an x and y component. Such a motion is shown in Fig. 4.4.

To combine the registration and motion correction problems together, we minimize

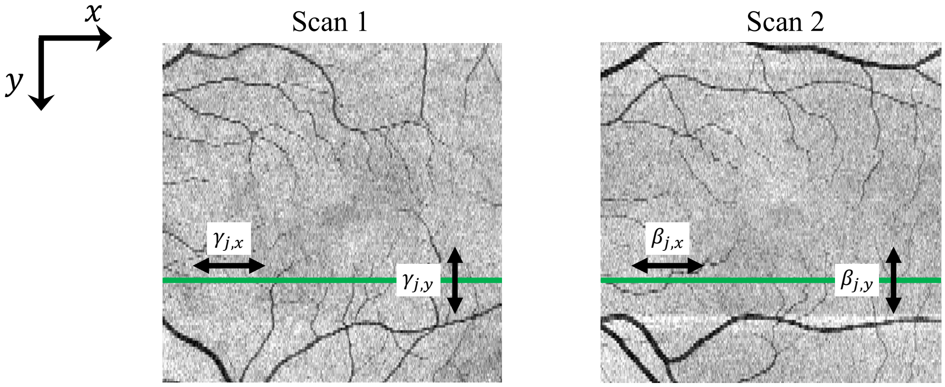


Figure 4.4: The assumed motion model for each scan. The green line represents the B-scan location, with each B-scan located at a different horizontal position.

the cost function

$$C = \sum_{i=1}^{n} \left\| \left(\mathbf{p}_i + \sum_{j=1}^{\lfloor p_{i,y} \rfloor} \boldsymbol{\gamma}_j \right) - s\mathrm{R} \left(\mathbf{q}_i + \sum_{j=1}^{\lfloor q_{i,y} \rfloor} \boldsymbol{\beta}_j \right) - \mathbf{t} \right\|^2 + \lambda \sum_{j=1}^{n_B} \left(|\boldsymbol{\gamma}_j| + |\boldsymbol{\beta}_j| \right). \tag{4.1}$$

The term $\sum_{j=1}^{\lfloor p_{i,y} \rfloor} \boldsymbol{\gamma}_j$ represents the overall displacement of vessel point $\mathbf{p}_i$ based on the cumulative displacement of each B-scan up to that point. Since extracted vessel points may not lie exactly on a B-scan, due to the resizing of the FPI when segmenting the points, we use the floor operator $\lfloor \cdot \rfloor$ in the limit of the summation. The coefficient λ encourages displacement values to be zero as it gets larger, since the L1 norm induces sparsity. It is important to note that if we use a global displacement model instead of a cumulative one, the displacements would no longer be sparse. Sparsity is a desired feature for two reasons. Firstly, eye motion tends to be abrupt during a scan, with infrequent, large displacements. Secondly, the estimation of displacements for both scans is rather ill-posed. A displacement in one image can be counteracted by an opposite displacement at the same location in the other image (e.g., if $\boldsymbol{\gamma}_j = -\boldsymbol{\beta}_j$ and s and R are close to identity, they are not identifiable).

If no displacement truly exists at that point (e.g. if $\boldsymbol{\gamma}_j = \boldsymbol{\beta}_j = 0$), the sparsity constraint will encourage both coefficients to be zero.

The problem of minimizing Eq. 4.1 is solved by iterating between solving for the transformation parameters and then solving for the displacements until convergence, which usually occurs within 20 iterations. We use a value of $\lambda = 1$, empirically set with a preference to keep many of the coefficients set to zero. By careful inspection of Eq. 4.1, we see that by fixing the displacements and solving for s, R, and $\mathbf{t}$, we have a simple least-squares point-based registration problem. The second term can be ignored as it does not depend on the transformation parameters. In other words, we minimize the reformulated function

$$\tilde{C} = \sum_{i=1}^{n} \left\| \tilde{\mathbf{p}}_i - s\mathrm{R}\tilde{\mathbf{q}}_i - \mathbf{t} \right\|^2 , \tag{4.2}$$

where $\tilde{\mathbf{p}}_i$ and $\tilde{\mathbf{q}}_i$ are the motion corrected vessel points. This minimization is solved in closed form, e.g., using singular value decomposition or Procrustes alignment [114]. Next, by fixing the transformation parameters, Eq. 4.1 can be rearranged such that the displacements are estimated by solving a lasso, or L1 regularized regression problem [115]. Specifically, the first term can be rewritten as

$$\sum_{i=1}^{n} \left\| \left(\mathbf{p}_i + \sum_{j=1}^{\lfloor \mathbf{p}_i \rfloor} \boldsymbol{\gamma}_j \right) - s\mathrm{R} \left(\mathbf{q}_i + \sum_{j=1}^{\lfloor \mathbf{q}_i \rfloor} \boldsymbol{\beta}_j \right) - \mathbf{t} \right\|^2 = \left\| \mathbf{y} - \mathrm{X}\boldsymbol{\alpha} \right\|^2 \tag{4.3}$$

where $\mathbf{y} = \mathbf{p}_i - s\mathrm{R}\mathbf{q}_i - \mathbf{t}$, $\mathrm{X} = -\begin{bmatrix} \mathrm{I}_2 \otimes \mathrm{X}_1 & -s\mathrm{R} \otimes \mathrm{X}_2 \end{bmatrix}$, $\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\gamma}^\intercal & \boldsymbol{\beta}^\intercal \end{bmatrix}^\intercal$, and the x and y components of vectors are stacked such that $\mathbf{y} = \begin{bmatrix} \mathbf{y}_x^\intercal & \mathbf{y}_y^\intercal \end{bmatrix}^\intercal$. I_2 is a 2×2 identity matrix while the design matrices X_1 and X_2 are $n \times n_B$ with a structure such that the first $\lfloor \mathbf{p}_i \rfloor$

columns of row i have a value of 1 and the remaining columns have a value of 0. Thus, Eq. 4.1 reduces to

$$C = \|\mathbf{y} - \mathrm{X}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1, \tag{4.4}$$

the form for a lasso regression, which we solve using the Glmnet software package [115, 116].

As a final note, the problem of estimating displacements in both images is still ill-posed since we do not know whether to use a displacement of either $\boldsymbol{\gamma}_j$ or $-\boldsymbol{\beta}_j$, when the other displacement set to zero (and s and R are again close to identity). Assuming there was true motion in the first volume, we could incorrectly introduce motion in the second volume to produce a good registration, minimizing Eq. 4.1. To alleviate this problem, we use the phase correlation motion correction (PCMC) method of [112] to provide both an initialization for the displacements, and to use as weights in the lasso regression. While PCMC can suffer from drift errors, it is useful for broadly identifying which B-scans have moved. In brief, it estimates the displacement of each B-scan image relative to the three proceeding images by finding the translation along the x-axis which maximizes the phase correlation between the images.

Since many of the small resulting displacements output by PCMC are unreliable due to noise, we modify its output by ignoring displacement estimates smaller than 3 pixels (≈ 35 μm). We denote these estimates (which are in the x-direction only) for each B-scan image as $\hat{\gamma}_{j,x}$ and $\hat{\beta}_{j,x}$. These values are then used to initialize our iterative registration algorithm described previously, resulting in a more accurate Procrustes registration at the first iteration.

We additionally incorporate the estimates $\hat{\gamma}_{j,x}$ and $\hat{\beta}_{j,x}$ as weights in the lasso regression. The final result of our method given the PCMC estimates is computed in the same way as described before, only using a different set of weights for each displacement term in Eq. 4.1. Specifically, we modify the second term of this equation to be

$$\lambda \sum_{j=1}^{n_B} \left(\nu_j |\gamma_{j,x}| + |\gamma_{j,y}| + \eta_j |\beta_{j,x}| + |\beta_{j,y}| \right), \tag{4.5}$$

with Gaussian-shaped weights $\nu_j = \exp - \frac{\hat{\gamma}_{j,x}^2}{2\sigma^2}$ and $\eta_j = \exp - \frac{\hat{\beta}_{j,x}^2}{2\sigma^2}$ and $\sigma = 10$. Larger motion estimates produce smaller weights, thereby acting to reduce the sparsity constraint when we have confident initial estimates.

4.1.2 Experiments and Results

To examine both the accuracy and consistency of our method, we looked at data from 26 healthy control subjects. Both eyes of all subjects were scanned twice, with the second scan occurring within an hour of the first. In total, 42 of the 52 possible pairs of images were used (considering both eyes), with some pairs not included due to a missing acquisition or poor image quality. Macular OCT data was acquired using a Zeiss Cirrus scanner (Carl Zeiss Meditec Inc., Dublin, CA), with each scan covering a 6 × 6 mm area centered at the fovea. Each scan has 1024 pixels per A-scan, 512 A-scans per B-scan, and 128 B-scans. For each pair of scans, landmark points were manually selected on FPIs at corresponding vessel bifurcations and corners to generate ground truth data for exploring the accuracy of the registration. An average of 37 points were selected from each pair of images, with a range of 18 to 45 points, depending on the complexity of the vessel pattern in each eye.

Table 4.1: Root mean square error (µm) of the manually selected blood vessel landmark positions after registration using different methods. Standard deviations are in parentheses.

Fovea alignment	**Rigid registration**	**Rigid reg. + motion correction**	**Proposed using manual points**
59.0 (30.5)	38.3 (19.3)	29.7 (10.8)	19.2 (3.3)

First, we looked at the accuracy of our method. To do this, we compared the average root mean square error (RMSE) of the manually selected landmark points after registration. Three methods of registering the data were compared: 1) alignment to the center of the fovea,[1] 2) the full proposed method without motion correction (registration only), and 3) the proposed method (registration plus motion correction). We also ran our method after replacing the automatically segmented vessel points $\mathbf{p}_i$ and $\mathbf{q}_i$ with the manually selected landmark points. This result provides both an indicator of the best possible performance for our method and also an overall estimate for how accurately the landmark points can be localized in an FPI. The results are shown in Table 4.1. Our method showed a significant improvement in accuracy as compared to both of the other methods when using a paired t-test ($p < 0.01$). Differences between errors when using the segmented points versus the manual points are due to the accuracy of both the vessel segmentation and the correspondence estimation using CPD. An example showing the landmark point alignment after registration using the four methods is shown in Fig. 4.5.

An important application of our method is the measurement of retinal thicknesses in longitudinal data. By segmenting the data first and then applying the registration and motion correction to the resulting thickness maps, corresponding areas will align better in

[1] The fovea center point is computed as the smallest vertical distance between the ILM and BrM.

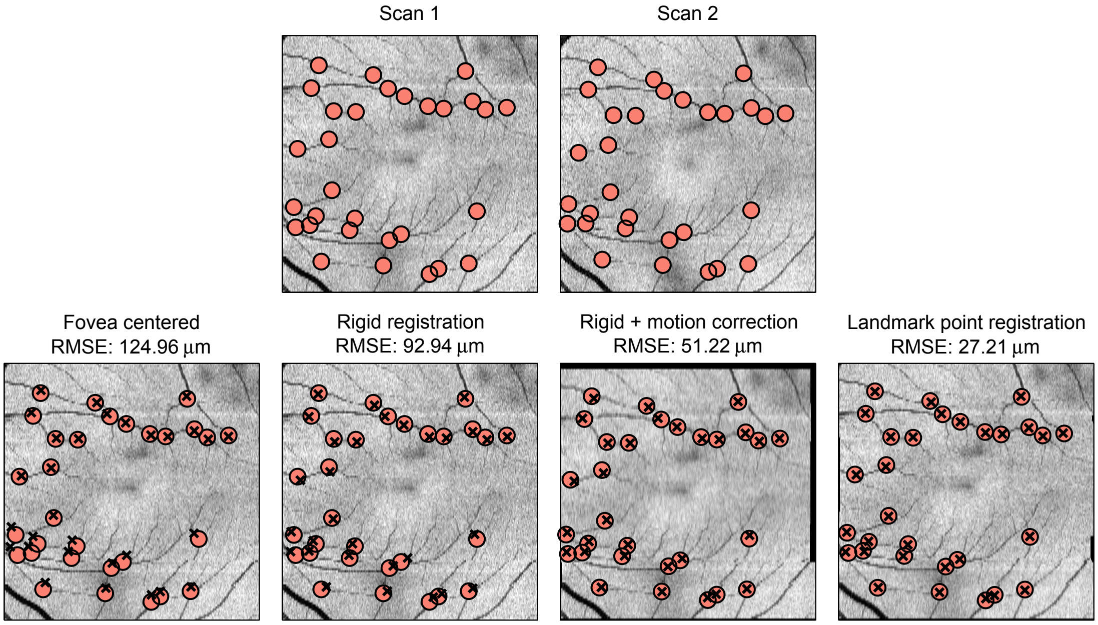


Figure 4.5: (Top row) FPIs from two successive scans with corresponding manually selected landmark points overlaid. (Bottom row) Landmark points from Scan 2 overlaid on Scan 1 after registration using different methods. Manual points are marked as red circles and registered points are marked as black ×'s.

the thickness maps, thereby improving the accuracy of any longitudinal comparisons. In our dataset, the time between scans was minimal, so we expect the difference in thickness between the scans to be close to zero. Any differences found are therefore due to either segmentation errors, or alignment errors. Since we cannot reduce the segmentation errors, which are mainly due to noise, we look to show improved results by only improving the alignment.

We applied the automated segmentation algorithm method described in Sec. 3.1 to all of the data, segmenting a total of eight layers. Looking only at the total retina thickness, we computed the average value within a 5 × 5 mm area centered at the fovea for each scan. The results are shown in Table 4.2, where we computed the average signed and unsigned change in thickness between the two successive scans. The proposed method was significantly

Table 4.2: Average signed and unsigned difference in total retinal thickness (μm). Standard deviations are in parentheses.

	Fovea alignment	**Rigid registration**	**Rigid reg. + motion correction**
Signed	−0.10 (0.82)	−0.08 (0.55)	−0.14 (0.46)
Unsigned	0.67 (0.48)	0.45 (0.34)	0.37 (0.31)

better (closer to 0) than the other methods when looking at the unsigned values ($p < 0.05$). Note that we did not compare to the result of using the manual landmark points in this experiment since the sparse nature of these points means we are not able to accurately localize B-scan motion (there are fewer landmark points than B-scan images). Thus, the thickness maps are likely to be incorrectly registered in areas where there are no landmark points. Looking at individual layers, we saw no significant differences when comparing the registration with and without the motion correction, but we did see thickness differences that were significantly closer to zero in the RNFL and GCIP layers when comparing the two registration methods versus fovea alignment ($p < 0.05$).

Finally, in Fig. 4.6, we show FPIs before and after registration for two subjects, with each row showing a different subject. The uncorrected FPIs from the first and second temporal scan are shown in Figs. 4.6(a) and 4.6(b), respectively. The same images are shown in Figs. 4.6(c) and 4.6(d) after motion correction and registration. Note that motion correction is applied to both scans, while registration is only applied to the second. Figure 4.6(e) shows the segmented boundaries from each scan after motion correction and registration overlaid on the first corrected FPI only. Motion artifacts are highlighted by arrows and have been corrected after running our algorithm.

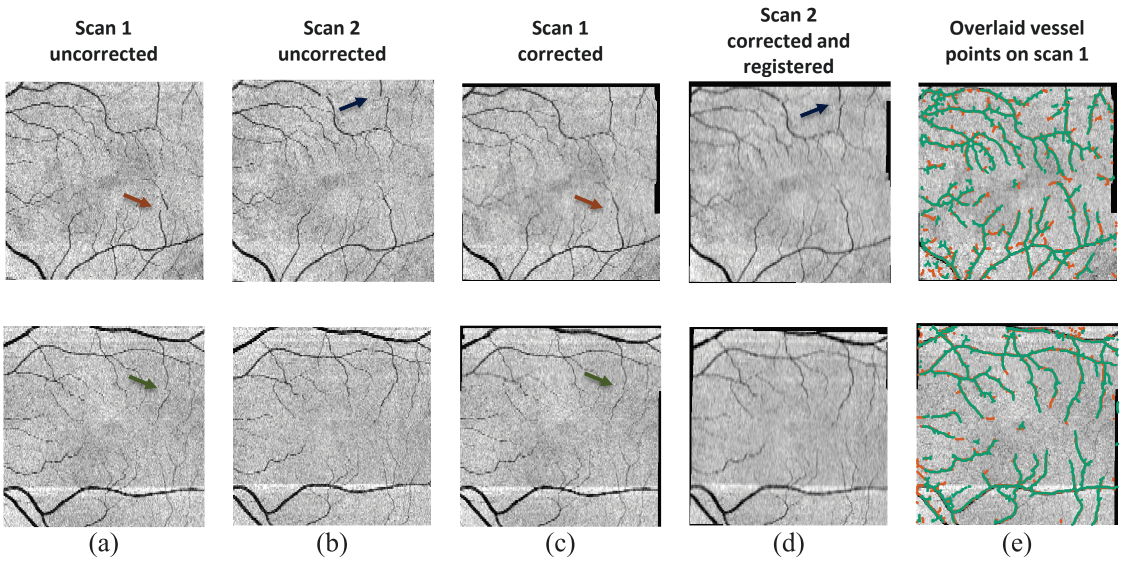


Figure 4.6: FPIs (a, b) before and (c, d) after motion correction and registration. Motion artifacts are indicated by arrows. In (e), segmented blood vessel points from the first (red) and second (green) scans are overlaid.

4.1.3 Conclusions and future work

We have developed a method for simultaneous registration and motion correction of longitudinal macular OCT data. Including motion correction improves the accuracy of registering longitudinal data, enabling more consistent thickness measurements between scans. While the motion correction did not show significant improvements when looking at specific layer thicknesses, the registration did prove to be important when compared against data aligned only to the fovea. The cohort of data included in our experiments was a healthy one with a minimal amount of motion artifacts. If scans with motion were explicitly included, we expect to see more improvement within each layer when using our method.

A critical step for the registration of OCT data is the segmentation and extraction of blood vessel points. Currently, this step leads to many points without correspondences, depending on the contrast of the vessels in each FPI. Thus, in future work, we will continue work on improving the segmentation of the vessels. There are also several parameters in the method, including the sparsity coefficient λ, and weight parameter σ, that need to be estimated in a more rigorous cross-validation framework. While the value of λ seems to be rather sensitive for accurately estimating motion, the value of σ is much more robust. Finally, we hope to look at simulated deformation experiments to further validate our methodology, in addition to running the method on data from non-healthy patients where motion may be more severe.

4.2 Longitudinal graph-based segmentation of macular OCT data

Many automated algorithms have been developed for segmentation of retinal layers in OCT data, but none incorporate longitudinal consistency. The accuracy of these algorithms is generally >4 μm, making them insensitive to detecting small temporal changes of the type seen in diseases like MS. Other factors, including blood vessel shadowing, scan misalignment, low SNR, the appearance of Henle's fiber layer [117], and ambiguities in layer boundary positions in the deeper retina [118], can also contribute to errors when comparing two scans acquired at different times. By enforcing consistency in the segmentation of longitudinal data, we can eliminate or reduce many of these sources of error.

In this work, we extend the RF+GC segmentation framework described in Sec. 3.1 to handle longitudinal data. We do this by connecting the separate 3D graphs constructed on the volume of each visit and carry out a simultaneous 4D segmentation. This idea was previously hypothesized as an application by Yin et al. [83], and used in practice for the segmentation of lung data by Petersen et al. [119]. The temporally connected graph edges serve to regularize the result over time. However, this simple connection of the graphs is not trivial because voxel/vertex correspondences over time are unknown. Therefore, the data must be registered before segmentation. To do this, we do not use traditional 3D image registration methods because of the highly anisotropic nature of OCT data, with through-plane resolution up to 25 times worse than the in-plane direction. Instead, we take advantage of the consistent geometry of the data, doing the registration in two steps, first aligning in the axial direction, and then in the transverse and through-plane directions.

4.2.1 Methods

Our method includes both registration and segmentation of longitudinal OCT data including multiple volumes from the same subject acquired at various times. The method itself simultaneously segments an arbitrary number of scans, with the only limitation being the increase in computational cost as more data is used.

Initial boundary segmentation

As a first step, we run the graph-based segmentation algorithm described in Sec. 3.1 on each volume cross-sectionally. This serves two purposes. First, it allows us to generate reference surfaces, which will be useful for registering the data to a common boundary. Second, it allows the final longitudinal graph segmentation to be made very efficient by reducing the search space for the graph algorithm.

Since running the full graph-based segmentation can be computationally expensive, we run it on highly downsampled data, producing a low resolution segmentation (LRS). To

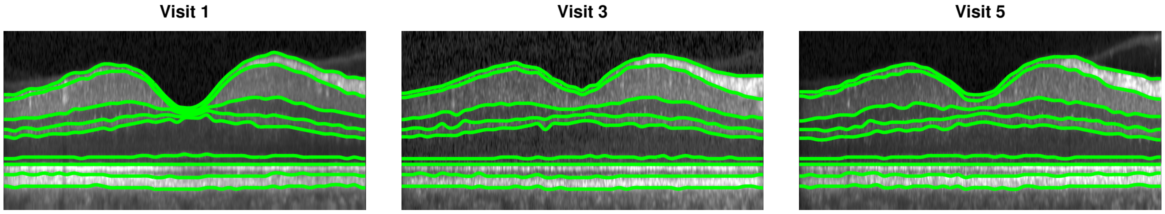


Figure 4.7: B-scan images from the same subject at three different visits after alignment to the IS-OS boundary (the third boundary from the bottom). Boundaries estimated from the LRS are overlaid.

get the LRS, we downsample the data by a factor of 20 in the x direction, thus making the LRS quick to compute while providing sufficient accuracy for later use. From the LRS, for each time point t, we end up with nine boundary estimates for each of the eight layers.

Axial alignment

To align the data in the axial, or z direction. We flatten the data to the IS-OS boundary, estimated by the LRS, for each visit. We use the IS-OS boundary since it is accurately estimated, even in the LRS, due to its sharp intensity gradient. After flattening, we assume that each longitudinal scan is aligned in the axial direction, i.e. $z = 1$ corresponds to the same vertical position for all t. We show an example of this axial alignment step, in addition to the LRS, in Fig. 4.7.

Fundus alignment by vessel registration

After axial alignment, the data is aligned in one out of the three dimensions of the data. The other two dimensions are aligned together by registering fundus projection images (FPIs) in a similar fashion to Sec. 4.1. This work uses slightly different methods for both creating the FPIs and for registration, however.

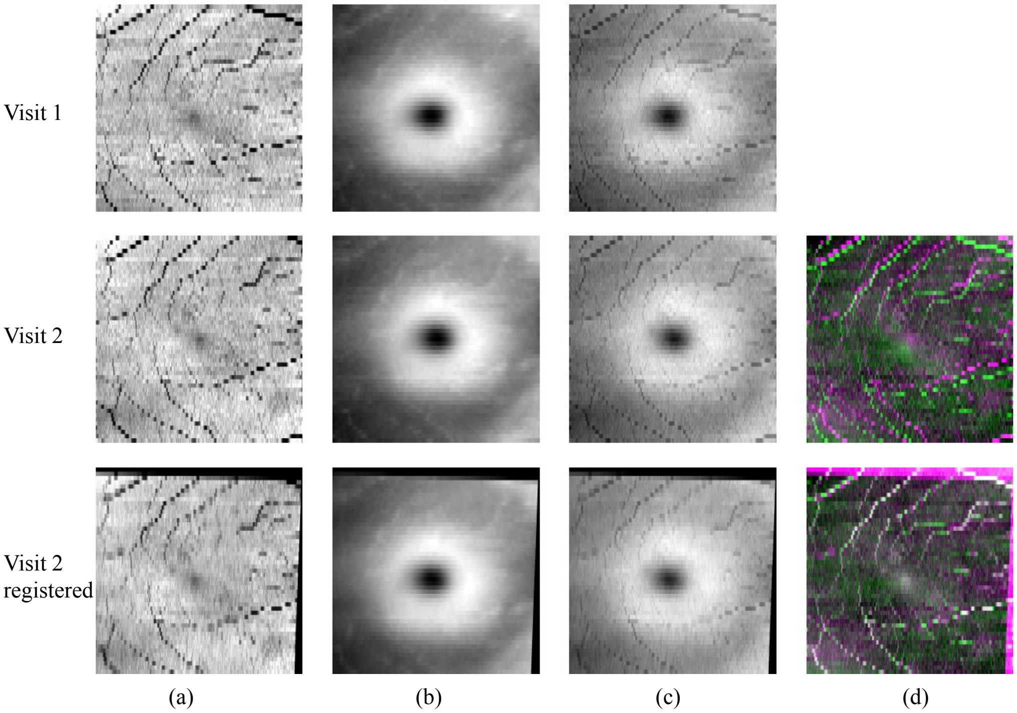


Figure 4.8: Fundus registration of 2 visits. Columns (a), (b), and (c) show the vessel shadow projection images $S(x, y, t)$, the retina thickness images $T(x, y, t)$, and the combined FPIs, respectively. In (d), we show color overlay images before and after registration. White indicates the improved alignment of the vessels.

The FPIs are constructed as a combination of total retina thickness maps, $T(x, y, t)$, and vessel shadow projections, $S(x, y, t)$, both generated using the LRS. The thickness maps are computed from the difference between the top and bottom boundary in the LRS (the ILM and BrM), while the vessel maps are computed from the sum of intensities in the OCT volume from the bottom boundary to 15 pixels above this boundary. After normalizing the intensity range of both $S(x, y, t)$ and $T(x, y, t)$ to $[0, 1]$, we get the final FPI as $F(x, y, t) = S(x, y, t) + \gamma T(x, y, t)$. Figure 4.8(a–c) shows an example of S, T, and F at two different visits from a single subject.

To register the FPIs together, we use an intensity-based approach, which is different than the point-based approach used previously. Intensity-based registration has the advantage of not needing to extract the blood vessel points, however, it is more sensitive to the initialization of the images, especially since the vessels are very thin. Using the thickness maps $T(x, y, t)$ are meant to counteract this sensitivity by providing a slowly varying structure for a wider convergence basin of registration.

The FPIs at each visit are registered pairwise to the baseline FPI using an affine transformation and the mutual information similarity measure. Figure 4.8(d) shows a result with overlaid FPIs from two visits before and after registration. Note that we empirically use a value of $\gamma = 2$ when generating the FPIs, which exhibits good registration performance.

Longitudinal graph-based segmentation

Given both the axial alignment and fundus registration, we know the transformation between voxels in any pair of OCT volumes. Since the scans are only aligned based on a single surface in the axial direction (the IS-OS), we assume that any differences in the remaining surfaces are either due to noise, or physiological changes, perhaps due to disease. By regularizing any changes between the segmentations at separate visits, we hope to smooth out the noise while maintaining any real changes that occur. We segment the final boundaries by adapting the graph-based method described in Sec. 3.1, using the same RF probability maps as input into a modified longitudinal graph.

For each of the longitudinal scans, we construct a separate graph on the image data following the methods outlined in Sec. 2.2. Solving the segmentation problem on each of these graphs separately gives us the traditional cross-sectional graph (CSG) segmentation. Connecting the graphs from adjacent visits will allow us to do a simultaneous longitudinal graph (LG) segmentation.

To extend the graph segmentation algorithm to longitudinal data, we add additional edges between the graphs at adjacent visits. Ideally, we could simply add bidirectional, non-zero weighted, horizontal edges between corresponding voxels at consecutive time points. These edges act to provide a regularization force which penalizes differences in the segmentation result over time. Such an edge is similar to the smoothness constraint described in Sec. 2.2.4 and shown in Fig. 2.5(c). A final segmentation that is identical at each visit would not incur any smoothness penalty since it would not cut any of the added edges.

Unfortunately, due to the large distance between B-scan images in our dataset, the voxels (or graph vertices) will not be in direct correspondence after registration (see Fig. 4.9, where we see that after registration, each B-scan, or horizontal line, will not align perfectly between datasets). Therefore, we cannot simply add a single edge between vertices to directly connect the longitudinal graphs. Connecting the vertices of one volume with the *nearest* vertex in the next volume is also not satisfactory due to the spacing. Instead, given that the voxel $\mathbf{x} = (x, y, z)$ at time t_1 corresponds to a (non-integer valued) point $\mathbf{x}' = (x', y', z')$ at time t_2 (after the FPI registration), we connect vertex $v(\mathbf{x}, t_1)$ to the four nearest vertices at t_2 found from $floor(x')$, $ceil(x')$, $floor(y')$, and $ceil(y')$ (see Fig. 4.9). The weight of these edges is set to a value inversely proportional to the distance between the vertices such that the four weights add up to w, which we empirically set $w = 0.1$.

