Contributions to Structural and Functional Retinal Imaging via Fourier Domain Optical Coherence Tomography

by

Jason Michael Tokayer

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)

December 2013

Acknowledgements

I would like to take this opportunity to express my gratitude and appreciation to a number of people who supported me over the years. I would first like to thank my co-advisors, Dr. Antonio Ortega and Dr. David Huang, for their insight, guidance and patience as I navigated down a windy road towards a Ph.D. Their vastly different mentoring styles equally enriched my graduate school experience. Not only did they help me achieve my Ph.D, they also prepared me for my career beyond graduate school.

I would like to thank Dr. B. Keith Jenkins both for serving on my dissertation committee as well for exploring new avenues of research in biomedical optics with me. I would also like to thank Dr. Bartlett Mel for the insightful and thought provoking questions that he posed during my dissertation defense.

I am also grateful for a number of my colleagues. Dr. Pankaj Mishra was by my side at the beginning of my graduate school tenure and taught me by example how to adapt to a life of research. Dr. Yali Jia showed me how to transform my research ideas and lab work into significant publications. Also, Dr. Al-Hafeez Dhalla has been there to answer all of my questions no matter how ridiculous they were. He has been especially inspirational with his thoughtfulness and rigorousness. I would also like to thank Garth Tormoen for repeatedly drawing my blood so that I could complete my experimental work.

I would also like to thank my family and my wife's family for their support.

Finally, and most importantly, I would like to thank my wife Alana. Her support and encouragement has been the primary driving force behind my push through graduate school.

Her tolerance of my occasional bad moods and sleepless nights is a testament to her devotion and love. I am especially grateful for the time she took to read and revise my thesis.

Table of Contents

List of Figures

List of Tables

Abstract

Optical coherence tomography (OCT) is a depth-resolved cross-sectional imaging modality capable of imaging biological tissue at the micron-scale *in vivo*. OCT has found widespread use in ophthalmology because of the translucent nature of retinal tissue as well as OCT's high resolution and non-contact implementation. OCT is analogous to ultrasound, except that it uses light instead of sound to illuminate a sample. OCT measures travel time of light that is backscattered from structures at different depths or axial distances (axial scans) within the sample. Cross-sectional images are generated by scanning the optical beam in the transverse direction and performing successive axial scan measurements.

Structural imaging of the human retina has provided new insights into ophthalmic disease. For example, retinal nerve fiber layer thickness is utilized in glaucoma progression analysis. Thus, segmentation of OCT retinal images, which is complicated by low image contrast and high levels of speckle noise, is crucial for the study and diagnosis of ocular diseases. Measurement of retinal blood flow is also important for the study of many ocular pathologies including glaucoma and diabetic retinopathy. Doppler OCT, which utilizes measured Doppler frequency/phase shifts to infer blood flow speed, has consequently found widespread use in the ophthalmic community. The Doppler effect is only sensitive to movement along the incident beam's axis. In the case of the human macular region, however, many blood vessels are nearly perpendicular to the incident beam, which makes the measured Doppler frequency shifts small and difficult to detect. For this reason, measurement of blood flow in the macula using Doppler OCT is a challenging task.

This thesis encompasses contributions to both structural and functional OCT retinal imaging. One common theme found throughout this thesis is the generation of synthetic or simulated OCT data in order to quantitatively evaluate various image processing algorithms. For each simulation application the synthetic data is first validated against *in vivo* retinal data or *in vitro* experimental data and then used to extrapolate the results and draw inferences about algorithm performance.

We first design and implement a novel sparsity-based retinal layer segmentation algorithm that identifies the boundaries between retinal layers in OCT structural images. In contrast to most state of the art retinal layer segmentation algorithms, our algorithm does not require overly restrictive *a priori* assumptions about the expected layer structure and can thus possibly be used to detect unexpected structures that arise in pathological cases. One difficulty with quantitative evaluation of segmentation algorithms applied to *in vivo* retinal images is that the ground truth segmentation is not known. Motivated by this fact, we design a novel method for generating synthetic structural retinal images for which the ground truth segmentation is known. Our technique improves upon an existing synthetic image generation method both by using a nonparametric smooth representation of the average intensity values in the image and by accurately modeling speckle noise. After verifying that synthetic images generated with our method accurately represent real *in vivo* retinal images, we show that our segmentation algorithm accurately identifies the retinal layer boundaries in the synthetic images and that our algorithm is robust to image noise.

We next examine functional OCT retinal imaging by studying various methods for measuring blood flow velocities in retinal vessels. We develop a two dimensional simulation of a blood vessel immersed in tissue that mimics functional imaging in a clinical setting more accurately than existing one dimensional simulations of retinal blood flow. This retinal vessel simulation enables quantitative evaluation of the accuracy of Doppler OCT algorithms by providing the ground truth blood flow velocities that Doppler OCT techniques aim to measure. We illustrate the utility of the simulation with two case studies. First, we evaluate the accuracy of two commonly used Doppler OCT algorithms as vessel diameters and associated flow speeds become increasingly small. We show that the phase-resolved Doppler OCT algorithm performs best for large blood vessels while neither algorithm performs well for very small blood vessels and slow blood flow velocities. This is the first time that a quantitative evaluation of the estimation accuracy of the Doppler OCT algorithms is performed using synthetic data with known ground truth blood flow velocities. In the second case study we examine the effect of transverse step size between adjacent axial scans on the phase-resolved Doppler OCT algorithm. We show that the phase-resolved Doppler OCT algorithm systematically underestimates blood flow velocities with increasing transverse step size and that this effect is more pronounced at higher velocities. We propose two techniques for correcting measured velocities that can be used to improve the accuracy of the phase-resolved Doppler OCT algorithm.

As previously mentioned, Doppler OCT techniques are only sensitive to velocities along the direction of the incident beam (axial velocities). While Doppler OCT is commonly used to measure blood flow near the optic disc, its application to flow estimation in the human macula is limited because many of the blood vessels in this region are nearly perpendicular to the incident beam. In order to overcome this limitation, we introduce a novel method for visualizing blood flow called split-spectrum amplitude-decorrelation angiography (SSADA). In contrast to Doppler phase-based methods, SSADA relies on intensity information only and is sensitive to both transverse (perpendicular) and axial flow velocities. After establishing that SSADA can be used to image flow in the human macula, we perform phantom flow experiments to establish a linear relationship between SSADA measurements and preset blood flow velocities that enables velocity quantification and thus blood flow measurement in the human macula with SSADA.

Chapter 1

Introduction

1.1 OCT Background

Optical coherence tomography (OCT) is a biomedical imaging technique based on low coherence interferometry capable of producing both structural and functional images of biological tissue with micrometer resolution [1]. Depth gating within an imaged sample is achieved by interfering broadband light of low coherence backscattered from a sample and a reference mirror. When the pathlengths of backscattered light from the reference and sample arms are within a source coherence length, the light will interfere and produce a depth resolved backscattered intensity profile of the sample. OCT measures the time of flight of light backscattered from within a sample. Cross-sectional OCT images are generated by scanning the optical beam transversally. An illustration of OCT retinal image formation is shown in Figure 1.1.

Depth resolutions of typical OCT imaging systems range from $5-10$ μm in air. Ultrahigh-resolution OCT imaging systems can achieve even better resolution. Depth penetration into biological tissue is on the order of a couple of millimeters. Compared to ultrasound imaging, OCT has lower penetration depth but better resolution. Compared to confocal microscopy, OCT has better penetration depth but lower resolution [2].

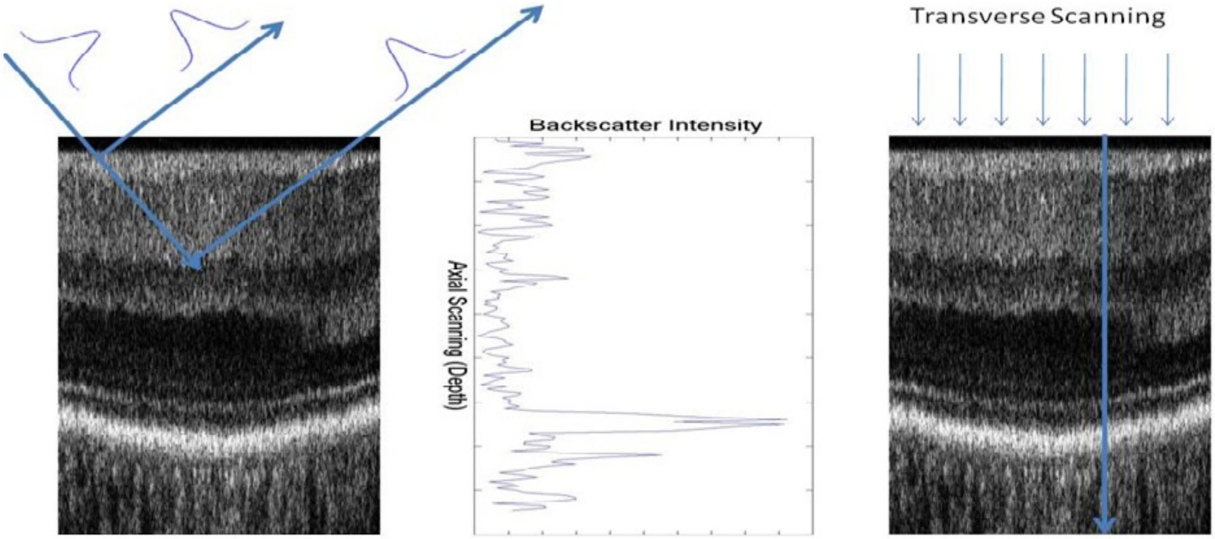


Figure 1.1: Formation of OCT image. OCT measures the time of flight of backscattered light and can resolve backscattered light from different depths within a sample. Left panel: light backscattered from deeper in the sample takes longer to return. Middle panel: Example intensity depth profile from the vertical blue line in the right panel. Right panel: transverse scanning produces a cross sectional image

First generation OCT systems used a mechanically scanned reference mirror to obtain backscattered reflectivity from different depths within a sample. Since the signals were acquired sequentially in time, these systems are known as time-domain systems. Because of the mechanical nature of scanning as well as the low imaging speeds and corresponding high imaging times, *in vivo* measurements suffered from much patient movement. More recently, OCT detection techniques were devised that do not require mechanical scanning of a reference mirror. This resulted in a significant increase in imaging speeds, ten to fifty times faster. In contrast to time-domain OCT, which detects light at different depths sequentially, these newer detection techniques measure all of the depths simultaneously by acquiring backscattered light as a function of frequency and using the Fourier transform relation between it and time/depth. Thus, these techniques are termed Fourier-domain optical coherence tomography (FDOCT). With high imaging speeds, FDOCT now enables *in vivo* three-dimensional imaging [2].

OCT has found widespread use in ophthalmology because of the translucent nature of retinal tissue as well as its high resolution and non-contact implementation. OCT tomograms enable quantification of morphological and functional characteristics of the human retina, some of which enhance the diagnostic capabilities for various eye diseases. For example, retinal nerve fiber layer thickness is utilized in glaucoma progression analysis. An illustration of the human eye is shown in Figure 1.2.

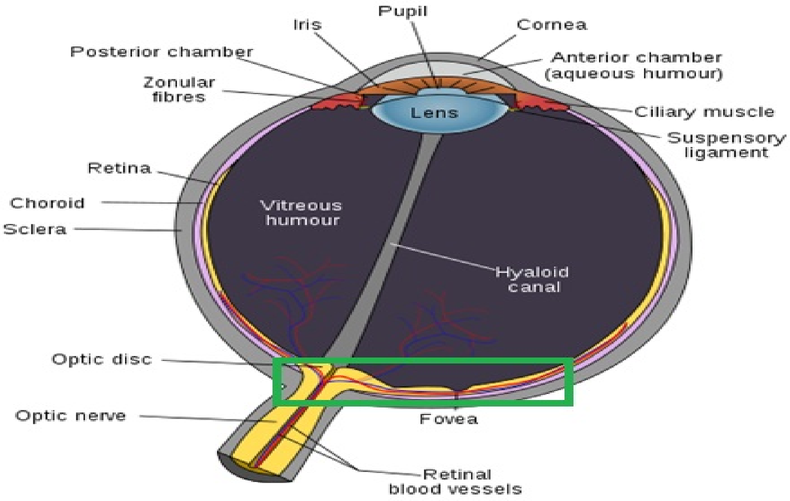


Figure 1.2: The human eye (from [3]). The focus of this work is the macula with the fovea at its center, which is denoted by a green box.

The green box highlights the posterior eye, which is the region of interest for the contributions of this work. The retina is a complex, light-sensitive, layered tissue that lines the inner surface of the eye. Most of this work focuses on the central subregion of the retina, the macula lutea. The most light sensitive structure in the macula is the fovea that is approximately 200 μm thick [4]. Nerve fiber layer thickness in the human macula is commonly used as a marker for glaucoma [5]. Also, reduced blood flow in the macula may be related to diabetic retinopathy [6].

1.2 Scope and Contributions

Various studies of both structural and functional human retinal imaging are presented in this thesis. A common theme throughout most of this work is the use of synthetic data to provide quantitative evaluation of various image processing algorithms used for retinal layer segmentation and the study of retinal blood flow.

The following list provides a summary of the key contributions presented in this thesis.

- We design and implement a novel sparsity-based retinal segmentation algorithm for OCT structural images. A Sparse Bayesian Learning algorithm is used to identify the discontinuities between adjacent retinal layers. The boundaries between retinal layers are then found using a graph theoretic approach. The algorithm is able to segment the nine layer boundaries without making overly restrictive assumptions about the retinal anatomy. This is the first segmentation algorithm that does not specify *a priori* the number of layers to be found. Rather, the algorithm is guided by the data, which makes it more general than many OCT retinal segmentation algorithms so that the segmentation of pathological cases could possibly be performed with our algorithm.

- We present a technique for generating synthetic OCT structural images. A model of OCT reflectivities is defined assuming locally homogeneous regions of reflectivity. Speckle patterns are implemented via simulation. Additive complex Gaussian noise is generated to complete the model. The intensity distributions of the synthetic data are validated with *in vivo* retinal images. The spatial correlation of the images, or speckle sizes, are also validated with *in vivo* retinal images. Our technique accurately models the speckle noise pattern that is determined by the OCT imaging system parameters. These synthetic images provide a ground truth segmentation that can be used for quantitative evaluation of various retinal layer segmentation algorithms. We show that our segmentation algorithm accurately identifies the retinal layer boundaries in the synthetic images and that our algorithm is robust to image noise.

- We develop a novel two dimensional simulation of a blood vessel immersed in tissue. Our simulation mimics functional imaging in a clinical setting more accurately than existing one dimensional simulations of retinal blood flow. We subsequently use the simulation to generate synthetic OCT retinal blood flow velocity data that enables quantitative evaluation of Doppler OCT velocity estimation algorithms. We present two case studies that illustrate the utility of the retinal vessel simulation:
    - We present a quantitative comparison between two Doppler OCT algorithms (phase-resolved versus moving-scatterer-sensitive). After validating the simulation with *in vivo* retinal data, we show that the phase-resolved algorithm outperforms the moving-scatterer-sensitive algorithm for large vessel diameters and blood flow velocities. We also show that neither algorithm provides accurate blood flow estimates for small vessel diameters and slow blood flow velocities, indicating that Doppler OCT has limited utility for flow measurements in the human macula where blood velocities are small.
    - We study the effect of transverse step size on measured flow velocities in phase-resolved Doppler OCT. We show that the phase-resolved Doppler OCT algorithm systematically underestimates blood flow velocities with increasing transverse step size and that this effect is more pronounced at higher velocities. We propose two techniques for correcting measured velocities that can be used to improve the accuracy of the phase-resolved Doppler OCT algorithm.

- Motivated by the limited application of Doppler OCT algorithms for macular flow estimation, we introduce a novel method for visualizing blood flow called split-spectrum amplitude-decorrelation angiography (SSADA). We illustrate the ability of the SSADA algorithm to visualize the capillary network in the human macula. Then, using flow phantom experiments, we derive a linear model that relates the SSADA measurements to flow velocities. This model enables flow velocity quantification with SSADA and could potentially be applied for macular blood flow quantification.

The organization of this thesis is as follows. In Chapter 2 the theory of optical coherence tomography is discussed. Chapter 3 focuses on the segmentation of OCT structural images. First, we introduce our novel retinal layer segmentation algorithm. We then develop our synthetic image generation technique and test the accuracy and noise robustness of our segmentation algorithm on synthetic OCT structural images for which the ground truth segmentation is known. In Chapter 4 we introduce the retinal vessel simulation and present two cases studies that illustrate its utility. In Chapter 5 we discuss the SSADA algorithm. Chapter 6 presents conclusions and future work.

Chapter 2

OCT Theory

Most of the following analysis is based on [2]. A generic OCT system schematic is illustrated in Figure 2.1.

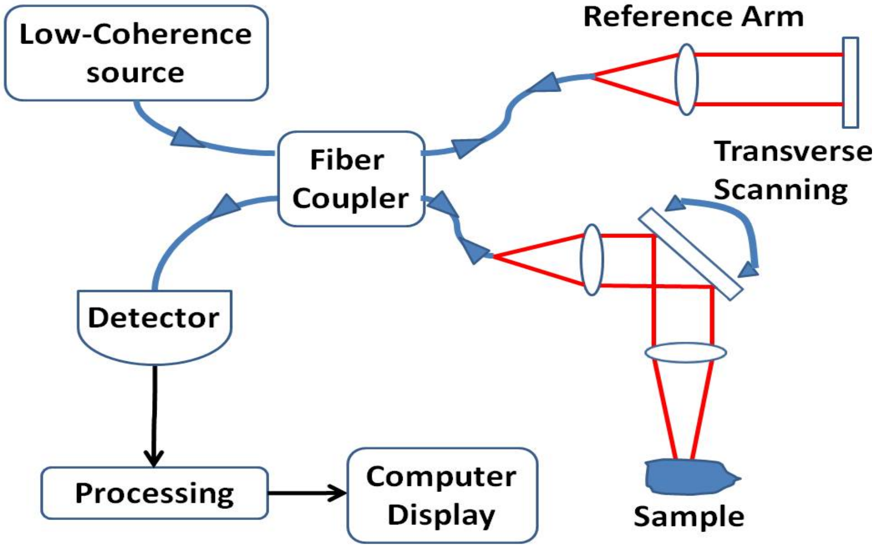


Figure 2.1: Fiber optic OCT system schematic. Blue lines indicate fiber optic paths, red lines represent free-space optical paths and black lines represent electronic signal paths

Light from a low-coherence source is directed into a fiber optic coupler that splits the light into a reference and sample arm. The reference arm light travels through the fiber and illuminates a mirror. The light is then backreflected back into the fiber and travels back towards the coupler. The sample arm light travels through the fiber and is focused onto a sample upon exiting the fiber. Backscattered light is collected by the fiber and combined with the reference arm light at the coupler. The combined light is detected by the detector, which passes it into a processing module that computes the depth resolved reflectivity profile of the sample. Depth profiles are commonly referred to as axial scans or A-lines. The input beam is incident on the sample along the axial direction and this is typically taken to be the Z direction. The plane that is perpendicular to the input beam is referred to as the transverse plane. The lateral scanner in Figure 2.1 can be used to alter the transverse position of the beam in either the X or Y directions. Typically, a two dimensional cross-sectional image or B-scan is acquired by scanning the beam along the X direction and computing A-lines at equally spaced points along the trajectory. A volumetric scan is acquired by incrementing the Y position of the beam after each B-scan acquisition. This raster scan pattern is shown in Figure 2.2.

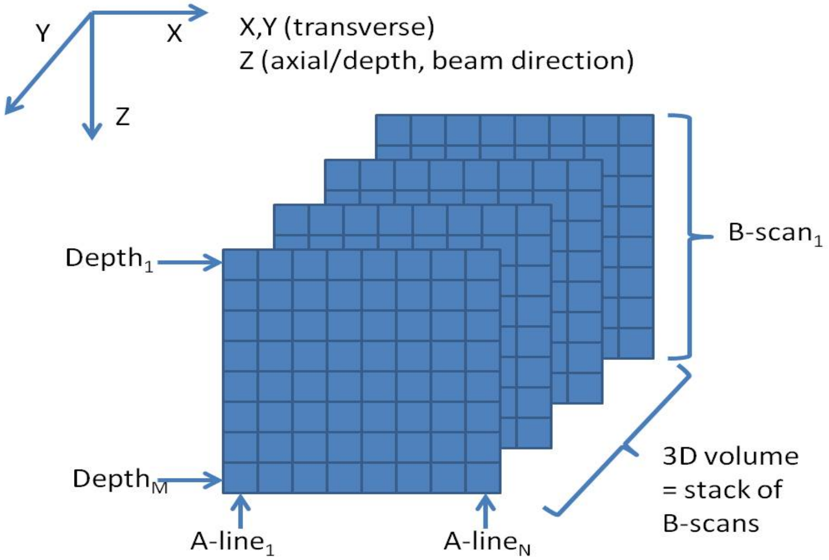


Figure 2.2: Raster scan pattern illustration. A depth (Z) profile is referred to as an axial scan or A-line. Scanning the input beam along the X (transverse) direction and acquiring A-lines at equally spaced points along the trajectory produces a cross-sectional image or B-scan. In this graphic, a B-scan has N A-lines, each of which contains a signal from M different depths. Incrementing the Y (trnsverse) position of the beam after each B-scan acquisition produces a volumetric scan.

A few additional scan patterns used in this thesis are briefly mentioned next. First, M-mode imaging refers to repeated A-line acquisitions without scanning the beam transversally so that multiple A-lines are acquired at the same transverse beam position over time. Acquiring multiple B-scans in sequence is called M-B mode scanning. In some cases, the volumetric data acquired during a raster scan is aggregated over depth, either by averaging or maximal projections, to create a transverse only en-face image. Finally, a circular scan protocol refers to the case for which the beam is scanned in a circle in the transverse plane.

2.1 Fourier Domain OCT

2.1.1 Axial Ranging

In the following theoretical discussion about axial ranging with OCT, the transverse beam extent is ignored. Consider the Michelson interferometer in Figure 2.3.

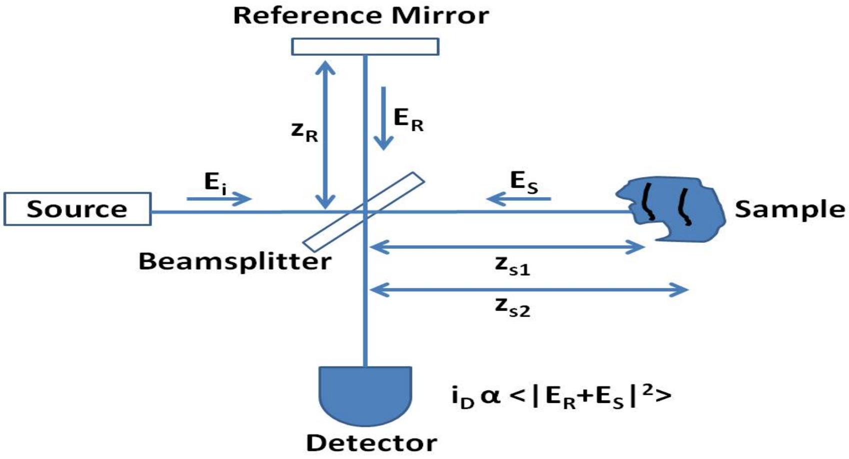


Figure 2.3: Schematic of Michelson interferometer used in OCT. The $z = 0$ position is located at the beamsplitter.

The illumination beam of the interferometer is a polychromatic plane wave with electric field given by:

$$E_i = s(k, \omega)\exp(i(kz - \omega t)), \tag{2.1}$$

where $s(k, \omega)$ is the amplitude, k is the wavenumber and ω is the angular frequency. The wavenumber is related to the source wavelength λ by $k = 2\pi/\lambda$ and the angular frequency is related to the temporal frequency by $\omega = 2\pi\nu$. The wavenumber and angular frequency are coupled by the index of refraction $n(\omega)$ through:

$$k(\omega) = \frac{n(\omega)\omega}{c}, \tag{2.2}$$

where c is the speed of light. For this theoretical introduction, dispersion is neglected so that $k = n\omega/c$. Let r_R denote the amplitude reflectivity of a reference mirror placed at a distance z_R from the beam splitter. The electric field arriving at the beamsplitter after reflecting off of the mirror can be written as:

$$E_R \propto E_i r_R \exp(ik2z_R), \tag{2.3}$$

where the factor of 2 results from the round-trip pathlength to and from the mirror. The sample is characterized by a depth dependent reflectivity $r_S(z_S)$. For simplicity, this reflectivity is approximated with a series of N discrete reflectors at different depths. Thus, the sample reflectivity can be written as:

$$r_S(z_S) = \sum_{n=1}^{N} r_{S_n}\delta(z_S - z_{S_n}), \tag{2.4}$$

where $\delta(x)$ denotes the discrete Dirac delta function and equals 1 at $x = 0$ and is zero everywhere else [7]. The reflected beam arriving at the beamsplitter from the sample is the coherent sum of all of the reflected beams from different depths within the sample and can be written as:

$$E_S \propto E_i \sum_{n=1}^{N} r_{S_n} \exp(ik2z_{S_n}). \tag{2.5}$$

The photodetector is a square law detector which means that it detects the square of the incident electric field or its intensity. Let $R_R = |r_R|^2$ denote the power reflectivity of the reference mirror and $R_{S_n} = |r_{S_N}|^2$ denote the power reflectivity for each reflector. The photocurrent generated by the superimposed sample and reference fields is given by:

$$\begin{aligned} I_D(k) &\propto \rho\langle|E_R + E_S|^2\rangle \\ &\propto \rho\langle|E_R|^2 + |E_S|^2 + 2\Re\{E_R E_S^\star\}\rangle \\ &\propto \rho\left[S(k)\left(R_R + R_{S_1} + R_{S_2} + ...\right)\right] \\ &\quad +\rho\left[S(k)\textstyle\sum_{n=1}^{N}\sqrt{R_R R_{S_n}}\left(\cos(2k(z_R - z_{S_n}))\right)\right] \\ &\quad +\rho\left[S(k)\textstyle\sum_{n\neq m=1}^{N}\sqrt{R_{S_n} R_{S_m}}\left(\cos(2k(z_{S_n} - z_{S_m}))\right)\right], \end{aligned} \tag{2.6}$$

where $S(k) = \langle|s(k,\omega)|^2\rangle$ encodes the power spectral density of the light source ρ is the responsivity of the detector and $\Re$ denotes the real part operator. In Equation 2.6 the summation consists of three distinct terms: the first term is the DC term and has no associated phase, the second term is the cross-correlation or interference term that represents the interference between the reference and sample beams, and the third term is the autocorrelation term that represents the interference of the sample beam with itself. In this theoretical analysis the spectral density will be approximated by a Gaussian function. Then, the spectral density and its Fourier transform pair can be written as [7]:

$$\begin{aligned} S(k) &= \mathcal{F}(\gamma(z)) \propto \exp(-4\ln2\left(\tfrac{k-k_0}{\Delta k}\right)^2), \\ \gamma(z) &= \mathcal{F}^{-1}(S(k)) \propto \exp(-z^2\Delta k^2), \end{aligned} \tag{2.7}$$

where $k_0 = 2\pi/\lambda_0$ denotes the source center wavenumber/wavelength, Δk denotes the full-width at half-maximum (FWHM) of the source spectral density and the phase delay is dropped. The function $\gamma(z)$ is called the coherence function and is characterized by its FWHM δz_0, which defines the round trip coherence length of the light source. The coherence length can be written as a function of the source spectral bandwidth as:

$$\delta z_0 = \sqrt{\frac{4\text{ln}2}{\Delta k^2}}. \tag{2.8}$$

In Fourier domain OCT, the photocurrent in Equation 2.6 is captured and processed using the Fourier transform to reconstruct an approximation to the sample reflectivity profile $\sqrt{R_S(z_S)}$. In spectral domain OCT (SDOCT), the combined beam is spectrally dispersed by a spectrometer and captured on a charge coupled device (CCD) camera. In swept source OCT (SSOCT), the spectral components are captured sequentially on a single detector by sweeping the source wavenumber.

Making use of the convolution property of the Fourier transform as well as the Fourier transform of a cosine function [7], the inverse Fourier transform of Equation 2.6 can be calculated as:

$$\begin{aligned} I_D(z) \quad &\propto \quad [\gamma(z)\left(R_R + R_{S_1} + R_{S_2} + ...\right)] \\ &+ \quad \left[\gamma(z) * \textstyle\sum_{n=1}^{N} \sqrt{R_R R_{S_n}}\left(\delta(z \pm 2(z_R - z_{S_n})\right)\right] \\ &+ \quad \left[\gamma(z) * \textstyle\sum_{n\neq m=1}^{N} \sqrt{R_{S_n} R_{S_m}}\left(\delta(z \pm 2(z_{S_n} - z_{S_m})\right)\right], \end{aligned} \tag{2.9}$$

where the ρ term was dropped and $*$ denotes a convolution. Using the sifting property of the Dirac delta function [7] this can be rewritten as:

$$\begin{aligned} I_D(z) \quad \propto & \quad [\gamma(z)\,(R_R + R_{S_1} + R_{S_2} + ...)] \\ + & \quad \left[\textstyle\sum_{n=1}^{N} \sqrt{R_R R_{S_n}}\,(\gamma(z \pm 2(z_R - z_{S_n}))\right] \\ + & \quad \left[\textstyle\sum_{n \neq m=1}^{N} \sqrt{R_{S_n} R_{S_m}}\,(\gamma(z \pm 2(z_{S_n} - z_{S_m}))\right]. \end{aligned} \tag{2.10}$$

Equation 2.10 is referred to as an OCT A-scan and is the point spread function (PSF) of an OCT system. It represents a single depth profile. A few important results from Equation 2.10 are discussed next. First, the zero position of the A-line appears at the position of the reference reflector z_R. Second, the displacement of each sample reflector from the reference position is doubled due to the round trip distance to each reflector. Third, there are mirror image artifacts due to the fact that the Fourier transform of a real signal is symmetric. Most importantly, each reflector is broadened out to a width equal to the coherence length due to the convolution with the coherence function. The coherence length δz_0 is thus defined as the axial resolution of an OCT imaging system. By Equation 2.8, a larger source bandwidth leads to a shorter coherence length. Thus, in order to increase the system axial resolution, the source bandwidth should be very large. The fundamental quality that differentiates OCT from other forms of optical microscopy is its ability to resolve in depth in this way. A sample interferogram and resulting A-line with reflectors located at two different depths are illustrated in Figure 2.4.

Typically, the DC term had the largest magnitude and the autocorrelation terms are negligible compared to the interferometric term. In order to recover the interferometric term and the sample reflectivity profile $\sqrt{R_S(z_S)}$, a background scan is first acquired with the signal from the sample arm blocked, which leaves only the DC term. Then, after unblocking the sample arm and acquiring a spectral interferogram, the background is subtracted and

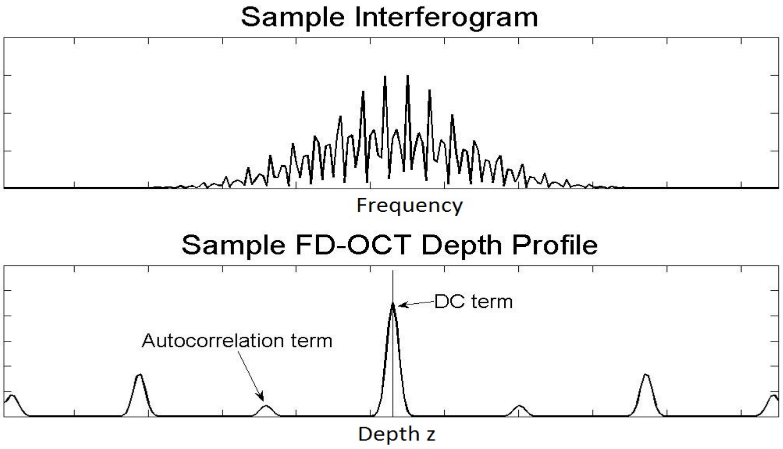


Figure 2.4: Sample A-Scan for SD-OCT. Two discrete reflectors are located at two different depths. In the interferogram in wavenumber/frequency space, this can be seen from the two modulating frequencies of the Gaussian source spectrum. In the depth profile, or A-line, the discrete reflectors are broadened by the finite axial resolution δz_0.

the resulting signal is Fourier transformed. In this way the DC term is removed and the resulting signal represents the sample reflectivity profile $\sqrt{R_S(z_S)}$.

2.1.2 Transverse Resolution

The transverse resolution of OCT is defined as the focal spot size of the incident beam on the sample. If the source emits a Gaussian beam [8] then the $1/e^2$ intensity transverse resolution is given by:

$$\omega_0 = \frac{4\lambda_0 f}{\pi D}, \tag{2.11}$$

where f denotes the focal length of the objective lens that focuses light onto the sample and D denotes the diameter of the collimated beam on the lens. The transverse resolution is independent of the axial resolution. This is in contrast to conventional microscopy where these values are coupled. The depth of focus, or the range over which the beam size remains approximately equal to the spot size, varies with the square of the transverse resolution. Thus, there is a tradeoff between the focal diameter and the focal zone of the sample beam;

the smaller the focal diameter (or the better the transverse resolution), the shorter the focal zone.

2.2 Practical Spectral-Domain OCT Processing

2.2.1 Coherence Volume

In practice, focused light reflected from the sample emanates from within a three dimensional coherence volume, defined by the beam width ω_0 in the transverse dimensions X and Y and by the coherence length δz_0 in the axial dimension Z. An example coherence volume is illustrated in Figure 2.5.

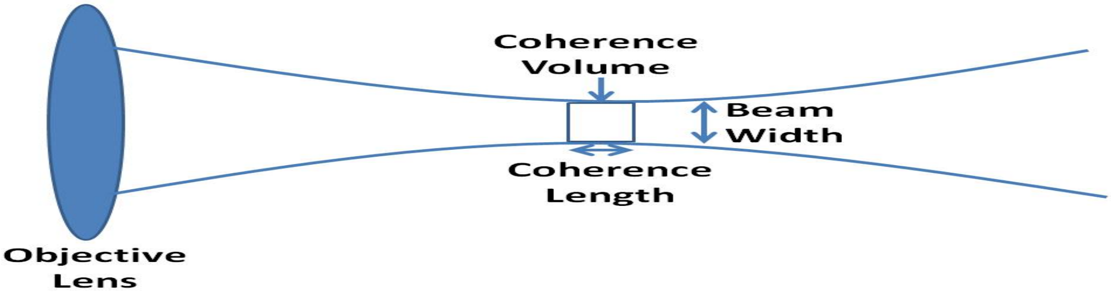


Figure 2.5: Coherence volume definition. Note that the beam width is the transverse extent in both the X and Y (not shown) dimensions.

Neglecting the DC and autocorrelation terms, the interferometric signal as a function of wavenumber emanating from a coherenec volume can be written as [9]:

$$I(x_b, y_b, k) \propto \Re\left\{\iiint S(k) r(x, y, z) e^{-4\ln 2\left(\frac{(x-x_b)^2+(y-y_b)^2}{\omega_0^2}\right)} e^{-i2kz} dx dy dz\right\}, \tag{2.12}$$

where (x_b, y_b) denotes the transverse position of the beam, (x, y, z) denotes the coordinate of a reference frame fixed to the sample, $r(x, y, z)$ is the backscattering coefficient of the sample,

ω_0 is the $1/e^2$ diameter of the beam intensity profile and the other terms have been previously defined. Equation 2.12 assumes that the zero path length difference in the interferometer is located at the beginning of the sample, or $z = 0$.

The complex OCT signal as a function of depth Z is obtained after Fourier transformation of Equation 2.12 and is given by:

$$I(x_b, y_b, Z) \propto \iiint r(x, y, z) e^{-i2k_0(z-Z)} e^{-8\left(\frac{(x-x_b)^2+(y-y_b)^2}{\omega_0^2}+\right)} e^{-4\ln2\left(\frac{(z-Z)^2}{\delta z_0^2}\right)} dxdydz, \qquad (2.13)$$

where δz_0 denotes the FWHM axial resolution. The amplitude of $I(x_b, y_b, Z)$ is proportional to a coherent sum of all of the light backscattered from within a coherence volume centered at $Z = z$.

2.2.2 Resampling and Dispersion Compensation

In practice, the reconstruction of an A-line is complicated by the fact that (a) the acquired spectra are unevenly spaced in wavenumber and (b) there is a dispersion mismatch between the reference and sample arm. Both of these effects lead to a broadening of the PSF in the axial dimension [2]. The resampling error leads to a depth dependent broadening, whereas the dispersion error leads to a depth independent broadening. The spectra in Equation 2.12 are evenly spaced in the array pixels p of the CCD camera. Thus, the acquired wavenumbers are a function of array pixels, or $k = k(p)$. The dispersion effect is manifested in an additional nonlinear phase term in Equation 2.12.

The purpose of resampling and dispersion compensation is to recover $I(k)$ that is evenly spaced in k prior to the Fourier transform computation in Equation 2.13. Most techniques to overcome the resampling problem rely on a precise calibration of the acquired wavelengths in the spectrometer using a diffraction grating equation [10]. For the experiments performed in this work, a different technique is used that does not rely on the precise spectrometer alignment. OCT interferograms are computed using two mirror reflections at different depths z_1 and z_2. In this case, the sample reflectivity is given by $r(x_b, y_b, z_j) = \delta(x_b, y_b, z_j)$ for $j = \{1, 2\}$. Plugging this into Equation 2.12, substituting $k(p)$ for k and adding the dispersion effect gives:

$$\tilde{I}(x_b, y_b, k(p)) \propto \Re\left\{S(k(p))e^{-i2k(p)z_j}e^{i\theta(k(p))}\right\}. \tag{2.14}$$

Only the real part of the signal is measured. However, the complex signal in Equation 2.14 can be recovered using the Hilbert transform [7]. After Hilbert transformation, the phases ϕ_j of the resulting signals are computed, unwrapped and subtracted to give:

$$\Delta\phi(k(p)) = 2k(p)(z_1 - z_2) = 2k(p)\Delta z, \tag{2.15}$$

where the fact that the dispersion is independent of depth has been used so that $\theta(k)$ vanishes. Using the known physical distance Δz, Equation 2.15 can be solved for $k(p)$, which gives the unevenly spaced wavenumbers as a function of CCD array pixel. The total wavenumber range is given by $k(p_{max}) - k(p_{min}) = K$. If there are A array pixels on the CCD camera, then the evenly spaced wavenumbers can be found by:

$$k'_a = k(p_{min}) + (a-1)\frac{K}{A-1}, \tag{2.16}$$

for $a = \{1, 2, ..., A\}$. The uneven wavenumber sampling issue is resolved by resampling the interferogram in Equation 2.14 to the evenly spaced wavenumbers in Equation 2.16 using spline interpolation. This resampling removes the depth dependent broadening, as illustrated in Figure 2.6.

After the resampling is complete, dispersion compensation is performed. The relation between the phase $\theta(k)$ and the multiple orders of dispersion can be described by the Taylor series expansion as:

$$\theta(k) = \theta(k_0) + \theta^{'}(k_0)(k - k_0) + a_2 \cdot \theta^{''}(k_0)(k - k_0)^2 + a_3 \cdot \theta^{'''}(k_0)(k - k_0)^3 + ..., \qquad (2.17)$$

where $a_n = (n!)^{-1}$. The first two terms describe a constant offset and group velocity, respectively, and are not related to dispersive broadening. The third term represents second order or group-velocity dispersion. Dispersion mismatch in sample and reference arms is largely compensated by this term, although adjustment of higher order dispersion can be necessary as well, especially when an ultra-broadband source is used. The nonlinear terms in Equation 2.17 lead to dispersive broadening and must be compensated for by multiplying the resampled interferogram by a corrective phase factor $\theta_c(k)$. In order to compute this correction, the Hilbert transform of a single mirror reflection and its unwrapped phase are computed. Then, a linear fit ϕ_{LIN} is applied to the unwrapped phase. The residual of this curve, which represents the nonlinear terms in Equation 2.17 that lead to dispersive broadening is assigned to $\theta_c(k)$. Then, to compensate each A-line, each resampled interferogram is multiplied by $e^{-i\theta_c(k)}$ to compensate for dispersion. Dispersion compensation leads to decrease in the axial width of the point spread function, as shown in Figure 2.6.

2.2.3 Retinal OCT Imaging

Retinal images are acquired by using the eye as the sample and focusing the beam onto the retina. Structural images are obtained by taking the magnitude of Equation 2.13. This process is illustrated in Figure 2.7

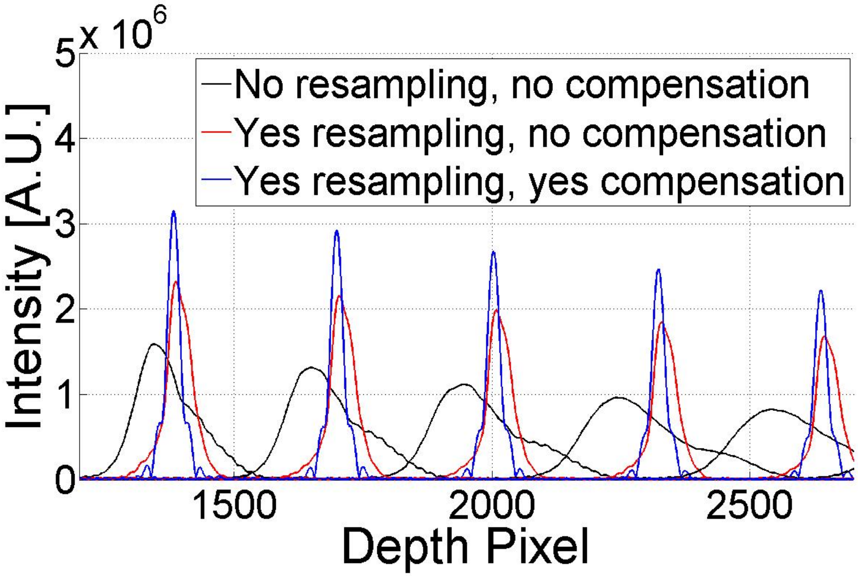


Figure 2.6: Resampling and dispersion compensation example. The original point spread functions (PSFs) broaden with depth. After resampling, the depth dependent broadening is removed. With resampling and dispersion compensation, the widths of the PSFs are small indicating good axial resolution. Note that the signal intensity falls off with depth. A complete treatment of this phenomenon is given in [2].