Since the physical locations of the vertices in each graph are maintained, the segmentation is carried out in the native space of each volume, instead of on an interpolated

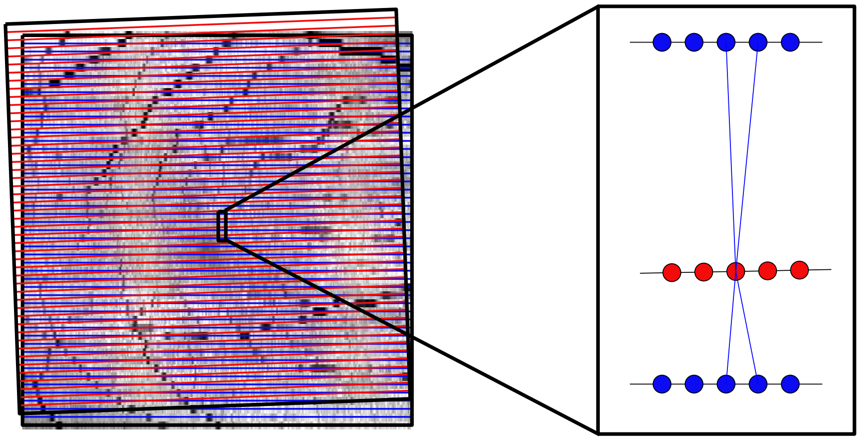


Figure 4.9: On the left are the outlines of two fundus images after alignment, with B-scan locations represented by horizontal lines. After registration, the vertices no longer have direct correspondences for connecting the graphs. Therefore, we connect vertices at one visit (red) to the four nearest vertices in the next visit (blue). The weighting of each edge is inversely proportional to the distance, encouraging the final segmentation to look more similar to closer vertices.

grid. To interpolate the grid would require a 3D interpolation of the RF probability output, which is undesirable due to the large spacing between images. Using the described graph with the added temporal edges between scans at visits t_i and t_{i+1} $\forall i \in \{1, \ldots, N_t - 1\}$, we are able to simultaneously solve for the segmentation at all visits of the patient using a single minimum s-t cut [81].

As a final note, we substantially reduce the memory and time requirements of the 4D LG segmentation problem by masking out pixels that are further than 3 pixels from the LRS boundaries. Masking out these pixels both reduces the search space for the algorithm and substantially reduces the size of the longitudinal graph. We also use the same mask for all visits by taking the union of each; this reduces the possibility that a segmentation error in the LRS will affect the final result.

4.2.2 Experiments and Results

For this study, we used macular OCT data (20° × 20°) acquired from a Heidelberg Spectralis scanner, which covered approximately a 6×6 mm area centered at the fovea. The volumetric dimensions of each scan are $1024 \times 49 \times 496$ voxels in the x, y, and z directions. The voxel spacing is approximately $5.8 \times 123 \times 3.9$ µm.

We explored the performance of our LG segmentation on two cohorts. The first included scans of 13 eyes from 7 healthy control (HC) subjects, with all scans repeated one year later or sooner. The second data pool contained 34 eyes of 17 patients with MS, each scanned between 3 and 5 times at intervals of 3 to 12 months. We looked at the total retinal thickness as measured from the ILM to the BrM to explore the overall effect of the algorithm. The thickness values were averaged within a 5×5 mm square centered at the fovea, where the center of the fovea was defined as the position with the smallest total thickness near the center of the volume. Results of running both the CSG and LG segmentation algorithms on both cohorts are shown in Fig. 4.10 and Fig. 4.11, where we see the change in total retina thickness relative to the baseline scan.

In the control population (Fig. 4.10), the longitudinal regularization reduces the standard deviation of the change in thickness from 1.35 to 0.73 or by 46%. Since this is a control population, we do not expect the measurements to change much over 1 year, and therefore, the LG produces a better result. We must be careful to note, however, that setting the regularization parameter ω to a large value encourages identical segmentations. As a result, this experiment must be examined in conjunction with looking at longitudinal results on the MS cohort, where we expect to see changes due to disease.

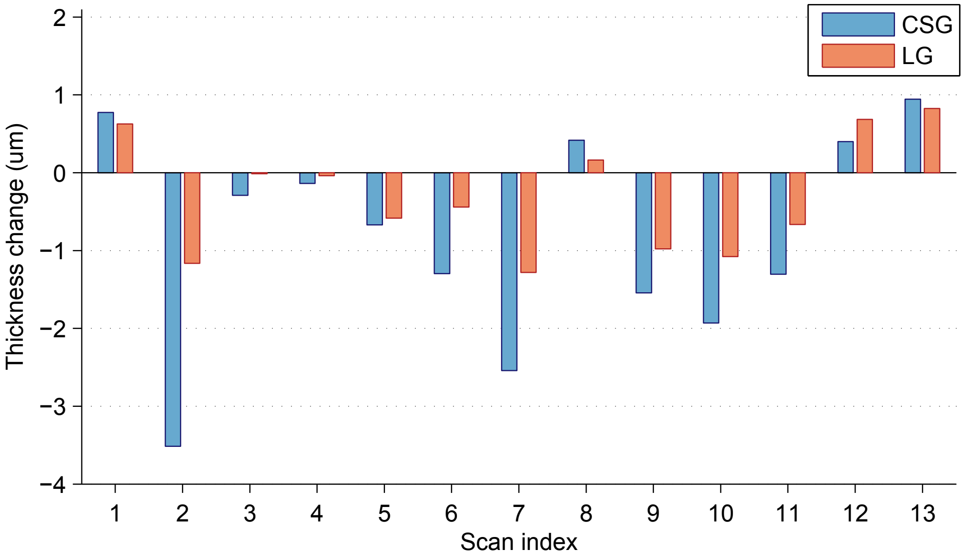


Figure 4.10: Plot of the change in total retina thickness as compared to the thickness at the baseline visit for 13 HC scans using both the CSG and LG segmentation methods. Repeat scans were acquired approximately 12 months after the baseline.

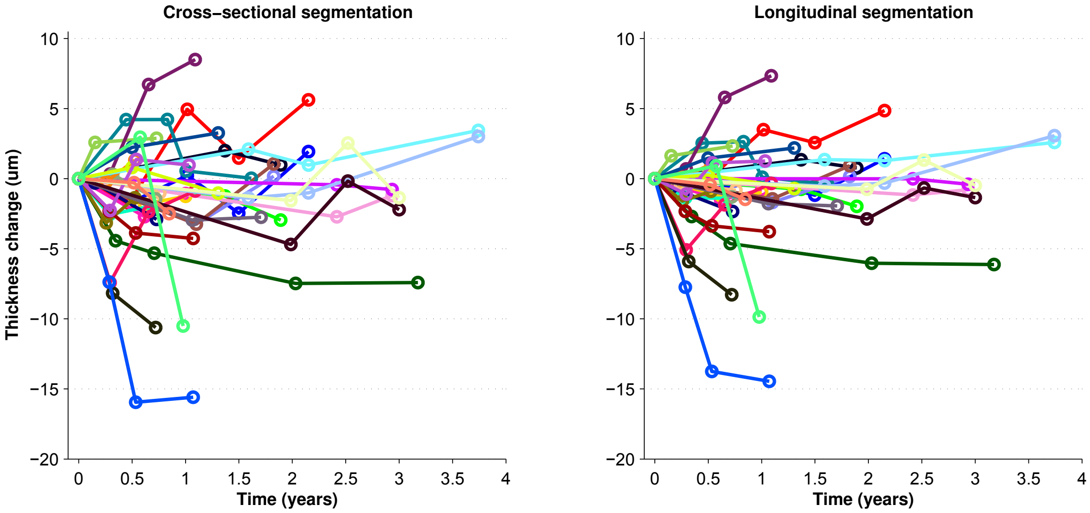
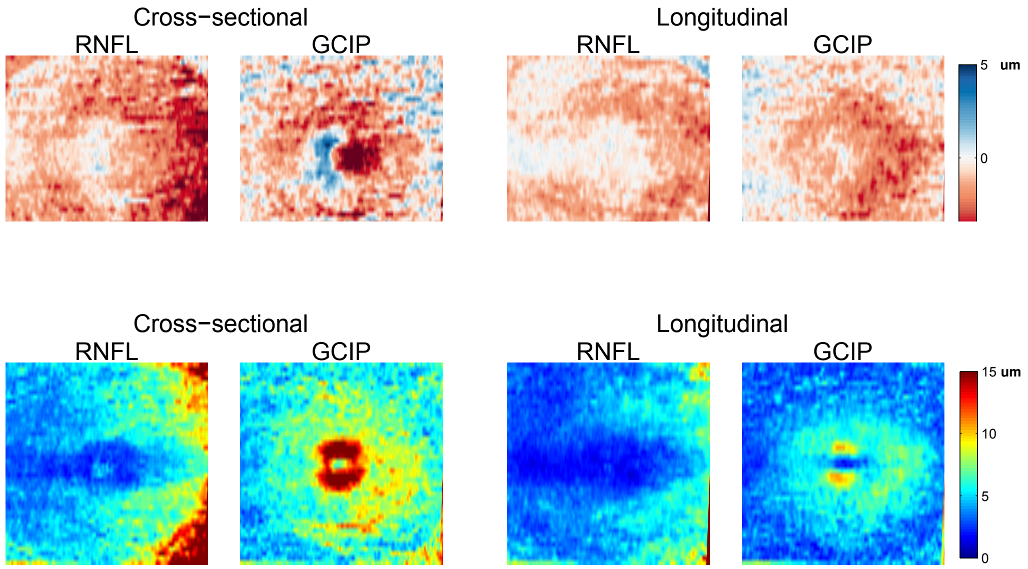


Figure 4.11: Plot of the change in total retina thickness as compared to the thickness at the baseline visit for 34 MS scans. Results for the CSG and LG segmentation methods are shown, with each color representing a single subject.

For the MS results, shown in Fig. 4.11, we see a broad smoothing of the results using the LG method as compared to the CSG. In other words, the trends for individual thickness trajectories are the same, but the noise is reduced. Since the changes are not trending towards zero, as we saw with the control data, this suggests that true changes are not being removed and we are not over regularizing. One way to measure the increased consistency of the LG segmentation is to look at the residual error of fitting a straight line each subjects results. In this case, the root mean square fitting error is reduced from an average of 1.83 μm to 1.13 μm ($p < 10^{-10}$ using a two-sided paired t-test). Our results also agree with the literature that the overall changes in retinal thickness due to MS are small [15, 108], but a larger cohort of data will need to be analyzed.

In the previous experiment, we looked at the average thickness over the entire macular area. Next, we explored how individual retinal layers change in a localized manner. Figure 4.12 shows the average change in the thickness of the RNFL and GCIP layers over one year (relative to the baseline scan) in the MS cohort for both the CSG and LG methods. We only show these layers since they are known to be the most affected by MS [15] and have the most atrophy in our dataset. We see that the magnitude of the changes in the LG method are smaller than those using the CSG, but the standard deviation is also much smaller. This result allows us to have more confidence in localizing changes using the longitudinal method. Also note that the GCIP shows an area of increased thickness near the fovea using the CSG method which is removed using the LG method; a result which is more consistent with what is known about changes due to MS. This result also shows us that global measures of thickness change (e.g. averaging over the entire retina) are less sensitive



(b) Standard deviation of change

Figure 4.12: Comparison of the mean and standard deviation of the change in the RNFL and GCIP layer thicknesses using the CSG and LG methods on the MS cohort of scans that have one year of data (24 scans).

to finding longitudinal changes since changes tends to be localized to specific areas. By looking at a specific regions of the retina, we get a better picture of how it is changing over time.

Finally, in Fig. 4.13 we show a comparison of running the CSG and LG algorithms. We see the negative effect that the disappearance of Henle's fiber layer has on the segmentation of the OPL and ONL. The LG regularization maintains consistency over time, even when the boundaries are not clearly visible. Note that this boundary appears more clearly in the other longitudinal scans, which are not shown.

4.2.3 Conclusions

We have extended our previously developed graph-based technique for the simultaneous segmentation of longitudinal OCT data of the retina. We register the data in

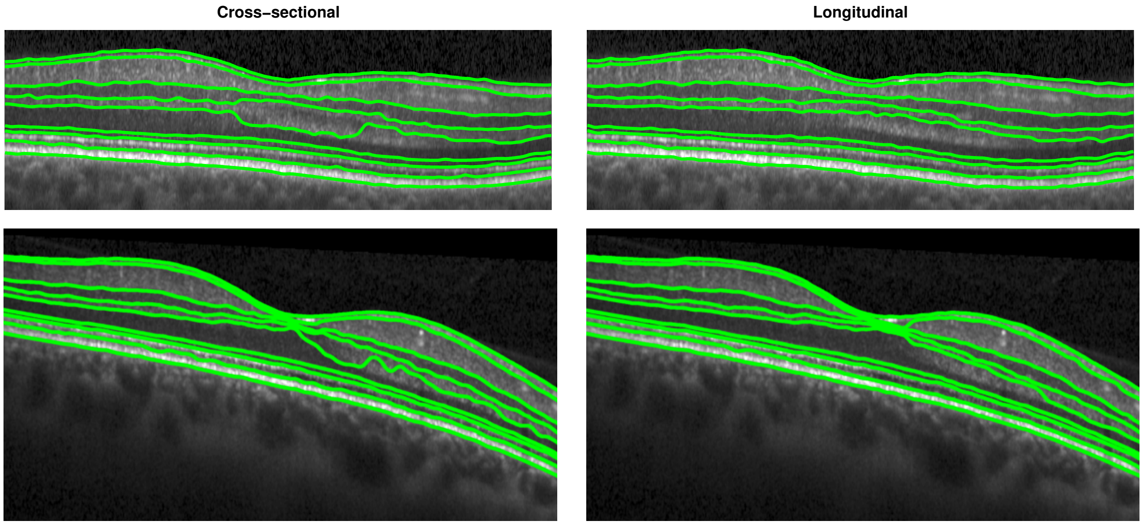


Figure 4.13: Improvement in the ONL segmentation for the two cases (different subjects) of the appearance and disappearance of Henle's fiber layer, with CSG result on the left and LG result on the right. Information from multiple visits (not shown) enforces consistency over time.

two steps, first axially, then in the fundus plane, and use the resulting correspondences to connect the graphs. Longitudinal segmentation relies on accurate alignment, so if either of our alignment steps fails, the segmentation will be incorrect. However, our alignment method was quite robust even for low quality data.

The most important parameter of this method is the regularization parameter ω. If it is set too large then changes in a subject with a large amount of atrophy or edema will not be found. If it is set too small then the results will not be different than running the method cross-sectionally on each dataset. It would be beneficial to allow ω to vary both spatially, with a larger value for more stable regions and layers, and temporally, making it inversely proportional to the time between scans. It would also be interesting to modify the weight based on the output of the RF classifier, with larger weights when connected vertices have a large RF probability.

4.3 Summary

In this chapter, we have described two methods for the registration of OCT data, with the goal of both methods being to produce more stable and accurate longitudinal thickness measurements. Beginning in Sec. 4.1, our first method aligns the data in 2D. By doing this, the registration result can be applied directly to the 2D thickness maps generated from a segmentation of the data. Comparing thickness measurements after registration produces more stable results than simply aligning the data at the fovea, which is typically done by most scanner software. Using longitudinal data also enables us to simultaneously do motion correction, which is a problem when looking at data acquired on older scanners without eye tracking technology. Since the structure of the blood vessels is unchanged between visits, we are able to correct for any motion by aligning corresponding points in the data. Overall, we showed that including motion correction produces more accurate registration results, and also more stable thickness results, by looking at a cohort of data which was scanned within a close time period (and thus, no change is expected). In the future, we will need to evaluate this method on MS data to see if thickness trends in this disease become more apparent or stable.

Since the segmentation results used in the experiments for the registration method developed in Sec. 4.1 were generated independently for each volume, we next extended our graph-based segmentation method in Sec. 4.2 to work simultaneously on longitudinal data. Thus, we were able to improve the consistency of the results using registration, as before, and also by generating a consistent segmentation between scans. It should be noted that the two methods are not incompatible with each other. The 2D segmentation of Sec. 4.1

could be used to register the data before running the segmentation. The main difference between the two registration components is that one is a point-based method and one is an intensity-based method. When using data from a Zeiss Cirrus scanner, there are more B-scan images and so the FPIs have better quality. As a result, the blood vessel points are more easily extracted. With the Heidelberg Spectralis data, the FPI is generated using fewer B-scans, meaning the quality of the blood vessels in each image is very poor. Intensity-based approaches tend to work better in this case. Going back to the segmentation result, the added regularization introduced by the 4D graph structure enabled much more consistent results. Not only do the thickness maps produce a better picture of disease progression, as we saw with MS, but we are also able to correct for large scale errors due to image artifacts. Ultimately, we hope to incorporate our longitudinal segmentation in more studies in the future.

Chapter 5

Automatic segmentation of microcystic macular edema in OCT

Microcystic macular edema (MME) is a condition found in a subset of MS patients whereby small cystic changes occur in the INL of the macula [17, 120]. These cystic lesions, which are called *pseudocysts*, appear in approximately 5% of MS patients [120]. Although the biological origin of these pseudocysts is not known, their presence has been found to correlate well with disease severity (MS severity score) and to predict an increased recurrence of MS attacks [17, 120]. Additionally, the appearance of MME is not restricted to MS patients. It has been noticed in eyes of patients suffering from neuromyelitis optica, Leber's hereditary optic neuropathy, glaucoma, and several other diseases [121]. Such a diversity of conditions further adds to the uncertainty surrounding MME. Therefore, the ability to accurately identify, localize, and quantify the presence of the pseudocysts found in MME is an important step in understanding how and why these changes occur. Both identifying the initial appearance of MME and tracking the changes over time will be crucial to this process.

To the best of our knowledge, there are no reports describing the automatic segmentation of pseudocysts in MME. As described in Sec. 1.4.2, several methods for approaching the more general problem of fluid and cystoid segmentation have been developed [32–37]. An important reason why these methods may not be successful when looking at MME is that the cysts found using these algorithms for their targeted diseases are generally much larger than the pseudocysts found in MS.

As an additional challenge, the primary focus with respect to MME has been on data acquired from a Heidelberg Spectralis scanner, which uses multi-frame averaging to improve the quality of each image. Unfortunately, this feature has the negative impact of averaging away the pseudocysts, reducing the contrast with the surrounding retinal tissue. Simpler methods of classifying the pseudocysts based on intensity only will therefore be less accurate.

In this chapter, we present a segmentation algorithm for the detection of pseudocysts due to MME in macular OCT scans acquired on the Spectralis scanner. Our algorithm leverages the ability of an RF classifier to learn the probability that a pixel belongs to a pseudocyst given a set of features. Section 5.1 describes the algorithm including preprocessing steps. Section 5.2 covers the experiments and results and finally, Section 5.3 provides a brief discussion and conclusions.

5.1 Methods

5.1.1 OCT data

Macular OCT data from nine MME subjects totaling twelve scans was used for this study (three subjects had MME in both eyes). All data was acquired on a Heidelberg Spectralis scanner (Heidelberg Engineering, Heidelberg, Germany). Each scan covered a 6 × 6 mm area of the macula using 49 B-scans (cross-sectional images), each with 1024 A-scans (vertical scan lines) and 496 pixels per A-scan. The axial resolution was 3.9 µm. The scanner's automatic real-time (ART) setting was used to average at least 12 B-scans at each location.

Manual segmentation of the MME data was generated by a trained rater who delineated all of the pseudocysts in each of the 12 volumes. A second rater delineated five volumes for comparison with the first rater. Pseudocysts were defined as small, hypointense, cystic lesions with identifiable boundaries and with other MME areas appearing locally in at least one adjacent B-scan (i.e. no isolated pseudocysts) [120]. Note that due to the large spacing between adjacent B-scans, individual pseudocysts are generally only visible in one image (i.e. we rarely see the same pseudocyst spanning multiple B-scans). Due to the averaging done by the scanner, the degree to which the pseudocysts appear 'dark' is highly variable. Thus, areas that were only slightly darker than the surrounding tissue were not necessarily labeled as pseudocysts, since minute intensity or texture differences may be artifactual. Table 5.1 provides an overview of all of the manually segmented data. Since a dichotomy of low and high pseudocyst count emerged from the data, we additionally classified subjects into these two groups for analytical purposes.

Table 5.1: Overview of the data used for this study including the statistics of the pseudocysts as labeled by the first manual rater. (OD - right eye, OS - left eye)

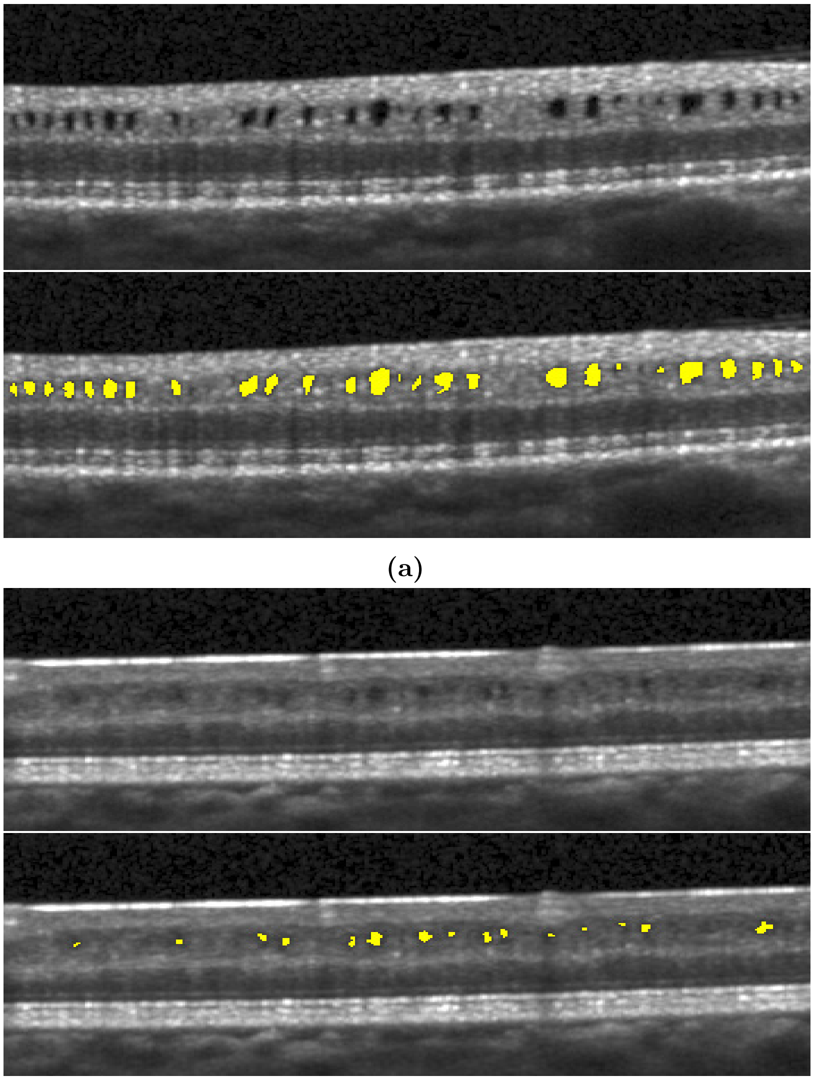
Low pseudocyst density subjects					
Subject	**Eye**	**# of pseudocysts**	**# of pseudocyst pixels**	**Median size (px)**	**Size range [min, max]**
1	OD	7	79	14	[6, 14]
2	OD	29	442	14	[4, 50]
2	OS	10	121	11	[5, 23]
3	OS	20	246	11	[6, 22]
4	OD	5	81	18	[4, 27]
4	OS	27	374	12	[5, 40]
Mean		16.3	224	13.3	

High pseudocyst density subjects					
Subject	**Eye**	**# of pseudocysts**	**# of pseudocyst pixels**	**Median size (px)**	**Size range [min, max]**
5	OD	239	3342	12	[3, 48]
6	OD	528	16793	24	[4, 444]
7	OS	301	7858	20	[3, 209]
7	OD	87	1578	15	[3, 92]
8	OD	214	5081	19	[4, 152]
9	OS	707	28621	32	[6, 254]
Mean		346	10546	20.3	

Illustrative B-scans from two separate MME subjects and their corresponding manual segmentations are shown in Fig. 5.1. In Fig. 5.1(b), we see that the ART can have the effect of averaging away some of the pseudocysts from the image, making them difficult to identify.

5.1.2 MME segmentation overview

Our overall method follows a pixel classification approach. For each pixel in a given B-scan, we compute several different features which a classifier then uses to decide which



(a)

(b)

Figure 5.1: B-scan images and corresponding pseudocyst manual segmentations from two different subjects. The averaging done by the scanner makes the pseudocysts in (b) more difficult to distinguish.

class the pixel belongs to—the two classes being pseudocyst or background. We use RF to do this classification [51], which outputs a probability that we will take advantage of in our algorithm.

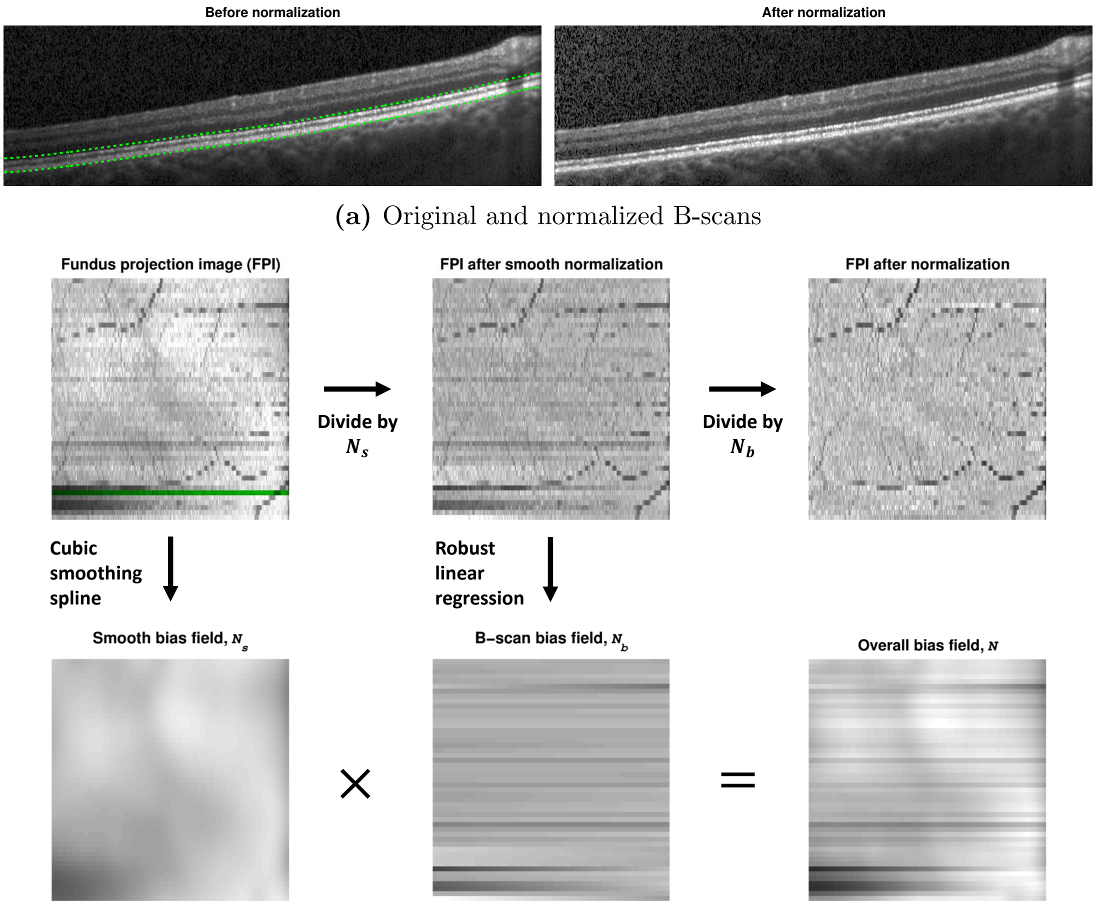
Before classifying the pixels, we first normalize the intensities of each volume to provide better consistency both between subjects and between B-scans of the same subject (Sec. 5.1.3). This normalization is different than that done in our layer segmentation method in Sec. 3.1. The reason for this change is that the previous method targeted normalizing the intensities of all the layers, while here, we focus on normalizing specifically to improve consistency of the MME. After this normalization, we describe the details of the RF features, as well as for training the RF in Sec. 5.1.4. Finally, in Sec. 5.1.5, we describe how the final segmentation is generated from the RF output.

5.1.3 Intensity normalization

Since the pseudocysts appear strictly darker than the surrounding INL tissue, it is important to normalize these intensities to make them more consistent. In the ideal case where we have the boundaries of the INL, we could normalize the intensities of this layer directly. Unfortunately, without running a layer segmentation algorithm, which may have trouble due to the appearance of the pseudocysts, we must rely on another method of normalization. Specifically, we use the RPE boundary, which is found in the same way that it was found as an initial step for layer segmentation described in Sec. 3.1.1.