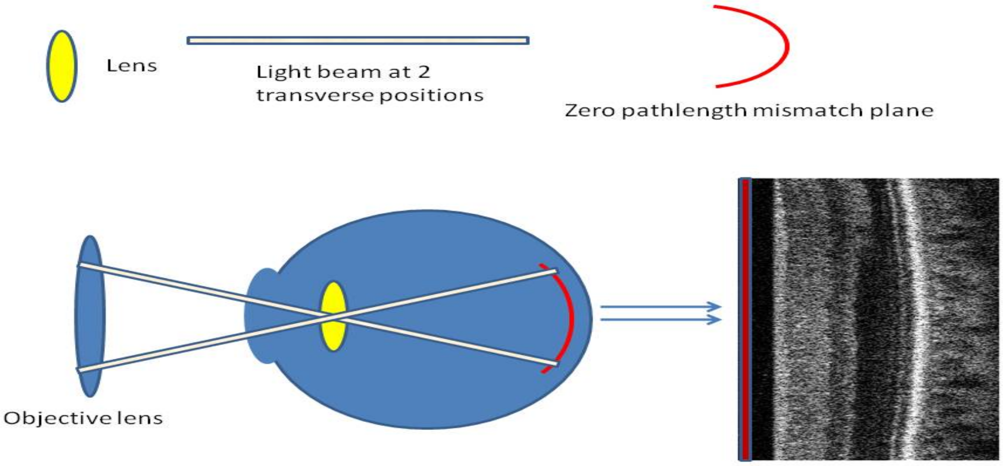

Lens
Light beam at 2 transverse positions
Zero pathlength mismatch plane
Objective lens

Figure 2.7: Cross-sectional imaging of the human retina.

2.2.4 Laboratory SDOCT System

The flow experiments presented in subsequent chapters of this thesis were performed using a laboratory SDOCT system that is illustrated in Figure 2.8. It contains a superluminescent diode source with a center wavelength of 840 μm and a bandwidth of 49 μm. The theoretical axial resolution of the system is 6.4 μm in air, while the measured axial resolution is 8.8 μm in air. The deviation from the theoretical value is likely due to the non-Gaussian spectrum profile as well as aberrations that reduce the spectral resolution of the spectrometer. A collimator is placed in the sample arm that produced a $1/e^2$ intensity diameter of approximately 1.0 mm, and is focused down to 20 μm spot size using a 20 mm focal length lens. The sample arm probe beam is focused onto a glass capillary tube (Wale Apparatus) with an outer diameter of 330 μm and an inner diameter of 200 μm. No anti-reflection material was used to coat the glass. The capillary is placed on top of a piece of paper and attached to a ball and socket mounting platform (Thorlabs). A syringe pump (Harvard Apparatus) is used to control the flow of human blood through the tube. The recombined field is spectrally dispersed by a spectrometer and detected by 1024 pixels on a line-scan camera. The data from the camera is transferred to a personal computer for data processing. The time between two sequential A-scan acquisitions is 56 μs which corresponds to a 17 kHz repetition rate.

2.3 Doppler OCT

Doppler OCT is a functional extension of OCT that enables the measurement of flow in biological tissues [11]. Doppler OCT provides both structural and functional information as a function of depth. The high imaging speeds of spectral-domain OCT make it possible to use OCT for *in vivo* blood flow measurements [12].

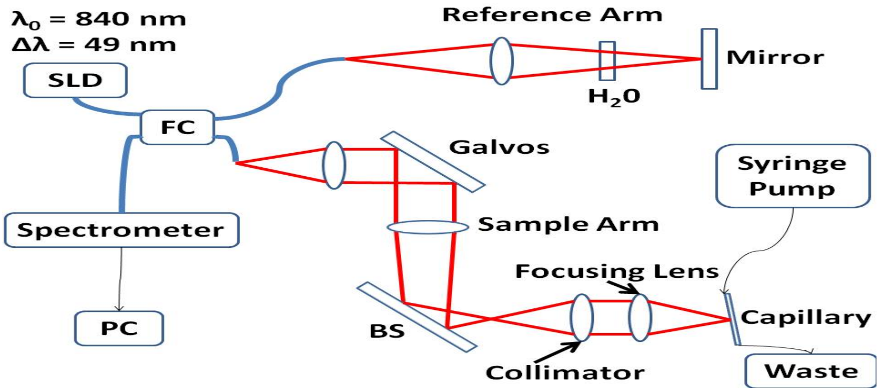


Figure 2.8: Spectral domain OCT system schematic used in flow phantom experiments. SLD superluminescent diode, FC 2x2 fiber coupler, PC personal computer, H_20 water cell, Galvos scanning mirror galvanometers, BS - beamsplitter. This system was originally designed for retinal imaging so that a water cell is in the reference arm for dispersion matching and a beamsplitter is in the sample arm to allow for slit lamp illumination of the retina. The retinal imaging system was modified by adding a focusing lens in the sample arm.

Doppler Effect

Doppler OCT relies on the Doppler Effect which states that moving particles will induce a Doppler frequency shift in scattered light according to [12]:

$$\Delta f = \frac{1}{2\pi}(\vec{k}_s - \vec{k}_i) \cdot \vec{V}, \tag{2.18}$$

where $\vec{k}_i$ and $\vec{k}_s$ are the respective wave vectors for the incident and scattered light and $\vec{V}$ is the velocity vector of the moving particles. In OCT, only backscattered light is detected and therefore the direction of $\vec{k}_s$ is the opposite of the direction of $\vec{k}_i$. Using this relation and the Doppler angle θ between the incident beam and the velocity vector Equation 2.18 can be rewritten as:

$$\Delta f = -2V cos(\theta) n/\lambda_0, \tag{2.19}$$

where n is the index of refraction of the medium and λ_0 is the center wavelength of the source. Notice that the Doppler angle is needed to calculate the total velocity vector. Figure 2.9 illustrates the Doppler effect.

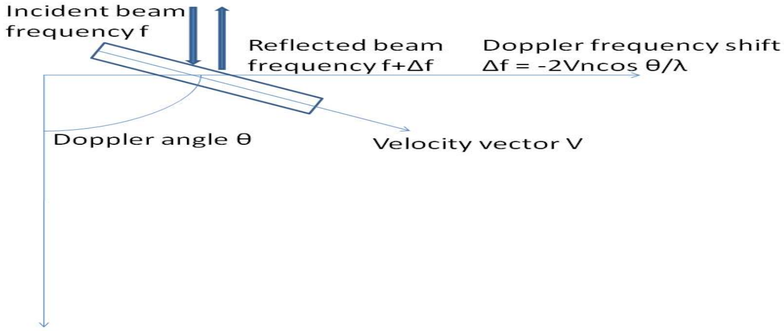


Figure 2.9: Illustration of the Doppler Effect. Moving red blood cells inside a blood vessel induces a Doppler frequency shift in the backscattered light. The Doppler frequency shift is proportional to the axial velocity and is thus insensitive to transverse motion.

Phase-Resolved Algorithm

The most common method for calculating Doppler frequency shifts in moving blood vessels is the phase-resolved algorithm [13–17]. Suppose that a single particle at position (x_j, y_j, z_j) within the sample backscatters the incident beam. If the particle moves a distance (dx, dy, dz) between two sequential A-line acquisitions A_j and A_{j+1} then the complex OCT amplitudes in Equation 2.13 can be written as:

$$\begin{aligned} A_j(Z) &\propto a_j(x_j, y_j, z_j)e^{i2k_0(z_j-Z)} \\ A_{j+1}(Z) &\propto a_{j+1}(x_{j+1}, y_{j+1}, z_{j+1})e^{i2k_0(z_{j+1}-Z)} \\ &\approx a_j(x_j, y_j, z_j)e^{i2k_0(z_j+dz-Z)}, \end{aligned} \tag{2.20}$$

provided that the amplitude difference between the two A-lines is small relative to the phase effect. The principle behind the phase-resolved (PR) algorithm is that the phase change between sequential A-line scans at the same depth can be used to calculate the Doppler frequency shift as:

$$\Delta f = \frac{\angle A_{j+1} - \angle A_j}{2\pi\tau} = \frac{\Delta\phi}{2\pi\tau}, \tag{2.21}$$

where τ is the time difference between A-scans. Ideally, the A-scans in Equation 2.21 would be acquired at the same transverse beam position. In practice, however, because of total imaging time and OCT imaging system limitations, the A-lines are acquired with a transverse distance offset Δx. The effect of this transverse step size will be studied in Section 4.4.

In general, the average phase shifts between sets of A-scans is used to improve velocity estimation. Because of phase wrapping, the PR algorithm can uniquely detect frequency shifts up to $\Delta f = 1/2\tau$, which is approximately 9 kHz for the laboratory phantom system described previously. Clearly then a shorter time between A-scans will increase the maximum detectable shift. The minimum detectable shift is determined by the phase noise of the OCT system [16].

The phase averaging implementation of the PR algorithm typically computes the phase differences between adjacent A-lines and then averages them over a window. A more robust averaging technique was suggested by [18] and is now commonly used for PR averaging. In this vector averaging technique, Doppler phase shifts are calculated according to:

$$\Delta\phi = \arctan(\frac{\Im(\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} A(m,n)A^*(m,n+1))}{\Re(\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} A(m,n)A^*(m,n+1))}), \tag{2.22}$$

where $A(m,n)$ is the complex reflectivity at the pixel (m,n) and M and N are the averaging windows in the axial and transverse dimensions, respectively.

Flow Calculation

Computation of the total velocity vector $\vec{v}(x, y, z)$ requires both the estimated Doppler frequency shifts as well as the Doppler angle θ between the incident beam and the blood vessel as in Equation 2.19. Two adjacent B-scans are typically required to calculate the Doppler angle between the incident light beam and the blood vessel [19]. After reconstructing the total velocity vector $\vec{v}(x, y, z)$, the velocity flux, or volumetric flow rate through a surface, can be computed with a surface integral as:

$$F = \int_S (\vec{v} \cdot \vec{n}) dS, \tag{2.23}$$

where $\vec{n}$ is the normal to the surface. In typical OCT B-scans, the XZ or transverse-depth plane is the integration surface. However, to calculate the volumetric flow, the integration should be calculated in a plane that is normal to the blood vessel direction. If β is the angle between the integration plane and the blood vessel normal, the volumetric flow rate is given by [19]:

$$F = \iint \vec{v}(x, z) cos\beta dxdz. \tag{2.24}$$

For most of the results presented in this thesis, neither the Doppler angle θ nor the integration plane correction angle β is computed. For example, in Section 4.3 the flow ratio between two Doppler OCT algorithms will be studied. In this case, neither angle is required since the effect of these angles cancel out when computing the ratio of estimated flow rates. In Section 4.4, the effect of transverse step size on Doppler OCT flow velocity measurements is examined. Since the volumetric flow rate is proportional to the sum of the Doppler phase shifts inside a vessel, this effect can be studied without explicitly computing the angles.

Chapter 3

Segmentation of OCT Retinal Images

3.1 Introduction

OCT tomograms enable detection of morphological structures of the human retina. An example of an OCT structural image of the retina is shown in Figure 3.1.

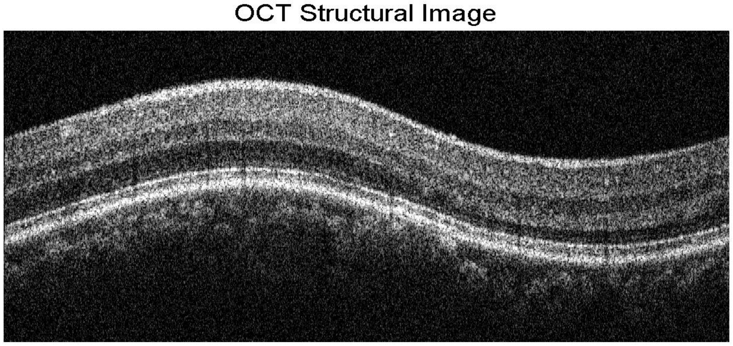


Figure 3.1: Example of an OCT structural image of the human retina.

Many of the retinal structures are critical for the diagnosis and study of ocular pathologies. For example, OCT can be used to detect macular holes and retinal detachments [20,21]. Also, one symptom of glaucoma is loss of ganglion cells in the retina, which causes a thinning in the nerve fiber layer. Thus, retinal nerve fiber layer thickness is utilized in glaucoma progression analysis [22, 23]. Segmentation of OCT retinal images is thus critical to the study of many ocular diseases. However, segmentation of OCT retinal tomograms is complicated by high levels of speckle noise. Narrowband detection systems, including OCT, radar and ultrasound, are all subject to the presence of speckle [24]. Briefly, speckle can be caused by the interference of waves backscattererd by particles within the same coherence volume. The result is a random intensity fluctuation about a mean value. Speckle reduces image fidelity and consequently segmentation algorithm performance. Other factors that degrade the performance of segmentation algorithms include poor and uneven illumination conditions and low contrast boundaries.

In Section 3.2 we introduce a novel retinal layer segmentation algorithm for identifying retinal layer boundaries in OCT tomograms. For the first time, a sparse representation using piecewise constant segments is computed to approximate each column of the image. Then, a graph theory and dynamic programming technique is applied to connect the segments, column by column, into layers that extend across the image. Using the identified layers, the boundaries between the retinal layers are extracted, resulting in an image segmentation. The primary advantage of our segmentation algorithm is the relaxation of many structural assumptions employed by other algorithms. Specifically, our algorithm does not specify the number of layers for which to look or the order in which to search for the layers. Conversely, state-of-the-art segmentation algorithms search for a specified number of retinal layer boundaries and typically employ an iterative boundary detection scheme during which a single boundary is identified using information about other previously identified boundaries. The information used relies on *a priori* assumptions about overall layer structure that hold in high fidelity images of healthy retinas but may not extend to poor imaging or pathological conditions. Relaxing some of the structural constraints results in a more general algorithm that could possibly be applicable to pathological images and even to other imaging modalities. Some of the material presented in Section 3.2 was published in [25].

One difficulty in evaluating retinal layer segmentation algorithms is the lack of an objective error metric to assess the accuracy and noise and parameter sensitivity of the segmentations. The accuracy of most of the algorithms is tested against subjective manual segmentations by one or more experts [26]. In order to overcome this limitation, we develop a synthetic retinal image generation technique in Section 3.3. A simplified image model is derived that is composed of a locally smooth component, a speckle component and an additive noise component. Each component is generated independently and then combined to produce a synthetic image. The synthetic images are validated by comparing the intensity distributions and axial and transverse speckle sizes with *in vivo* retinal images. Our technique improves upon an existing synthetic image generation method both by using a nonparametric smooth representation of the average intensity values in the image and by accurately modeling speckle noise. The use of a nonparametric representation of the image intensity provides a more flexible data generation method. The synthetic images are generated with known ground truth segmentations and are thus useful for objective quantitative evaluation of retinal layer segmentation algorithms. In order to illustrate its utility, we test our novel segmentation algorithm using synthetic retinal images in Section 3.4. We show that our segmentation algorithm identifies the nine retinal layer boundaries accurately and that our algorithm is robust to image noise.

3.2 Retinal Layer Segmentation Algorithm

3.2.1 Introduction

As stated in Section 3.1, segmentation of the retinal layers in OCT structural images is critical for the study of ocular pathologies. OCT image segmentation methods are capable of locating between 4 and 9 retinal layers [23, 27, 28]. Early segmentation approaches operated by searching for intensity peaks and valleys in each column of a despeckled retinal image [29,30]. These methods suffer in low contrast cases for which peaks may not be detectable and so erroneous boundaries can result. In addition these methods rely on assumptions about relative layer positioning, which may not hold in a pathological case.

More recently, graph theory and dynamic programming have been applied to OCT retinal image segmentation. Garvin et al. [27] segmented a 3D composite image into five layers by mapping the segmentation problem to one of finding a minimum cost closed set in a 3D graph. Both Yang et al. [28] and Chiu et al. [26] define a 2D graph on the image grid and locate boundaries by cutting the graph iteratively based on intensity and/or gradient information. While accurate segmentation results have been reported, these graph methods suffer from overly restrictive assumptions. Specifically, these algorithms rely on iterative boundary detection during which a single boundary is identified using information about other previously identified boundaries. The information used relies on *a priori* assumptions about overall layer structure that hold in high fidelity images of healthy retinas but may not extend to poor imaging or pathological conditions. Relaxing some of the structural constraints could result in a more generic algorithm applicable to pathological images and even to other imaging modalities. Additionally, these graph based approaches suffer from complexity scaling. Specifically, the number of nodes in a graph is typically equal to the number of pixels in the image. Thus, graph complexity scales with the image size even though the number of layers does not. As OCT systems increase in speed and scans over the same area increase in definition, the number of pixels in each image increases as well. It might therefore be advantageous to devise a method for transforming the image from the pixel domain to a domain that captures the layer characteristics while maintaining a dimension on the order of the number of layers.

In order to overcome the aforementioned limitations, we develop a novel sparsity-based retinal layer segmentation algorithm. The notion of sparsity is pervasive throughout the biomedical imaging literature [31]. OCT retinal images are naturally sparse in the layer domain, an abstraction of the set of retinal layers, in the sense that there are only a handful of layers in each image. We leverage this natural sparsity by transforming the input image from the pixel space to a much smaller space while still capturing layer properties that are useful for image segmentation, such as intensity and gradient information. The sparse approximation can be viewed as a preprocessing step so that any subsequent boundary search can be performed in a lower dimensional space. Other graph search algorithms [26–28] could thus benefit from this complexity reduction.

After computing a sparse representation of an OCT retinal image, we perform an iterative multi-boundary search using graph theory and dynamic programming techniques. The significant contribution of this multi-boundary search is the relaxation of some structural assumptions that are typically enforced in other graph search methods. Most retinal segmentation algorithms rely on the specification of the number of boundaries for which to look and some order in which to look for them. Conversely, our implementation proposed in this thesis only assumes that there are relatively few, primarily parallel retinal layers that are separated by smooth boundaries and are distinguished by information such as mean intensity and edge/gradient magnitude. Thus, our algorithm is fairly general and could possibly be used for other types of images such as ultrasound and synthetic aperture radar.

3.2.2 Procedure

The goal of our novel segmentation algorithm is to identify all of the boundaries between the various retinal layers in an OCT structural image without making overly restrictive assumptions about the expected layer structure. Throughout this section, a retinal image is assumed to have X columns and Z rows. The first row lies at the top of the image and the last row lies at the bottom. A superscript x will be used to denote a particular column. B_j

will be used to denote the jth boundary ordered from the top of the image to the bottom. We assume that all boundaries extend across the entire image so that $B_j = [z_j^1, z_j^2, ...z_j^X]$, where z_j^x is the row location of the jth boundary in column x. The algorithm iteratively searches for the boundaries between the retinal layers until no new reliable boundaries are found. At the beginning of each algorithm iteration, the boundary search region is limited to lie in between each pair of adjacent boundaries B_j and B_{j+1} found in previous algorithm iterations. A sub-image is then created by flattening the image to boundary B_j, which means that pixels in the original image are mapped to a new grid so that B_j lies along the first row of the sub-image. The flattened image is then smoothed and downsampled to reduce noise and the number of sub-image pixels. Then, a sparse approximation to each column in the sub-image is computed using the Sparse Bayesian Learning algorithm. This step relies on the assumption that intensities of adjacent retinal layers are distinct and produces a compact representation of each sub-image column with a set of piecewise constant segments. The first row of each piecewise constant segment is called a breakpoint. A graph theoretic approach is then used to group the segments in each column into layers that extend across the width of the image. The grouping is based on segment similarity in terms of intensity and edge information as well as smoothness constraints. The boundaries between each of the layers are then extracted and tested for smoothness. If no new boundaries pass the smoothness test then the algorithm halts and returns all of the identified retinal layer boundaries. Each new boundary that passes the smoothness test is refined using the technique in [26]. Our algorithm then limits the search region again to lie in between each pair of adjacent boundaries and iterates until no new boundaries are identified.

Figure 3.2 depicts the flowchart for the segmentation algorithm. The individual modules are described in detail in each of the following subsections.

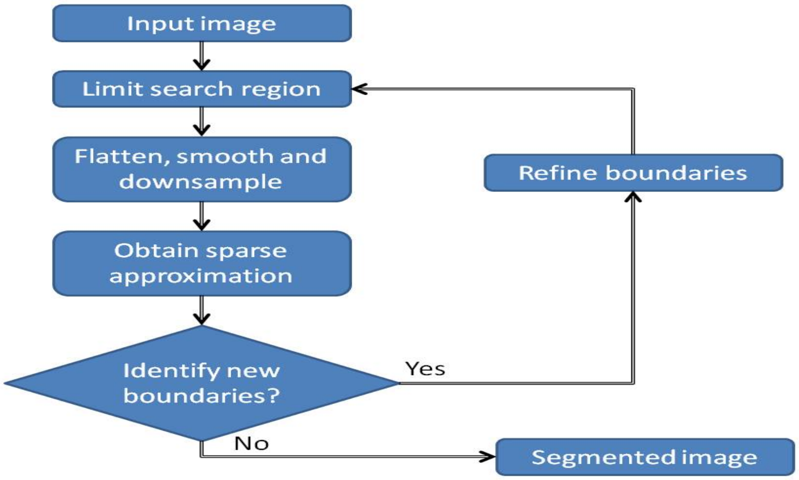


Figure 3.2: Flowchart for segmentation algorithm.

3.2.2.1 Limit Search Region

In OCT imaging of the human retina, the retina and choroid occupy less than a third of the total imaging depth. Thus as a first step it is advantageous to isolate the retina and choroid and reduce the dimensions of the image. This is done by smoothing the image with a Gaussian filter and thresholding with Otsu's method [32] to create a binary mask of the region of interest (ROI). For each column x, the top-most and bottom-most pixels in the ROI are identified and used to create temporary boundaries $\tilde{B}_1$ and $\tilde{B}_2$. Then, the maximum width $\tilde{W}_{1,2}$ over all columns between $\tilde{B}_1$ and $\tilde{B}_2$ is computed and $\tilde{B}_2$ is redefined as $\tilde{B}_2 = \tilde{B}_1 + \tilde{W}_{1,2} - 10$. The constant offset is subtracted in order to reduce the influence of the irregularly shaped choroid on the subsequent steps. The reduced search region is then defined as all of the pixels that lie on or in between $\tilde{B}_1$ and $\tilde{B}_2$. The region is illustrated in Figure 3.3.

3.2.2.2 Flatten, Smooth and Downsample

In the first algorithm iteration, the temporary boundaries $\tilde{B}_1$ and $\tilde{B}_2$ define the limited search region. In subsequent iterations, the non-temporary boundaries B_j and B_{j+1} will

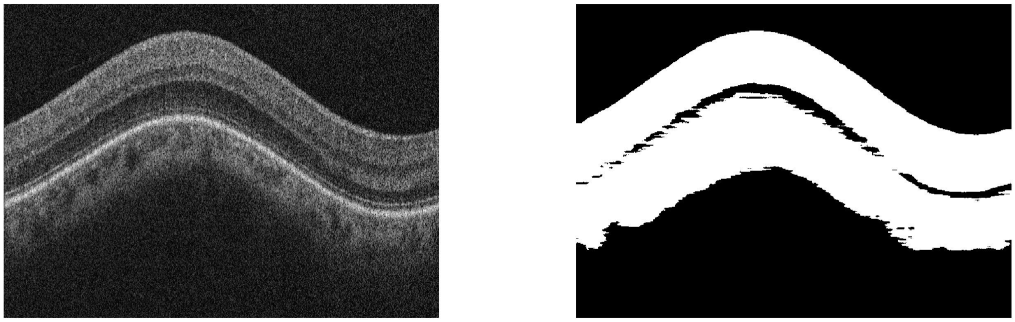


Figure 3.3: Fast identification of retina and choroid. The image is smoothed and thresholded to create a binary mask that identifies the retina and choroid regions.

define the limited search region. In this and all subsequent sections, the temporary notation is removed for brevity.

After the search region is reduced, a sub-image is created by flattening the image to the top boundary B_j. This flattening defines a mapping so that B_j lies along the first row in the flattened sub-image. The image is flattened because the retinal layers are primarily parallel and can be oriented along the transverse dimension by flattening. In order to compute the flattened image, the maximum width $W_{j,j+1}$ across all columns between B_j and B_{j+1} is first computed as $W_{j,j+1} = \max_x(B_{j+1}^x - B_j^x) + 1$. Then, the mapping from a pixel (x, z) to the flattened image is formally defined by

$$M(x,z) = z - B_j^x + 1, \forall\ z \in \left\{B_j^x, B_j^x + 1, ..., B_j^x + W_{j,j+1}\right\}. \tag{3.1}$$

The next steps in the algorithm aim to reduce the number of pixels in the image without compromising the spatial features of interest. Specifically, the images examined in this work consist of 4000 columns that span over approximately 6.9 millimeters, resulting in a physical transverse pixel size of 1.7 microns. The retinal layers are large scale image feature that span across the entire image. In order to keep the computation time low, 20x downsampling is performed which changes the transverse pixel size to 34.5 microns. This pixel size is small enough to enable retinal layer identification. Prior to downsampling, the flattened images are smoothed with a Gaussian filter with a width of 200 columns in order to (a) reduce speckle noise and (b) reduce the additional downsampling noise. After smoothing the flattened image, downsampling is performed. Flattening, smoothing and downsampling results are shown in Figure 3.4. Note that these techniques are commonly used to improve segmentation algorithm performance [23, 26].

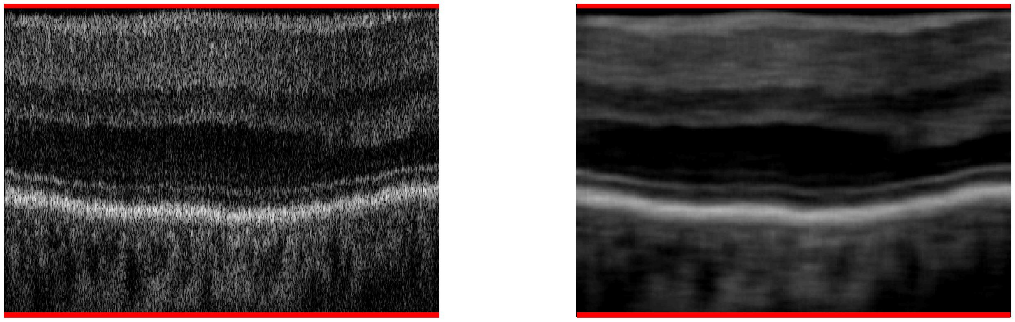


Figure 3.4: Smoothing and downsampling of flattened image. The top and bottom boundaries used to compute the flattened image are shown in red. On the first iteration of the segmentation algorithm, these boundaries lie at the top and bottom rows of the flattened image. In subsequent iterations, the top boundary will lie along the top row of the flattened image, but the bottom boundary need not lie along the bottom row.

3.2.2.3 Obtain Sparse Approximation

Formulation

Motivated by the assumptions that (a) the layers are primarily horizontal after image flattening and (b) the layer intensities are approximately constant, on average, after noise removal, each column is sparsely approximated with a weighted sum of Heaviside step functions [7]. The sparse approximation provides a compact representation of each layer column while preserving the layer distinguishing properties such as mean and edge information. Formally, a vector of M intensities I^X in the flattened sub-image is observed and a solution to the problem

$$\hat{w}^X = argmin_w ||I^X - Fw^X||_2 + \lambda ||w^X||_0 \tag{3.2}$$

is sought. In Equation 3.2, F is an M x M normalized basis of step functions, λ is a tradeoff parameter between goodness of fit and sparsity and $||w||_0$ is the ℓ_0 norm and is equal to the number of nonzero weights. M is determined by the height of the flattened image. The ith vector in the basis is defined by

$$f_i(m) = \begin{cases} -\sqrt{\frac{M-i}{iM}} & \text{if } m \leq i \\ \sqrt{\frac{i}{M(M-i)}} & \text{otherwise,} \end{cases} \tag{3.3}$$

when $i > 0$ and

$$f_0(m) = \frac{1}{\sqrt{M}}. \tag{3.4}$$

Except for the $i = 0$ basis vector, each basis vector f_i is a function with a single discontinuity. If the following basis selection algorithm incorporates the basis vector f_i by assigning if a nonzero weight, then the sparse approximation is said to have a breakpoint at position i. The goal of the problem in Equation 3.2 is to closely approximate the vector of intensities I^X with as few basis vectors as possible. A basis vector f_i is excluded from the approximation only if the corresponding weight w_i is zero. Note that a positive w_i indicates an increase in intensity at the breakpoint and a negative w_i indicates a decrease in intensity.

Sparse Bayesian Learning

The sparse Bayesian learning (SBL) algorithm is employed to compute the maximum *a posteriori* estimates of the weights within a Bayesian framework. SBL is used because it has been shown [33] to perform well with dictionaries of high coherence/collinearity, as is the case with a basis of unit step functions.

The SBL algorithm runs on each column of the image sequentially. The fast implementation of SBL used here is described in [34]. Briefly, the implementation begins with all weights set to zero except the vector f_i that has the largest inner product with the observed vector I^X. It is said that vector f_i is in the model and all other vectors are not. Then at each iteration the algorithm either (a) adds a basis vector to the model, (b) removes a basis vector from the model or (c) updates the weight of a basis vector that is already in the model. This implementation is unique in the sense that all but one of the weights are set to zero initially, whereas for most other implementations all weights are nonzero initially. This is the key factor behind the speed of the implementation. SBL runs for a specified number of iterations and returns the weight vector. In the first iteration of the segmentation algorithm, the number of SBL iterations is specified by a parameter $SBLIter$. In subsequent iterations of the segmentation algorithm, the number of SBL iterations scales with the number of rows in the relevant flattened image. For example, suppose that in the first segmentation algorithm iteration the flattened image between temporary boundaries $\tilde{B}_1$ and $\tilde{B}_2$ has 100 rows. If on a subsequent segmentation algorithm iteration the flattened image between boundaries B_k and B_{k+1} has 50 rows, then SBL runs for $SBLIter/2$ iterations on this flattened image. The SBL iteration scaling is done this way because it is assumed that there will be fewer breakpoints found if the size of the flattened image that represents the limited search region is smaller.

Another parameter required for SBL implementation is an estimate of the image noise variance. Fortunately, a region that consists of noise only is readily identified in OCT retinal images. As previously mentioned, the retina and choroid compose only about one third of an OCT retinal tomogram. The region above the retina can be used to estimate the noise variance.

The SBL algorithm solves Equation 3.2 sequentially by adding and/or removing basis vectors. Basis vectors are removed by setting the corresponding weights to zero. Since the flattened image to which SBL is applied contains two previously identified adjacent boundaries B_k and B_{k+1}, it is imperative that the weights corresponding to these boundaries be nonzero. The SBL implementation used in this work allows for specification of a fixed basis F_B which defines a subset of the basis F that must be included in the model. In order to prevent the two previously identified boundaries from being excluded from the model, they are added to the fixed basis F_B. Note that this does not apply to the temporary boundaries $\tilde{B}_1$ and $\tilde{B}_2$ used in the first iteration of the segmentation algorithm.

Backwards Elimination

Because of the significant amount of smoothing employed prior to applying the SBL algorithm, many edges stretch across multiple breakpoints. An example is shown in Figure 3.5. In order to reduce the number of breakpoints in the SBL approximation without sacrificing any boundary information, a backwards elimination algorithm is applied to combine close adjacent breakpoints that have the same sign. Assume that there are K breakpoints in a column x. Suppose that in this column there is a sequence of breakpoints located at positions z_i thru z_j such that (a) all of the weights w_i thru w_j have the same sign and (b) $z_{k+1} - z_k \leq mnBPDst$ for all $k \in \{i, i+1, ..., j-1\}$. Denote this sequence by A_{ij}. If none of the breakpoints in A_{ij} are in the fixed basis F_B, then A_{ij} is replaced with a single breakpoint located near the center of mass of the sequence CM_{ij}. The procedure for this replacement is as follows:

1. Compute $CM_{ij} = \sum_{k=i}^{j} w_i z_i / \sum_{k=i}^{j} w_i$.

2. Find $r = \text{argmin}_k |z_k - CM_{ij}|$ for $k \in \{i, i+1, ..., j\}$.

3. Compute the cumulative weights for all breakpoints in the column $c_{k+1} = \sum_{m=0}^{k} w_m$ for $k \in \{0, 1, ..., j-1\}$. Set $c_0 = 0$.

4. Update weight $w'_r = c_{j+1} - c_i$.

5. Remove breakpoints $k \in \{i, i+1, ..., j\}$, $k \neq r$.

This process replaces the sequence of breakpoints with a single breakpoint whose weight is updated to the cumulative weight of the sequence.

On the other hand, if any of the breakpoints in A_{ij} are in the fixed basis F_B previously discussed, then those are retained and the others are removed. This replacement procedure is as follows:

1. Compute the cumulative weights for all breakpoints in the column $c_{k+1} = \sum_{m=0}^{k} w_m$ for $k \in \{0, 1, ..., j-1\}$. Set $c_0 = 0$.

2. If one vector is in the fixed basis then update its weight. If two vectors are in the fixed basis then update only the second weight $w'_r = c_{j+1} - c_i$.

3. Remove all breakpoints not in F_B.

Note that by construction of F_B, at most two breakpoints can be in the fixed basis. If there are two breakpoints in F_B then the first one must lie at the top of the image. By convention, the weight of this breakpoint is not updated.

An example of the SBL algorithm applied to a single image column and the subsequent backward elimination step is illustrated in Figure 3.5. Notice that the number of breakpoints is significantly reduced by applying the backwards elimination procedure.

The SBL algorithm and backwards elimination procedure are run on each column of the input image. After completing all X columns, a sparse image approximation is obtained. An example of a sparse image approximation is shown in Figure 3.6.

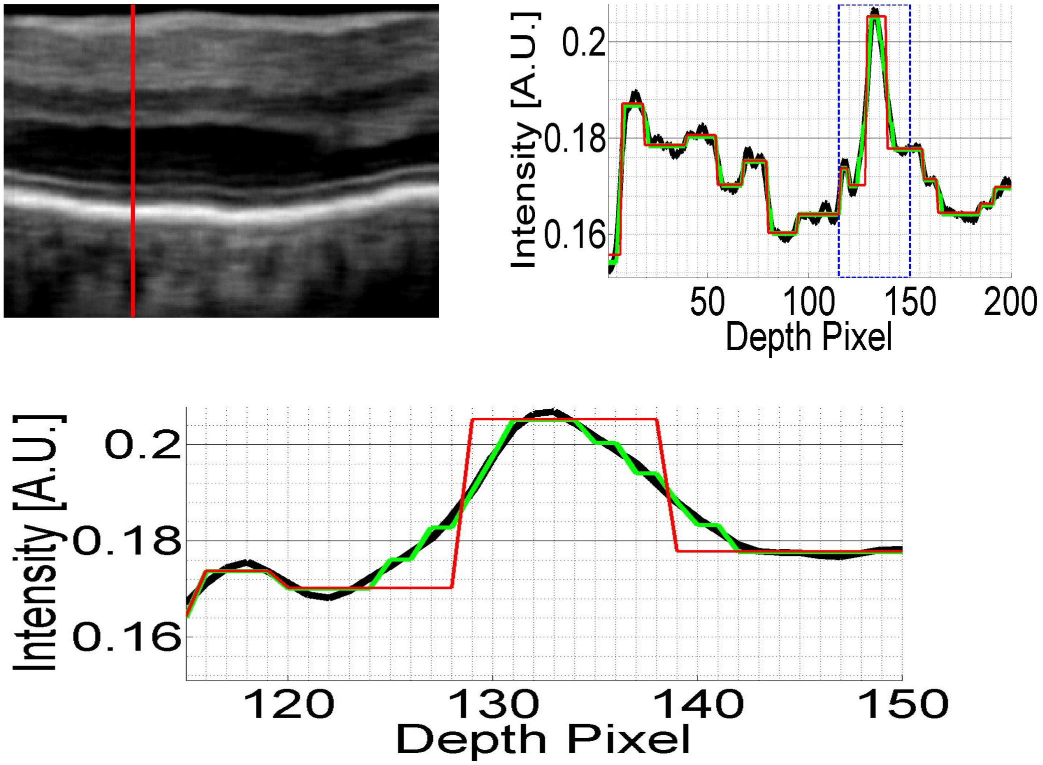


Figure 3.5: Sparse approximation of a sample A-line after Sparse Bayesian Learning and backwards elimination. Top left panel: sample B-scan with image column denoted in red. Top right panel: The black line denotes intensity in image column. The green line denotes the result after applying SBL. The red line denotes the result after backwards elimination. Bottom panel: Zoomed in version of the blue dotted box in the top right panel. The effectiveness of the backwards elimination procedure is clearly illustrated by the reduction in the number of breakpoints.

As previously mentioned, one primary advantage in computing a sparse approximation of an OCT retinal image is the reduction of the dimension of the search space used in subsequent boundary searching. Specifically, the number or rows in the images tested in this thesis is 768. After running SBL, approximately 20 breakpoints are identified in each column. The number unique intensity values is thus reduced by a factor of 38 for our compact representation. This compact representation could improve the computer memory consumption for other graph based segmentation algorithms.

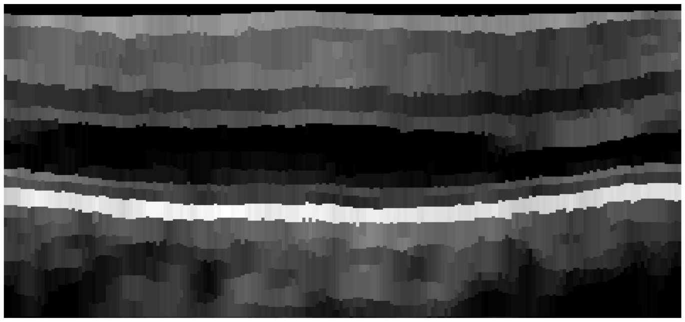


Figure 3.6: Sparse approximation of a B-scan.

3.2.2.4 Identify New Boundaries

The purpose of this module is to group the intra-column piecewise constant segments obtained by SBL into layers that extend across all of the image columns and then extract the retinal layer boundaries from the groupings. First, some definitions and notation are provided to aid the understanding of the rest of this section. Then, we formulate the boundary identification problem using a greedy approach by assuming that at column x all of the segments in columns 1 thru $x-1$ have been correctly grouped into layers and then trying to find the best way to extend the layers with the segments in column x. In order to find the best solution we construct a graph that describes the different ways that the layers can be extended by the segments and apply a shortest path algorithm to find the optimal extension. One advantage of our graph-based formalism over most others is that a single graph cut can segment multiple retinal layers which means that our graph search has built in boundary interaction constraints that are typically implemented separately. From the shortest path graph cut, we then compute the layer groupings and subsequently extract the boundaries between the retinal layers. As a final step in identifying new boundaries, a smoothness test is applied to each boundary and those that do not pass are discarded.

Definitions and Notation

The sparse approximation of the each column x can be compactly represented by a set of N_S^x segments S_i^x, each associated with a mean intensity value $\mathbf{M}(S_i^x)$, position $\mathbf{P}(S_i^x)$, width $\mathbf{W}(S_i^x)$ and gradient or edge height $\mathbf{E}(S_i^x)$. Note that the edge height can be positive or negative and is defined as the height of the discontinuity at a particular position. The superscript x identifies in which column the segment resides, and will be used only when the ambiguity may cause confusion. The position $\mathbf{P}(S_i)$ is defined as the location of the top-most pixel in the segment and the edge height $\mathbf{E}(S_i)$ is defined as the height of the jump at the first pixel in the segment. These conventions are illustrated in Figure 3.7.

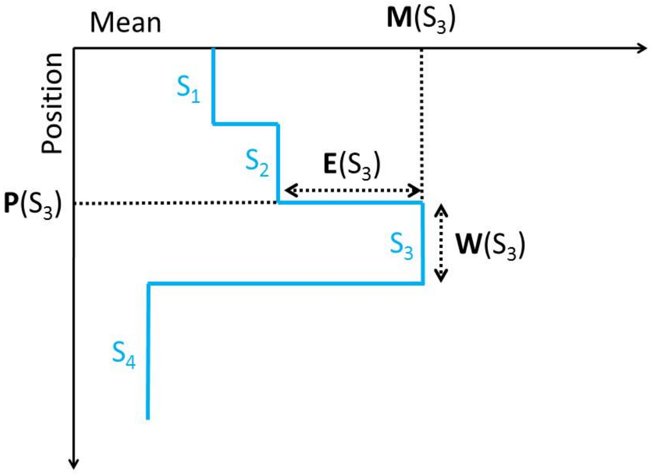


Figure 3.7: Definitions and notation for segments and related quantities. The SBL approximation resulted in 4 segments S_1 through S_4., with associated mean value $\mathbf{M}$, position $\mathbf{P}$, front edge height $\mathbf{E}$ and width $\mathbf{W}$.