To normalize the data, we follow the a similar method used to normalize the RP data in Sec. 3.2, only the intensities are scaled based on the RPE layer only, and not the RNFL. Since we use one layer, there is no z component to the estimated bias field, $N(x, y)$,[1] which is applied as $\hat{I}(x, y, z) = \frac{I(x,y,z)}{N(x,y)}$, where I and $\hat{I}$ are the original and normalized intensities, respectively. The bias field consists of two components, one varying smoothly between B-scans, N_s, and one varying within each B-scan, N_b. The underlying

[1] Note the equivalence between $N(x, y)$ here and $N_{\mathrm{RPE}}(x, y)$ in Sec. 3.2.1.



(a) Original and normalized B-scans

(b) Bias field estimation from RPE intensities

Figure 5.2: (a) B-scan images before and after intensity normalization. The dashed lines in the original image represent the area used to compute the fundus projection image (FPI) shown in (b). (b) Estimation of the bias field and its decomposition into two components using the FPI. The green line on the FPI shows where the B-scan in (a) was acquired.

RPE intensities in each A-scan are given as the median value within 80 μm of the lower RPE boundary, which is shown between the dashed green lines in Fig. 5.2(a). In 2D, these intensity values create a fundus projection image (FPI).

The steps for estimating the bias field are shown in Fig. 5.2(b), which follow those described in Sec. 3.2.1. The final FPI after normalization and the bias field are shown in the right of Fig. 5.2(b). We see that the FPI contains only a uniform intensity in addition to the blood vessels remaining intact. An example of the resulting intensity normalized B-scan is shown on the right of Fig. 5.2(a).

5.1.4 Random forest features and training

In total, 18 features are used by the RF classifier to identify the pseudocysts, with 16 of those being derived from intensities and 2 from the spatial position. The list of features used is given in Table 5.2. Since the pseudocysts are generally identified by their dark intensity, several multi-scale intensity-based features were included (features 1–12). For the morphological operators, the closing operator acts to remove the pseudocysts from the image, providing a contrast to the opening operator which enhances the pseudocysts. Examples of these features can be found in Figs. 5.3(a)–(f). Features 15 and 16 are computed for each A-scan and thus have the same value for each pixel in an A-scan. The FPI (Fig. 5.3(g)) is useful since the MME regions show up darker due to the pseudocysts producing a shadow below them. The signal-to-noise ratio (SNR) feature is used to learn where the quality of the data is particularly poor. Using this feature helps to reduce the number of false positives in low SNR areas since these areas are generally very dark.

Finally, we augment the 16 previously described intensity features with 2 spatial features. The first measures the relative distance each pixel lies between the inner and outer retina boundaries, which were again found using the initial method described in Sec. 3.1.1. An example is shown in Fig. 5.3(h). The second spatial feature measures the radial distance in the x-y plane of each pixel from the fovea (Fig. 5.3(i)). Together these features help to identify the MME based on where the pixel is in the retina. Since MME is generally only found within the INL, the first feature will be particularly helpful to identify where the

Table 5.2: List of the 18 features used by the random forest classifier.

#	Feature	Details
1-4	Intensity	Gaussian smoothed at 4 scales ($\sigma = 0,\ 5,\ 10,\ 15$ µm)
5-8	Gradient magnitude	
9-12	Laplacian	
13	Morphological opening	20 µm disc structuring element
14	Morphological closing	20×20 µm square structuring element
15	Fundus projection image	Single value for each A-scan
16	Signal-to-noise ratio	
17	Retina distance	See text
18	Radial distance to fovea	Computed in the x-y fundus plane

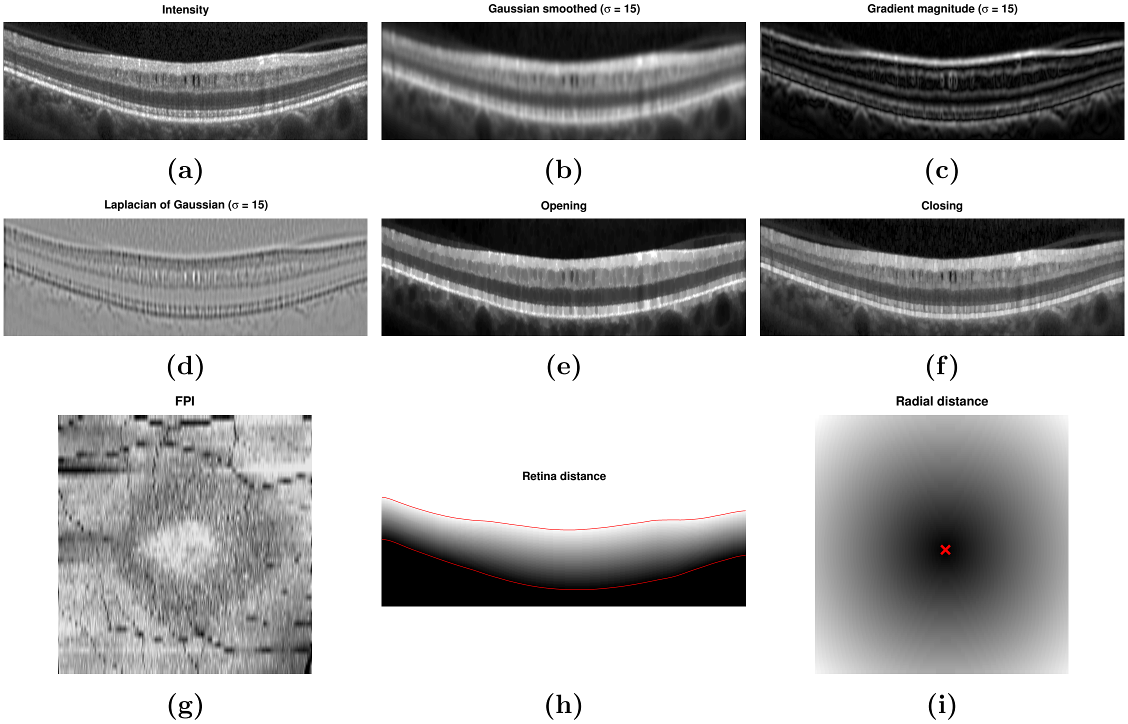

Intensity
Gaussian smoothed (σ = 15)
Gradient magnitude (σ = 15)
(a)
(b)
(c)
Laplacian of Gaussian (σ = 15)
Opening
Closing
(d)
(e)
(f)
FPI
Retina distance
Radial distance
(g)
(h)
(i)

Figure 5.3: Example images for several of the features used by the classifier to find the MME. The SNR feature is not shown since it was fairly uniform for this subject.

pseudocysts are. The pseudocysts also appear mostly within an annulus around the fovea. Thus, the radial distance will discourage the algorithm from finding pseudocysts outside of this region.

To train the RF, we used 60 trees, each grown with a minimum terminal leaf size of 10 samples. At each node in the tree, four features were randomly chosen ($m = 4$) and used to determine the best split. The algorithm's performance was fairly robust to the selection of these parameters. For training, all of the pseudocyst pixels from the manual segmentations were included along with a random selection of 20% of all background pixels taken from within a mask region described as follows. Given the position of the fovea, the background mask included all pixels within a radial distance of between 0.45 and 3 mm of the fovea (from feature 18) along with all pixels within a relative distance of 40% and 90% of the axial position from the outer to inner retina boundaries (from feature 17). Overall, this region encompassed all pseudocysts that were manually delineated.

5.1.5 Final MME segmentation

We follow a three-stage thresholding scheme to produce the final segmentation of the data. First, the RF probabilities are thresholded at a value of 0.5 to generate candidate cystic lesions. This threshold was empirically chosen to represent a majority voting scheme commonly used for binary classification problems. A second threshold is used to remove any connected regions (defined using 8-connectivity) that do not have any pixels with a probability of greater than 0.85. Thus, we encourage only those pseudocysts that are highly probable. Due to the noise and variability of the pseudocyst appearance, we do not expect a high probability everywhere within a given lesion. Note that this idea is similar to the idea of hysteresis thresholding, used previously for layer segmentation in Sec. 3.1.1. As a final step, we remove all connected components with fewer than 5 pixels, thus removing spurious areas potentially found due to noise. All remaining pixels are then labeled as pseudocysts. This final threshold makes sense since there were very few (25 in total) manually delineated pseudocysts smaller than this threshold.

5.2 Experiments and results

To explore the performance of our algorithm, we used a leave-one-out approach for training the classifier. Since we had data from nine subjects, we left one *subject* out instead of one *scan* to avoid any bias that including the opposite eye may introduce. Thus, each classifier was trained on data from eight scans. For subjects with MME in both eyes, either the left or right eye was randomly used. However, evaluations were done on all 12 scans. We also divided the results on the MME data into the two low and high density groups (see Table 5.1 for an overview of this data). This dichotomy allows us to better understand the performance of the method.

To additionally show how the algorithm performs on data without MME, we ran the algorithm on 10 healthy control (HC) subjects and 10 MS patients, with each scan manually examined and found to not have any pseudocysts. The classifier for these experiments was trained on data from all of the MME subjects (9 scans).

To evaluate our experiments, we looked at several different measures based on the number of pseudocysts found including precision (Pr), recall (Re), and F-measure (F-m).

These are defined as follows:

$$\begin{aligned}
\text{Pr} &= \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\#\ \text{correct}}{\#\ \text{found}} \\
\text{Re} &= \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\#\ \text{correct}}{\#\ \text{true}} \\
\text{F-m} &= \frac{2 \cdot \text{Pr} \cdot \text{Re}}{\text{Pr} + \text{Re}}
\end{aligned}$$

where the true positive (TP), false positive (FP), and false negatives (FN) were computed in a per cyst manner (as opposed to per pixel). If any portion of a pseudocyst found by the algorithm overlapped with one from the ground truth, it was counted as a TP. We can interpret precision as the probability that a pseudocyst found by the algorithm is real and recall as the probability that a real pseudocyst is found by the algorithm. The F-measure is an overall measure of overlap, similar to the Dice coefficient, with the difference being that it is computed over all pseudocysts found instead of all pixels found.

5.2.1 Results

Table 5.3 lists the results of the MME segmentation algorithm, divided into the low and high density groups, as well as the overall results. The measures are generally lower for the low density subjects with a larger spread as measured by the interquartile range (IQR). Specifically, we see that the low density group maintains a high confidence when a pseudocysts is found (higher precision), but the MME is, in general, more difficult to find (lower recall).

We also compared the total MME volume, computed as the total number of MME pixels in each scan, between the manual and automatic segmentation results. Comparing

Table 5.3: Median precision, recall, and F-measure values over all cross-validation runs with IQR in parentheses.

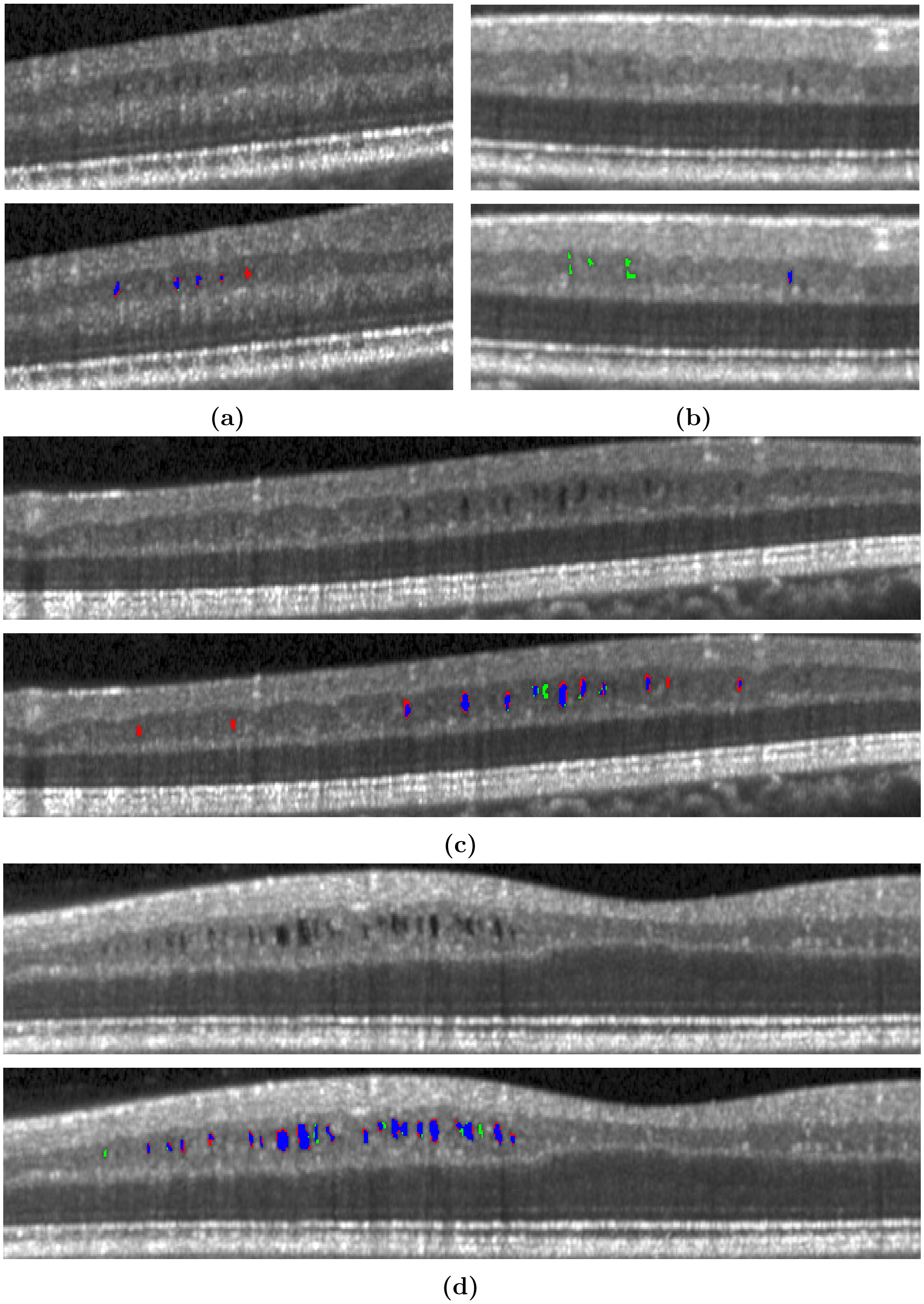
Group	Precision	Recall	F-measure
Low density	0.84 (0.75, 0.90)	0.63 (0.45, 0.78)	0.73 (0.60, 0.81)
High density	0.88 (0.77, 0.92)	0.86 (0.80, 0.88)	0.85 (0.80, 0.88)
Overall	0.85 (0.76, 0.91)	0.79 (0.63, 0.87)	0.80 (0.73, 0.85)

Table 5.4: The difference in the number of pseudocysts found and the overall difference in MME volume. Differences were computed as (algorithm − truth). Values represent the median over all scans with IQR in parentheses.

	Difference in # of pseudocysts found	Volume difference (px)	Volume difference (%)
Low density	-4 (-10, -1)	-19 (-75, 40)	-11.6 (-22.9, 26.7)
High density	3 (-14, 28)	413 (-1058, 458)	9.6 (-6.3, 25.3)
Overall	-4 (-12, 7)	15.5 (-88, 413)	0.0 (-14.6, 26.0)

these, we get a correlation coefficient of 0.98 ($p < 1 \times 10^{-7}$), which indicates excellent agreement. Table 5.4 shows the volume differences numerically, where we see that the algorithm consistently underestimates the volume for the low density subjects and overestimates for the high density subjects. Since the low density subjects generally have smaller pseudocysts (see Table 5.1), which are more difficult to segment, the total volume from the algorithm is generally smaller than the truth. The large value for the volume difference in the high density subjects can also partially be attributed to overestimation of the pseudocyst size by the algorithm.

In Fig. 5.4, we display the results of the algorithm on one B-scan each from four subjects, two low and two high density subjects. While the algorithm clearly has no trouble finding the larger, darker pseudocysts, many of the false negatives and false positives are due to having a much brighter intensity. Indeed, many of the false positives could be argued



(a)

(b)

(c)

(d)

Figure 5.4: B-scan images showing the original images and those with the results overlaid. The algorithm is shown in red, the manual segmentation in green, and the agreement of the two in blue. Images (a) and (b) are from low pseudocyst density subjects while (c) and (d) are from high density subjects.

Table 5.5: The results of running the algorithm on non-MME data. Shown are the total number of estimated pseudocysts found and the total MME volume.

	# of Pseudocysts		**Volume (px)**	
	Median (IQR)	[Min, Max]	Median (IQR)	[Min, Max]
HC	1 (0, 1)	[0, 5]	3 (0, 8)	[0, 52]
MS	1 (0, 2)	[0, 5]	4 (0, 20)	[0, 103]

to be real pseudocysts but were simply not designated as such by the rater. Also note the general appearance of a red ring around the pseudocysts indicating that the algorithm found slightly larger lesions.

5.2.2 Non-MME data

The results of running the segmentation algorithm on the non-MME data are shown in Table 5.5. The median number of pseudocysts found was the same for the HC and MS subjects with the IQR being slightly larger for the MS subjects.

Given all of the data analyzed, a simple classifier for predicting whether or not a subject has MME can be created based on the number of pseudocysts found by the algorithm. If we set the minimum number of pseudocysts found at 6 in order to classify a subject as having MME, we get a true positive rate (TPR) of $11/12 = 0.92$ and a true negative rate (TNR) of $20/20 = 1$, correctly predicting all of the non-MME as non-MME. Since thresholding in this way can misidentify those with a few pseudocysts early in the disease, we can instead look at the total RF probability of all high probability pixels in each volume (e.g. the sum of the probability of all pixels with a probability > 0.85). In this case, setting a threshold on the total probability in each volume at 16, we can get the same TPR and TNR as with the threshold based on number of pseudocysts.

5.2.3 Rater comparison

For an inter-rater comparison, each rater manually segmented five of the MME scans. Under the assumption that the first rater is the ground truth, recall, precision, and F-measure values are 0.99, 0.53, and 0.68, respectively, for the second rater. This means that the second rater found nearly all of the same pseudocysts as the first rater, but was much more lenient in their definition of what a pseudocyst is. In total, 1182 pseudocysts were found by the first rater and 1977 were found by the second rater, with median pseudocyst sizes of 19 and 27, respectively. Therefore, the second rater found more pseudocysts and they were larger as well. Another common measure of delineation overlap is the Dice coefficient, which produces values between 0 and 1, varying from no agreement and to complete agreement, respectively. Comparing the two raters, the average value over all scans was 0.53. Note that this measure is on a per-pixel basis as opposed to per-pseudocyst like the other measures. Overall, these differences highlight the difficulty in the task at hand. The most likely explanation is that the averaging done by the scanner blurs the boundaries, as is shown in Fig. 5.1(b), which creates uncertainty about what should be called a pseudocyst.

5.2.4 Algorithm design

Here, we explore the importance of different aspects and parameters of the algorithm. In the first step, we normalize the intensities of the OCT data using an estimate of the intensity values in the RPE. Comparing against the method we previously used for layer segmentation (Sec. 3.1.1), we find that the median F-measure is higher for the presented method (0.80 vs. 0.78), although the difference is not significant ($p = 0.13$, one-sided paired t-test). The difference in precision, however, is significant (0.85 vs. 0.80, $p = 0.013$). Additionally, when comparing performance on the non-MME data, the number of pseudocysts found were significantly fewer with the RPE normalization ($p = 0.025$).

Another important parameter of the algorithm is the value of the threshold for the second stage of the segmentation algorithm where only the high probability pseudocysts are retained. Fig. 5.5 shows a plot comparing the F-measure, precision, and recall values across a set of thresholds from 0.5 to 0.9 in increments of 0.05. The final threshold used was chosen as the one that maximized the F-measure. There is a trade-off between precision and recall as the threshold changes. At a value of 0.5, the recall is largest since no pseudocysts are removed. As the threshold increases, the recall decreases and precision increases until the F-measure peaks at a threshold of 0.85. We also note that this peak value also has the smallest IQR (not shown).

One of the more important aspects of the algorithm is the choice of features used by the classifier. Some features are clearly more important than others. Looking at the variable importance measure output by the classifier, the four most important features were the Laplacian of Gaussian with $\sigma = 15$ and $\sigma = 10$, pixel intensity, and A-scan distance. Some features like the estimated SNR had little effect on the overall accuracy of the algorithm, but were useful specifically in segmenting the poor quality scans. For instance, when we exclude this feature, the maximum number of pseudocysts found in the non-MME data jumps from 5 to 79.

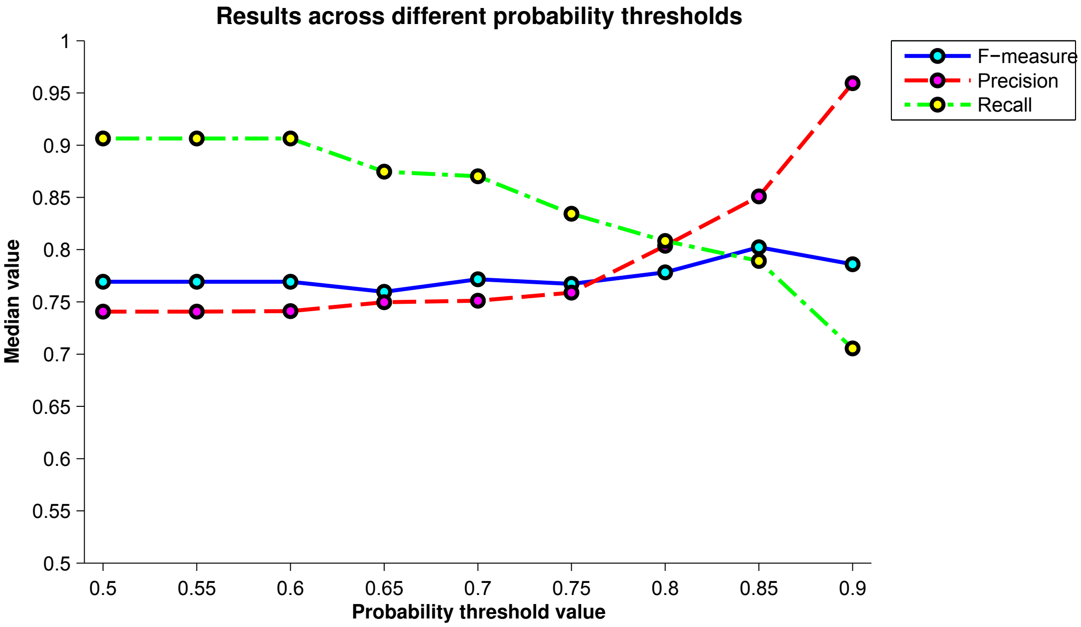


Figure 5.5: Plot of the median F-measure, precision, and recall values over all scans for varying values of the high probability threshold. The maximum F-measure value was used to determine the final threshold.

5.3 Discussion and conclusion

In this work, we developed an automated algorithm for segmentation of MME in macular OCT scans with a focus on data from the Spectralis scanner. The performance was quite good, finding 79% of the pseudocysts within the data. Using a simple classifier based only on the number of pseudocysts found, the classifier correctly labeled all of the non-MME data. While the algorithm is straightforward, using RF in a pixel-wise fashion, the addition of a novel intensity normalization method proved to increase the performance of the algorithm.

The major limitation of this study is the number of scans available for validation. Since MME appears in only a small percentage of all MS subjects, the number of available scans was small. Additionally, several datasets had to be excluded due to poor quality, further reducing the number of usable scans. More data would not only allow us to do a more comprehensive evaluation and increase our understanding of the variability of the algorithm, but it would also allow us to include more data in the training of the RF. More training data could improve both the accuracy and robustness of the algorithm.

As a word of caution with respect to segmentation of MME in Spectralis data: longitudinal analysis may prove to be difficult because each scan is averaged differently at each visit. The amount of averaging can directly influence the number of pseudocysts found by the algorithm since they can be “averaged away”. We still believe the algorithm and the data to be useful, however, as we are able to detect the presence and severity of MME. While previous authors have suggested that ART values larger than 12 are sufficient for detecting MME [120,122], this value proved to limit the ability of the algorithm to find all of the true pseudocysts. While some degree of averaging is important to reduce the amount of noise, the best practice for acquiring data to quantify MME would be to reduce the amount of averaging performed. Alternatively, the raw data could be used to do averaging in a smarter way that would not remove the pseudocysts.

In the future, we hope to explore two avenues of extending the presented work. First, the algorithm will be adapted to segment MME in data from other scanners, in particular, the Zeiss Cirrus. Preliminary work on this front has already been done [123]. Since these images do not undergo any averaging, the data from this scanner has a significant amount of noise making the pseudocysts more difficult to identify. Second, we hope to use the results of our algorithm to improve the performance of the RF+GC layer segmentation algorithm on MME subjects. The performance of this segmentation algorithm is inadequate to assess the thickness changes of the INL in these subjects, which is important since a thickened INL might be related to the presence of MME [17, 124]. Preliminary work on this idea has been done [107], where the graph structure was adapted to estimate the layer boundaries and MME simultaneously.

Finally, we note that although the proposed algorithm has been developed for segmentation of MME specifically, it should be able to identify larger cysts found in other eye diseases. Removal of the spatial features may be necessary in these cases, especially if the cysts are not expected to appear in a consistent location within the retina.

Chapter 6

Macular flatspace for improved automated processing of OCT data

In this chapter, we develop the idea of macular flatspace (MFS), which provides a computational domain enabling efficient and accurate processing of macular OCT data. After transformation to MFS, each of the retinal layers appears flat, with boundaries in approximately the same vertical position in data from different subjects. An example of a macular OCT image before and after this transformation is shown in Fig. 6.1.

MFS acts as a standardized computational space allowing for consistent processing across subjects. It also removes the curvature of the retina—which can vary significantly across acquisitions—allowing different regions of the volume to be treated in the same manner. We also note that the coordinate system in MFS becomes meaningful relative to the "coordinate system" of the retina; traversing the x and y axes in an MFS image corresponds to movement within and between layers, respectively.

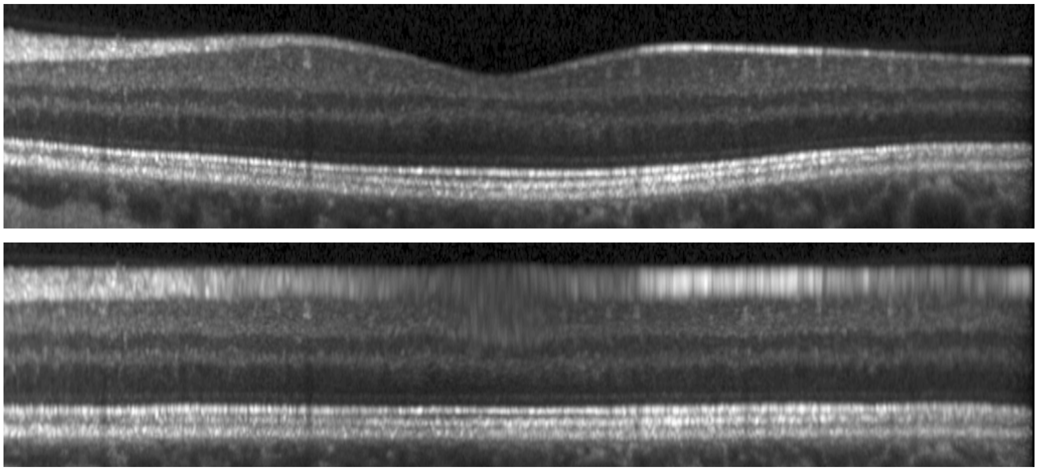


Figure 6.1: A B-scan image in (top) native space and (bottom) MFS.

The idea of using MFS to simplify the process of automated analysis is similar to the idea of placing imaging data in a common coordinate system in other domains, like neuroimaging for processing of brain MRI and functional MRI data. Such a spatial normalization process can be done by either rigid or affine registration, but often times, the data is aligned to a template space, with examples being the Talairach [125] and the Montreal Neurological Institute (MNI) [126] templates. These templates define landmarks in the brain that are be aligned, forming a common computational space.

We present two applications of using MFS: intensity inhomogeneity correction in Sec. 6.1, and layer segmentation in Sec. 6.2. It is important to note that these two methods were developed separately, and therefore the MFS is constructed in slightly different ways for each application.