A layer is defined as a set of similar segments grouped within and across columns. Within a single column x, a set of adjacent segments $S_{i:j}^x$ can be assigned to a layer L_k^x. A layer L_k^x has an associated mean value, position, width and edge height that are determined by its constituent segments. For example, the width of a layer in a particular column equals the sum of the widths of the segments that reside in that layer and column. Formally, if adjacent segments i thru j, denoted by $S^x_{i:j}$ are assigned to layer k, then these quantities can be written as:

$$\begin{aligned}
\mathbf{M}(L^x_k) &= \mathbf{M}(S^x_{i:j}) &= \textstyle\sum_{p=i}^{j}\mathbf{M}(S^x_p)\mathbf{W}(S^x_p)/\sum_{p=i}^{j}\mathbf{W}(S^x_p),\\
\mathbf{P}(L^x_k) &= \mathbf{P}(S^x_{i:j}) &= \mathbf{P}(S^x_i),\\
\mathbf{W}(L^x_k) &= \mathbf{W}(S^x_{i:j}) &= \textstyle\sum_{p=i}^{j}\mathbf{W}(S^x_p),\\
\mathbf{E}(L^x_k) &= \mathbf{E}(S^x_{i:j}) &= \mathbf{E}(S^x_i).
\end{aligned} \tag{3.5}$$

Again, the superscript x identifies in which column the segment/layer resides, and will be used only when the ambiguity may cause confusion. These conventions are illustrated in Figure 3.8.

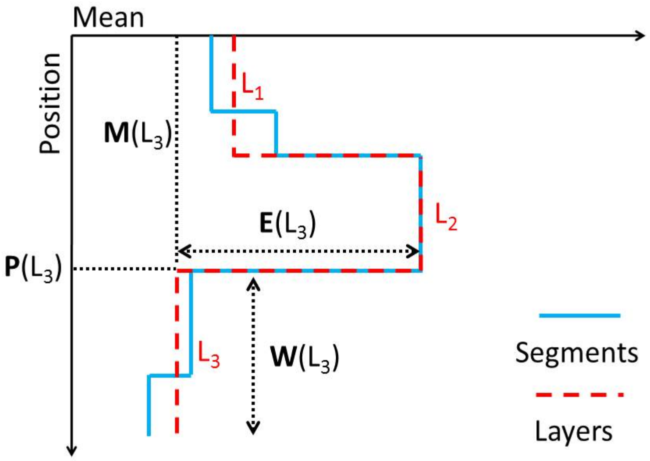


Figure 3.8: Definitions of layer quantities. The 5 segments in blue compose the 3 layers in red. A layer's mean, edge, width and position are determined by the segments assigned to that layer.

A layer extends across multiple columns. Let $\hat{L}^x_k$ denote the kth layer grouped across columns 1 thru $x-1$, or $\hat{L}^x_k = \cup_{i=1}^{x-1} L^x_i$. When viewed across columns, the mean, position, width and edge height of $\hat{L}_k^x$ are defined by a Gaussian weighted average of all of the corresponding values in the previous columns. Letting $\mathbf{Y}(\hat{L}_k^x)$ denote any of the quantities $\mathbf{M}(\hat{L}_k^x)$, $\mathbf{P}(\hat{L}_k^x)$, $\mathbf{W}(\hat{L}_k^x)$ or $\mathbf{E}(\hat{L}_k^x)$, the aggregated mean, position, width and edge height are formally defined by:

$$\mathbf{Y}(\hat{L}_k^x) = \sum_{i=1}^{x-1} \exp\left(-\frac{(i-(x-1))^2}{2\sigma_G^2}\right) \cdot Y(L_k^i), \tag{3.6}$$

where σ_G is the Gaussian standard deviation and is set to 8.5 pixels which corresponds to a full-width at half-maximum of 20 pixels.

Two or more adjacent layers may also be grouped within a column, similar to the way segments are grouped within a column. In this case the layer properties, namely mean, position, width and edge height, are defined similarly to Equation 3.5. Because these quantities will be referenced later on, these definitions are repeated for the aggregated layers below. Analogous equations can be given for the non-aggregated layers by replacing $\hat{L}$ by L.

$$\begin{array}{rcc} \mathbf{M}(\hat{L}_{i:j}^x) & = & \sum_{p=i}^{j} \mathbf{M}(\hat{L}_p^x)\mathbf{W}(\hat{L}_p^x) / \sum_{p=i}^{j} \mathbf{W}(\hat{L}_p^x), \\ \mathbf{P}(\hat{L}_{i:j}^x) & = & \mathbf{P}(\hat{L}_i^x), \\ \mathbf{W}(\hat{L}_{i:j}^x) & = & \sum_{p=i}^{j} \mathbf{W}(\hat{L}_p^x), \\ \mathbf{E}(\hat{L}_{i:j}^x) & = & \mathbf{E}(\hat{L}_i^x). \end{array} \tag{3.7}$$

A summary of the definitions and notation for segments and layers is provided in Table 3.1.

Formulation

A greedy approach is taken to build up the layers by connecting segments across columns. It is assumed that at column x all segments in the previous columns have been grouped into layers. Then, the goal is to find the best way to extend the N_L^x layers, found in columns 1 thru $x-1$ with the N_S^x segments in column x by assigning each segment to one of the layers.

Notation	Description
S_i^x	ith segment in column x
$S_{i:j}^x$	Group of adjacent segments i thru j in column x
N_S^x	Number of segments in column x
L_i^x	ith layer in column x
$L_{i:j}^x$	Group of adjacent layers i thru j in column x
$\hat{L}_i^x$	Union of ith layer across columns 1 thru $x-1$
$\hat{L}_{i:j}^x$	Union of group of adjacent layers across columns 1 thru $x-1$
N_L^x	Number of layers in column $x-1$
$\mathbf{M}(\cdot)$	Mean intensity of a segment(s)/layer(s)
$\mathbf{P}(\cdot)$	Position of beginning/top of segment(s)/layer(s)
$\mathbf{W}(\cdot)$	Width of segment(s)/layer(s)
$\mathbf{E}(\cdot)$	Front edge height of segment(s)/layer(s)

Table 3.1: Summary of segment and layer definitions and associated notation. Segments are obtained from the sparse approximation of the image and are confined to a single column. A layer is defined as a set of similar segments that are grouped within and across columns. Note that the mean, position, width and edge height of the union of layers across columns are defined using the Gaussian weighted average in Equation 3.7.

This process is iterated column by column until all segments have been grouped into layers. The problem formulation is illustrated in Figure 3.9.

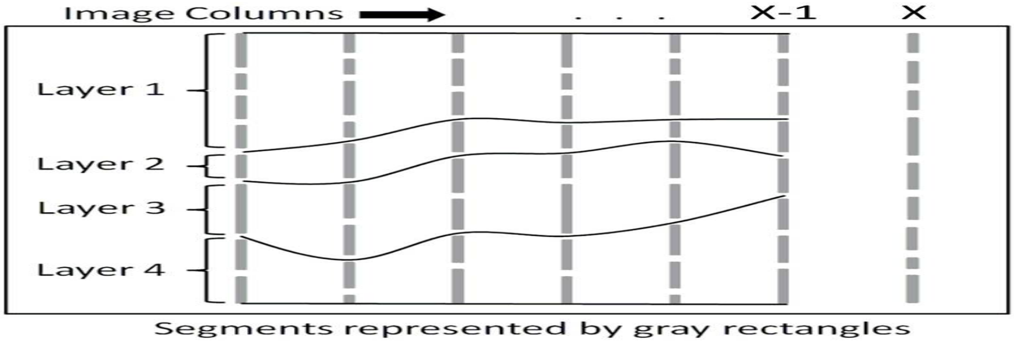


Figure 3.9: Layer building formulation.

In order to solve this assignment problem a graph-based approach [35] is used. Graph-based methods require specification of (a) nodes, (b) transitions/edges, (c) transition/edge costs and (d) a graph search algorithm. The node and transitions structure can best be understood using the matrix representation shown in Figure 3.10. It is assumed that the segment and layer indices are ordered from the top to the bottom of the image. Then, by construction, the layers are extended sequentially beginning at layer 1 and ending with layer N_L.

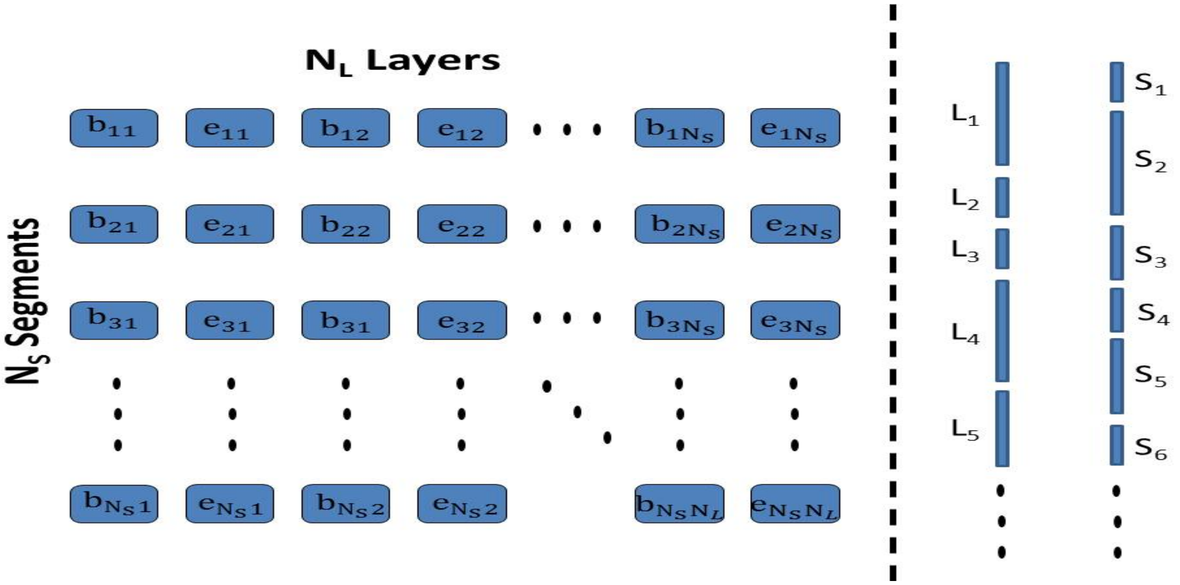


Figure 3.10: Illustration of matrix structure of the graph.

For this graph problem, two types of nodes are defined: (1) beginning node b_{ij}, which indicates that segment i begins layer j, and (2) end node e_{ij}, which indicates that segment i ends layer j. Traversing through node b_{ij} indicates that the top-most pixel of layer j in the previous columns coincides with the top-most pixel of segment i in the current column x. Similarly, traversing through node e_{ij} indicates that the bottom-most pixel of layer j in the previous columns coincides with the bottom-most pixel of segment i in the current column x. Two hard constraints impose restrictions on the reachability of nodes to ensure that the assignment of segments to layers is physically sensible. If either constraint is satisfied then a node is considered reachable. The first constraint allows a segment to extend a layer only if some part of it lies adjacent to some part of the layer. Formally, segment i can be grouped in layer k if

$$\begin{aligned} \mathbf{P}(S_i) &\leq \mathbf{P}(\hat{L}_k) + \mathbf{W}(\hat{L}_k) - 1 \\ \mathbf{P}(\hat{L}_k) &\leq \mathbf{P}(\hat{L}_k) + \mathbf{W}(S_i) - 1 \end{aligned} \tag{3.8}$$

are both satisfied. Notice that the aggregated layer quantities defined in Equation 3.8 are used to impose this constraint. The second constraint can be broken into two parts; one for a beginning node and one for an end node. Specifically, a beginning node can begin a layer only if the first pixel in the corresponding segment is close to the first pixel in the layer. An end node can end a layer only if the last pixel in the corresponding segment is close to the last pixel in the layer. How close is defined by a parameter called *maxJump*. Let $\mathbf{P_E}(S_i) \equiv \mathbf{P}(S_i) + \mathbf{W}(S_i) - 1$ denote the end position of a segment and let $\mathbf{P_E}(\hat{L}_k) \equiv \mathbf{P}(\hat{L}_k) + \mathbf{W}(\hat{L}_k) - 1$ denote the end position of a layer. Formally, segment i can be grouped in layer k if

$$\begin{aligned} |\mathbf{P}(S_i) - \mathbf{P}(\hat{L}_k)| &\leq maxJump, \quad \text{for a beginning node} \\ |\mathbf{P_E}(S_i) - \mathbf{P_E}(\hat{L}_k)| &\leq maxJump, \quad \text{for an end node} \end{aligned} \tag{3.9}$$

A node is considered reachable if it satisfies either the constraint in Equation 3.8 or the constraint in Equation 3.9. An example of these hard reachability constraints in shown in Figure 3.11.

In order to assign segments i thru j to layer k, segment i must begin layer k and segment j must end layer k. In terms of the node structure, this requires a beginning-to-end transition from node b_{ik} to node e_{jk}, denoted by $b_{ik} \rightarrow e_{jk}$. Note that in this formulation, no segments are skipped so that this particular transition implies that all segments $\{i, i+1, ..., j\}$ are assigned to layer k. End-to-beginning transitions that do not have an analogous relation the segment grouping by do enforce continuous structure are also utilized. An additional beginning-to-end merge transition is also defined to allow for layer merging. Though introduced here, the significance of layer merging is made clear in the Initialization section

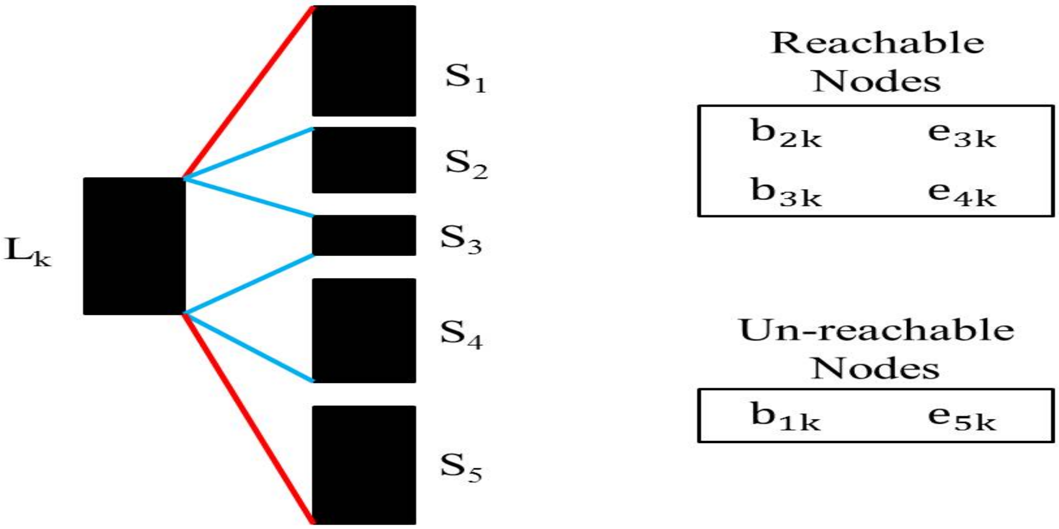


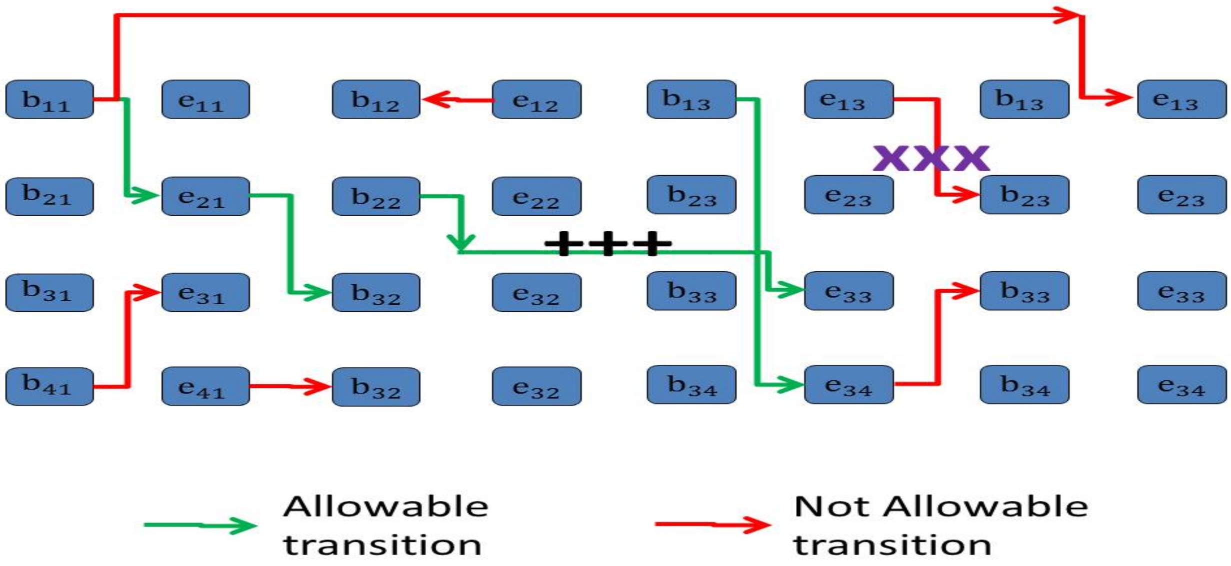
Figure 3.11: Node reachability constraint.

below. Physically, a merge transition means that two or more layers that have previously been distinct in columns 1 thru $x-1$ will now be treated as a single layer. This can be represented by a transition from node b_{ik} to node e_{jl} for $l > k$.

Hard constraints are imposed on the allowability of different transition types. For example, beginning-to-beginning and end-to-end transitions are never allowed. Also, graph traversal always proceeds from left to right in Figure 3.12 since layer k is extended before layer $k+l$. Additional beginning-to-end and end-to-beginning constraints are defined to ensure a physically sensible solution so that (1) the boundaries between layers do not cross, (2) all segments are assigned to layers and (3) positive edges connect only to positive edges and similarly for negative edges. Additionally, the maximum number of layers that can be merged in a single merge transition is set to 3 to keep the number of edges in the graph small. These constraints on the allowable transitions can be written formally as:

$$\begin{array}{lll} b_{ik} \rightarrow e_{jl} & \text{allowable only if} & j \geq i \;\&\; k \leq l \leq k+2, \\ e_{ik} \rightarrow b_{jl} & \text{allowable only if} & j = i+1 \;\&\; l = k+1 \;\&\; \text{sign}(\mathbf{E}(S_j)) = \text{sign}\,(\mathbf{E}(L_l)). \end{array} \quad (3.10)$$

Some allowable transitions and transitions that are not allowed are shown in Figure 3.12.



b_{11} e_{11} b_{12} e_{12} b_{13} e_{13} b_{13} e_{13}
b_{21} e_{21} b_{22} e_{22} b_{23} e_{23} b_{23} e_{23}
b_{31} e_{31} b_{32} e_{32} b_{33} e_{33} b_{33} e_{33}
b_{41} e_{41} b_{32} e_{32} b_{34} e_{34} b_{34} e_{34}
+++
XXX
Allowable transition
Not Allowable transition

Figure 3.12: Transition allowability constraint. The '+++' notation denotes an allowable merge transition. The 'xxx' notation denotes that this transition is not allowable because the sign of the edge at the beginning of segment 2 is not equal to the sign of the edge at the beginning of layer 3. Note that this violation cannot be seen in the figure.

Having described the nodes and transitions, transition costs are now defined. An edge from node n_i to node n_j in the graph exists only if node n_j is reachable as defined by Equation 3.8 and Equation 3.9 and the transition between nodes n_i and n_j is allowable, as defined in Equation 3.10. The cost of a graph transition is a measure of the similarity between a layer or layers and a set of segments. Since a beginning-to-end transition describes a grouping of segments into layers, the associated costs measure the similarity between the means and widths of the segments and layers. The beginning-to-end transition costs for the mean and width are defined by:

$$\begin{aligned} C_M(b_{ik} \rightarrow e_{jl}) &= \alpha_{M,(j,l)} \cdot |\mathbf{M}(S_{i:j}) - \mathbf{M}(\hat{L}_{k:l})|, \\ C_W(b_{ik} \rightarrow e_{jl}) &= \alpha_{W,(j,l)} \cdot |\mathbf{W}(S_{i:j}) - \mathbf{W}(\hat{L}_{k:l})|, \end{aligned} \tag{3.11}$$

for the allowable transitions to reachable node and are infinite (i.e., no edge) otherwise. The α quantities are normalizing constants for each layer-to-layer k to l transition and are defined by the inverse of the maximum value over all allowable transitions and segment pairs i and j. By normalizing in this way, the costs always lie in the unit interval. End-to-beginning transitions, on the other hand, describe a transition only to the first segments in the layer, and so that associated cost measures the similarity between the front edge height of the segments and layers. The end-to-beginning transition cost for the edge height is defined by:

$$C_E(e_{ik} \rightarrow b_{jl}) \quad = \quad \alpha_{E,(j,l)} \cdot |\mathbf{E}(S_{i:j}) - \mathbf{E}(\hat{L}_{k:l})|, \tag{3.12}$$

for the allowable transitions to reachable node and are infinite (i.e., no edge) otherwise.

A merge penalty P_M is also defined as an additional cost for beginning-to-end transitions. The purpose for this penalty can be understood by examining the merge transition in Figure 3.12. Beginning-to-end transitions with no merging move between adjacent columns in the graph matrix. On the other hand, a merge transition skips two or more columns in that matrix. This means that if a merge occurs, the shortest path across the graph will have fewer nodes than if there is no merge. Since the cost is nonnegative at every edge, this leads to a bias towards merging during the graph traversal. In order to combat this bias, a merge penalty is imposed that scales with the number of layers merged. For example, if a merge between two layers is penalized by P_M, then a merge between three layers is penalized by $2 \cdot P_M$. Recall from Equation 3.10 that up to 3 layers can merge at a time, and so $2 \cdot P_M$ is the maximum merge penalty imposed. The total cost of beginning-to-end and end-to-beginning transitions are defined by the sum of the respective terms from Equation 3.11 and Equation 3.12. These costs are defined by:

$$C(b_{ik} \rightarrow e_{jl}) = C_M(b_{ik} \rightarrow e_{jl}) + C_W(b_{ik} \rightarrow e_{jl}) + (k - l) \cdot P_M \tag{3.13}$$

and

$$C(e_{ik} \to b_{jl}) = C_E(e_{ik} \to b_{jl}). \tag{3.14}$$

With all nodes and costs defined, Dijkstra's algorithm [35] is employed to find the shortest path between two nodes in the graph. The source node is forced to be node b_{11} and the destination node is forced to be node $e_{N_S N_L}$. These restrictions mean that the first segment must begin the first layer and the last segment must end the last layer, both of which are natural physical restrictions. Figure 3.13 shows a sample traversal of the graph. The algorithm is run sequentially for each column from the left to the right of the image. Upon completion, all of the segments have been grouped into layers that extend across all of the image columns.

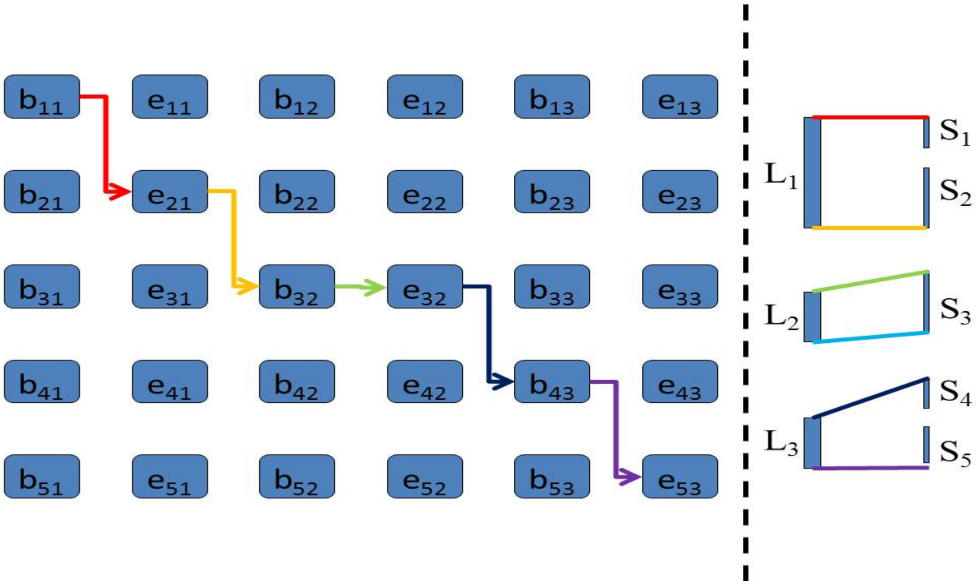


Figure 3.13: Sample graph traversal and corresponding layer extension. In this traversal, no layers are merged and the 5 segments extend the 3 layers.

Initialization

Using a greedy approach, it is assumed that at each column all of the segments in previous columns have been grouped into layers correctly. Thus the remaining issue to handle is initialization of the segments in the first column. Rather than trying to find the best initial assignment, each segment is assigned to its own distinct layer in the first column. The merge transition defined above allows some layers to merge as the algorithm progresses across columns. In this way, the onus of initializing the correct layer grouping is mitigated. Note that if layers i and j merge at column x, then all of the layer properties $\mathbf{M}$, $\mathbf{P}$, $\mathbf{W}$ and $\mathbf{E}$ are combined analogously to Equation 3.5 for each column 1 thru x. An example of merging layers is shown in Figure 3.14.

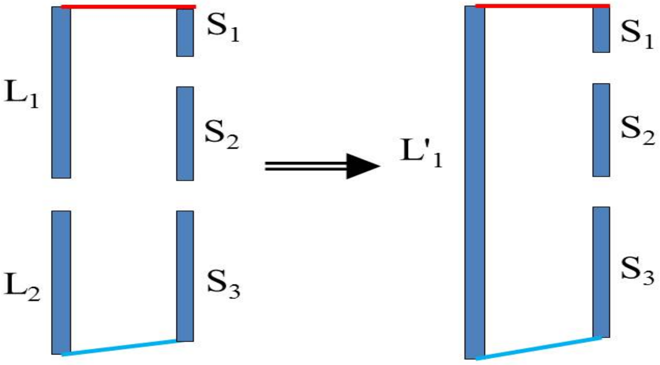


Figure 3.14: Example of layer merging. When layers merge the mean, position, width and edge are combined analogously to Equation 3.5.

Layer Building

After the segments in the first column are initialized to lie in their own distinct layers, the graphs are built and traversed on each column 2 thru x sequentially until all of the segments are grouped into layers. Figure 3.15 shows the results of the layer building process.

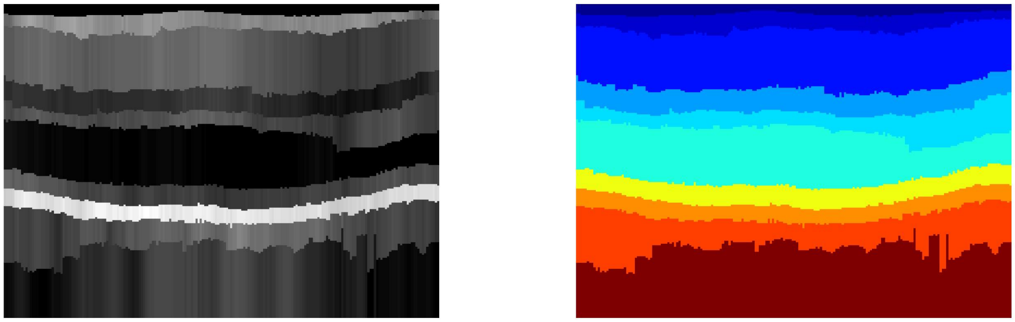


Figure 3.15: Results of layer building. Left panel: Color coded by mean value of a layer in each column. Right panel: labeled image where the identified layers are color coded.

Boundary Extraction and Upsampling

After the layer building process completes across all of the columns, all of the segments are grouped into layers. The boundaries between the layers are then extracted by computing, for each column, the first pixel in each layer. Then, these boundaries are upsampled using nearest neighbor interpolation to compute boundaries that extend across the original (full-width) image. Results of the first iteration are shown in Figure 3.16.

Boundary Pruning

Not all of the newly found boundaries are real boundaries between retinal layers. Since the layer building module model uses locally based cost functions, big picture inaccuracies might result. For example, a candidate boundary may not be smooth enough to constitute a physically sensible boundary. In order to eliminate such false positives, a smoothness metric

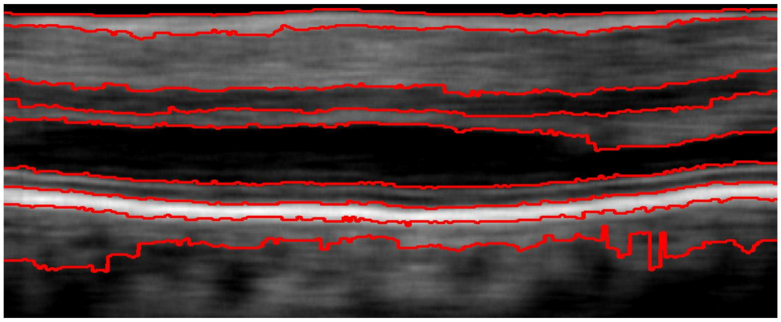


Figure 3.16: Extracted and upsampled boundaries. Upsampling is performed using nearest neighbor interpolation.

is computed for each potential boundary and compared to a threshold so that a boundary that is not smooth is pruned.

In order to compute its smoothness, a boundary $B_j = [z_j^1, z_j^2, ...z_j^X]$ is first low-pass filtered using a second order Butterworth filter with normalized cutoff frequency $\omega = 0.005$. Then, the mean-squared error (MSE) between the original boundary and its smoothed version is computed. If the MSE exceeds the threshold $smThr$, it is considered a false positive and is removed from further analysis. Figure 3.17 illustrates an example of boundary pruning.

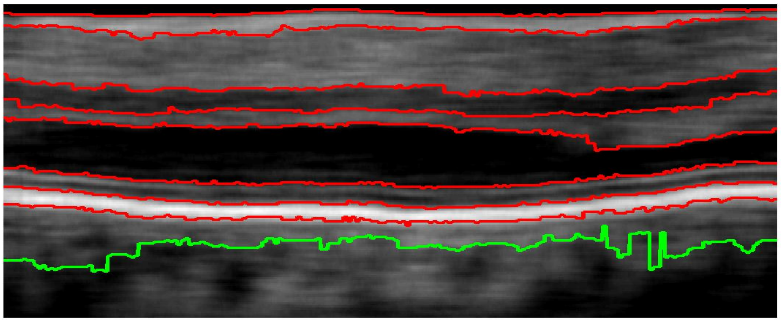


Figure 3.17: Sample boundary pruning. The red boundaries are retained while the green boundary is pruned because of a lack of smoothness.

3.2.2.5 Refine Boundaries

Due to the high level of smoothing implemented prior to applying SBL, there may be slight inaccuracies in the retained retinal layer boundaries. In order to improve the accuracy of boundary detection, a boundary refinement procedure using a fast graph-based approach based on the algorithm in [26] is employed. The algorithm is described in detail in the next paragraph. Briefly, the image is represented as a graph of nodes, where each node corresponds to a pixel. Pixels are connected to adjacent pixels with edges and Djikstra's algorithm is employed to identify a single shortest path graph cut that separates the top and bottom of the image. One especially advantageous aspect of the algorithm is that it provides automatic endpoint initialization. The retained boundaries are refined sequentially and independently.

For each boundary that survives the boundary pruning stage, a small region of $\Delta \pm 3$ pixels is identified and a sub-image with the boundary across the center is computed. The boundary extends across the width of the sub-image, and left-to-right is considered the forward direction. Each pixel in this sub-image defines a node in a new graph. As a modification to the implementation in [26], each node is connected to only 4 of its neighbors, 45°, 0°, -45° and -90°, so that each transition has either a down or forward component. The cost of each transition is based on a measure of the similarity of the vertical gradients between the two nodes. Formally, the transition cost c_{ij} for moving from node i to node j is defined by

$$c_{ij} = 2 - (\nabla_y I_i + \nabla_y I_j) + c_{min}, \tag{3.15}$$

where ∇_y denotes the vertical gradient operator, I_i denotes the intensity at the pixel in the image that corresponds to the ith node in the graph and $c_{min} = 10^{-5}$ is the minimum cost for a transition between two nodes. Note that the gradients are normalized to lie in the interval $[0,1]$ so that the cost function encourages transitions between two pixels that have large and similar vertical gradients.

As mentioned previously, the algorithm provides automatic endpoint initialization so that the beginning of the cut and end of the cut do not influence the boundary refinement. Specifically, an additional column of nodes to both sides of the image are added and connected to adjacent nodes as before. In this case, however, the cost of all allowable transitions to the 4 adjacent nodes is set to the minimum cost c_{min} so that the cut is able to traverse these columns with minimal resistance. The graph start node can be assigned as any node in the newly added column to the left of the image, and the end node can be assigned as any node in the newly added column to the right of the image. Once the image is segmented, the two additional columns can be removed, leaving an accurate cut without endpoint initialization error. An example graph traversal is shown in Figure 3.18.

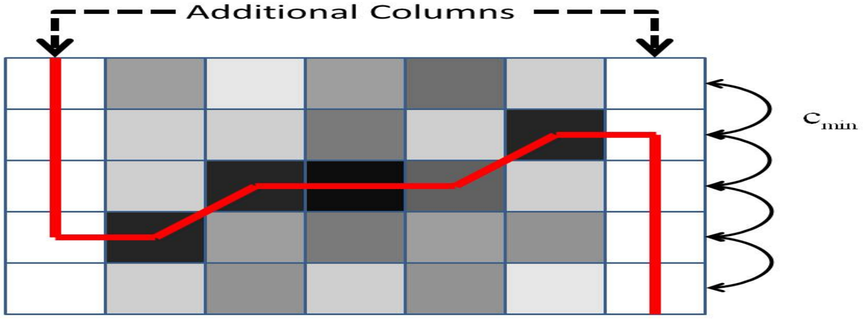


Figure 3.18: Sample graph cut using algorithm in [26]. By adding the additional columns to the left and right of the image through which the graph cut can move freely (at a cost c_{min}), the endpoint initialization is automated.

The boundaries are sequentially refined using the described algorithm. The results of this refinement are shown in Figure 3.19.

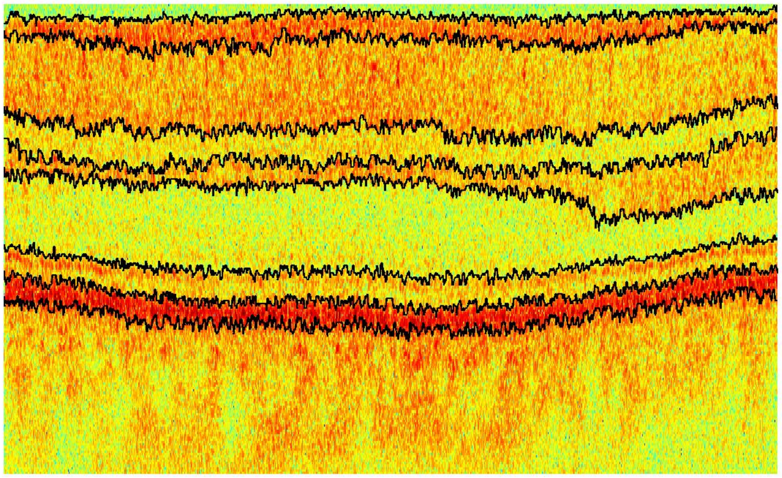


Figure 3.19: Refined boundaries. This result is shown in color so that the resulting refinement is clearly demonstrated.

3.2.2.6 Limit Search Region or Return

If no new boundaries are retained after boundary pruning, then the segmentation algorithm stops and returns the boundaries identified during the previous algorithm iterations, if any. If new boundaries are identified then they are grouped with the previously identified boundaries and sorted from the top to the bottom of the image. Then, for each pair of adjacent boundaries B_j and B_{j+1}, the search region is limited by flattening the image to B_j. Note that in this flattened image, there are some pixels that are below boundary B_{j+1}. Recall that in each column x, B_j^x and B_{j+1}^x are in the fixed basis and thus discontinuities must be found by SBL at these locations. On the other hand, no discontinuity should be found at any pixel $z^x > B_{j+1}^x$ since it does not lie in between the two boundaries B_j and B_{j+1}. So, all of the basis vectors with breakpoints below B_{j+1}^x are removed from the basis. An example of the search region limitation and reduced basis is shown in Figure 3.20.

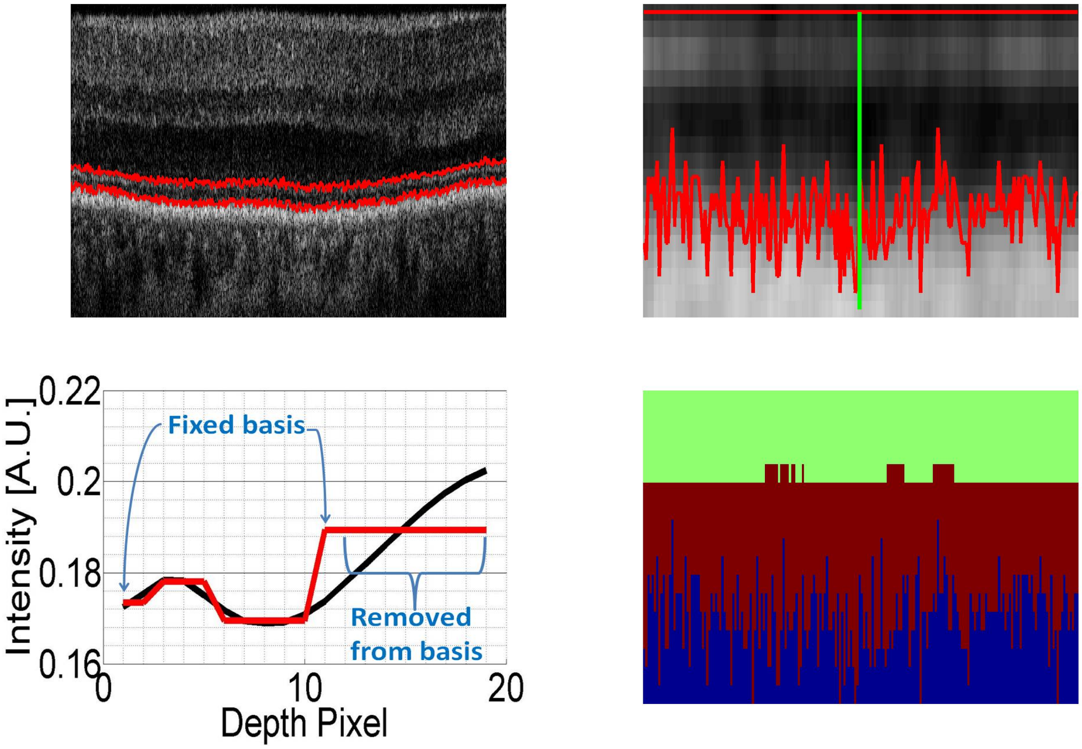


Figure 3.20: Limited search region between adjacent boundaries. Top left panel: adjacent boundaries identified in red. Top right panel: image flattened to first boundary, which lies on a horizontal line in the flattened image. The jagged red boundary is the second boundary in the left panel. The green line is an example A-scan. Bottom right panel: Sample A-line from the top right panel is shown in black and the resulting SBL approximation is shown in red. The fixed basis must be retained by SBL and includes the top and bottom boundaries from the top panel. The region labeled removed from basis are the basis vectors that lie underneath the last fixed basis vectors and are removed prior to the SBL run. Bottom right panel: resulting segmentation after running SBL and new boundary identification. Notice that a boundary (green to red) that was missed in the first algorithm run is found after reducing the search region.

The search region limitation and subsequent rerun of the entire segmentation algorithm is repeated for each pair of adjacent boundaries. This entire process repeats itself until no new boundaries are found. All identified boundaries are returned upon algorithm completion.

3.2.2.7 Summary of Parameters

The segmentation algorithm employs a number of parameters. These parameters are summarized in Table 3.2.

Parameter Name	**Value**	**Description**
$SBLIter$	30	Number of iterations of SBL algorithm on first search region limitations
$P_M(2P_M)$	4	Penalty for merging two (three) layers
$smThr$	3.5	Smoothness threshold used for boundary pruning
$mnBPDst$	4	Backwards elimination parameter used to combine close adjacent segments with same sign edge
$maxJump$	4	Reachability constraint parameter describing the maximum allowed jump of a boundary in pixels
ω	0.005	Cutoff frequency for Butterworth filter
$dsFact$	20	Downsample factor
$filtSz$	200	Transverse length of Gaussian filter used to smooth image

Table 3.2: Parameters used in segmentation algorithm

3.2.3 Summary

We demonstrated a novel retinal layer segmentation algorithm for OCT retinal images using sparsity constraints. The algorithm relaxes anatomical assumptions about the number of retinal layers as well as the relative positions of the layers made by most other segmentation algorithms. In order to provide a framework for objective quantitative evaluation of our algorithm, a method for generating synthetic retinal images with known segmentations is presented next.

3.3 Synthetic Retinal Image Generation

3.3.1 Introduction

Quantitative analysis of structural OCT images has become increasingly important for the study of various ocular pathologies [21,36]. For example, retinal layer thinning is an indicator of glaucoma [5]. Measurements of retinal layer thickness can be made after the layers are segmented using the magnitude of an OCT image. However, quantitative evaluation of the accuracy of segmentation algorithms usually relies on subjective segmentations made manually [26]. For objective quantitative evaluation, the real, or ground truth, boundaries must be known so that an objective error metric, such as mean-squared error between the ground truth boundaries and the boundaries identified by a segmentation algorithm, can be defined. The ability to generate synthetic retinal images with known ground truth boundaries would thus be useful for objective quantitative evaluation of segmentation algorithms.