6.1 Intensity inhomogeneity correction of SD-OCT data using macular flatspace

Automated methods for processing retinal OCT data have increased in importance in recent years as their use becomes more widespread. Methods for tasks such as layer segmentation (see Ch. 3), image registration (see Ch. 4), and fluid segmentation (see Ch. 5) are actively being used to look at and quantify changes in the retina. Such automated algorithms rely on the consistency of intensities within images and between subjects to perform optimally. Unfortunately, OCT images often have different intensity values between subjects and scanners. Figure 3.2(a), shown previously, and Fig. 6.2 show examples of intensity differences between OCT images. In Fig. 6.2, the images were acquired on the same patient on different scanners. In general, differences in intensity are attributed to differences in scanner settings and protocols, in the opacity of the ocular media, and possibly in tissue properties like attenuation, which can change due to disease [127, 128].

OCT images also suffer from intensity inhomogeneity problems, leading to variability *within* a single scan of the same subject. Intensity inhomogeneity occurs for a variety of reasons: off-axis acquisition resulting in signal loss [129], tissue attenuation [130], orientation of the cellular structure [117], vignetting due to the iris [131], material inhomogeneity in the eye's lens, cornea, and vitreous fluid [129], misalignment when averaging multiple images to improve image quality, and even dirt on the scanner eyepiece. Since the sources of intensity inhomogeneity are not necessarily consistent with each other, a systematic approach to correct for it is not as straightforward as in magnetic resonance imaging (MRI), where the inhomogeneity is well understood [59]. Basic methods for correcting OCT data from

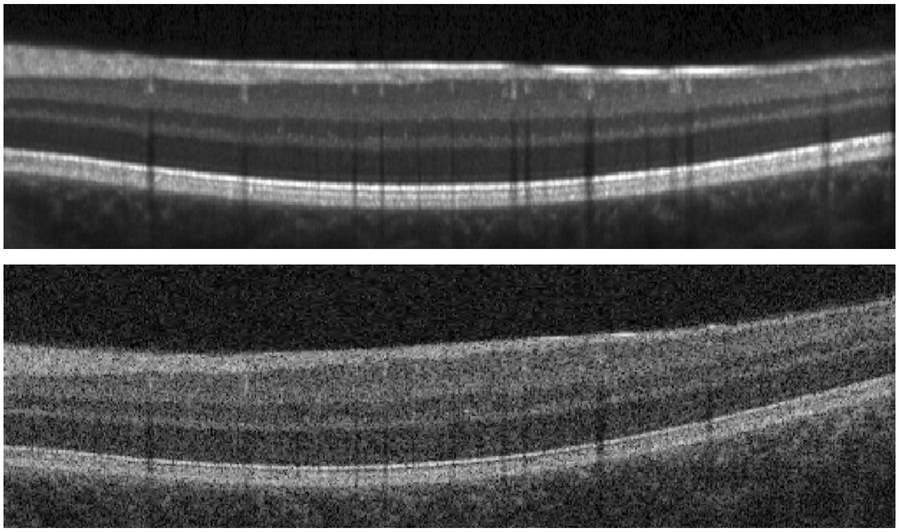


Figure 6.2: B-scan images acquired on the same subject at (approximately) the same location on two different scanners demonstrating the variability in the intensity profile. The images were acquired on (top) a Heidelberg Spectralis scanner and (bottom) a Zeiss Cirrus scanner.

well-characterized physical sources like attenuation [130, 132] may not correct for other sources of inhomogeneity.

A few methods have been developed to correct inhomogeneity in retinal OCT data. In [133], the projected intensity pattern in the fundus plane was used for illumination correction of the data; however, the correction did not vary with depth. Novosel et al. [134] use the attenuation correction method in [130] as a pre-processing step prior to running their layer segmentation method. The N4 algorithm [135], originally developed for MRI data, was used by Kaba et al. [136], but it was not able to remove all inhomogeneity as shown in their presented figures.

For intensity normalization, we previously presented three slightly different methods, in Sec. 3.1.1, Sec. 3.2.1, and Sec. 5.1, as preprocessing methods to our layer and MME segmentation work. In the first method (Sec. 3.1.1), images were normalized based on a robust estimate of the maximum intensity within each image, which is similar to other prior work, rescaling the intensity values either across the volume or in individual B-scans, to have a fixed range [19, 26]. These methods do not use the intensity range of specific layers when normalizing the data and thus inconsistencies still arise. In our other two methods, we normalized each A-scan based on the intensity values within the RPE and RNFL layers, providing a more robust normalization. The full scope of the 3D volumetric information was not used however, which means normalization within each layer may not be accurate. In contrast, the work of Chen et al. [137] used histogram matching to normalize the intensities of different scans, and showed improved stability of layer segmentation across a range of images with different quality. However, matching histograms may not be robust, especially in the presence of inhomogeneity, which blurs the peaks of the histograms.

In this work, we propose a method for both inhomogeneity correction and normalization of macular OCT data which we call N3 for OCT (N3O). At the core of our method is the N3 algorithm, which was developed for inhomogeneity correction of brain MR data [59] and has been shown to be competitive with other state-of-the-art algorithms [138, 139]. Since direct application of N3 does not produce satisfactory results on OCT data, we adapted the method, making several changes to improve both performance and efficiency.

6.1.1 Methods

Overview

The goal of this work is to correct macular OCT data so that the pixel intensities of each layer are consistent both within and across subjects. Our approach proceeds in three steps. First, we convert each B-scan into the MFS computational domain—in this domain, the second step of inhomogeneity correction is improved. The correction step estimates and then removes a smoothly varying gain field. Finally, the data is normalized so that the intensity values of the corrected data lie in a predefined range consistent across subjects.

Macular Flatspace

In order to transform the image to MFS, where the layers appear flat, we require an estimate of the boundary positions. Because intensity correction is a *pre-processing* step, we assume that no layer segmentation is available. However, we can estimate the top and bottom retinal boundaries, the ILM and BrM, respectively, as done previously in Sec. 3.1, and use these two boundaries to predict where the interior boundaries will be. For this, we use a separate regression model to find each boundary within each column, or A-scan, of an image. Figure 6.3 shows an example image with estimated boundaries as dashed green lines, which are computed based only on the solid red outer retina boundaries. Given the boundary positions, we construct a transformation going from the regression boundaries to a flattened position defined by the average position of each boundary in the native space.

Boundary estimation We estimate the boundary positions within an A-scan assuming that the thickness, $t_i(x)$, of layer i ($i \in \{1, \ldots, 8\}$) can be predicted given the total retina thickness, $t(x) = b_2(x) - b_1(x)$, where b_1 and b_2 are the initial estimates of the ILM and BrM boundaries, respectively, and x indexes the A-scans. Specifically, we use a quadratic model expressed as

$$t_i(x) = \alpha_{i,1}(x) + \alpha_{i,2}(x)t(x) + \alpha_{i,3}(x)t^2(x) \tag{6.1}$$

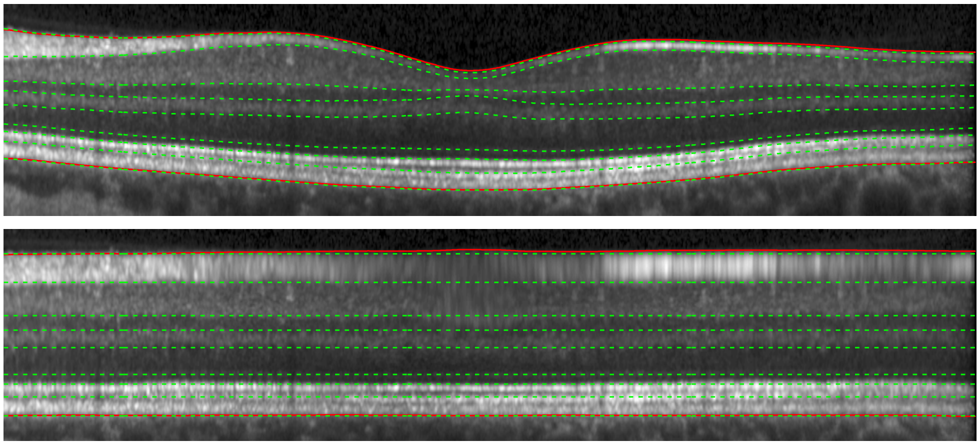


Figure 6.3: A B-scan image in native space and MFS with retinal boundaries overlaid as solid lines in red and regression estimated boundaries overlaid as dashed lines in green.

where $\alpha_{i,j}$ are the regression coefficients. A similar regression model estimates $t_0(x)$, the distance from $b_1(x)$ to the true ILM boundary, correcting for any bias in the b_1 estimate compared to ground truth data. Given an input set of retina boundaries and the result of each regression, the estimated *boundary positions* are given as

$$r_j(x) = b_1(x) + \sum_{i=0}^{j-1} t_i(x) \tag{6.2}$$

where $j \in \{1, \ldots, 9\}$ indicate the boundaries in order from the ILM to the BrM. We note that while these boundaries are estimated using the regression model here, if an automated segmentation result was available, it could be used to produce a more accurate flatspace result. Since inhomogeneity correction is intended as an efficient preprocessing step, we assume not layer segmentation is available, which would take several minutes using the method developed in Sec. 3.1.

The regression model in Eq. 6.1 is trained using data from manually segmented macular OCT scans. Since this model is spatially varying over A-scans, we first align each scan to the fovea, providing a central reference point. Since the scans were acquired having a consistent orientation, no further alignment was done aside from flipping right eyes to appear as left eyes. Any remaining variability is expected to be captured by the regression.

Given T manually segmented volumes, each having a fixed size of $L \times M \times N$ voxels,[1] there are MN A-scans and $3MN$ coefficients to estimate for the quadratic regression of a single layer. We solve for the coefficients using a regularized least squares system of equations formulated as

$$\arg\min_{\boldsymbol{\alpha}_i} \|\mathrm{A}\boldsymbol{\alpha}_i - \mathbf{t}_i\|^2 + \lambda \|\Gamma\boldsymbol{\alpha}_i\|^2 \tag{6.3}$$

where $\mathbf{t}_i$ is a $TMN \times 1$ vector of the manually delineated boundary points, A is a $TMN \times 3MN$ block diagonal matrix of the form

$$\mathrm{A} = \begin{bmatrix} \mathrm{V}_1 & & 0 \\ & \ddots & \\ 0 & & \mathrm{V}_{MN} \end{bmatrix} \tag{6.4}$$

with Vandermonde matrices V_k on the diagonal with the form

$$\mathrm{V}_k = \begin{bmatrix} 1 & t_{1k} & t_{1k}^2 \\ \vdots & \vdots & \vdots \\ 1 & t_{Tk} & t_{Tk}^2 \end{bmatrix} \tag{6.5}$$

where each row uses thickness values from a different training subject, and $\boldsymbol{\alpha}_i$ is a $3MN \times 1$ vector containing the quadratic coefficients at each spatial location. The regularization

[1] All of our training data had the same size. For testing data acquired with a different number of A- or B-scans, we can resize the coefficient maps accordingly.

matrix Γ penalizes differences in the coefficients of neighboring A-scans (i.e. the first order difference in the two orthogonal directions of the data) and λ is a coefficient balancing the fit of the data with the smoothness of the result. A cross-validation scheme described in Sec. 6.1.2 was used to choose its value. Solving the problem in Eq. 6.3 can be done efficiently as a sparse system of equations using the QR decomposition (e.g. as implemented using the `mldivide` function in MATLAB).

MFS Transformation Given an A-scan at location x, the transformation from native space to MFS is constructed by mapping the regression boundaries so that they are flat. Specifically, we use the learned values of $\boldsymbol{\alpha}_i$ from Eq. 6.3 to compute the boundaries $\{r_i(x),\ i = 1, \ldots, 9\}$ using Eqs. 6.1 and 6.2. This process is outlined in Fig. 6.4. Given this set of points in native space and a corresponding set of points in flat space, $\{f_i(x),\ i = 1, \ldots, 9\}$, we need to find a *smoothly varying* and *monotone* transformation $r = T(f)$ such that $r_i = T(f_i)$ for all i.[2] The monotone requirement preserves layer ordering and prevents folding in the transformation. Since a piecewise cubic Hermite interpolating spline [140] fits these requirements, we use this method for interpolating the transformation between boundaries. (For further details see the MATLAB function `pchip`.) In previous work, this transformation was defined using linear interpolation [60, 141], which can produce discontinuity artifacts at boundaries.

For the values of f_i, we use the average value of r_i over all A-scans (i.e. the native space boundaries are mapped to their average position). Since this definition defines positions in μm, the size of a pixel along an A-scan in MFS is arbitrarily set to be 4 μm,

[2] We define the transformation from MFS to native space since the mapping of the volume into MFS uses a pull-back transformation at each pixel defined in the opposite direction. However, since the mapping is invertible due to its monotonicity, we can compute the transformation in both directions.

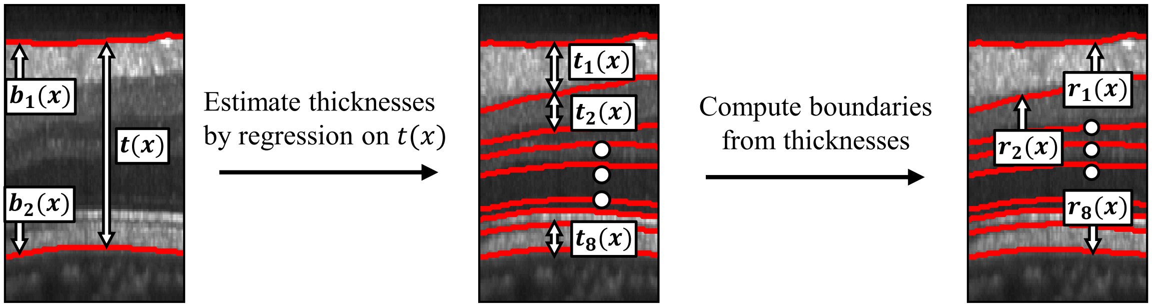

$b_1(x)$
$b_2(x)$
$t(x)$
Estimate thicknesses by regression on $t(x)$
$t_1(x)$
$t_2(x)$
$t_8(x)$
Compute boundaries from thicknesses
$r_1(x)$
$r_2(x)$
$r_8(x)$

Figure 6.4: Estimation of the regression boundaries: given the initial retinal boundaries, $b_1(x)$ and $b_2(x)$, the thickness of each layer, $t_i(x)$, is estimated from the total retina thickness $t(x)$ using the learned regression model. The boundary locations $r_i(x)$ are then estimated from the cumulative summation of the layer thicknesses.

which is close to the digital resolution of the data. Padding is also added to the ILM and BrM at a fixed distance of 60 μm. The resulting MFS data has approximately 130 pixels per A-scan, depending on the subject.

N3 Inhomogeneity Correction

We briefly describe the details of N3 [59] here, before detailing our modification for OCT data. The inhomogeneity model is assumed to be multiplicative with the intensity of an image v at position $\mathbf{x}$ given by $v(\mathbf{x}) = u(\mathbf{x})b(\mathbf{x}) + n(\mathbf{x})$, where u is the corrected/underlying image, b is a smoothly varying gain field, and n is normally distributed noise. By taking a logarithm of the data, an additive model is created, leading to the model $\log v(\mathbf{x}) = \hat{v}(\mathbf{x}) = \hat{u}(\mathbf{x}) + \hat{b}(\mathbf{x})$. The additive field $\hat{b}$ is assumed to be smoothly varying following a zero-mean Gaussian distribution. The algorithm iterates over three steps of estimating $\hat{u}$ from $\hat{v}$ given $\hat{b}$ in the previous iteration, sharpening the distribution of $\hat{u}$ using the assumed normal distribution of $\hat{b}$, and smoothing the resulting estimate of $\hat{b}$ from the sharpened $\hat{u}$ by fitting a cubic B-spline surface to the data. Iterations continue until either the field estimate converges to within a given threshold or the number of iterations reaches a specified limit.

To adapt N3 to work for OCT data, we incorporate two significant modifications to the original algorithm. First, instead of initializing the gain field to unity as in [59], we use an average MFS image for initialization to improve the convergence of the algorithm by providing a target with homogeneous layer intensities. Second, while we maintain the smoothing step of the original algorithm at every iteration, we use a slightly different and more efficient B-spline smoothing model. This new model produces similar results to the original method at a fraction of the computational cost.

Initialization By averaging over all A-scans in an MFS-converted OCT volume, we create an average A-scan profile that is then replicated back to the size of the original data to produce a template MFS image. This template image serves as a guide to what the original data should look like without any inhomogeneity. We can then compute an initial estimate of the gain field by dividing the input image by the template image. An example of these three images is shown in Fig. 6.5. This initialization has artifacts since the boundaries in the initialization are not perfectly flat. However, the iterative refinements of the algorithm allow for convergence to an accurate estimate of the true gain field, which produces corrected intensities similar to those of the template image.

B-spline smoothing model In N3, the estimated gain field is smoothed by fitting a tensor cubic B-spline surface to the data. In two dimensions, this B-spline function is written as

$$\mu(x, y) = \sum_i \sum_j \alpha_{ij} B_i(x) B_j(y) \tag{6.6}$$

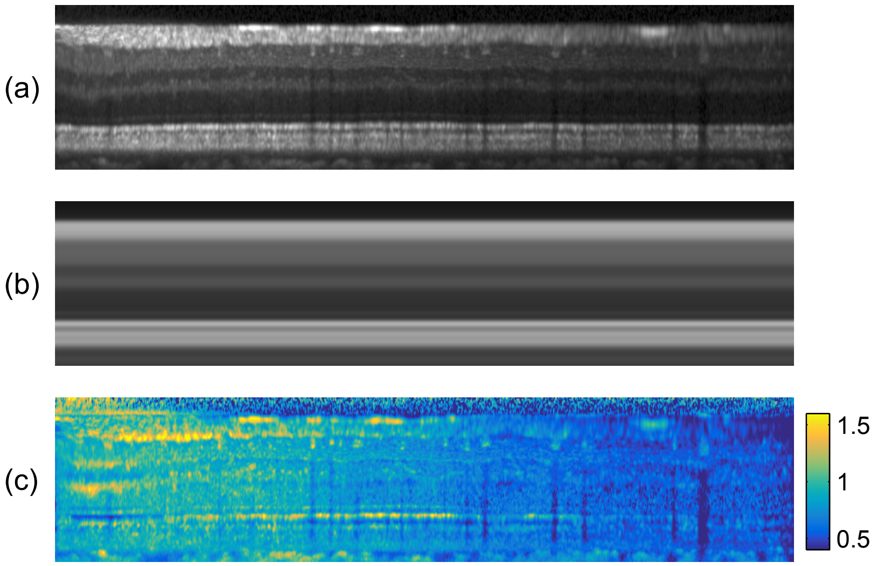


Figure 6.5: The data (a) in MFS is divided by (b) the template image to generate (c) the initial gain field.

where $B_i(x)$ is a 1D B-spline along the x dimension, centered at control point i, with a similar definition for $B_j(y)$, and α_{ij} are weights of the 2D tensor B-spline function centered at the control point indexed by i and j. Control points are equally spaced over the data with the spacing between points in each of the two directions given as algorithm parameters. Smoothness of the fit is enforced both by increasing the distance between the control points, and by adding a regularization term to the least squares fitting problem.

The B-spline model is fit to the data by solving for the B-spline coefficients that minimize an energy function as

$$\arg\min_{\boldsymbol{\alpha}} E(\boldsymbol{\alpha}) + \beta R(\boldsymbol{\alpha}) \tag{6.7}$$

where $E(\boldsymbol{\alpha})$ measures the average error of the B-spline fit, $R(\boldsymbol{\alpha})$ measures the roughness of the fit, and β is a balance parameter. In [59] the data term was defined as

$$E(\boldsymbol{\alpha}) = \frac{1}{N}\sum_{n=1}^{N}(I_n - \mu(\mathbf{x}_n))^2 \tag{6.8}$$

where I_n is the measurement of the intensity at coordinate $\mathbf{x}_n = (x_n, y_n)$, and the roughness term is defined by the thin plate bending energy as

$$R(\boldsymbol{\alpha}) = \frac{1}{A}\int_C \sum_{i=1}^{2}\sum_{j=1}^{2}\left(\frac{\partial^2\mu(\mathbf{x})}{\partial x_i \partial x_j}\right)^2 d\mathbf{x} \tag{6.9}$$

where the integral is over the region C containing the data, which has area A [59].

We use the same form of the data energy $E(\boldsymbol{\alpha})$, but a different form for $R(\boldsymbol{\alpha})$, following the work of Eilers et al. [142]. Specifically, instead of minimizing the energy over the entire B-spline surface μ, we minimize over only the B-spline coefficients $\boldsymbol{\alpha}$. The regularization term used for this problem has the form (using the notation of the original paper)

$$R(\boldsymbol{\alpha}) = \sum_i \|\boldsymbol{\alpha}_{i\bullet} D_2\|^2 + \sum_j \|D_2 \boldsymbol{\alpha}_{\bullet j}\|^2 \tag{6.10}$$

where D_2 is a second order difference matrix and $\boldsymbol{\alpha}_{i\bullet}$ is a row vector containing values of α_{ij} over all j, with a similar definition for $\boldsymbol{\alpha}_{\bullet j}$ as a column vector. Assuming that the number of control points is much smaller than the number of pixels in the image, this form of regularization constructs a far smaller matrix allowing the problem to be solved in an efficient way [142]. While this model produces slightly different results than the original N3, we have empirically found that a similar result can be produced by tuning the regularization parameter β. When tuned to provide similar results using the same control point grid, the fit using the new model is about 150 times faster than running the fit using the old model.[3]

Intensity Normalization

Since N3 only acts to sharpen the peaks of a histogram, the intensity ranges of corrected images are not necessarily consistent across scans. As we do not assume any knowledge of the segmentation of the layers, other than the retinal boundaries used in our MFS step, we cannot simply scale the intensity values within each layer to a predefined value. Instead, we scale the intensities based on the peak values found in the histograms of the vitreous region above the ILM and in the RPE.

Since the peak value in the vitreous histogram is consistent and easy to find, we use the maximum value in the histogram directly for scaling; let this value be I_1. While the RPE generally contains bright intensities, there are often two peaks in its histogram since darker intensity values also appear due to its proximity to the choroid, the appearance of blood vessel shadows, and the dark bands of the photoreceptor layers. As a result, using the peak value of the histogram may produce a value not representative of the RPE band. We counteract this possibility in three ways: 1) Restrict the region we compute the histogram over to be from the BrM to 25 μm above it, which may not fully encompass the RPE layer; 2) Compute the histogram using a kernel density estimate in which the kernel is Gaussian with a relatively wide bandwidth of 0.05; 3) Find all peaks in the histogram and choose the one centered at the largest value, which we denote as I_2. We normalize the data by contrast stretching, mapping values in the range $[I_1, I_2]$ to the range $[0.1, 0.65]$. These values are arbitrary, however they produce images with an intensity range that is consistent with those

[3] Run times come from comparing our implementation of N3 incorporating [142] in MATLAB with the implementation of [59] available at `https://github.com/BIC-MNI/N3`.

in the native data.

6.1.2 Experiments

To evaluate the performance of our algorithm, N3O, we constructed three separate experiments. 1) We created synthetic OCT data free of any inhomogeneity and looked at the accuracy of using N3O to recover randomized gain fields having different characteristics, which are added to the data. 2) We looked at the variability of the intensities within each layer before and after running N3O to explore how the stability and consistency of the intensities changes in real data. 3) We ran an intensity based registration algorithm [48] on unprocessed and N3O processed data to show the improvement in performance offered by N3O.

OCT Data

The data used in our experiments were acquired using either a Zeiss Cirrus scanner or a Heidelberg Spectralis scanner. All images from the respective scanner were scanned using the same protocol. Both Cirrus and Spectralis data cover a 6×6 mm area of the macula centered at the fovea, with the Cirrus imaging to a depth of 2 mm and the Spectralis to a depth of 1.9 mm. B-scan images have a size of 1024×512 pixels for the Cirrus data with 128 equally spaced B-scans per volume. The Spectralis images have a size of 496×1024 pixels with 49 equally spaced B-scans per volume. The Spectralis scanner also used the automatic real-time (ART) setting where a minimum of 12 B-scan images of the same location were averaged to reduce noise.

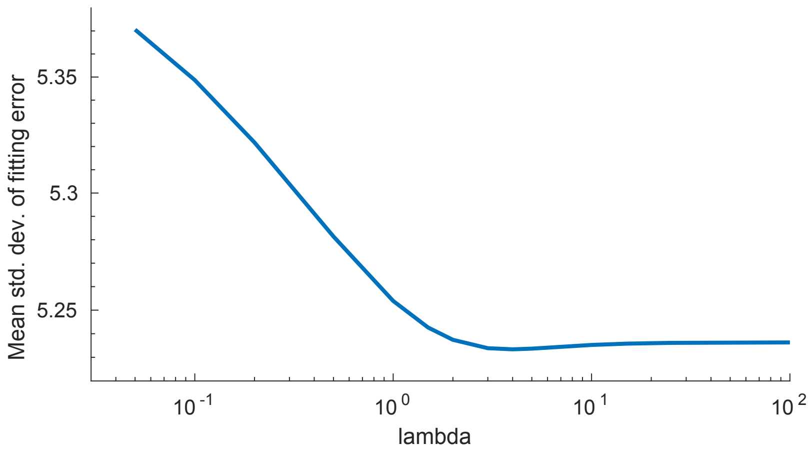


Figure 6.6: A plot of the average standard deviation of the fitting error over different values of λ using ten fold cross validation on the training data. A minimum was found at $\lambda = 4$.

Algorithm details

Training for the MFS was done using manually segmented Spectralis data from 41 subjects. Each boundary has one pixel per A-scan and therefore has 496×49 points over the volume. Rather than modifying the regularization in Eq. 6.3 to account for the anisotropy of the data, we resized the boundary maps to have a size of 224×224 pixels ($224 = \sqrt{1024 \cdot 49}$) before computing the regression. We set $\lambda = 4$ in Eq. 6.3, based on a ten fold cross validation by minimizing the average standard deviation of the fitting error in the subjects left out of a fold. A plot of the average standard deviation versus lambda is shown in Fig. 6.6. Note that we did not use mean squared error (MSE) to determine λ since a small bias to the resulting fit is acceptable as it would still produce a flat result. However, the MSE versus λ plot took a minimum at the same value.

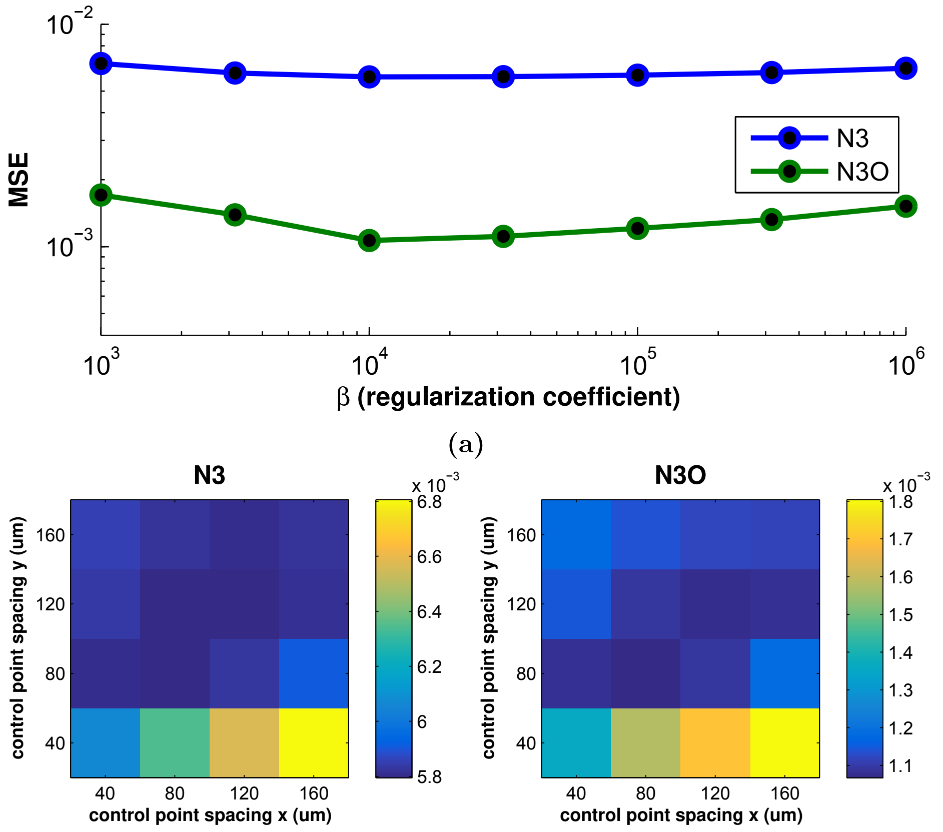
We used default parameters from the original N3 algorithm for the number of iterations (50) and the convergence threshold (0.001) and used a FWHM of 0.1 for the

Gaussian distribution used for sharpening. Downsample factors of 2 were used for each dimension in Cirrus B-scans and a factor of 4 horizontally in Spectralis B-scans (no vertical downsampling). For the experiment of running N3 on native space data, the region of the retina between the estimated ILM and BrM boundaries was used as a mask for running the algorithm.

Finally, the values of the B-spline control point spacing in the x and y directions and the regularization parameter β in Eq. 6.7 were chosen by tuning the parameters over a range of values and choosing those with the best performance. Evaluation was done on an independent set of simulated OCT data, generated as described for experiments described next. This set consisted of six synthetic OCT volumes (three Spectralis and three Cirrus), with five gain fields randomly generated using each of the four described models added to each volume. Thus, for a given set of parameters, the mean squared error between the true and estimated gain fields were averaged over all 120 data sets. The results of fixing the control point spacing to 80 μm in each direction and searching over different values of β, as well as fixing $\beta = 10^4$ and searching over the 2D space of control point values is show in Fig. 6.7. The values that worked best for both N3 and N3O were control point spacing in both directions of 80 μm and $\beta = 10^4$. Note that since both of these parameters characterize the smoothness of the gain field, an increase in one value with a decrease in the other will produce similar results.

Gain Field Recovery from Synthetic Data

For the first experiment, we created several sets of synthetic OCT data free of inhomogeneity and therefore useful for estimating the performance of N3O by applying



(a)

(b)

Figure 6.7: Plots of the mean squared error (MSE) of the gain fields recovered using N3 and N3O while (a) varying β and fixing the control point spacing, and (b) varying the control point spacing in the x and y directions while fixing β.

artificially generated gain fields and comparing the recovered field to the true field. The artificial gain fields were generated randomly by following one of two different inhomogeneity models, one having decreased intensities near the edges of the data field-of-view, simulating effects of curvature and vignetting, and one having decreased intensities over different regions of the data, with the size and number of regions varying randomly. In addition to these two global inhomogeneity models, we included an additional inhomogeneity field independently to each B-scan to simulate effects due to scan averaging, raster-scan acquisition, and eye movement.

Synthetic Data To create synthetic OCT data, we followed a similar process to how the template images were created for the N3 initialization step, as described in Sec. 6.1.1. Specifically, an input volume was converted to MFS where all of the A-scans were then averaged to create a 1D template A-scan. This template was then replicated back to the size of the original volume and transformed back to native space.

Noise was added to each image according to the OCT speckle model described by Serranho et al. [143]. Details of the algorithm are left to the cited paper. The algorithm has several parameters, and we used a different set for each scanner's data. For the Cirrus data, we used the same parameters described in the paper but amplified the scale of the additive noise by 25%. For the Spectralis data, we changed algorithm parameters $\beta_1 = 0.1$ and $\beta_3 = 0.6$. We also decreased the amplitude of the additive noise by 50% and smoothed the final noise field with an isotropic Gaussian kernel ($\sigma = 2.5$ μm). These changes are heuristic in nature, designed only to produce visually similar images to those we acquire on living persons using these two scanners.

Example synthetic B-scans from both scanners are shown in Fig. 6.8. We note that blood vessels are removed by this process. Blood vessels could be added back into the data, however we did not do this so as to assess the performance purely based on layer intensities.

Artificial Gain Fields The gain fields are randomly generated as 2D patterns on the top-down fundus plane of the data. The pattern is then projected down through the data such that either the gain has the same value throughout the entire A-scan, or it has the same value only within the RPE region. When restricted to the RPE region, we also smooth the resulting gain field so that it is not discontinuous at the boundaries. An example of this

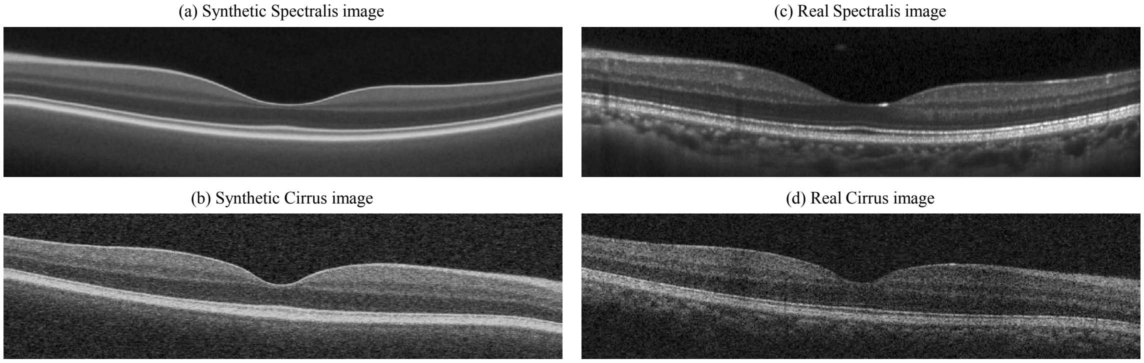

(a) Synthetic Spectralis image
(c) Real Spectralis image
(b) Synthetic Cirrus image
(d) Real Cirrus image

Figure 6.8: Examples of synthetic OCT data generated from real scans acquired by the (a) Spectralis and (b) Cirrus scanners next to their corresponding real images (c, d).

process is shown in Fig. 6.9 where we show the gain field pattern in 2D and then projected through the volume.

Two different types of randomized gain fields, which we will denote as Type 1 and Type 2, were added to the synthetic data. While these two types of fields are modeled after realistic patterns found in OCT data, they are exaggerated to be relatively extreme cases. The Type 1 pattern reduces the intensities around the edge of the field-of-view of the data. Specifically, the multiplicative gain field has a unity value within a circle of radius $3\sqrt{2}$ mm (thus, the circle can circumscribe the square 3×3 mm area of the data). The center of this circle is then randomly placed between 2 and 3 mm of the center of the scan. The gain field decays in a Gaussian shape outwardly from the edge of the circle with a variance such that the smallest value over the entire image is scaled equal to 0.2.

The Type 2 gain field pattern includes random areas of decreasing intensity simulating local areas of inhomogeneity, which commonly arise in the data. Specifically, we randomly include between 1 and 4 spots, centered randomly within the central 5 mm area of the data. The spots are modeled as anisotropic Gaussian functions with a randomly chosen

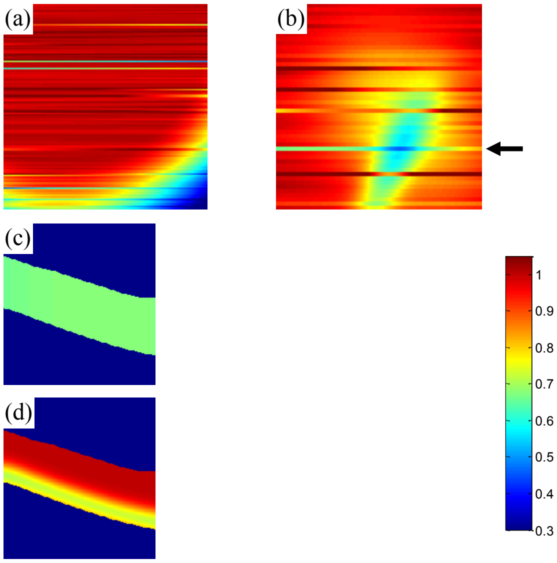


Figure 6.9: Fundus views of an example (a) Type 1 and (b) Type 2 gain field pattern. (c, d) The gain field in (b) projected through the volume, either covering the whole retina or only the RPE region. The B-scan shown in (c) corresponds to the row of the fundus image in (b) indicated by the arrow.

standard deviation value between 0.25 and 2 in each direction, as well as being oriented in a random direction, and having a peak magnitude between 0.2 and 0.5.

Finally, to model inhomogeneity patterns that differ between B-scans, we linearly vary fields across each B-scan. Specifically, the gain values on the left and right edge of each image are chosen randomly from a normal distribution with a mean value of 1 and a standard deviation of 0.02 for 85% of the images, and with a standard deviation of 0.2 for 15% of the images. Thus, the added inhomogeneity pattern is smaller for most of the data, which is an effect commonly seen in data acquired from the Spectralis scanner.

Experiments We generated ten synthetic OCT scans of healthy controls, with five coming from each of the Spectralis and Cirrus scanners. For each synthetic scan, we randomly generated ten inhomogeneity fields using each of the two models. We further restricted each of the fields to be applied either to the entire retina or to only the RPE region. Restricting only to the RPE mimics changes in the intensity of only a single layer or region, which is similar to attenuation differences over varying thicknesses. Thus, a total of 400 synthetic data sets were created. Note that noise was added to the synthetic images *after* applying the respective gain field.[4]

To evaluate the algorithm performance, we compute the average RMS error across all scans as

$$\text{RMS} = \sqrt{\frac{1}{n}\sum_{i\in M}\left(b_{\text{gt},i} - \omega b_{\text{est},i}\right)^2} \tag{6.11}$$

where the summation is over the n pixels in the retina masked region M, $b_{\text{gt},i}$ and $b_{\text{est},i}$ are the ground truth and estimated gain fields, respectively, indexed by pixel i, and ω is a normalization factor accounting for a scale difference between the two fields. The value of ω is computed in closed form by minimizing the sum-of-squared-differences [144]. Finally, we evaluate the performance using the simulated data after running N3O versus running N3 on the native space data with initialization using a unity gain field and the modified smoothing model described in Section 6.1.1 (thus we do not compare against the original N3 algorithm, but our modified version).

Consistency and Contrast of Layer Intensities

For the second experiment, we measure the variability of intensities within each layer before and after running N3O. While not a direct measure of inhomogeneity correction, measuring this variability provides a surrogate measure of algorithm performance since a

[4] While speckle can be viewed as a tissue property affected by inhomogeneity, the noise in real images appears amplified after inhomogeneity correction, and thus, this model may be appropriate

stated goal was to increase the consistency of intensities within each layer.

To measure the intensity variability, we computed the coefficient of variation (CV) of the intensity values within each layer, with lower CV values indicating better performance (more stability). CV measures the ratio of the standard deviation to the mean value and is independent of the absolute scaling of the estimated gain field and thus is directly comparable between algorithms and data sets. Layers were defined based on the results of using a validated automated segmentation algorithm on the data [53]. To reduce the effect of boundary errors, the resulting segmentation labels for each layer were eroded by one pixel.

An image having a uniform intensity everywhere would be best according to CV, so we therefore also look at the contrast between adjacent layers to show that we maintain the differences between layer intensities after running N3O. The contrast between adjacent layers i and j is defined as $C_{ij} = \left|(\bar{\mathrm{I}}_i - \bar{\mathrm{I}}_j)\right| / \left|(\bar{\mathrm{I}}_i + \bar{\mathrm{I}}_j)\right|$ where $\bar{\mathrm{I}}_i$ denotes the average intensity within layer i. Larger values of C_{ij} indicate increased levels of contrast. Since contrast is not invariant to linear transformations of the intensity, we rescaled the intensity range within each image so that the vitreous region has a value of 0.2 and the RPE region has a value of 0.65. We did not use histograms for normalization as in Sec. 6.1.1, since the histogram is less reliable in the presence of inhomogeneity.

In our experiments, we analyzed 80 scans from each of the Spectralis and Cirrus scanners—160 subjects in total. The 80 scans consisted of 40 healthy control subjects and 40 multiple sclerosis (MS) subjects. While there was some overlap in subjects scanned on both scanners, many were separate. We compared the results using N3O with those on the original data, using only the intensity normalization strategy described in Section 6.1.1, and an attenuation correction method [132]. While the work of Girard et al. [132] does not explicitly correct for inhomogeneity, it does aim to standardize the intensities across scans by estimating the attenuation coefficient of each pixel as part of a multiplicative model.

Registration and Segmentation

As a final experiment, we assess the performance of segmenting OCT data by label transfer using image registration. Label transfer segmentation, sometimes called atlas-based segmentation, produces a segmentation of an unlabeled image by registration to an image that has a manual segmentation. After registration, the segmentation of the labeled data is transferred through the registration to the unlabeled subject. While it has been used for segmentation of OCT data previously [48], atlas-based segmentation is still uncommon for this application with the previous work showing inferior accuracy to other state-of-the-art methods. Nonetheless, it has been extensively utilized successfully for other modalities like CT and MRI [145, 146].

For registration, we use a deformable registration method developed by Chen et al. [48], which, after an affine alignment step to align the outer retinal boundaries, uses one-dimensional radial basis functions to model deformations along each A-scan. Since a full registration is carried out separately on each A-scan, a regularization term is used to ensure that the resulting deformations are smoothly varying between A-scans. Further details of the full registration algorithm can be found in [48].

To evaluate the results of the label transfer segmentation on the OCT data, we registered a set of 5 randomly chosen subjects to each of a separate set of 10 subjects, with the subjects chosen from the same cohort of data used in the consistency experiments. Since the registration algorithm has not yet been validated using Cirrus data, this experiment was restricted to only Spectralis data. Evaluation was done by looking at the average unsigned boundary error between the registered data and the ground truth segmentations produced over all 50 registrations.

6.1.3 Results

In Table 6.1, we show the results of running N3 and N3O on the synthetic OCT data. We see that N3O performs better than N3 in every case with statistical significance shown for every comparison ($p < 10^{-10}$ using paired two-tailed t-tests). Looking within specific experiments, recovery of the whole scan field had a smaller error than recovery of the field restricted to the RPE for N3O. This is because the restricted RPE field had an abrupt change in the gain field which could not be recovered as accurately due to the gain field smoothing step of N3. We also see similar performance for recovery of the second type of gain field as compared to the first type. While the second type appeared to be locally more difficult where the gain field changes in smaller regions, looking globally averaged away these differences.

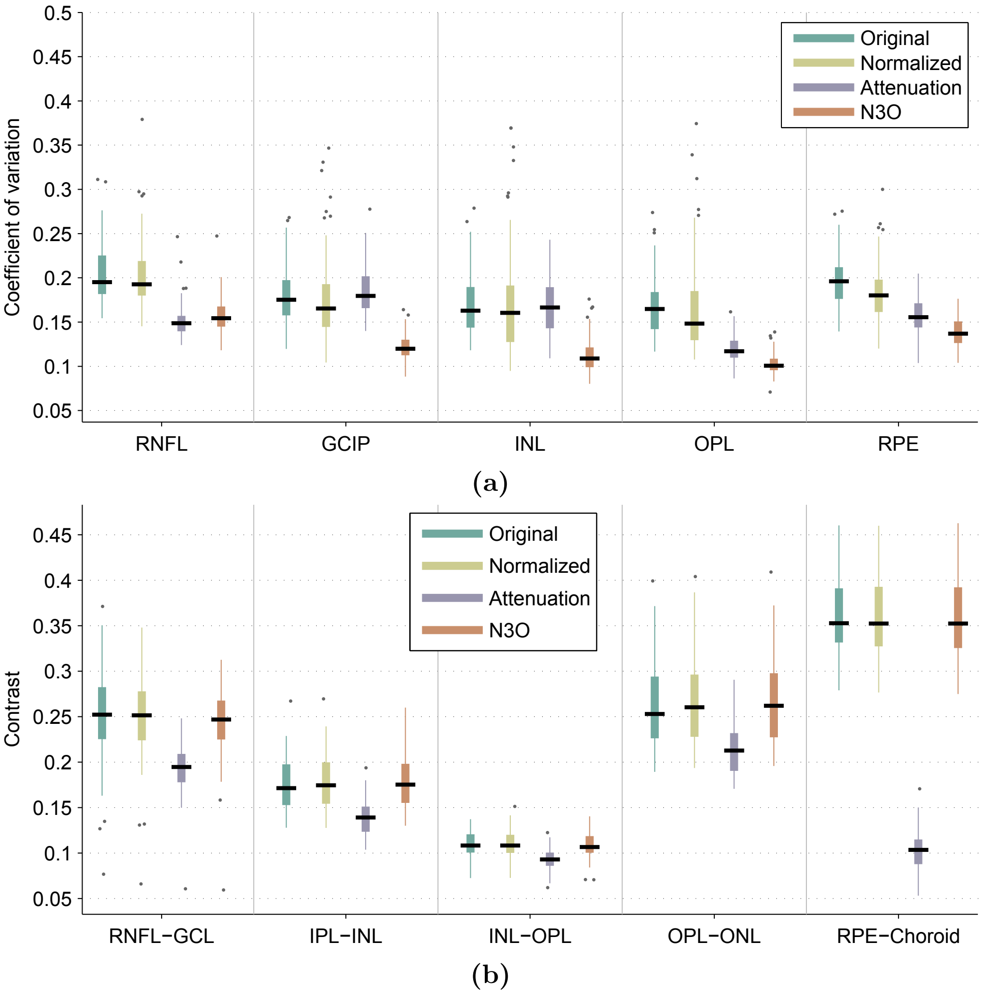
We observe from Table 6.1, that the gain field is more accurately recovered on synthetic Cirrus data as opposed to Spectralis data. This, on the surface, is counter-intuitive as Cirrus data is noisier than Spectralis data. We believe that this result is due to the Spectralis and Cirrus data having different histogram profiles as well as different contrasts between each of the layers. Since N3 acts on the histogram of the data, differences between the histograms from these two scanners produce different results. When the Spectralis noise model was applied to simulated Cirrus data, and vice versa, we see the expected result with

Table 6.1: Mean RMS error in the recovered gain field. Results for each field type and method are averaged over 100 trials. All results show significant improvement using N3O over N3 ($p < 10^{-10}$ using a paired two-tailed t-test).

	Whole Scan		**RPE**		
	Type 1	**Type 2**	**Type 1**	**Type 2**	**Mean**
Spectralis					
N3	0.095	0.076	0.047	0.049	0.067
N3O	0.015	0.014	0.036	0.042	0.027
Cirrus					
N3	0.094	0.078	0.046	0.048	0.067
N3O	0.010	0.009	0.030	0.039	0.022

the smaller Spectralis noise performing better than with the Cirrus noise.

Figure 6.10(a) shows the result of computing both the CV and the contrasts of the layer intensities on the original data and the data after intensity normalization, after attenuation correction, and after running N3O. The presented results are only shown for the Spectralis data on a restricted set of layers. Results for all of the layers and for the Cirrus data are provided in Appendix 6.A. Overall, we see that N3O has significantly better CV values than the first two methods for all layers ($p < 10^{-9}$), and significantly better than the attenuation correction for all layers except the RNFL, OPL, and ISOS layers ($p < 10^{-6}$), where the attenuation correction was significantly better ($p < 0.01$). The Cirrus data showed similar results, with added significance over attenuation correction for all layers ($p < 0.01$). Interestingly, the normalization only result on the Cirrus had worse CV values than in the original data (see Fig. 6.A1), which is likely due to noise in the data affecting the histogram peak estimation. Since the histogram peaks are sharpened after running N3, N3O did not have this problem. Thus, intensity normalization by peak finding is not a recommended strategy for uncorrected OCT data.



(a)

(b)

Figure 6.10: Box and whisker plots of (a) the coefficient of variation of the intensities within a select set of layers and (b) the contrast between successive layers.

Figure 6.10(b) also shows that the contrast of the data is not negatively affected by running N3O. While some layers showed significant differences when compared to the original data, the contrast values were consistent with each other to within a median value of 0.011 for every layer. We also see that while attenuation correction helps to remove inhomogeneity in the data by normalizing to attenuation values (thus improving the CV), some layers end up with less contrast, for instance, the RPE to choroid interface.

Finally, we show the results of the label transfer segmentation experiment in

Table 6.2: Mean absolute errors (μm) in boundary positions between the ground truth segmentation and the segmentation results after registration. Standard deviation across subjects are in parentheses. Using N3O produces a significantly better result compared to using the original intensity data for every boundary ($p < 0.05$).

	ILM	**RNFL-GCL**	**IPL-INL**	**INL-OPL**	**OPL-ONL**
Original	3.90 (0.68)	9.84 (2.83)	11.95 (5.07)	10.77 (4.49)	9.11 (4.09)
N3O	3.65 (0.51)	7.78 (2.51)	7.22 (3.30)	6.63 (1.73)	6.00 (1.21)

	ELM	**IS-OS**	**OS-RPE**	**BrM**
Original	5.86 (1.47)	4.29 (1.43)	6.37 (1.53)	4.94 (2.04)
N3O	4.84 (1.00)	3.48 (1.47)	5.91 (1.53)	2.83 (0.53)

Table 6.2. Here, we see that N3O improves the performance of image registration of OCT data on average with significantly better results for every boundary ($p < 0.05$). Figure 6.11 shows an enlarged portion of a sample registration result displaying both the subject and target as well as the results of the registration using the original intensity image and the N3O corrected image. While N3O improves the registration result, there is still room for improvement, as indicated by some of the larger values in Table 6.2. This problem is mainly due to the variability inherent in the registration. The algorithm parameters were not tuned to optimize the performance of the segmentation, and better results would likely be possible if they were.

6.1.4 Discussion and Conclusion

We have proposed a method for inhomogeneity correction and intensity normalization of macular OCT data. While the previously developed N3 method is used for correction, it required adaptation for OCT by converting the data to a macular flat space before running it. MFS allows the estimated gain field to vary smoothly within layers thereby improving

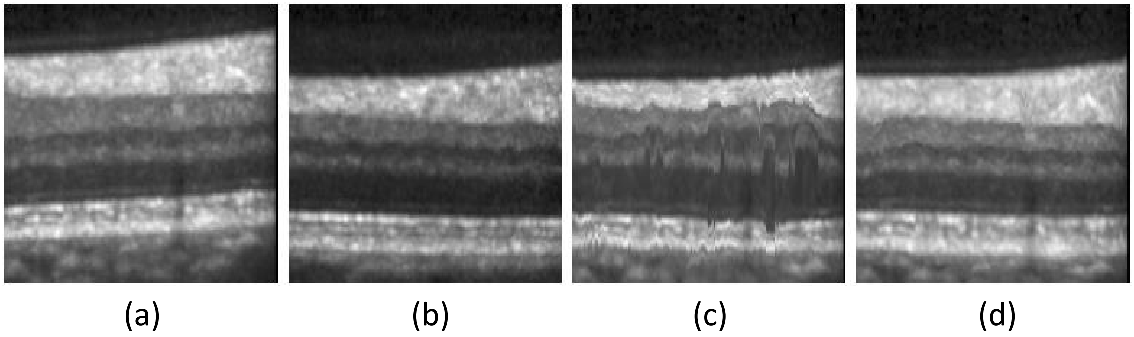


Figure 6.11: Cropped registration result between (a) subject and (b) target images with (c) the result without intensity correction and (d) the result after running N3O on both images prior to registration.

the consistency of the resulting intensities. It also allows us to initialize the gain field in a meaningful way by creating a template image to estimate the initial field. In addition to the use of MFS, we improved the performance of N3 by modifying the model used to smooth the gain field at every iteration. This modification produces similar results to the original algorithm with improved efficiency.

Experiments showed that N3O can accurately recover gain fields applied to synthetic data. While the process used to create the synthetic data did not follow physical principles of how OCT is generated, realistic images were created with the performance of the algorithm further validated by looking at the intensity consistency as well as the results of registration. Since simulation of OCT was not a research aim of this work, we believe the model used in our experiments provides sufficient evidence for the performance of N3O. In the future, validation based on imaging a phantom could be used. However, questions about how realistic the phantom is with respect to the shape, texture, and number of layers will likely still arise.

Additionally, the assumption that the gain field is a multiplicative model may not be appropriate, but the resulting images showed improved consistency according to our initial goal of reducing the variability of intensities within each layer. There is also evidence to show that a multiplicative model might be correct in some sense since the attenuation correction of OCT data as modeled in [130] and [132] is multiplicative.

To further validate our method based on the performance of automated methods, we used the application of image registration, which depends entirely on the intensity of the data. While alternative cost functions, like mutual information and cross correlation, may be less sensitive to intensity variations, the method we used was validated using the sum of square differences measure and we wanted to maintain consistency in our comparison to previous work [48].

When initially undertaking this research, we hypothesized that N3O would improve the accuracy of our retinal layer segmentation algorithm [53]. However, our experiments (not described here) have shown no significant differences in the results. Since this algorithm uses a classifier to learn boundary positions based on a variety of features, including both intensity and spatially varying contextual features, there is a learned robustness to intensity differences. We believe that this accounts for the lack of statistical significance. Other work has shown that intensity normalization is important for generating consistent results using other segmentation algorithms [137], and therefore, we believe N3O would be an important pre-processing step for such methods.

In the future, we hope to extend the algorithm to work fully on 3D data, instead of running independently on 2D B-scan images. While we showed good performance when running the algorithm in 2D, we expect improvements in the consistency of the results between adjacent images to improve with a 3D model. The difficulty of implementing this idea is that the gain field will need two components, one varying in 2D within each image independently, and one varying smoothly between images in 3D, since we see both types of problems in OCT data, depending on the source of the inhomogeneity.

Finally, we note the computational performance of our algorithm. The method takes on average 0.06 seconds per B-scan image for the Spectralis data and 0.12 seconds for the Cirrus data, with code written in MATLAB. Approximately 40% of this time is spent on converting the data to and from MFS (e.g. interpolation), with the remaining time spent on the N3 inhomogeneity correction and intensity normalization. Performance was measured on a 1.73 GHz quad core computer running Windows 7 and ultimately can be improved by both conversion to a low-level programming language and through parallelization for each B-scan.

6.2 Use of an adaptive, patient-specific graph for layer segmentation in retinal OCT

In this work, we enhance the retinal segmentation algorithm described in Sec. 3.1 by converting the data to MFS and then running the segmentation, which allows for reduced constraints in the resulting graph search problem. In our original work describing this method [60], we looked at this problem as deforming, or adapting, the underlying graph that the segmentation is carried out on to account for the size, orientation, and curvature of each individual's retina. In fact, adapting the graph in this way is equivalent to converting the data to MFS; these are simply two alternative ways to think about the problem. Such

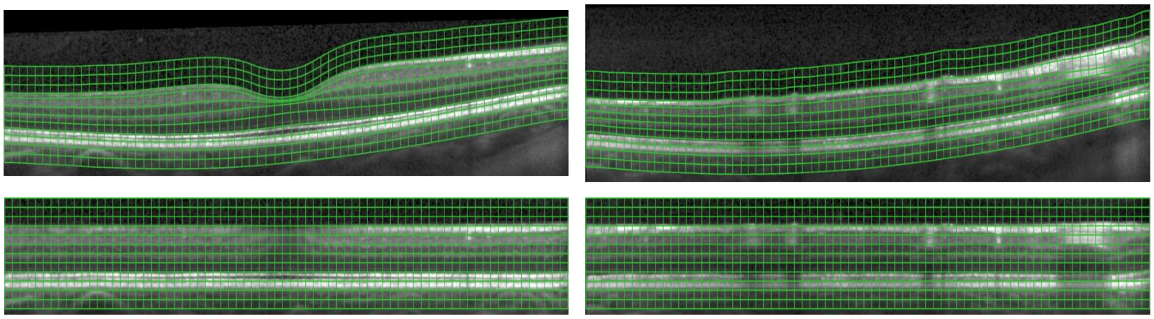


Figure 6.12: (top row) A subject adaptive graph is overlaid on OCT images. By carrying our our RF+GC segmentation on this graph, we produce the same result as if the graph was constructed on data converted to MFS (bottom row).

an idea is shown in Fig. 6.12 where we see the subject adaptive graph on top, and the appearance of the same graph if the vertices were laid out as a regular grid on the bottom. Therefore, throughout this section, we will refer to adapting the graph the segmentation is carried out on, as opposed to converting the data to MFS. The equivalence will be made clear at the end of the Methods section in Sec. 6.2.1.

In the traditional graph-based segmentation approach, a graph is constructed by placing vertices, or nodes, at the location of each voxel, with edges added to enforce constraints. Here, we instead place vertices at fixed locations between estimates of the top and the bottom surfaces of the retina, the ILM and BrM, respectively.

There are three main benefits to constructing the graph in this new way. Firstly, we control the scale of the graph, allowing for subvoxel resolution where necessary (for example, the fovea). Secondly, by allowing the graph to take the shape of the retina, graph construction is simplified, enabling a more natural inclusion of smoothness constraints. Finally, we are able to smoothly fill in areas having a lack of boundary evidence (for example, in the shadow of a blood vessel).