In this section we present a novel method for generating a synthetic OCT retinal image that mimics an *in vivo* OCT retinal image. The algorithm uses a segmented, manually or automatically, retinal image as input and produces different instantiations of the image at a prespecified noise level. Although the algorithm relies on an input segmented image, the segmentation itself need not be exact in the sense that the identified boundaries are precisely equal the retinal boundaries. However, the synthetic images will be generated so that its boundaries can be used as the ground truth solution for subsequent segmentation. Our technique improves upon the method presented in [37] in two ways. First,the authors compute a smooth approximation to the input retinal image using parametrized surface fitting for the vitreous, retina and choroid separately. The implementation of this technique relies on the ability to capture the retinal structure in parametric form. In cases where the retinal structure is altered, for example due to the presence of cysts, the parametrized functions no longer model the structure accurately. Conversely, we compute a nonparametric smooth approximation to the input image that can capture unexpected structures as well. Thus, our technique is more general and could be applied to generate retinal images in pathological cases. A second limitation of the work in [37] is the generation of spatially correlated speckle noise. Specifically, the authors in [37] generate additive Gaussian noise and then smooth it at different spatial scales in order to create the typical grainy appearance of speckle noise. Their method does not accurately reflect the image formation process and requires ten independent parameters to be tuned to generate speckle. On the other hand, the speckle noise pattern generated with our technique is dictated by the OCT imaging system parameters as is the case in clinical OCT imaging. Furthermore, we are able to validate the axial and transverse speckle sizes of our speckle noise with clinical OCT images. Thus, our synthetic image generation technique more accurately mimics real OCT retinal images.

3.3.2 Model of OCT Retinal Image

Assuming that a sample consists of discrete point scatterers, the noiseless spectral-domain complex OCT signal emanating from a coherence volume centered at a point $(x_0, 0, z_0)$ within the sample can be written as:

$$A \propto \sum_{p \in CV_0} r_p \exp(-i2k_0(z_p - z_0))G_p, \tag{3.16}$$

where p is the pth particle index, r_p is its reflectivity and (x_p, y_p, z_p) is its position in the sample, CV_0 denotes the coherence volume centered at $(x_0, 0, z_0)$, $k_0 = 2\pi/\lambda_0$ is the source center wavenumber, and G_p is a Gaussian function defined by:

$$G_p = \exp\left(-8\left(\frac{(x_p - x_0)^2 + (y_p)^2}{\omega_0^2} + \frac{(z_p - z_0)^2}{\omega_z^2}\right)\right), \tag{3.17}$$

where ω_0 and ω_z are the $1/e^2$ transverse and axial resolutions, respectively. Assuming a locally homogeneous reflectivity so that $r_p = R$ for all $p \in CV_0$, the local intensity variation is due to speckle only. In this case, Equation 4.9 can be rewritten as:

$$A \propto R \cdot \sum_{p \in CV_0} \exp(-i2k_0(z_p - z_0))G_p = R \cdot S, \tag{3.18}$$

where S models the speckle caused by coherent scattering due to the interference of backscattered waves emanating from within the coherence volume. Then the measured OCT signal magnitude can be modeled as:

$$\hat{I} = |R \cdot S + N|, \tag{3.19}$$

where N models incoherent scattering such as multiple scattering, scattering by out-of-focus light and subwavelength scattering as well as system noise [24].

3.3.3 Procedure

The following sections describe methods for generating synthetic R, S and N in Equation 3.19. First, we compute a smooth approximation to the input image using piecewise constant regions. Unlike the work in [37], we use a nonparametric smooth approximation that is capable of representing structures that are difficult to capture parametrically. Furthermore, the smooth approximation framework allows for different levels of detail to be used to approximate the different retinal layers in the input image. Because we don't have a complete model for the image formation process for OCT retinal images, we leverage the data in the input segmented image to generate different instantiations of synthetic images. A scattered data interpolation technique is thus employed in order to compute a smooth approximation to the input image. Scattered data interpolation is commonly implemented by first computing a triangulation over the scattered data and then interpolating within each triangle. In our implementation, a random subset of the pixels in each retinal layer is first computed. Then, the pixels in each layer are mapped to a rectangular grid and a triangulation is computed over the random subset of pixels. The mapping to a rectangular grid, although not required, guarantees that all triangles in the triangulation lie entirely in the rectangular grid which simplifies the algorithm implementation. Next, the intensities of the pixels in each triangle are replaced with the respective root-mean-square value. This step preserves the root-mean-square intensity of the entire image. The rectangular region is then mapped back to the original image space. This process is repeated for each retinal layer until a smooth approximation to the entire input image is computed.

The next step in our synthetic image generation method is the generation of the speckle noise component S. Our novel technique for speckle noise generation utilizes the simulation developed in Section 4.2 and accurately reflects the real speckle noise generation process. Next, the additive noise component N is generated using the fact that this noise process is known to be a zero mean complex Gaussian process. R is computed next as a scaling factor to the smooth image approximation so as to preserve the mean squared value of the input image in the synthetic image. Finally, the components are combined using Equation 3.19 to produce the final synthetic magnitude image.

3.3.3.1 Smooth Approximation

In order to generate the local intensity in a synthetic image, a smooth approximation I_{SM} to the input segmented image is computed. This smooth approximation is used to model local reflectivity information. In order to maintain the layered structure present in the retina, the approximation is computed for each layer separately. Our smooth approximation also maintains the root-mean-square (RMS) intensity value over all of the retinal layers in the input image. A number of options exist for generating such a smooth approximation. As previously discussed, we employ a nonparametric technique using a piecewise constant representation to generate the smooth approximation for each layer. The process to compute the smooth piecewise constant approximation of the image is illustrated in Figure 3.21.

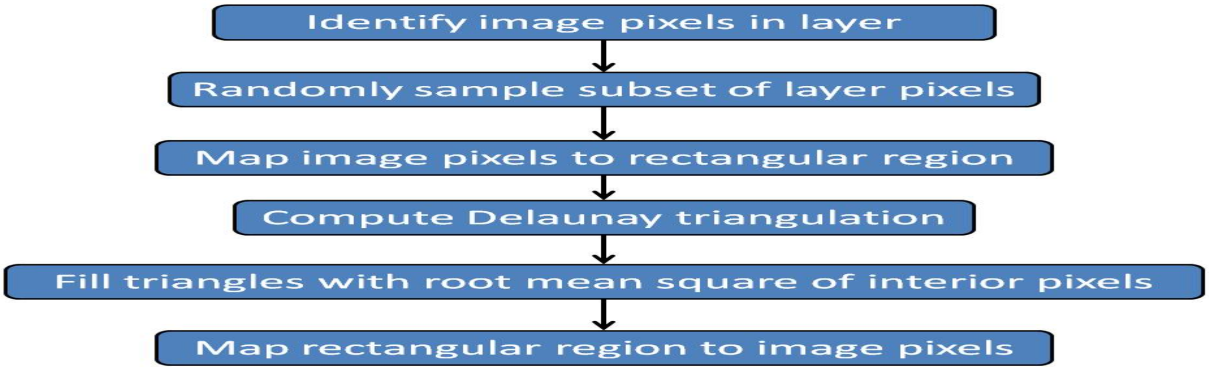

Identify image pixels in layer
Randomly sample subset of layer pixels
Map image pixels to rectangular region
Compute Delaunay triangulation
Fill triangles with root mean square of interior pixels
Map rectangular region to image pixels

Figure 3.21: Flowchart for computing smooth approximation of OCT image.

Prior to computing the smooth approximation, the layer labeled as the choroid is split into 2 regions so that the choroid with high signal can be generated without the low signal sub-choroid. An example of this is illustrated in Figure 3.22. Since all of the intensity variation in the vitreous and subchoroid is due to additive noise, the smooth approximation is set to zero in these regions.

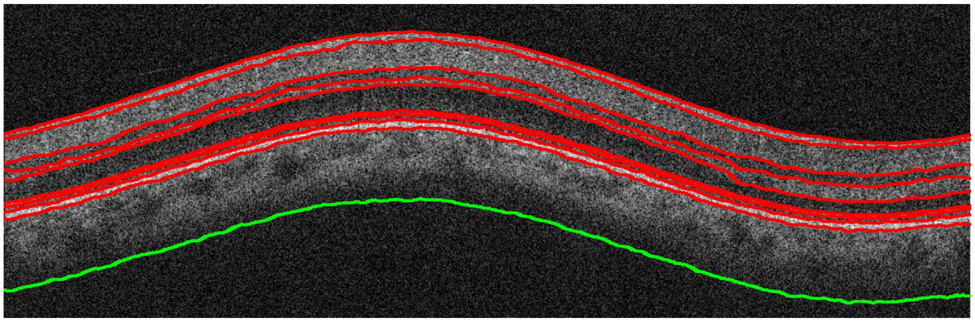


Figure 3.22: Sample segmented image used for synthetic image generation. The choroid is split into 2 regions to isolate high signal and low signal regions. The artificial boundary is indicated in green.

As a first step, the indices of all of the pixels in a retinal layer are identified using the segmentation of the input image. Then, a subset of these points is chosen randomly according to a uniform distribution. The smooth approximation framework allows for different levels of detail to be used to approximate the different retinal layers in the input image. For example, the choroid is a rough surface and thus requires a high level of detail to represent accurately. On the other hand, the retinal layers are relatively smooth surfaces and thus do not require as high a level of detail. The level of detail in the smooth approximation for each layer is determined by the number of points in the random subset of layer pixels. For the retinal layers (excluding the choroid), the number of pixels in the random samples range from 20 to 120, depending on the number of pixels in the layer. On the other hand, because the choroid requires a high level of detail to approximate realistically, the number of samples in the random subset is approximately 1300. These values were determined empirically and can be changed should a layer require more or less detail.

The corners of each layer are added to the random sample. These corners are defined by the top and bottom pixels in the first column of the layer and the top and bottom pixels in the last column of the layer. A layer with random samples is illustrated in Figure 3.23.

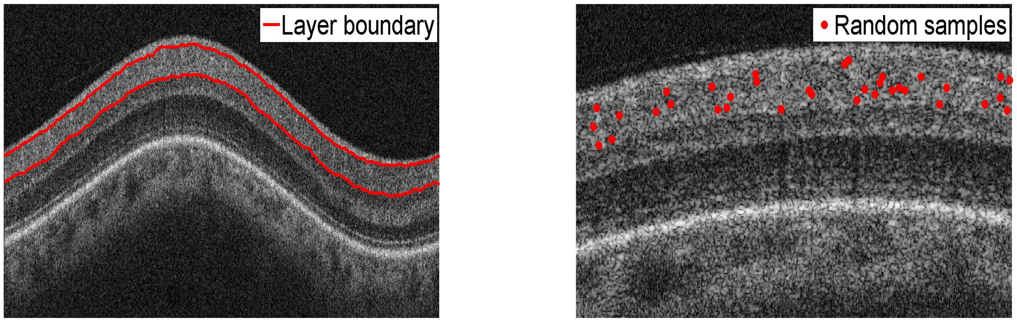


Figure 3.23: Sampling of pixels a in layer.

The next step is to construct and apply a mapping from the pixels in a particular layer in the image space to a rectangular grid. As previously mentioned, the purpose of applying this mapping is so that all triangles in the subsequent triangulation remain enclosed by the layer boundaries which simplifies the implementation. Assume that the image has M rows and N columns. A boundary between each pair of adjacent layers contains a single pixel in each image column. Thus, the bth boundary B_b is defined by a set of points $\{r^{B_b}(c_n)\}_{n=1}^{N}$, where c_n is the nth column index and r is the particular row of the bth boundary in the nth column. Layer L is enclosed by boundaries B_L above and B_{L+1} below. By convention, boundary B_L is in the Lth layer. Then, the set of all pixels $\{r_m(c_n)\}$ in a particular layer L is given by:

$$\{r_m(c_n) : r^{B_L}(c_n) \leq r_m(c_n) < r^{B_{L+1}}(c_n)\}_{n=1}^{N}. \tag{3.20}$$

Let $r_{min}^{L} \equiv \min_n r_m^{B}(c_n)$ be the highest pixel in the layer and $r_{max}^{L} \equiv \max_n r_m^{B+1}(c_n) - 1$ be the lowest pixel in the layer. These notation conventions are illustrated in Figure 3.24.

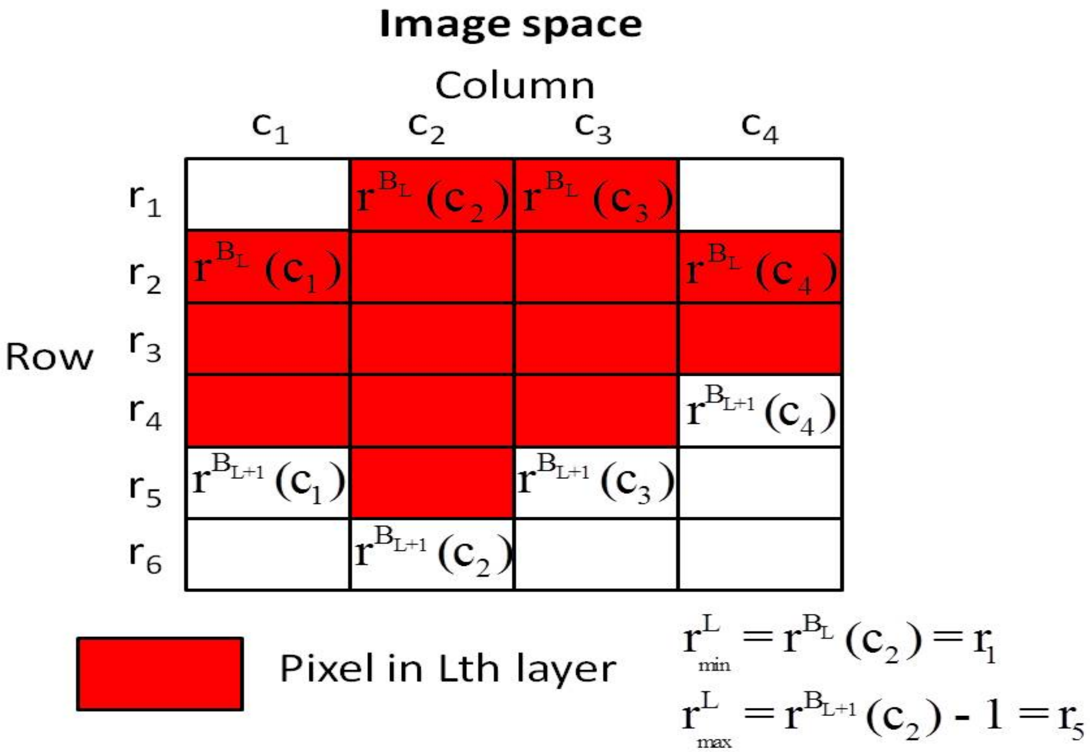


Figure 3.24: Notation conventions for mapping of layer to rectangular region. $r^{B_L}(c_j)$ denotes the row of the jth column of the Lth boundary. The boundary has a single row in each column of the image. By convention, the Lth boundary lies in the L layer. r_{min}^{L} and r_{max}^{L} denote the top most and bottom most pixels in the Lth layer, respectively.

The mapping maps all of the points in a layer to a rectangular grid defined by $[r^L_{min}, r^L_{max}] \times [c_1, c_N]$. Formally, the mapping is defined by

$$r^*_m(c_n) \equiv \lceil((r^L_{max} - r^L_{min}) \cdot \frac{r_m(c_n) - r^{B_L}(c_n)}{r^{B_{L+1}}(c_n) - 1 - r^{B_L}(c_n)} + r^L_{min})\rceil, \tag{3.21}$$

where the $\lceil x \rceil$ denotes the ceiling function defined by the smallest integer greater than or equal to x. An illustration of the mapping is shown in Figure 3.25. Note that after the mapping is applied, the top and bottom boundaries of the layer lie in a horizontal line. Also note no pixels are mapped to some of the positions in the rectangular grid.

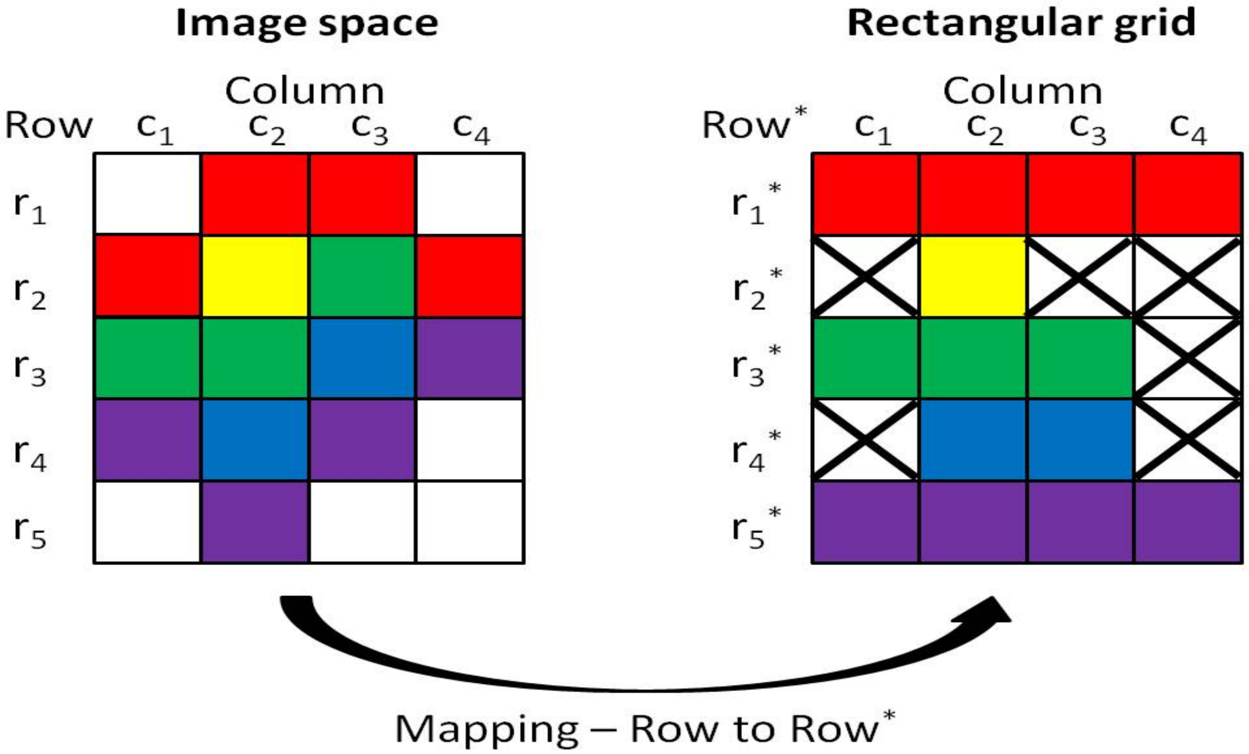


Figure 3.25: Mapping from image space to rectangular grid. The pixels in the top boundary of the layer are colored in red. The pixels in the bottom boundary of the layer are colored in purple. The white pixels in the image space denote pixels that are not in the layer. Note that after the mapping is applied both the top and bottom boundaries lie along horizontal lines along the top and bottom of the rectangular grid. Note also that since the mapping is not a surjection, some pixels in the rectangular grid are not mapped to by image pixels. These pixels are denoted in the right panel with a black x.

After the mapping is applied to all of the pixels in the region a Delaunay triangulation [38] is applied to the random samples. A common way to interpolate scattered data is to define a surface over a triangulation [39]. A Delaunay triangulation was chosen here because of its computational speed and, more importantly, the minimum angle result which states that over all possible triangulations, the Delaunay triangulation maximizes the minimum angle in each triangle [40]. This property discourages long, flat triangles. The left panel in Figure 3.26 illustrates an example of Delaunay triangulation defined over the samples. As a next step, each triangle is filled with the RMS of the intensities of all of the pixels inside the triangle. In order to identify whether a point is inside a given triangle, a technique based on [41] is used. Briefly, three inequalities are established that represent the condition that a point (x, y) lies on the same side of the infinite line along one of the triangle's three sides as the triangle does. If all of the inequalities hold true, then (x, y) must lie inside the triangle. Note that the vertices of the triangles always lie in more than one triangle. In order to preserve the RMS intensity value of the input image in the smooth approximation and in order to prevent additional scaling needed if a vertex were used in the RMS computation for multiple triangles, each vertex is assigned to a single triangle on a first-come first-served basis.

After the pixels in the rectangular grid are filled, they are mapped back to the original image space. The entire process for computing a smooth approximation is repeated for each retinal layer (excluding the vitrous and subchoroid). A sample smooth image is shown in Figure 3.27.