6.2.1 Methods

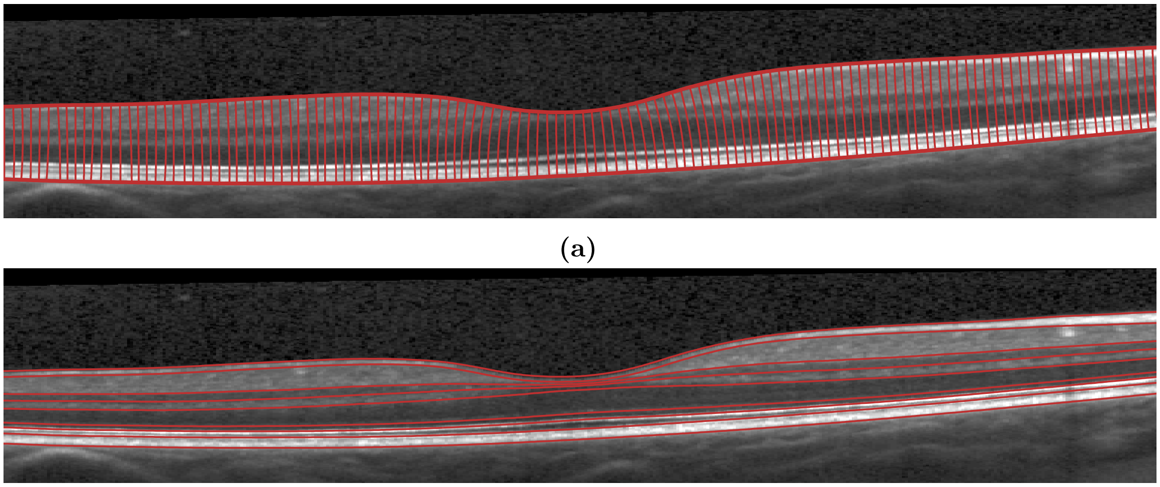
This layer segmentation algorithm works by generating an adaptive, patient-specific graph to carry out an optimal graph-based segmentation algorithm. To do this, we assume that there is an initial segmentation for the inner and outer boundaries of the retina, which are once again found as described in Sec. 3.1.

Adaptive graph construction

There are three steps for generating a patient-specific adaptive graph. In the first step, we place vertices along streamlines generated from the inner to the outer retina surface (Fig. 6.13(a)). Such a streamlining approach has previously been used to aid in the computation of cortical thickness of brain MRI scans [147], as well as in the construction of a grid for annular tissues [148]. In the second step, we use a regression model to produce an estimate of the location of each layer boundary between the two outer boundaries (Fig. 6.13(b)). A *separate* regression model is learned at each point on the surface of the ILM, thereby incorporating spatial information into the estimated positions. Finally, we fill in the graph with nodes placed along the streamlines based on the regression.

Streamline generation We generate streamlines from the ILM to the BM by solving Laplace's equation between the two surfaces [147]. This approach has the property that the streamlines are guaranteed not to cross each other; a property that lends itself nicely to construction of a grid. Laplace's equation takes the form

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \tag{6.12}$$



(a)

(b)

Figure 6.13: Images showing the construction of our adaptive grid. (a) Streamlines are generated between initial estimates of the outer retinal boundaries. (b) Estimates of each boundary as determined by using a regression model.

subject to the boundary conditions $u(S_{\text{ILM}}) = 0$ and $u(S_{\text{BrM}}) = 1$, where u represents the harmonic function to be solved for, and S_{ILM} and S_{BrM} are the locations of the inner and outer retina surfaces, respectively. Since the retina does not form a closed object, we additionally enforce boundary conditions along the sides of each image to maintain their initialized values (described next). Also note that since the out-of-plane resolution of our data is disproportionally larger than the in-plane resolution, we only solve the equation in 2D, on each B-scan independently. Were the data closer to isotropic, then solving the equation would make more sense in 3D.

We solve Eq. 6.12 for u using a red-black Gauss-Seidel method. The interior values of u (between S_{ILM} and S_{BrM}) are initialized to the proportional distance of each pixel from the top surface to the bottom, along each A-scan. This initialization provides a good approximation leading to fast convergence. We then construct streamlines between the two surfaces by integrating over the tangent field of u [147]. A tangent vector field, $\vec{N}$, between the two surfaces is constructed from

$$\vec{N} = \frac{\nabla u}{\|\nabla u\|}. \tag{6.13}$$

By integrating $\vec{N}$ from every point on S_{ILM} to some point on S_{BrM}, we can generate correspondence trajectories, or streamlines, between the two surfaces. We carry out the integration using a Runge-Kutta 4th order scheme. Finally, to account for not having isotropic pixel size, we modify both of the previous equations to incorporate the resolution of the data [149]. An example of the streamlines generated between the retinal boundaries is shown in Fig. 6.13(a). Notice that the streamlines flow orthogonally from each surface.

Boundary regression model With the streamlines computed, we estimate the positions of the seven interior retinal boundaries along each streamline using a regression model trained on manual segmentation data. This regression takes the form

$$p_i(x) = \alpha_{i,1}(x) + \alpha_{i,2}(x)t(x), \tag{6.14}$$

where $p_i(x) = d_i(x)/t(x)$ is the relative distance along streamline $x \in \mathcal{X}$ from the ILM to surface $i \in \{1, ..., 7\}$, $d_i(x)$ is the total distance along the streamline from the ILM, $t(x)$ is the total length of the streamline, and $\mathcal{X}$ is the set of all streamlines across the retina. To align each subject spatially for the regression model, the thicknesses are aligned to the center of the fovea. Note that this model can be directly be compared with the MFS model used in Sec. 6.1.1. By multiplying both sides of Eq. 6.14 by the total thickness $t(x)$, we end up with a similar cubic regression model, except no intercept is used in this case. Equivalence can

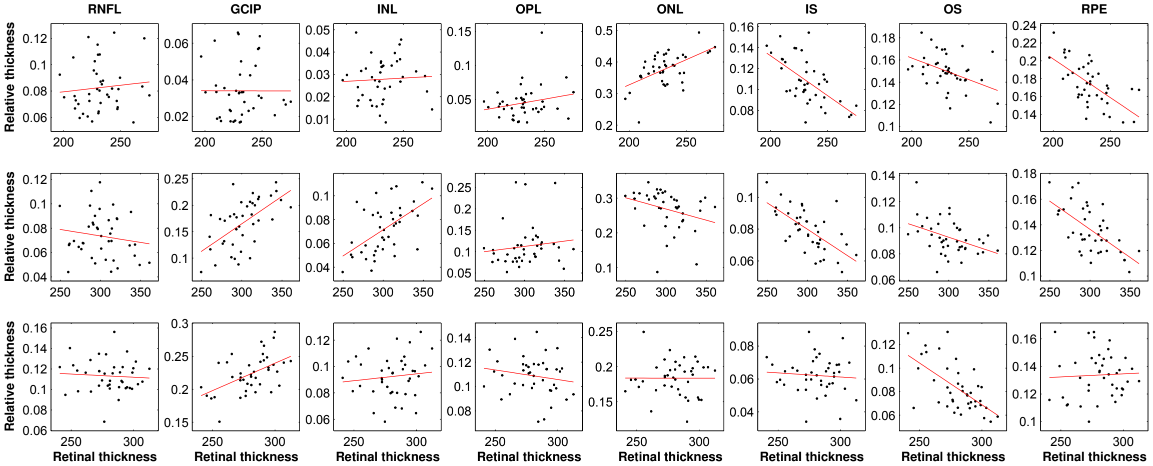


Figure 6.14: Examples of fitting the regression model at three separate locations in the retina (from three separate streamlines). Results are shown for each layer (top row) at the fovea center, (middle row) 0.5 mm from the fovea in the nasal direction to the fovea, and (bottom row) 1.5 mm from the fovea in the temporal superior direction. Each black dot represents the measurement from a separate subject (by manual segmentation), with total retina thickness (i.e. the total length of the streamline, in µm) on the x-axis, and the thickness of the layer with respect to the total distance from the ILM to the BM on the y-axis.

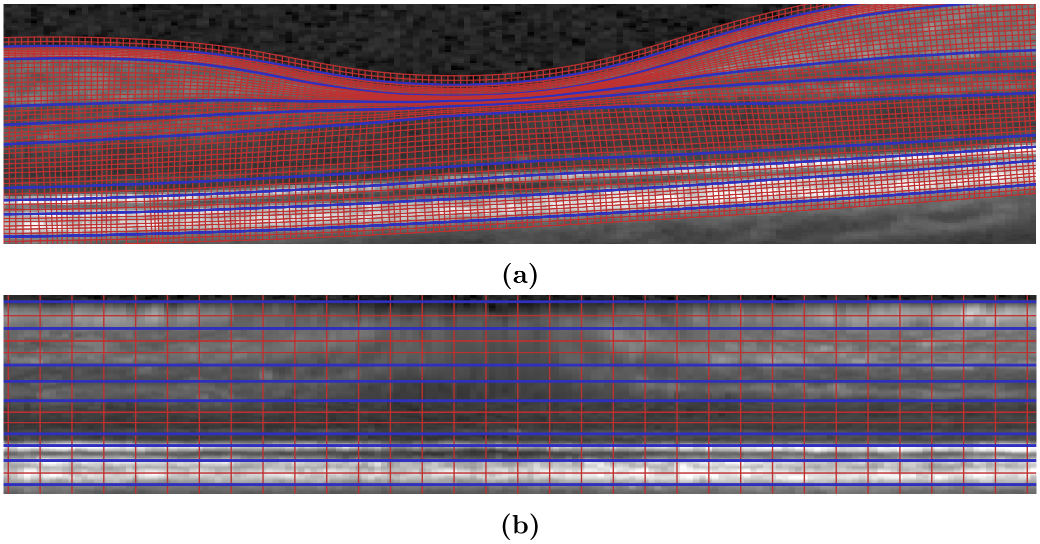
be made when computing $d_i(x) - d_{i-1}(x)$, looking at individual layer thicknesses instead of absolute distances.

Figure 6.14 shows an example of the regression on $p_i(x) - p_{i-1}(x)$, the relative thickness of each layer at three locations on the retina using all of the training data included for fitting the model. From this figure, we see that while there is a lot of variability between subjects, the data often follows a linear trend. A line with a strong positive slope means that as the total retina thickness increases, the layer thickness increases at a greater rate than the total thickness. For example, we estimate the GCIP thickness as 10% of the distance from the ILM to BM if the total distance is 250 µm and 20% of the distance if the total thickness is 330 µm.

Final graph construction Given a set of streamlines between the retinal boundaries and a learned regression model for each streamline, we place "base nodes" in the graph according to Eq. 6.14. In other words, for boundary i we place a vertex at a distance $d_i(x)$ along streamline x. We denote these vertices as base nodes since they anchor the graph at locations consistent with the location of each boundary, also allowing us to fill in the rest of the graph with an specified number of vertices between each base node. All of the base nodes for a respective boundary are connected with horizontal edges between adjacent A-scans, allowing them to follow the natural shape of the retina (see Fig. 6.13(b)).

The rest of the graph is filled in between the base nodes by adding N_i vertices along the streamlines between the base-nodes surrounding layer i. Nodes are also added above and below the top and bottom boundaries to allow for small changes to the fit of these surfaces. An example of a final constructed graph overlaid on a B-scan is shown in Fig. 6.15(a). Note that, for display purposes, this is not the full density of the final graph we use for our algorithm. We choose N_i such that the average distance between nodes within a layer is equal to s (i.e., if the average distance between base nodes within a layer is 40 µm, and we would like an average of $s = 4$ µm between nodes, $N_i = 40/4 - 1 = 9$ equally spaced nodes would be added). Since the distance between base nodes is variable between streamlines, the actual distance between inserted nodes is different across the retina, providing better resolution in areas where it is more useful, like where the layers come together at the fovea.

If we visualize the deformed graph on a regular lattice grid with only horizontal and vertical edges, like in Fig. 6.15(b), we see that the effect of constructing the graph in this way is to flatten the data to each boundary. Since the underlying data ultimately needs



(a)

(b)

Figure 6.15: (a) The final grid overlaid on the retina, constructed after filling in the graph along the streamlines between the regression estimates (shown in blue). The vertices of the graph are located at the intersection of the lines. We denote base nodes as those at the intersection of the regression (blue) lines. (b) When looking at the deformed grid as a 4-connected rectangular lattice, we can think of the deformation as flattening the data to each boundary (significantly downsampled for visualization).

to be interpolated to this new adaptive graph, we end up with an identical model to MFS. Indeed, the remaining segmentation problem becomes one of finding boundaries in MFS, and unflattening them after to produce a result in the native image space.

Segmentation algorithm

Our final segmentation method follows closely from our previous work (Sec. 3.1), using the RF+GC method. The difference between the current method and previous method is in the use of the adaptive graph, described above, and modified edge weights within the graph. Since the graph nodes do not fall on the regular grid of the OCT volume, we use linear interpolation to extract the probability values corresponding to each node.

By using the adaptive graph, or MFS, setting up the final graph-search segmentation becomes much simpler. Following the graph construction described in Sec. 2.2, we use hard constraints to limit the the minimum and maximum distance between adjacent surfaces. Since our graph contains a fixed number of vertices between base nodes, we simply set the maximum distance (δ^u) to be $2\,(N_i + 1)$ voxels for layer i. The minimum distance (δ^l) is set to one voxel to allow for the possibility of surfaces coming together near the fovea. Additionally, we limit the maximum distance a surface may change between adjacent pixels ($\Delta^l_{x,y}$ and $\Delta^u_{x,y}$). This distance is set to one voxel in each direction since, as a result of the implicit boundary flattening, the surfaces should not change much between A- and B-scans.

Finally, we add soft constraints to our graph to penalize surfaces that deviate from the natural curvature provided by our adaptive grid. This soft constraint uses finite, non-zero edge weights to horizontally connected edges between adjacent streamlines. We assign a constant weight of w edge. Such an edge is similar to the ones used to connect our longitudinal graphs in Sec. 4.2.1, and also described in Sec. 2.2.4 and shown in Fig. 2.5(c).

6.2.2 Experiments and Results

To evaluate our algorithm, we used macular OCT scans ($20^\circ \times 20^\circ$) from the right eye of 38 subjects. All scans were acquired using a Spectralis OCT scanner. Of the 38 subjects, 15 were healthy controls and 23 were diagnosed with MS. Each volume contained 49 B-scans, 1024 A-scans, 496 pixels per A-scan, although we downsampled the lateral direction to 512 A-scans. The total imaged area of the macula was approximately 6×6 mm. All nine layer boundaries were manually segmented on all B-scans of all subjects by a single trained rater.

The regression model from Sec. 6.2.1 was trained using all of the manual segmenta-

Table 6.3: Values of the mean absolute error (standard deviation) for each boundary for the adaptive grid algorithm in comparison to using the voxel grid (as in Sec. 3.1). Values are in μm and standard deviations are computed across subjects.

	ILM	**RNFL-GCL**	**IPL-INL**	**INL-OPL**	**OPL-ONL**
Voxel Grid	3.89 (0.72)	4.06 (0.62)	3.85 (0.49)	3.68 (0.68)	3.52 (0.82)
Adaptive Grid	3.49 (0.65)	3.42 (0.72)	3.81 (0.55)	3.83 (0.64)	3.32 (0.94)

	ELM	**IS-OS**	**OS-RPE**	**BM**	**Overall**
Voxel Grid	3.05 (0.40)	2.45 (0.54)	4.60 (1.84)	3.35 (1.68)	3.60 (1.14)
Adaptive Grid	2.81 (0.39)	2.24 (0.61)	4.28 (1.89)	3.20 (1.63)	3.38 (1.15)

tion data. A cross-validation strategy could have been employed, but we found the regression parameters to be robust to the data used. The RF classifier was trained in the same way as described in our previous work in Sec. 3.1.

Since the RF was trained using seven subjects, we examined the results of our algorithm on the remaining 31 subjects. We first explored the effect of changing two parameters in our algorithm, s and w, the average distance between vertices and the amount of smoothness, respectively. We changed the values of s from 2 μm to 6 μm in increments of 1 μm, while we changed the value of w from 0 to 1 in increments of 0.2. For each pair of values, we looked at the average absolute error across all subjects and all boundaries. Fig. 6.16 shows the average and standard deviation of the errors as heatmaps. The minimum error was found for values of $s = 3$ and $w = 0.2$. These are the values used to produce the numerical results in Table 6.3. It makes sense that a smaller value of s produced the best accuracy since lower values of s lead to higher resolution of the grid. Since our manual segmentation protocol leads to smoother segmentations, it also makes sense that some amount of smoothing improves the results.

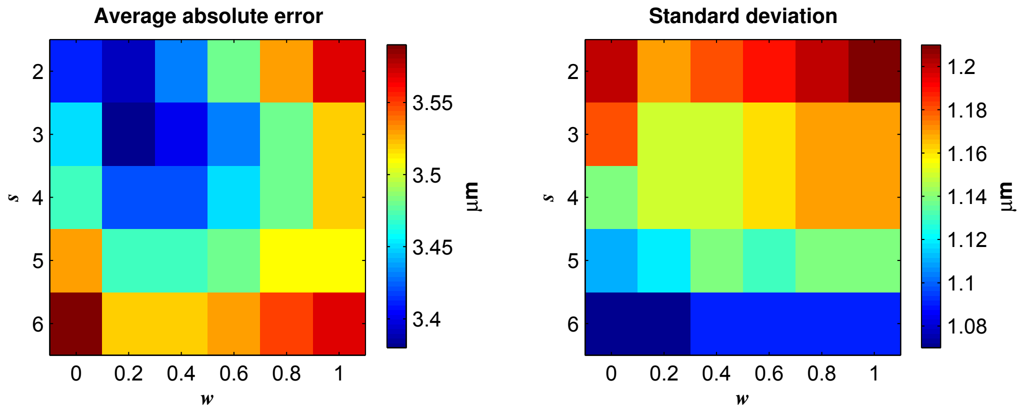
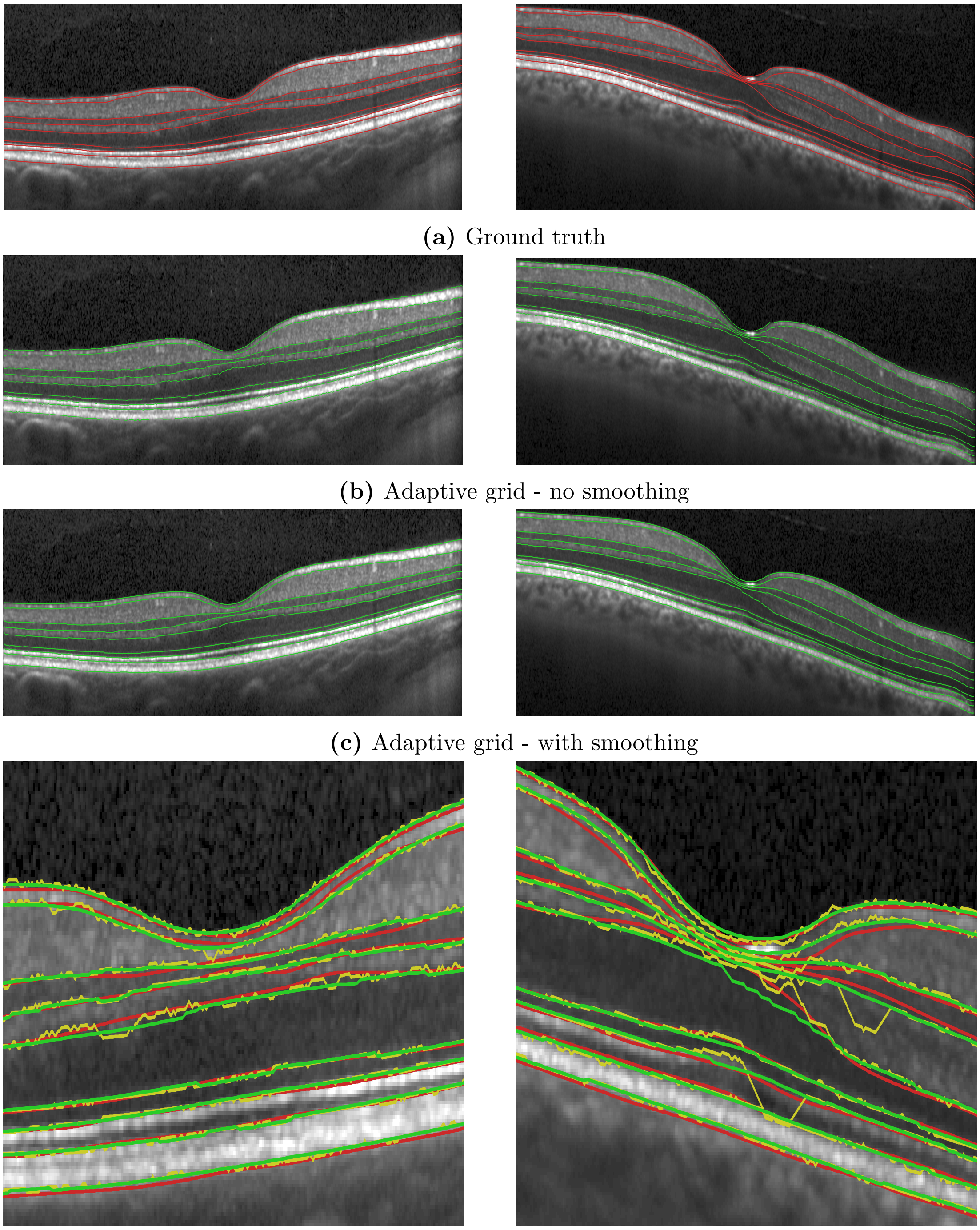


Figure 6.16: Images showing the value of the average absolute error (left) and standard deviation (right) as the parameters for smoothness (w) and for grid size (s) are changed.

In Table 6.3 we compare the results of running the segmentation on the adaptive grid for each subject to the results of not using the adaptive grid—i.e., using the voxel grid as done by our previous segmentation method (Sec. 3.1). The values represent the mean absolute error over all subjects, with distances computed along A-scans. Looking only at the overall errors, we see that the OS-RPE boundary has the largest error. This is likely due to the ambiguity which sometimes arises as there are multiple dark bands in this region. Overall, we see that many of the remaining boundaries have errors within the axial resolution of our data (≈ 3.9 μm). In comparison to the voxel grid method, the new method has smaller errors for 8 out the 9 boundaries. Fig. 6.17 shows the results of both algorithms on two B-scan images. Both the manual and automatic segmentation results are displayed for comparison. In particular, the results in the right column of Fig. 6.17 show the new algorithm performing better in two areas where the voxel grid algorithm performed poorly due to errors in the pixel classification. Note some of the discretization effects of our adaptive graph on the final segmentation that arise due to the graph construction not lying on the pixel boundaries, which could be smoothed out using smaller values of s or with additional post-processing such as smoothing.

6.2.3 Conclusions

A new layer segmentation method was developed by using a subject adapted graph. It has many advantages over previously developed methods including added smoothness without additional complexity, the ability to smoothly fill in areas with low boundary probabilities by smoothly following the retina shape, and finally, the possibility for subvoxel resolution in areas where the layers converge together. We showed accuracy that is as good or better than our previous algorithm, which was shown to be competitive with the current state-of-the-art. One of the main drawbacks of the method is the additional computational burden from streamlining. Fortunately, every streamline can be generated independently, which would allow for a large speedup with parallel processing. Alternatively, fast algorithms for computing correspondence trajectories using a hybrid Euler-Lagrange approach could be used [148]. Finally, since training was required for several steps of the algorithm (streamline regression and random forest classification), it is not likely to work with data containing gross pathological issues. This atypical data could be included in the training process, but other possibilities for improvement include removing the spatial component from the classifier training and using a nonlinear regression model for the graph construction.


(a) Ground truth

(b) Adaptive grid - no smoothing

(c) Adaptive grid - with smoothing

(d) Ground truth (red), voxel grid (yellow), adaptive grid (green)

Figure 6.17: Two B-scan images with overlaid segmentations from (a) the manual ground truth, (b) the voxel grid algorithm, and (c) the deformed grid algorithm. The results from a zoomed in region of the fovea are shown in (d). Images are from separate subjects and have been scaled 3× in the vertical direction.

6.3 Summary

In this chapter, we introduced the concept of MFS and described two applications in which it can be used for. By transforming macular OCT data to MFS, data from different patients looks similar, with all of the retinal layers appearing flat. This outcome is very beneficial to development and application of automated image processing methods. By providing a common space where every scan is similar, we do not need to incorporate the shape variability of different patients into the algorithm itself. Instead, any variability appears as slight deviations from the layers being perfectly flat. Additionally, processing like smoothing can be done exclusively within or between layers by smoothing in the horizontal or vertical directions. Oriented smoothing to match the shape of the retina is no longer necessary, as was used with the RF features in our original layer segmentation method (Sec. 3.1). Also, any algorithm constraints on the shape of the retina, like those used in the graph structure, can be simplified, or even removed.

In Sec. 6.1, we used MFS to improve the performance of the N3 method, which was originally developed to work for brain MRI data [59]. By converting the data to MFS, we were able to both utilize a more accurate initialization for the algorithm, and improve the performance of the smoothing step within the algorithm to better handle the shape variability and thin structure of the retina. Going forward, we expect inhomogeneity correction and intensity normalization to be an important step for any automated image processing method. As such, we previously utilized simpler methods to normalize the intensities as a preprocessing step for several of the methods developed in this thesis, however, we believe that going forward, N3O will be the best choice to use since it is efficient and uses a principled, validated method to do the corrections.

In Sec. 6.2, we used MFS to improve the performance of our previously developed layer segmentation method utilizing RF and a graph-theoretical model. While the work was motivated by the idea of adapting, or deforming the underlying voxel grid the data lies on, this idea is conceptually equivalent to using MFS.

We want to make note of three distinct differences in the MFS formation used between Sec. 6.1 and Sec. 6.2. First, a slightly different regression model was used, with an added intercept term included in the regression model in Sec. 6.1. This allows for improved modeling of the shape of the retina, but added regularization is needed to enforce spatial smoothness and prevent overfitting. Another difference between the models is that cubic interpolation was used instead of linear interpolation when converting the data to flatspace in Sec. 6.1. Using linear interpolation tends to produce discontinuity artifacts when the layers are stretched by a large amount. Finally, in Sec. 6.1, data from the each A-scan remained in the same A-scan in MFS, while in Sec. 6.2, the data was distorted such that each point was placed along streamlines according to the solution of Laplace's equation. The reason for using streamlines is to improve modeling of the shape of the retina. However, there is added cost to computing the streamlines, which adversely affects performance of algorithms utilizing MFS. The best MFS model was not explicitly compared, but going forward, we expect to converge on a single one for all applications, which include both of those described here.

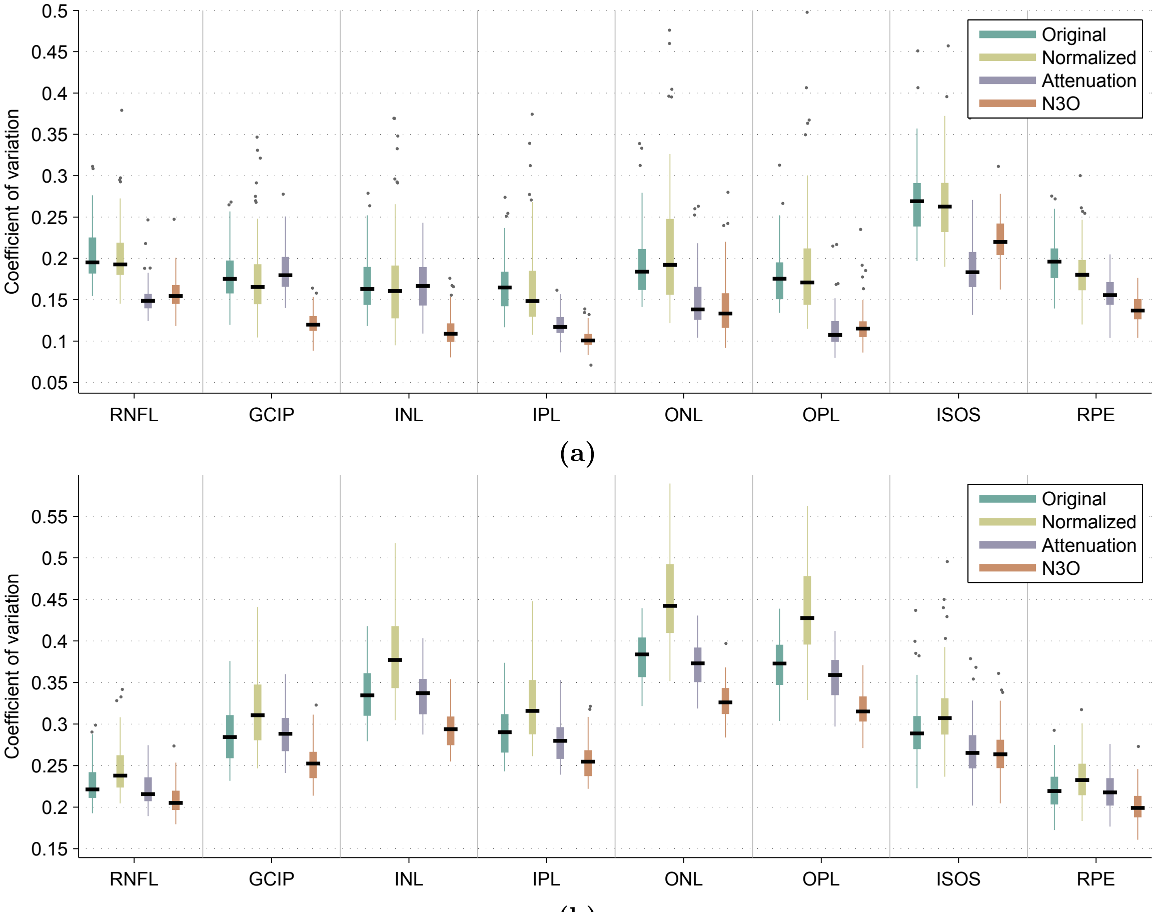
As a final note, we expect MFS to be useful for many different applications in macular OCT processing, beyond those of intensity normalization and layer segmentation.

Two potential applications are MME segmentation and registration, both of which we previously developed methods for. For MME, MFS might allow the pseudocysts to be more easily found within specific regions of the data. For registration, we expect that the full 3D image registration problem will be made considerably simpler when looking at two images in MFS. Prior work already requires that the outer retinal boundaries be aligned as an initial affine step [48], but alignment of all of the boundaries would further place the data into a good initial alignment. Finally, we note that in addition to being used in our own graph-based segmentation framework, MFS also been successfully utilized to improve the performance of other layer segmentation algorithms, including one using deformable models [141].

6.A Full set of consistency experiment results

Box and whisker plots showing the coefficient of variation of the intensities within all segmented layers for both the Spectralis and Cirrus data are shown in Fig. 6.A1. N3O was found to have statistically significantly lower CV values compared to the original and normalized results for all layers ($p < 10^{-9}$), and compared to the attenuation coefficient for all layers except the RNFL, OPL, and ISOS layers in the Spectralis data ($p < 10^{-6}$), where the attenuation coefficient result is significantly lower ($p < 0.01$).

Box and whisker plots showing contrast between all successive layers for both the Spectralis and Cirrus data are shown in Fig. 6.A2. Results were similar between the original, normalized, and N3O results, with a maximum median difference of 0.011 when comparing either the original or normalized results against N3O. The attenuation correction results

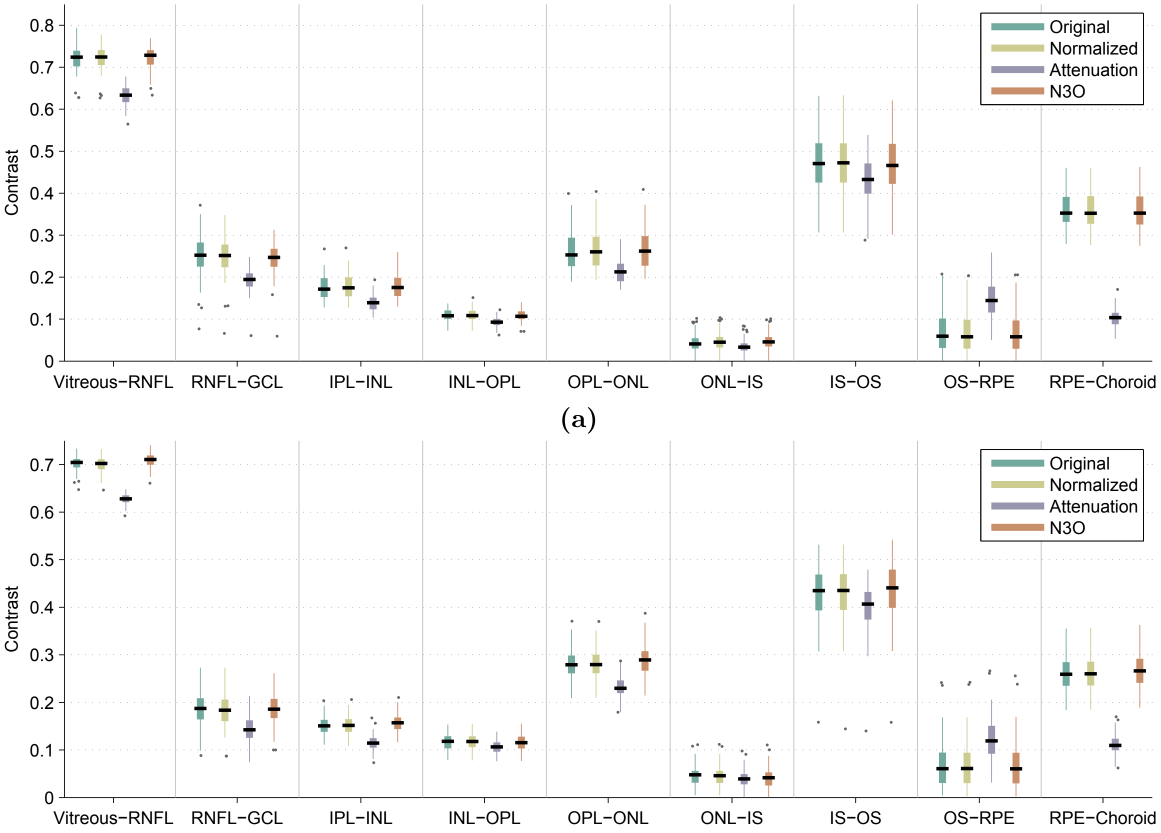


(a)

(b)

Figure 6.A1: Box and whisker plots of the coefficient of variation of the intensities within all segmented layers for (a) the Spectralis data and (b) the Cirrus data.

showed decreased contrast for all layers except between the OS and RPE.



(a)

(b)

Figure 6.A2: Box and whisker plots of the contrast between all successive layers for (a) the Spectralis data and (b) the Cirrus data.

Chapter 7

Conclusions and future work

7.1 Summary

In this thesis, we have developed several methods for improved automated analysis of macular OCT data. Chapter 2 began with primers on RF classification and graph-based segmentation, which were critical tools used throughout the thesis, setting the groundwork for the developed algorithms. In Chapter 3, we introduced a layer segmentation framework, utilizing RFs to learn boundary probabilities, which were then refined either by boundary tracking or a graph-based algorithm to produce a final segmentation. We also showed an adaptation of this method for patients having retinitis pigmentosa, which degrades the photoreceptor layers. In Chapter 4, we developed methods for registration and segmentation of longitudinal data. Registration proved to be important for both aligning thickness maps and for aligning data prior to doing a simultaneous longitudinal segmentation, which extended our prior work in Chapter 3. In Chapter 5, we looked at the segmentation of pseudocysts in patients with MME, with this method also utilizing the RF classifier. Finally, in Chapter 6, we introduced macular flatspace, enabling data from different subjects to be closely aligned, reducing the burden of automated methods for learning these differences. MFS transforms the data to have visually flat layers, allowing individual layers to be easier targeted.

7.2 Key ideas and results

7.2.1 Layer segmentation

1. We developed a layer segmentation framework that proved to be robust, working on a variety of different types of data. An RF classifier was used to produce boundary probabilities, which were then refined to generate a final segmentation. The refinement step used either a Canny-based tracking algorithm (RF+CAN) or a previously developed graph search surface segmentation method (RF+GS). Both methods showed a high degree of accuracy with average boundary errors of less than 3.4 μm over all layers. While not statistically different in terms of accuracy, RF+GS showed several regions having a smaller error variance spatially compared to RF+CAN, indicating that it produced fewer outlier results.

2. Our layer segmentation framework, RF+GC specifically, was adapted for data from patients with RP, a disease in which the photoreceptor layers show significant atrophy. Image quality was also severely degraded in many cases. Adaptation of the method was done at each stage of the algorithm. Despite the added difficulty in segmenting this data, our algorithm performed well with an average absolute boundary error of 4.22 μm, which is about 25% worse than the errors computed on normal data. Our

method also showed about 0.8 μm better accuracy in delineating four of the boundaries when compared to one of the only previously developed methods for RP data in the literature [103].

3. The use of machine learning proved to be critical to the success of our method. By using manually delineated images from several subjects, the algorithm was able to learn differences in the appearance of each boundary based on the output of several features. Using only a separate set of training data, we found comparable segmentation performance between data acquired from two different scanners, each having very different noise properties. Specifically, Bhargava et al. [106] used our method and found that when evaluated on subjects scanned on both scanners, 5 out of 7 layers showed an average difference in thickness of less than 1.31 μm, with the other two layers having a difference of 2.16 and 4.71 μm, a reasonable result considering the increased noise level of the second scanner.

4. The method is efficient, producing a segmentation in 3–4 minutes. Further gains in computational time could be made through efficient coding, parallelization, and selective masking of the layers, as done for the longitudinal segmentation. If increased performance is not essential, a different set of algorithm parameters (the MSP) can be used to reduce the run time by 41%. However, the cost of using these parameters is a significantly lower accuracy by 0.63 μm ($p < 0.001$).

7.2.2 Registration and segmentation of longitudinal data

1. Registration of longitudinal OCT data is important for examining changes in the retina over time. For registration of Cirrus data, a point-based registration method was used in conjunction with motion correction. Blood vessel points were segmented and extracted from an FPI generated from the volumetric data, and the CPD algorithm was used to estimate correspondences between points at each visit. Looking at manually selected landmark points, the method showed an accuracy of 29.7 μm, and the addition of the motion correction component accounted for a significant increase in accuracy by 22% ($p < 0.01$).

2. To evaluate the importance of registration and motion correction of data with respect to comparing layer thicknesses, we looked at data from 26 healthy control subjects, each scanned twice over the span of one hour. In this data, any change in thickness between the scans is partially attributed to misregistration of the data. After registration, the difference in thickness values between scans was significantly closer to zero than without registration ($p < 0.05$), and the added incorporation of motion correction showed a significantly better result in two of the layers ($p < 0.05$), with an average improvement of 0.08 μm.

3. We extended the RF+GC framework to improve the segmentation of longitudinal data. This segmentation was done simultaneously on multiple registered volumes to enforce the consistency of the delineated layers between scans. Despite the added computational burden of simultaneously segmenting multiple volumes, the method was made efficient using a low resolution cross-sectional segmentation, allowing the

algorithm to run in less time than for each of the volumes separately. Specifically, it takes about 10 minutes to segment 4 longitudinal volumes simultaneously, while it takes about 3–4 minutes to segment a single scan using the previously developed RF+GC framework.

4. To evaluate the performance of the longitudinal segmentation framework, we showed improvement in the stability of thickness measurements made over time as compared to measurements made cross-sectionally at each visit. For this experiment, the data was registered before comparing the results to reduce the effect of misregistration when looking at the results. Looking at a cohort of 13 scans from 7 healthy subjects scanned twice over an interval of approximately one year, the longitudinal method showed reduced variability in the thickness measurements by 46% over the cross-sectional analysis. A similar experiment showed significantly better stability for measurements made on a cohort of MS patients (38% improvement, $p < 10^{-10}$).

5. Looking towards providing clinically relevant results using our longitudinal layer segmentation, we showed that the change in thickness of the RNFL and GCIP layers can be localized in patients with MS with lower variability, providing more confidence that we are finding true changes. Data from 17 MS patients was used with each scanned between 3 and 5 times at intervals of between 3 and 12 months. In Figure 4.12, we showed that without incorporation of longitudinal consistency, many localized areas of increased thickness are found, which appear to be artifactual since these layers are known to atrophy over the course of the disease.

7.2.3 MME segmentation

1. We developed the first automated algorithm for finding MME in macular OCT data. An RF classifier was again used to do the segmentation. Careful design of the features used by the RF was needed, with intensity-based features being most important due to the large contrast between the MME and the surrounding tissue. Overall, the algorithm was able to find 79% of all pseudocysts when looking at manual segmentation data. Accuracy was higher in subjects that had more pseudocysts, likely because the cysts were larger on average in these images.

2. The probabilistic output of the RF was utilized to improve the performance of the segmentation. While the final pseudocysts were identified using a majority vote (≥50% probability), we excluded any cysts that did not have at least one pixel with ≥85% probability. This second threshold was important since many false positives surpassed the first threshold but not the second. Figure 5.5 indicates the importance of this threshold, showing that a trade-off between precision and recall exists as the threshold is varied from 0.5 to 0.9.

3. Looking at 10 healthy and 10 MS scans without the appearance of MME, we found a median false positive rate of one pseudocyst per volume, with a maximum number of 5 in a given scan over all of the data. In order to classify a subject as having MME, setting a threshold on the number of pseudocysts at 6 leads to a true positive rate of 92% and a true negative rate of 100% when combining all of the non-MME and MME scans.

4. MME is difficult to segment when multi-frame averaging is enabled on the scanner. Since consecutive images may not be in identical positions due to minute eye movements, the pseudocysts become more difficult to see. Ambiguities between what is and is not a cyst lead to both false positives and false negatives by the algorithm that could be true in reality due to rater misclassification. This discrepancy was shown by the relatively small overlap value between two raters manual segmentation results. Specifically, while one rater found 99% of the same pseudocysts as the second rater, they delineated 67% more pseudocysts than the other rater overall. When looking at the dice coefficient, a measure of overlap between the raters, a relatively low value of 0.53 was found, where a value of 1 indicates complete agreement.

5. The incorporation of intensity normalization as a preprocessing step proved to be important for improving the performance of our method. A separate normalization scheme was needed compared to the one used for our RF-based layer segmentation, with the new method normalizing based on the intensity values in the RPE layer. Incorporation of the normalization significantly improved the precision of the algorithm by 5% ($p < 0.05$) and also significantly reduced the number of pseudocysts found in the non-MME data ($p < 0.05$).

7.2.4 Macular flatspace

1. Macular flatspace (MFS) is a normalized coordinate space that transforms macular OCT images such that all of the layers appear flat. As one application of MFS, we developed N3O, a method for doing both intensity inhomogeneity correction and

intensity normalization of macular OCT data. By converting the data to MFS before running the N3 algorithm, we were able to improve the performance of the method by 63% when looking at the ability of the method to recover artificially generated gain fields on simulated OCT data. We also showed that N3O significantly decreased the variability of the intensity values across each layer when compared to the original data ($p < 10^{-9}$).

2. As an application of using N3O, we showed that the results of registering OCT scans from different subjects were significantly improved when N3O was applied to both of the images before being registered ($p < 0.05$). Looking at a set of 15 subjects, the performance of the registration improved by an average of 2.08 μm for all layers when looking at the results of a label transfer experiment to produce a segmentation of the registered images. Initial experiments showed that application of N3O as a preprocessing step to the RF+GC method for layer segmentation did not yield improved results, perhaps indicating the ability for the RF to implicitly handle different types of intensity inhomogeneity.

3. As a second application, we integrated MFS into our RF+GC layer segmentation framework. The motivation for this work was to adapt the underlying graph structure to the shape of the retina, which results in a practically equivalent formulation to converting the data to MFS. Running RF+GC in MFS allows us to relax the hard constraints in the graph and incorporate soft constraints in a simple manner by penalizing vertical changes. Overall, we found that the segmentation performed 6% better when using the MFS than running on the original data, with an average

improvement in accuracy of 0.22 μm over all of the boundaries.

7.3 Clinical significance

While the work presented in this thesis did not directly study the effects of disease on the retina, it presented several methods which are helpful for answering clinical hypotheses. In many cases, we used a cohort of MS patients for validation of our algorithms, which enables us to have confidence in studies of this disease using our methods. Indeed, several studies have thus far been done using the results of our RF+GC method to look at layer thicknesses in MS. A study by Al-Louzi et al. (abstract at ECTRMS 2015 with a journal article in submission by Button et al. [150]) looked at the effect of disease modifying therapies on the rate of retinal atrophy in different layers. Their findings showed that different therapies lead to different rates of change, which might help to better understand how these therapies are affecting the course of the disease.

Another important application of our work is for multicenter studies. Such studies are becoming a more common way to study disease, since the larger sample size allows the disease to be studied in more detail. One difficulty, however, is that each center may use different scanners and protocols to acquire their data. Therefore, automated methods need to work for such varied data, and also provide consistent results so that they may be compared across sites. Since our RF+GC method utilizes a robust machine learning framework, it can be applied to any collection of images provided there is training data to teach the algorithm. In Bhargava et al. [106], it was used to compare data acquired from two different scanners. In this work, the differences in thickness measures between the scanners were less than 1.3 μm for 5 out of the 7 layers that were looked at. Another issue between clinical sites is the use of different protocols for data acquired from the same scanner. Fortunately, the images are of similar quality meaning the same algorithm can be used, but awareness of the spatial position of each B-scan becomes important, especially if the region-of-interest is slightly different. Since spatial position is encoded by the RF as a feature, our method is again easily adapted for this data.

7.4 Limitations and future work

The presented work is far from comprehensive and should be looked at with respect to the problems it was designed to solve and the data it was evaluated with. Specifically, we looked at both healthy and MS data, most of which appears visually similar. While variability exists with respect to the shape of the eye, the retinal structure remains remarkably unchanged. Any pathology in the data is likely to cause many of the developed algorithms to have problems. Specifically, the initial step of boundary estimation is not robust to these changes, as we saw with the RP data where the BrM boundary was not estimated well. Going forward, a machine learning approach to finding these boundaries would likely enhance our ability to find them in a variety of types of data.

One area that will need to be more comprehensively evaluated in the future is the FPI-based registration used in the longitudinal methods described in Chapter 4. Since accurate registration is critical for both producing accurate thickness measures, and for accurate longitudinal segmentation, we need to better understand the limits of the registration. In both cases, the methods are not robust to outlier data when the FPI generation fails due to segmentation errors and volumetric artifacts. The intensity-based method also uses FPIs generated as a combination of the blood vessel images and the thickness maps. The use of the thickness maps should ideally not be used in the registration, since the registration may try to match the thicknesses, which may not align the data well. Alternative methods like incorporating multiple scales, or combining intensity and point-based methods should be explored.

There are many areas of research left to be done with respect to MS. While our methods were validated using some of this data, little disease-specific analysis was done to look at how the disease affects the eye in the macula. While several studies exist in the literature on this topic, we believe that our improved methods for analyzing the data could lead to new and interesting results. Our preliminary work on longitudinal data indicated that our results showed more stability in finding changes over time, and therefore a larger population study should be done, looking at all of the layers instead of the few that we know to change with MS. We were also able to segment MME for the first time and are hoping to build up a larger database of subjects to do a comprehensive study of the relationship between MME and MS.

Finally, we are excited to explore the many different potential applications of MFS, which we believe can improve many automated methods designed to process macular OCT data. By design, any two volumes look similar after conversion to MFS, which enables a lot of simplicity to algorithmic development around this data. We have motivated the use of MFS by intensity correction and layer segmentation, but foresee improvements in MME segmentation, as well as in segmentation of larger cysts in the retina, and in full volumetric image registration.

7.5 Conclusion

Automated image processing of retinal OCT data is hugely important, particularly for large scale studies where lots of data needs to be processed. Being able to process this data quickly and accurately allows us to better study disease and its effects on the eye. Better accuracy not only gives more confidence in the results, but it also enables smaller sample sizes to be used when looking for changes. We have developed a set of algorithms to make these things possible. Specifically, we tackled the problems of layer segmentation, useful for computing layer thicknesses, MME segmentation, allowing us to quantify MME burden for the first time, and image registration, which aligns data from the same subject for improved longitudinal measures. We additionally extended our layer segmentation framework to work for pathological data (RP) and for longitudinal data. Going forward, we hope these tools will be useful to the community at large, and have made our work accessible to the public through open source releases at `http://www.nitrc.org/projects/aura_tools/`. Overall, better disease understanding and patient care are two of the most important outcomes of our work, and it is exciting to see how these areas are improved as a result of this work.

Bibliography

[1] C. R. Keeler, “The ophthalmoscope in the lifetime of Hermann von Helmholtz,” *Arch. Ophthalmol.*, vol. 120, no. 2, pp. 194–201, 2002.

[2] T. J. Bennett and C. J. Barry, “Ophthalmic imaging today: an ophthalmic photographer’s viewpoint - a review,” *Clin. Experiment. Ophthalmol.*, vol. 37, no. 1, pp. 2–13, 2009.

[3] D. D. Donaldson, “A new camera for stereoscopic fundus photography,” *Trans. Am. Ophthalmol. Soc.*, vol. 62, pp. 429–458, 1964.

[4] D. Huang, E. Swanson, C. Lin, J. Schuman, W. Stinson, W. Chang, M. Hee, T. Flotte, K. Gregory, C. Puliafito, and a. et, “Optical coherence tomography,” *Science*, vol. 254, no. 5035, pp. 1178–1181, 1991.

[5] J. V. Forrester, A. D. Dick, P. G. McMenamin, F. Roberts, and E. Pearlman, *The Eye: Basic Sciences in Practice*, 4th ed. Edinburgh; New York: Elsevier, 2016.

[6] S. J. Ryan, A. P. Schachat, C. P. Wilkinso, D. R. Hinton, S. R. Sadda, and P. Wiedemann, *Retina*, 5th ed. London: W.B. Saunders, 2013.

[7] S. B. Syc, S. Saidha, S. D. Newsome, J. N. Ratchford, M. Levy, E. Ford, C. M. Crainiceanu, M. K. Durbin, J. D. Oakley, S. A. Meyer, E. M. Frohman, and P. A. Calabresi, "Optical coherence tomography segmentation reveals ganglion cell layer pathology after optic neuritis," *Brain*, vol. 135, no. 2, pp. 521–533, 2012.

[8] C. A. Puliafito, M. R. Hee, C. P. Lin, E. Reichel, J. S. Schuman, J. S. Duker, J. A. Izatt, E. A. Swanson, and J. G. Fujimoto, "Imaging of macular diseases with optical coherence tomography," *Ophthalmology*, vol. 102, no. 2, pp. 217–229, 1995.

[9] J. M. Schmitt, S. H. Xiang, and K. M. Yung, "Speckle in optical coherence tomography," *J. Biomed. Opt.*, vol. 4, no. 1, pp. 95–105, 1999.

[10] National MS Society. (2016, May 19) National MS Society Fact Sheet. [Online]. Available: http://www.nationalmssociety.org

[11] E. M. Frohman, M. K. Racke, and C. S. Raine, "Multiple sclerosis—the plaque and its pathogenesis," *N. Engl. J. Med.*, vol. 354, no. 9, pp. 942–955, 2006.

[12] A. C. Arnold, "Evolving management of optic neuritis and multiple sclerosis," *Am. J. Ophthalmol.*, vol. 139, no. 6, pp. 1101–1108, 2005.

[13] E. M. Frohman, J. G. Fujimoto, T. C. Frohman, P. A. Calabresi, G. Cutter, and L. J. Balcer, "Optical coherence tomography: a window into the mechanisms of multiple sclerosis," *Nat. Clin. Pract. Neuro.*, vol. 4, no. 12, pp. 664–675, 2008.

[14] A. Petzold, J. F. de Boer, S. Schippling, P. Vermersch, R. Kardon, A. Green, P. A. Calabresi, and C. Polman, "Optical coherence tomography in multiple sclerosis: a systematic review and meta-analysis," *Lancet Neurol.*, vol. 9, no. 9, pp. 921–932, 2010.

[15] J. N. Ratchford, S. Saidha, E. S. Sotirchos, J. A. Oh, M. A. Seigo, C. Eckstein, M. K. Durbin, J. D. Oakley, S. A. Meyer, A. Conger, T. C. Frohman, S. D. Newsome, L. J. Balcer, E. M. Frohman, and P. A. Calabresi, "Active MS is associated with accelerated retinal ganglion cell/inner plexiform layer thinning," *Neurology*, vol. 80, no. 1, pp. 47–54, 2013.

[16] S. Saidha, S. B. Syc, M. A. Ibrahim, C. Eckstein, C. V. Warner, S. K. Farrell, J. D. Oakley, M. K. Durbin, S. A. Meyer, L. J. Balcer, E. M. Frohman, J. M. Rosenzweig, S. D. Newsome, J. N. Ratchford, Q. D. Nguyen, and P. A. Calabresi, "Primary retinal pathology in multiple sclerosis as detected by optical coherence tomography," *Brain*, vol. 134, no. 2, pp. 518–533, 2011.

[17] S. Saidha, E. S. Sotirchos, M. A. Ibrahim, C. M. Crainiceanu, J. M. Gelfand, Y. J. Sepah, J. N. Ratchford, J. Oh, M. A. Seigo, S. D. Newsome, L. J. Balcer, E. M. Frohman, A. J. Green, Q. D. Nguyen, and P. A. Calabresi, "Microcystic macular oedema, thickness of the inner nuclear layer of the retina, and disease characteristics in multiple sclerosis: a retrospective study," *Lancet Neurol.*, vol. 11, no. 11, pp. 963–972, 2012.

[18] D. Koozekanani, K. Boyer, and C. Roberts, "Retinal thickness measurements from optical coherence tomography using a Markov boundary model," *IEEE Trans. Med. Imag.*, vol. 20, no. 9, pp. 900–916, 2001.

[19] H. Ishikawa, D. M. Stein, G. Wollstein, S. Beaton, J. G. Fujimoto, and J. S. Schuman,

"Macular segmentation with optical coherence tomography," *Invest. Ophthalmol. Vis. Sci.*, vol. 46, no. 6, pp. 2012–2017, 2005.

[20] A. Mishra, A. Wong, K. Bizheva, and D. A. Clausi, "Intra-retinal layer segmentation in optical coherence tomography images," *Opt. Express*, vol. 17, no. 16, pp. 23 719–23 728, 2009.

[21] M. Garvin, M. Abramoff, R. Kardon, S. Russell, X. Wu, and M. Sonka, "Intraretinal layer segmentation of macular optical coherence tomography images using optimal 3-D graph search," *IEEE Trans. Med. Imag.*, vol. 27, no. 10, pp. 1495–1505, 2008.

[22] M. Garvin, M. Abramoff, X. Wu, S. Russell, T. Burns, and M. Sonka, "Automated 3-D intraretinal layer segmentation of macular spectral-domain optical coherence tomography images," *IEEE Trans. Med. Imag.*, vol. 28, no. 9, pp. 1436–1447, 2009.

[23] V. Kajić, B. Považay, B. Hermann, B. Hofer, D. Marshall, P. L. Rosin, and W. Drexler, "Robust segmentation of intraretinal layers in the normal human fovea using a novel statistical model based on texture and shape analysis," *Opt. Express*, vol. 18, no. 14, pp. 14 730–14 744, 2010.

[24] Q. Yang, C. A. Reisman, Z. Wang, Y. Fukuma, M. Hangai, N. Yoshimura, A. Tomidokoro, M. Araie, A. S. Raza, D. C. Hood, and K. Chan, "Automated layer segmentation of macular OCT images using dual-scale gradient information," *Opt. Express*, vol. 18, no. 20, pp. 21 293–21 307, 2010.

[25] S. J. Chiu, X. T. Li, P. Nicholas, C. A. Toth, J. A. Izatt, and S. Farsiu, "Automatic

segmentation of seven retinal layers in SDOCT images congruent with expert manual segmentation," *Opt. Express*, vol. 18, no. 18, pp. 19 413–19 428, 2010.

[26] I. Ghorbel, F. Rossant, I. Bloch, S. Tick, and M. Paques, "Automated segmentation of macular layers in OCT images and quantitative evaluation of performances," *Pattern Recognition*, vol. 44, no. 8, pp. 1590–1603, 2011.

[27] K. A. Vermeer, J. van der Schoot, H. G. Lemij, and J. F. de Boer, "Automated segmentation by pixel classification of retinal layers in ophthalmic OCT images," *Biomed. Opt. Express*, vol. 2, no. 6, pp. 1743–1756, 2011.

[28] B. J. Antony, M. D. Abràmoff, M. M. Harper, W. Jeong, E. H. Sohn, Y. H. Kwon, R. Kardon, and M. K. Garvin, "A combined machine-learning and graph-based framework for the segmentation of retinal surfaces in SD-OCT volumes," *Biomed. Opt. Express*, vol. 4, no. 12, pp. 2712–2728, 2013.

[29] P. A. Dufour, L. Ceklic, H. Abdillahi, S. Schroder, S. De Zanet, U. Wolf-Schnurrbusch, and J. Kowal, "Graph-based multi-surface segmentation of OCT data using trained hard and soft constraints," *IEEE Trans. Med. Imag.*, vol. 32, no. 3, pp. 531–543, 2013.

[30] Q. Song, J. Bai, M. Garvin, M. Sonka, J. Buatti, and X. Wu, "Optimal multiple surface segmentation with shape and context priors," *IEEE Trans. Med. Imag.*, vol. 32, no. 2, pp. 376–386, 2013.

[31] R. Bernardes and J. Cunha-Vaz, Eds., *Optical Coherence Tomography: A Clinical and Technical Update.* Berlin, Heidelberg: Springer, 2012.

[32] D. Fernandez, “Delineating fluid-filled region boundaries in optical coherence tomography images of the retina,” *IEEE Trans. Med. Imag.*, vol. 24, no. 8, pp. 929–945, 2005.

[33] Y. Zheng, J. Sahni, C. Campa, A. N. Stangos, A. Raj, and S. P. Harding, “Computerized assessment of intraretinal and subretinal fluid regions in spectral-domain optical coherence tomography images of the retina,” *Am. J. Ophthalmol.*, vol. 155, no. 2, pp. 277–286, 2013.

[34] X. Chen, M. Niemeijer, L. Zhang, K. Lee, M. Abramoff, and M. Sonka, “Three-dimensional segmentation of fluid-associated abnormalities in retinal OCT: Probability constrained graph-search-graph-cut,” *IEEE Trans. Med. Imag.*, vol. 31, no. 8, pp. 1521–1531, 2012.

[35] S. J. Chiu, C. A. Toth, C. B. Rickman, J. A. Izatt, and S. Farsiu, “Automatic segmentation of closed-contour features in ophthalmic images using graph theory and dynamic programming,” *Biomed. Opt. Express*, vol. 3, no. 5, pp. 1127–1140, 2012.

[36] G. Wilkins, O. Houghton, and A. Oldenburg, “Automated segmentation of intraretinal cystoid fluid in optical coherence tomography,” *IEEE Trans. Biomed. Imag.*, vol. 59, no. 4, pp. 1109–1114, 2012.

[37] M. Pilch, K. Stieger, Y. Wenner, M. N. Preising, C. Friedburg, E. Meyer zu Bexten, and B. Lorenz, “Automated segmentation of pathological cavities in optical coherence tomography scans,” *Invest. Ophthalmol. Vis. Sci.*, vol. 54, no. 6, pp. 4385–4393, 2013.

[38] D. Rueckert and J. A. Schnabel, "Medical image registration," in *Biomedical Image Processing*, M. T. Deserno, Ed. Berlin, Heidelberg: Springer, 2011, pp. 131–154.

[39] M. Niemeijer, M. K. Garvin, K. Lee, B. van Ginneken, M. D. Abràmoff, and M. Sonka, "Registration of 3D spectral OCT volumes using 3D SIFT feature point matching," in *Proc. SPIE Med. Imag.*, vol. 7259, 2009, p. 72591I.

[40] M. Niemeijer, K. Lee, M. K. Garvin, M. D. Abràmoff, and M. Sonka, "Registration of 3D spectral OCT volumes combining ICP with a graph-based approach," in *Proc. SPIE Med. Imag.*, vol. 8314, 2012, p. 83141A.

[41] J. Wu, B. Gerendas, S. Waldstein, G. Langs, C. Simader, and U. Schmidt-Erfurth, "Stable registration of pathological 3D SD-OCT scans using retinal vessels," in *Proc. Ophthalmic Medical Image Analysis 1st Int. Workshop, Held in Conjunction with MICCAI 2014*, Boston, MA, 2014.

[42] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2262–2275, 2010.

[43] W.-D. Vogl, S. Waldstein, B. Gerendas, C. Simader, A.-M. Glodan, D. Podkowinski, U. Schmidt-Erfurth, and G. Langs, "Spatio-temporal signatures to predict retinal disease recurrence," in *Proc. Information Processing in Medical Imaging (IPMI)*, vol. 9123, 2015, pp. 152–163.

[44] T. M. Jørgensen, J. Thomadsen, U. Christensen, W. Soliman, and B. Sander, "Enhancing the signal-to-noise ratio in ophthalmic optical coherence tomography by image

registration—method and clinical examples," *J. Biomed. Opt.*, vol. 12, no. 4, pp. 041 208–041 208–10, 2007.

[45] D. Alonso-Caneiro, S. A. Read, and M. J. Collins, "Speckle reduction in optical coherence tomography imaging by affine-motion image registration," *J. Biomed. Opt.*, vol. 16, no. 11, pp. 116 027–116 027–5, 2011.

[46] Y. Li, G. Gregori, B. L. Lam, and P. J. Rosenfeld, "Automatic montage of SD-OCT data sets," *Opt. Express*, vol. 19, no. 27, pp. 26 239–26 248, 2011.

[47] H. Bogunović, M. Sonka, Y. Kwon, P. Kemp, M. Abramoff, and X. Wu, "Multi-surface and multi-field co-segmentation of 3-D retinal optical coherence tomography," *IEEE Trans. Med. Imag.*, vol. 33, no. 12, pp. 2242–2253, 2014.

[48] M. Chen, A. Lang, H. S. Ying, P. A. Calabresi, J. L. Prince, and A. Carass, "Analysis of macular OCT images using deformable registration," *Biomed. Opt. Express*, vol. 5, no. 7, pp. 2196–2214, 2014.

[49] Y. Zheng, R. Xiao, Y. Wang, and J. C. Gee, "A generative model for OCT retinal layer segmentation by integrating graph-based multi-surface searching and image registration," in *Proc. MICCAI*, 2013, pp. 428–435.

[50] S. Lee, E. Lebed, M. Sarunic, and M. Beg, "Exact surface registration of retinal surfaces from 3-D optical coherence tomography images," *IEEE Trans. Biomed. Imag.*, vol. 62, no. 2, pp. 609–617, 2015.

[51] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[52] A. Lang, A. Carass, E. Sotirchos, P. Calabresi, and J. L. Prince, “Segmentation of retinal OCT images using a random forest classifier,” in *Proc. SPIE Med. Imag.*, vol. 8669, 2013, p. 86690R.

[53] A. Lang, A. Carass, M. Hauser, E. S. Sotirchos, P. A. Calabresi, H. S. Ying, and J. L. Prince, “Retinal layer segmentation of macular OCT images using boundary classification,” *Biomed. Opt. Express*, vol. 4, no. 7, pp. 1133–1152, 2013.

[54] A. Lang, A. Carass, A. Bittner, J. Prince, and H. Ying, “Automated segmentation of eight layers in 3D SD-OCT scans of retinitis pigmentosa patients and the relationship of GCL+IPL thickness to visual function,” in *Proc. ARVO 2015 Annual Meeting*. Denver, CO: Association for Research in Vision and Ophthalmology, May 2015.

[55] A. Lang, A. Carass, O. Al-Louzi, P. Bharghava, H. S. Ying, P. A. Calabresi, and J. L. Prince, “Longitudinal graph-based segmentation of macular OCT using fundus alignment,” in *Proc. SPIE Med. Imag.*, vol. 9413, 2015, p. 94130M.

[56] A. Lang, A. Carass, O. Al-Louzi, P. Bharghava, S. D. Solomon, P. A. Calabresi, and J. L. Prince, “Combined registration and motion correction of longitudinal retinal OCT data,” in *Proc. SPIE Med. Imag.*, vol. 9784, 2016, p. 97840X.

[57] E. K. Swingle, A. Lang, A. Carass, P. A. Calabresi, H. S. Ying, and J. L. Prince, “Microcystic macular edema detection in retina OCT images,” in *Proc. SPIE Med. Imag.*, vol. 9038, 2014, p. 90380O.

[58] A. Lang, A. Carass, E. K. Swingle, O. Al-Louzi, P. Bhargava, S. Saidha, H. S. Ying,

P. A. Calabresi, and J. L. Prince, “Automatic segmentation of microcystic macular edema in OCT,” *Biomed. Opt. Express*, vol. 6, no. 1, pp. 155–169, 2015.

[59] J. G. Sled, A. P. Zijdenbos, and A. C. Evans, “A nonparametric method for automatic correction of intensity nonuniformity in MRI data,” *IEEE Trans. Med. Imag.*, vol. 17, no. 1, pp. 87–97, 1998.

[60] A. Lang, A. Carass, P. A. Calabresi, H. S. Ying, and J. L. Prince, “An adaptive grid for graph-based segmentation in retinal OCT,” in *Proc. SPIE Med. Imag.*, vol. 9034, 2014, p. 903402.

[61] A. Lang, A. Carass, B. Jedynak, S. D. Solomon, P. A. Calabresi, and J. L. Prince, “Intensity inhomogeneity correction of macular OCT using N3 and retinal flatspace,” in *Proc. IEEE Int. Symp. Biomed. Imag.*, 2016, pp. 197–200.

[62] A. Lang, A. Carass, B. M. Jedynak, S. S. Solomon, P. A. Calabresi, and J. L. Prince, “Intensity inhomogeneity correction of SD-OCT data using macular flatspace,” *IEEE Trans. Med. Imag.*, 2016, In Submission.

[63] K. Li, X. Wu, D. Chen, and M. Sonka, “Optimal surface segmentation in volumetric images - a graph-theoretic approach,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 119–134, 2006.

[64] S. B. Kotsiantis, “Supervised machine learning: A review of classification techniques,” *Informatica*, vol. 31, pp. 249–268, 2007.

[65] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, pp. 273–297, 1995.

[66] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, pp. 297–336, 1999.

[67] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[68] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. New York: CRC press, 1984.

[69] A. Criminisi and J. Shotton, Eds., *Decision Forests for Computer Vision and Medical Image Analysis*. London: Springer, 2013.

[70] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[71] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. New York: Springer, 2009.

[72] T. Grubinger, A. Zeileis, and K.-P. Pfeiffer, "evtree: Evolutionary learning of globally optimal classification and regression trees in R," *Journal of Statistical Software*, vol. 61, no. 1, pp. 1–29, 2014.

[73] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning," *Foundations and Trends in Computer Graphics and Vision*, vol. 7, no. 2-3, pp. 81–227, 2012.

[74] S. Bernard, L. Heutte, and S. Adam, “A study of strength and correlation in random forests,” in *Proc. 6th International Conference on Intelligent Computing*, vol. 93, 2010, pp. 186–191.

[75] A. Liaw and M. Wiener, “Classification and regression by randomforest,” *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[76] M. R. Segal, “Machine learning benchmarks and random forest regression,” University of California, San Francisco, Tech. Rep., 2004. [Online]. Available: http://escholarship.org/uc/item/35x3v9t4

[77] Y. Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in ND images,” in *Proc. IEEE International Conference on Computer Vision*, vol. 1, 2001, pp. 105–112.

[78] D. M. Greig, B. T. Porteous, and A. H. Seheult, “Exact maximum a posteriori estimation for binary images,” *J. R. Stat. Soc. Series B*, vol. 51, no. 2, pp. 271–279, 1989.

[79] V. Kolmogorov and R. Zabih, “Computing visual correspondence with occlusions using graph cuts,” in *Proc. IEEE International Conference on Computer Vision*, vol. 2, 2001, pp. 508–515.

[80] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, 2001.

[81] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, 2004.

[82] L. R. Ford, Jr. and D. R. Fulkerson, *Flows in Networks.* Princeton, NJ: Princeton University Press, 1962.

[83] Y. Yin, X. Zhang, R. Williams, X. Wu, D. D. Anderson, and M. Sonka, “Logismos—layered optimal graph image segmentation of multiple objects and surfaces: Cartilage segmentation in the knee joint,” *IEEE Trans. Med. Imag.*, vol. 29, no. 12, pp. 2023–2037, 2010.

[84] A. Delong and Y. Boykov, “Globally optimal segmentation of multi-region objects,” in *Proc. IEEE International Conference on Computer Vision*, 2009, pp. 285–292.

[85] J.-K. Wang, R. H. Kardon, M. J. Kupersmith, and M. K. Garvin, “Automated quantification of volumetric optic disc swelling in papilledema using spectral-domain optical coherence tomography volumetric quantification of optic disc swelling,” *Invest. Ophthalmol. Vis. Sci.*, vol. 53, no. 7, pp. 4069–4075, 2012.

[86] Y. Boykov and O. Veksler, “Graph cuts in vision and graphics: Theories and applications,” in *Handbook of Mathematical Models in Computer Vision*, N. Paragios, Y. Chen, and O. Faugeras, Eds. New York: Springer, 2006.

[87] J.-C. Picard, “Maximal closure of a graph and applications to combinatorial problems,” *Management Science*, vol. 22, no. 11, pp. 1268–1272, 1976.

[88] D. S. Hochbaum, “A new–old algorithm for minimum-cut and maximum-flow in closure graphs,” *Networks*, vol. 37, no. 4, pp. 171–193, 2001.

[89] H. Ishikawa, “Exact optimization for Markov random fields with convex priors,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1333–1336, 2003.

[90] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, 1990.

[91] M. A. Mayer, J. Hornegger, C. Y. Mardin, and R. P. Tornow, “Retinal nerve fiber layer segmentation on FD-OCT scans of normal subjects and glaucoma patients,” *Biomed. Opt. Express*, vol. 1, no. 5, pp. 1358–1383, 2010.

[92] M. S. Nixon and A. S. Aguado, *Feature Extraction & Image Processing for Computer Vision*, 3rd ed. Academic Press, 2012.

[93] M. Varma and A. Zisserman, “Texture classification: Are filter banks necessary?” in *Proc. IEEE Computer Vision and Pattern Recognition*, 2003, pp. 691–698.

[94] J.-M. Geusebroek, A. Smeulders, and J. van de Weijer, “Fast anisotropic Gauss filtering,” *IEEE Trans. Imag. Proc.*, vol. 12, no. 8, pp. 938–943, 2003.

[95] M. Varma and A. Zisserman, “A statistical approach to texture classification from single images,” *Int. J. Comput. Vision*, vol. 62, no. 1-2, pp. 61–81, 2005.

[96] J. Canny, “A computational approach to edge detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, 1986.

[97] J. Huang, X. Liu, Z. Wu, H. Xiao, L. Dustin, and S. Sadda, “Macular thickness measurements in normal eyes with time-domain and fourier-domain optical coherence tomography,” *Retina*, vol. 29, no. 7, pp. 980–987, 2009.

[98] ETDRS Research Group, “Photocoagulation for diabetic macular edema. early treatment diabetic retinopathy study report number 1.” *Arch Ophthalmol*, vol. 103, pp. 1796–1806, 1985.

[99] A. Jaiantilal, “Classification and regression by randomforest-matlab,” 2009, http://code.google.com/p/randomforest-matlab.

[100] T. Sharp, “Implementing decision trees and forests on a GPU,” in *Proc. European Conference on Computer Vision (ECCV)*, 2008, pp. 595–608.

[101] D. Zikic, B. Glocker, E. Konukoglu, A. Criminisi, C. Demiralp, J. Shotton, O. Thomas, T. Das, R. Jena, and S. Price, “Decision forests for tissue-specific segmentation of high-grade gliomas in multi-channel MR,” in *Proc. MICCAI*, 2012, pp. 369–376.

[102] J. Lee, S. Chiu, P. Srinivasan, J. Izatt, C. Toth, S. Farsiu, and G. Jaffe, “Fully automatic software for retinal thickness in eyes with diabetic macular edema from images acquired by Cirrus and Spectralis systems,” *Invest. Ophthalmol. Vis. Sci.*, vol. 54, no. 12, pp. 7595–7602, 2013.

[103] Q. Yang, C. Reisman, K. Chan, R. Ramachandran, A. Raza, and D. Hood, “Automated segmentation of outer retinal layers in macular OCT images of patients with retinitis pigmentosa,” *Biomed. Opt. Express*, vol. 2, no. 9, pp. 2493–2503, 2011.

[104] R. Ramachandran, L. Zhou, K. Locke, D. Birch, and D. Hood, "A comparison of methods for tracking progression in X-linked retinitis pigmentosa using frequency domain OCT," *Trans. Vis. Sci. Tech.*, vol. 2, no. 7, pp. 1–9, 2013.

[105] A. Santos, M. Humayun, E. de Juan Jr, R. Greenburg, M. Marsh, I. Klock, and A. Milam, "Preservation of the inner retina in retinitis pigmentosa: A morphometric analysis," *Arch. Ophthalmol.*, vol. 115, no. 4, pp. 511–515, 1997.

[106] P. Bhargava, A. Lang, O. Al-Louzi, A. Carass, J. L. Prince, P. A. Calabresi, and S. Saidha, "Applying an open-source segmentation algorithm to different OCT devices in multiple sclerosis patients and healthy controls: implications for clinical trials," *Multiple Sclerosis International*, vol. Article ID 136295, pp. 1–10, 2015.

[107] B. J. Antony, A. Lang, E. K. Swingle, O. Al-Louzi, A. Carass, S. D. Solomon, P. A. Calabresi, S. Saidha, and J. L. Prince, "Simultaneous segmentation of retinal surfaces and microcystic macular edema in SDOCT volumes," in *Proc. SPIE Med. Imag.*, vol. 9784, 2016, p. 97841C.

[108] L. S. Talman, E. R. Bisker, D. J. Sackel, D. A. Long, K. M. Galetta, J. N. Ratchford, D. J. Lile, S. K. Farrell, M. J. Loguidice, G. Remington, A. Conger, T. C. Frohman, D. A. Jacobs, C. E. Markowitz, G. R. Cutter, G.-S. Ying, Y. Dai, M. G. Maguire, S. L. Galetta, E. M. Frohman, P. A. Calabresi, and L. J. Balcer, "Longitudinal study of vision and retinal nerve fiber layer thickness in multiple sclerosis," *Ann. Neurol.*, vol. 67, no. 6, pp. 749–760, 2010.

[109] R. J. Zawadzki, A. R. Fuller, S. S. Choi, D. F. Wiley, B. Hamann, and J. S. Werner,

"Correction of motion artifacts and scanning beam distortions in 3D ophthalmic optical coherence tomography imaging," in *Proc. SPIE*, vol. 6426, 2007, p. 642607.

[110] M. F. Kraus, B. Potsaid, M. A. Mayer, R. Bock, B. Baumann, J. J. Liu, J. Hornegger, and J. G. Fujimoto, "Motion correction in optical coherence tomography volumes on a per A-scan basis using orthogonal scan patterns," *Biomed. Opt. Express*, vol. 3, no. 6, pp. 1182–1199, 2012.

[111] J. Xu, H. Ishikawa, G. Wollstein, L. Kagemann, and J. Schuman, "Alignment of 3-D optical coherence tomography scans to correct eye movement using a particle filtering," *IEEE Trans. Med. Imag.*, vol. 31, no. 7, pp. 1337–1345, 2012.

[112] A. Montuoro, J. Wu, S. Waldstein, B. Gerendas, G. Langs, C. Simader, and U. Schmidt-Erfurth, "Motion artefact correction in retinal optical coherence tomography using local symmetry," in *Proc. MICCAI*, vol. 8674, 2014, pp. 130–137.

[113] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," in *Proc. MICCAI*, vol. 1496, 1998, pp. 130–137.

[114] P. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, 1992.

[115] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. R. Stat. Soc. Ser. B*, vol. 58, pp. 267–288, 1994.

[116] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *J. Stat. Softw.*, vol. 33, no. 1, pp. 1–22, 2010.

[117] B. J. Lujan, A. Roorda, R. W. Knighton, and J. Carroll, “Revealing Henle’s fiber layer using spectral domain optical coherence tomography,” *Invest. Ophthalmol. Vis. Sci.*, vol. 52, no. 3, p. 1486, 2011.

[118] R. F. Spaide and C. A. Curcio, “Anatomical correlates to the bands seen in the outer retina by optical coherence tomography: Literature review and model,” *Retina*, vol. 31, no. 8, pp. 1609–1619, 2011.

[119] J. Petersen, M. Modat, M. Cardoso, A. Dirksen, S. Ourselin, and M. de Bruijne, “Quantitative airway analysis in longitudinal studies using groupwise registration and 4D optimal surfaces,” in *Proc. MICCAI*, vol. 8150, 2013, pp. 287–294.

[120] J. M. Gelfand, R. Nolan, D. M. Schwartz, J. Graves, and A. J. Green, “Microcystic macular oedema in multiple sclerosis is associated with disease severity,” *Brain*, vol. 135, pp. 1786–1793, 2012.

[121] P. Bhargava and P. A. Calabresi, “The expanding spectrum of aetiologies causing retinal microcystic macular change,” *Brain*, vol. 136, no. 11, pp. 3212–3214, 2013.

[122] F. Kaufhold, H. Zimmermann, E. Schneider, K. Ruprecht, F. Paul, T. Oberwahrenbrock, and A. U. Brandt, “Optic neuritis is associated with inner nuclear layer thickening and microcystic macular edema independently of multiple sclerosis,” *PLoS ONE*, vol. 8, no. 8, p. e71145, 2013.

[123] E. K. Swingle, A. Lang, A. Carass, O. Al-Louzi, P. A. Calabresi, and J. L. Prince, “Segmentation of microcystic macular edema in Cirrus OCT scans with an exploratory longitudinal study,” in *Proc. SPIE Med. Imag.*, vol. 9417, 2015, p. 94170P.

[124] M. C. Burggraaff, J. Trieu, W. A. E. J. de Vries-Knoppert, L. Balk, and A. Petzold, "The clinical spectrum of microcystic macular edema," *Invest. Ophthalmol. Vis. Sci.*, vol. 55, no. 2, pp. 952–961, 2014.

[125] J. Talairach and P. Tournoux, *Co-planar Stereotaxic Atlas of the Human Brain.* New York: Thieme, 1988.

[126] D. L. Collins, P. Neelin, T. M. Peters, and A. C. Evans, "Automatic 3D intersubject registration of MR volumetric data in standardized Talairach space." *J. Comput. Assist. Tomogr.*, vol. 18, no. 2, pp. 192–205, 1994.

[127] K. A. Vermeer, J. van der Schoot, H. G. Lemij, and J. F. de Boer, "RPE-normalized RNFL attenuation coefficient maps derived from volumetric OCT imaging for glaucoma assessment," *Invest. Ophthalmol. Vis. Sci.*, vol. 53, no. 10, p. 6102, 2012.

[128] B. E. Varga, W. Gao, K. L. Laurik, E. Tátrai, M. Simó, G. M. Somfai, and D. Cabrera DeBuc, "Investigating tissue optical properties and texture descriptors of the retina in patients with multiple sclerosis," *PLoS ONE*, vol. 10, no. 11, pp. 1–20, 2015.

[129] J. F. Arevalo, Ed., *Retinal Angiography and Optical Coherence Tomography.* New York: Springer, 2009.

[130] K. A. Vermeer, J. Mo, J. J. A. Weda, H. G. Lemij, and J. F. de Boer, "Depth-resolved model-based reconstruction of attenuation coefficients in optical coherence tomography," *Biomed. Opt. Express*, vol. 5, no. 1, pp. 322–337, 2014.

[131] W. Drexler and J. G. Fujimoto, Eds., *Introduction to Optical Coherence Tomography.* Berlin, Heidelberg: Springer, 2008.

[132] M. J. A. Girard, N. G. Strouthidis, C. R. Ethier, and J. M. Mari, "Shadow removal and contrast enhancement in optical coherence tomography images of the human optic nerve head," *Invest. Ophthalmol. Vis. Sci.*, vol. 52, no. 10, p. 7738, 2011.

[133] M. F. Kraus, J. J. Liu, J. Schottenhamml, C.-L. Chen, A. Budai, L. Branchini, T. Ko, H. Ishikawa, G. Wollstein, J. Schuman, J. S. Duker, J. G. Fujimoto, and J. Hornegger, "Quantitative 3D-OCT motion correction with tilt and illumination correction, robust similarity measure and regularization," *Biomed. Opt. Express*, vol. 5, no. 8, pp. 2591–2613, 2014.

[134] J. Novosel, G. Thepass, H. G. Lemij, J. F. de Boer, K. A. Vermeer, and L. J. van Vliet, "Loosely coupled level sets for simultaneous 3D retinal layer segmentation in optical coherence tomography," *Med. Image Anal.*, vol. 26, no. 1, pp. 146–158, 2015.

[135] N. J. Tustison, B. B. Avants, P. A. Cook, Y. Zheng, A. Egan, P. A. Yushkevich, and J. C. Gee, "N4ITK: Improved N3 bias correction," *IEEE Trans. Med. Imag.*, vol. 29, no. 6, pp. 1310–1320, 2010.

[136] D. Kaba, Y. Wang, C. Wang, X. Liu, H. Zhu, A. G. Salazar-Gonzalez, and Y. Li, "Retina layer segmentation using kernel graph cuts and continuous max-flow," *Opt. Express*, vol. 23, no. 6, pp. 7366–7384, 2015.

[137] C.-L. Chen, H. Ishikawa, G. Wollstein, R. A. Bilonick, I. A. Sigal, L. Kagemann, and J. S. Schuman, "Histogram matching extends acceptable signal strength range on optical coherence tomography images histogram matching extends OCT data usability," *Invest. Ophthalmol. Vis. Sci.*, vol. 56, no. 6, p. 3810, 2015.

[138] J. B. Arnold, J.-S. Liow, K. A. Schaper, J. J. Stern, J. G. Sled, D. W. Shattuck, A. J. Worth, M. S. Cohen, R. M. Leahy, J. C. Mazziotta, and D. A. Rottenberg, “Qualitative and quantitative evaluation of six algorithms for correcting intensity nonuniformity effects,” *NeuroImage*, vol. 13, no. 5, pp. 931–943, 2001.

[139] Z. Hou, “A review on MR image intensity inhomogeneity correction,” *International Journal of Biomedical Imaging*, vol. 2006, pp. 1–11, 2006.

[140] F. N. Fritsch and R. E. Carlson, “Monotone piecewise cubic interpolation,” *SIAM J. Numerical Analysis*, vol. 17, pp. 238–246, 1980.

[141] A. Carass, A. Lang, M. Hauser, P. A. Calabresi, H. S. Ying, and J. L. Prince, “Multiple-object geometric deformable model for segmentation of macular OCT,” *Biomed. Opt. Express*, vol. 5, no. 4, pp. 1062–1074, 2014.

[142] P. H. C. Eilers, I. D. Currie, and M. Durbán, “Fast and compact smoothing on large multidimensional grids,” *Comput. Stat. Data Anal.*, vol. 50, no. 1, pp. 61–76, 2006.

[143] P. Serranho, C. Maduro, T. Santos, J. Cunha-Vaz, and R. Bernardes, “Synthetic OCT data for image processing performance testing,” in *Proc. 18th IEEE Int. Conf. Image Process.*, 2011, pp. 409–412.

[144] Z. Y. Chua, W. Zheng, M. W. Chee, and V. Zagorodnov, “Evaluation of performance metrics for bias field correction in MR brain images,” *J. Magn. Reson. Imaging*, vol. 29, no. 6, pp. 1271–1279, 2009.

[145] I. Isgum, M. Staring, A. Rutten, M. Prokop, M. A. Viergever, and B. van Ginneken, “Multi-atlas-based segmentation with local decision fusion — application to cardiac

and aortic segmentation in CT scans," *IEEE Trans. Med. Imag.*, vol. 28, no. 7, pp. 1000–1010, 2009.

[146] M. Cabezas, A. Oliver, X. Lladó, J. Freixenet, and M. B. Cuadra, "A review of atlas-based segmentation for magnetic resonance brain images," *Computer Methods and Programs in Biomedicine*, vol. 104, no. 3, pp. e158–e177, 2011.

[147] S. E. Jones, B. R. Buchbinder, and I. Aharon, "Three-dimensional mapping of cortical thickness using Laplace's equation," *Human Brain Mapping*, vol. 11, no. 1, pp. 12–32, 2000.

[148] K. Rocha, A. Yezzi, and J. Prince, "A hybrid Eulerian-Lagrangian approach for thickness, correspondence, and gridding of annular tissues," *IEEE Trans. Med. Imag.*, vol. 16, no. 3, pp. 636–648, 2007.

[149] O. Acosta, P. Bourgeat, M. A. Zuluaga, J. Fripp, O. Salvado, and S. Ourselin, "Automated voxel-based 3D cortical thickness measurement in a combined Lagrangian-Eulerian PDE approach using partial volume maps," *Med. Image Anal.*, vol. 13, no. 5, pp. 730–743, 2009.

[150] J. Button, O. Al-Louzi, A. Lang, P. Bhargava, S. D. Newsome, T. Frohman, L. J. Balcer, E. M. Frohman, J. Prince, P. A. Calabresi, and S. Saidha, "Disease modifying therapies modulate retinal atrophy in multiple sclerosis: a retrospective study," *Neurology*, 2016, In Submission.

Vita

Andrew R. Lang was born on September 27, 1984 in New London, CT. He grew up in Waterford, CT, graduating from Waterford High School in 2003. For his undergraduate studies, he attended Rensselaer Polytechnic Institute (RPI), graduating in 2007 with a Bachelor of Science degree in Electrical Engineering. Next, he received a Master of Applied Science degree in Electrical and Computer Engineering from Queen's University in Kingston, ON, Canada under the co-supervision of Dr. Purang Abolmaesumi and Dr. Parvin Mousavi. Since 2010, he has attended Johns Hopkins University in Baltimore, MD where he has been pursuing a Ph.D. in Electrical and Computer Engineering under the supervision of Dr. Jerry L. Prince. While there, he was named the Willis C. Gore Fellow in the Whiting School of Engineering. His research interests are in medical image analysis with a primary focus on analyzing retinal optical coherence tomography (OCT) images. He also enjoys working in the areas of machine learning and statistical modeling.