3.3.3.2 Speckle Noise Generation

The OCT simulation that will be introduced and detailed in Section 4.2 for a vessel in tissue is adapted for speckle noise generation here. As previously mentioned, our speckle noise generation technique more accurately reflects the real OCT retinal image formation process

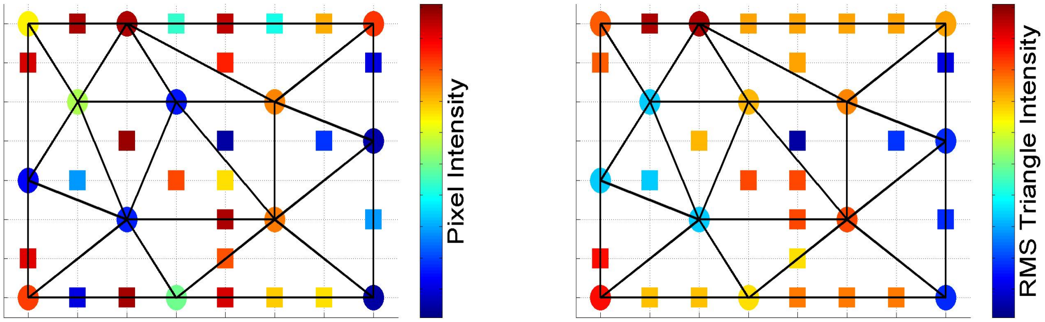


Figure 3.26: Example of Delaunay triangulation with filling. The triangulation is performed on a rectangular grid of pixels. The circles indicate pixels that are included in the random pixel sampling described previously. The squares indicate pixels that are in the layer but were not included in the sample. Left panel: original pixel intensity. Right panel: modified intensities computed by setting the intensity of each pixel equal to the root-mean-square (RMS) of the intensities of all of the pixels in a particular triangle.

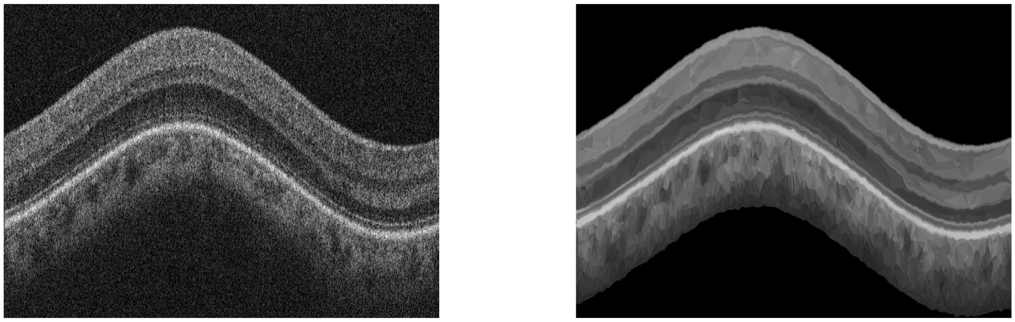


Figure 3.27: Triangulated smooth image approximation.

than does the work presented in [37]. A few modifications to the retinal vessel simulation are noted. Because the OCT signal magnitude and not the phase information is used for segmentation, only stationary scatterers were used by setting the vessel diameter to zero. Also, because the OCT amplitudes from the vitreous and sub choroid regions are dominated by noise, no speckle patterns were generated in these regions. A sample speckle pattern is shown in Figure 3.28.

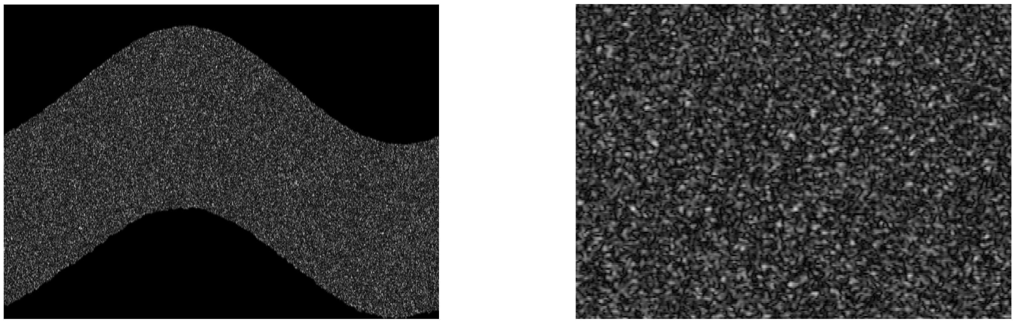


Figure 3.28: Simulated speckle pattern in retina. Left panel: speckle generated over entire retinal region. Right panel: zoomed in speckle pattern.

3.3.3.3 Additive Noise Model

A complex Gaussian noise process is used to model incoherent scattering and system noise in retinal OCT images [42]. The generation of this additive noise was performed efficiently in MatLab. The real and imaginary parts of N are independent and identically distributed Gaussian random variables with zero mean and variance σ^2. The magnitude of this noise process is known to follow a Rayleigh distrbution with parameter σ [42] with mean equal to $\sigma\sqrt{\frac{\pi}{2}}$ and variance equal to $\frac{4-\pi}{2}\sigma^2$. The second moment of the magnitude of the noise N is given by

$$\mathbb{E}(|N|^2) = 2\sigma^2. \tag{3.22}$$

In the vitreous, no oscillation in the signal intensity is expected and therefore all of its variation is due to noise. Thus, the statistics of the intensity distribution in the vitreous can be used to estimate the parameter σ. Specifically, given the set of V intensity samples in the vitreous $\{I_v\}$, the maximum-likelihood estimate of the parameter of the Rayleigh distribution is given by [43]

$$\hat{\sigma} = \sqrt{\frac{1}{2V}\sum I_v^2}. \tag{3.23}$$

This estimate is used to generate the complex Gaussian process.

In order to verify that the Rayleigh assumption is valid, the probability distribution function (pdf) of the intensities in the vitreous is estimated using a kernel smoothing technique [44] and compared to the Rayleigh distribution computed using the parameter $\hat{\sigma}$. These two curves are illustrated in the right panel of Figure 3.29.

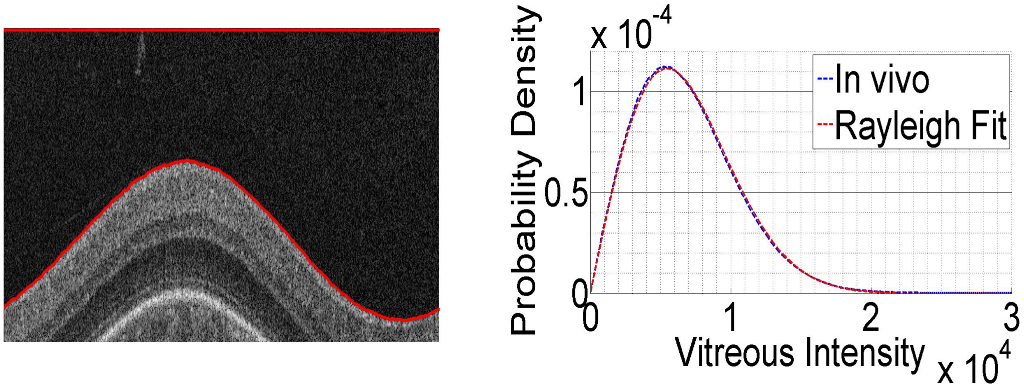


Figure 3.29: Validation of Rayleigh model for OCT signal noise. The intensity distribution in the vitreous (left panel) is estimated and compared to the Rayleigh probability distribution function (right panel) with parameter estimated by Equation 3.23.

For a quantitative evaluation of the noise model, a modified coefficient of variation of the root-mean-square deviation (CVRMSD) between the two curves in the right panel of Figure 3.29 is computed. Let $P(I)$ denote the estimated pdf of the vitreous intensity I and let $R(I)$ denote the Rayleigh pdf parameterized by $\hat{\sigma}$. The modified CVRMSD is defined by:

$$\mathrm{CVRMSD} = \frac{\sqrt{\sum_{v=1}^{V}(P(I) - R(I))^2}}{\frac{1}{2M}\sum_{v=1}^{V}(P(I) + R(I))} \cdot 100, \tag{3.24}$$

and is equal to 4.05%, which indicates a high degree of similarity between the model and the observed intensities.

3.3.3.4 Scaling the Smooth Approximation

In the previous sections, procedures for computing a smooth approximation, S and N in Equation 3.19 were presented. What remains is the choice of R is used to scale the smooth approximation. Taking the expected value of the magnitude squared of the phantom image $\hat{I}$ in Equation 3.19 gives

$$\mathbb{E}(|\hat{I}|^2) = |R|^2 \cdot \mathbb{E}(|S|^2) + \mathbb{E}(|N|^2), \tag{3.25}$$

where the fact that the coherent speckle noise S and incoherent scattering noise N are uncorrelated is used and that $\mathbf{E}(N) = 0$ since it is a zero mean noise process. Using Equation 3.22, the term $\mathbf{E}(|N|^2)$ is given by $2\sigma^2$. The term $\mathbf{E}(|S|^2)$ is difficult to calculate in closed form and so it is approximated from the data. The smooth approximation of the input image I was generated so that for each triangulated region, $\text{mean}(I_{SM}^2) = \text{mean}(I^2)$. In order to maintain a similarity between $\hat{I}$ and I the value of R is chosen so that the mean squared value in each triangulated region is preserved, or $\mathbf{E}(|\hat{I}|^2) = \text{mean}(I^2)$. Thus, the value of R is chosen to be

$$|R|^2 = \frac{\text{mean}(I_{SM}^2) - \mathbb{E}(|N|^2)}{\mathbb{E}(|S|^2)}, \tag{3.26}$$

A consequence of this choice is that retinal regions with high intensity will exhibit the speckle associated with S, whereas noise regions with low intensity will exhibit the noise properties similar to N. After computing R by taking the square root of Equation 3.26, the phantom image $\hat{I}$ is computed using Equation 3.19. A sample phantom image is illustrated in Figure 3.30.

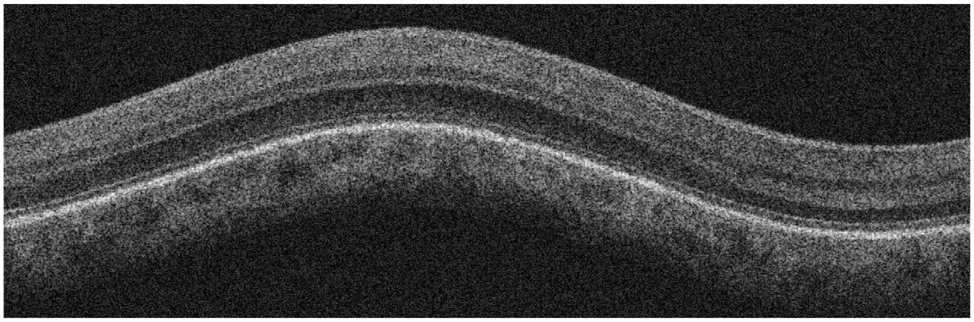


Figure 3.30: Sample phantom image

3.3.3.5 Signal-To-Noise Ratio

The signal-to-noise ratio (SNR) can be defined by

$$\mathrm{SNR} \equiv 20 \cdot \log_{10}\left(\frac{\mathrm{mean}|R \cdot S|}{\mathrm{std}(|N|)}\right), \tag{3.27}$$

where the mean of $|R \cdot S|$ is computed over the nerve fiber layer and the standard deviation of $|N|$ is equal to $\frac{4-\pi}{2}\sigma^2$ by construction. The phantom image generation implementation allows the user to change the image SNR. Specifically, a change in SNR by K dB can be written as

$$\mathrm{SNR}' - \mathrm{SNR} = 20 \cdot \log_{10}\left(\frac{\mathrm{std}(|N|)}{\mathrm{std}(|N'|)}\right) = 20 \cdot \log_{10}\left(\frac{\sigma}{\sigma'}\right) = K, \tag{3.28}$$

so that the new noise standard deviation is given by

$$\sigma' = (10^{-K/20})\sigma. \tag{3.29}$$

In order to implement this SNR change, the scaling parameter R is first computed using Equation 3.26 with the original noise level σ. Then, the new noise standard deviation computed using Equation 3.29 is used to generate the noise process N' and then Equation 3.19 is applied to compute the phantom image. An example phantom image with the SNR reduced from 15 dB to 2 dB is shown in Figure 3.31.

3.3.3.6 Comparison Between *In Vivo* and Synthetic OCT Data

The imaging parameters for the synthetic image generation, such as beam width, coherence length, axial and transverse step sizes, were chosen to mimic the imaging parameters of the clinical system used for the *in vivo* retinal images. To test the validity of the synthetic data, the *in vivo* and phantom images are qualitatively compared in Figure 3.32.

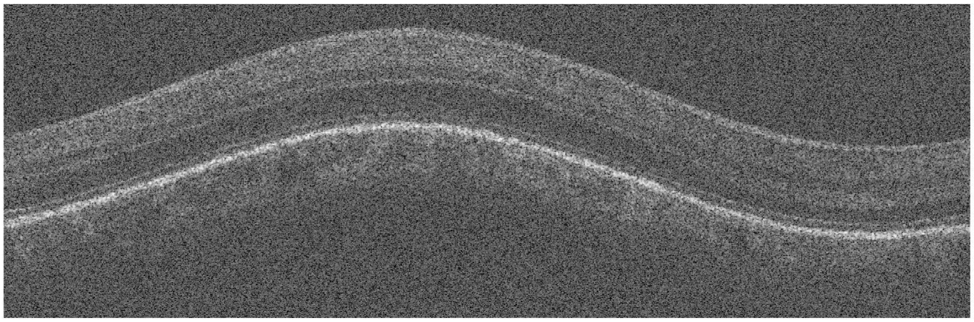


Figure 3.31: Sample phantom image with reduced signal-to-noise ratio (SNR). The SNR was decreased from 15 dB in the original image to 2 dB in this phantom image.

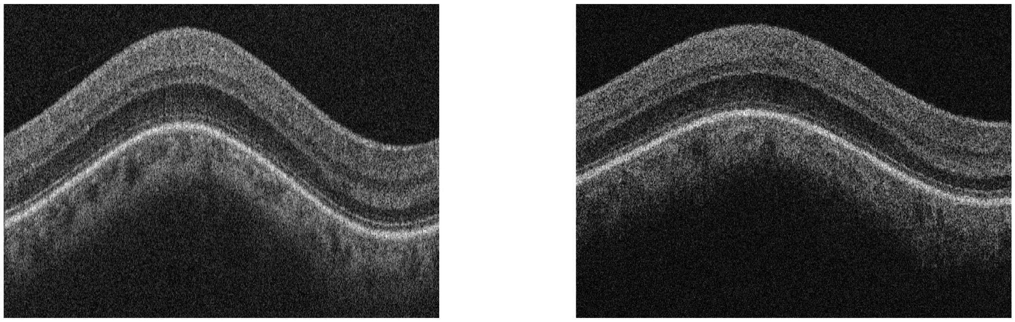


Figure 3.32: Visual comparison between *in vivo* and synthetic OCT Bscans. Left panel: *in vivo*. Right panel: Synthetic.

For a quantitative comparison, the distributions of intensities as well as the estimated speckle sizes are compared. As opposed to the intensity distribution in the vitreous, which is given by a simple Rayleigh density, the pdf of the intensities with the retinal layers is given by a complex expression with a Bessel function term [42]. Thus, instead of comparing the intensity distribution to a known parameterized distribution as was done for noise model validation, the pdfs of the ganglion cell complex intensity distribution between the *in vivo* and synthetic images are compared. The distributions are compared in Figure 3.33. For a quantitative comparison of the distributions, the CVRMSD is again computed as in Equation

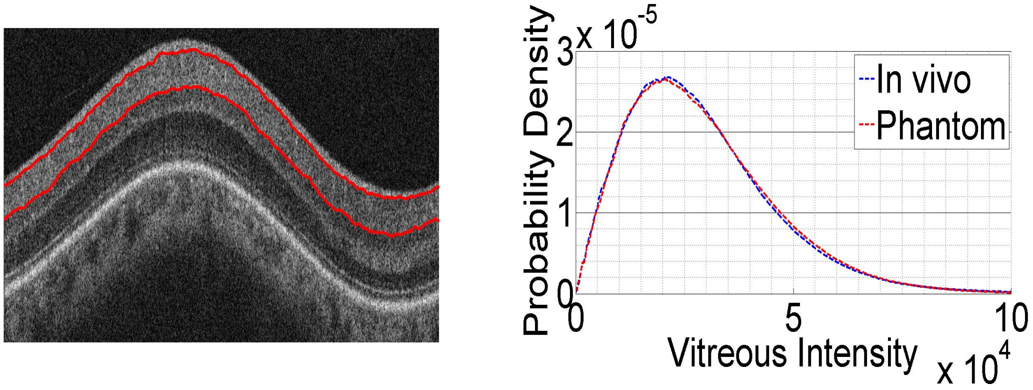


Figure 3.33: Comparison of GCL intensity distributions between *in vivo* and synthetic OCT data. Left panel: GCL region. Right panel: probability density estimates of intensity distributions.

3.24. In this case, the CVRMSD is equal to 4.62%, indicating a high degree of similarity between the *in vivo* and synthetic images.

For the comparison of the axial and transverse speckle sizes, normalized autocorrelations over a 50 μm by 250 μm window were computed in the ganglion cell layer [45]. Figure 3.34 shows a quantitative comparison. To quantify the speckle sizes, linear interpolation was used to compute with full-width at half-maximum (FWHM) widths in pixels. The computed axial speckle size was 1.98 pixels for the retinal image and 1.70 pixels for the synthetic image. The computed transverse speckle size was 6.30 pixels for the retinal image and 6.25 pixels for the synthetic image. The difference between the retinal and synthetic axial speckle sizes is likely due to dispersive broadening in clinical measurements that is not accounted for in the synthetic data generation.

3.3.4 Summary

We introduced a novel method for generating synthetic OCT retinal B-scan intensities. Our technique improves on the work in [37] by (a) computing a nonparametric smooth approximation to the input image that can represent unexpected changes in retinal layer structure

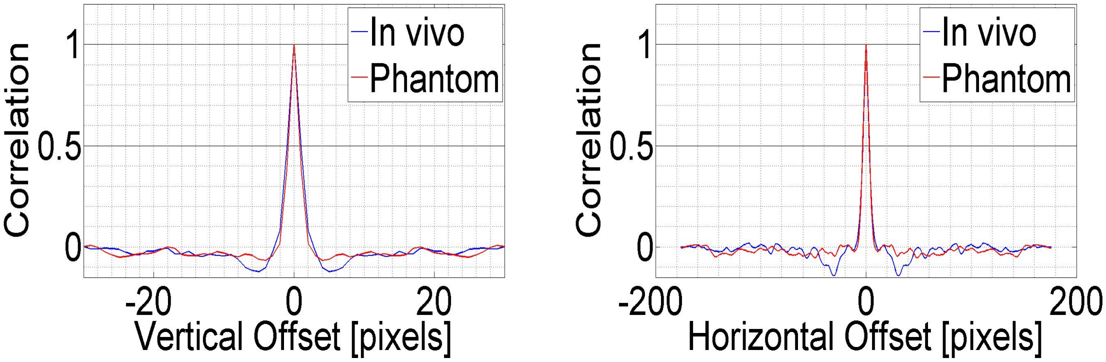


Figure 3.34: Comparison between speckle sizes in *in vivo* and synthetic OCT data.

and (b) modeling the speckle noise formation process accurately. Because the ground truth boundaries in the synthetic images are known, this method can be used for quantitative analysis of image processing algorithms when applied to retinal images. We next test our novel segmentation algorithm using both *in vivo* retinal images as well as synthetic retinal images generated by our method.

3.4 Evaluation of Segmentation Algorithm

3.4.1 Introduction

In this section, we evaluate the performance of our novel retinal layer segmentation algorithm presented in Section 3.2. First, we show that our segmentation algorithm is able to find the nine retinal layers (including the choroid) using *in vivo* retinal images obtained in our clinic. The phantom image generation technique presented in Section 3.3 is then used to test the novel segmentation algorithm. By doing so, a quantitative evaluation of the algorithm parameters, accuracy and noise sensitivity can be made automatically and objectively using the known ground truth boundaries. This objective evaluation is not possible using the *in vivo* retinal images because their ground truth segmentations are not known.

The synthetic retinal images used to test our segmentation algorithm were generated using the *in vivo* retinal images that were segmented by our segmentation algorithm. The imaging parameters settings used to generate the synthetic data were chosen to mimic the clinical system used for *in vivo* imaging.

3.4.2 Evaluation with *In Vivo* Retinal Images

Two normal eyes were scanned using the RT-Vue Fourier-Domain OCT system (Optovue Inc., Freemont, CA), which has 5 micron depth resolution and can perform 26000 axial scans per second. A double circular scan protocol [19] was used with diameters of 2.2 and 1.9 millimeters, respectively. The segmentation algorithm was applied to a single scan at each diameter of each eye. In total, four images were segmented.

Figure 3.35 shows two sample segmented images, one from each eye. Both segmentations resulted in the identification of the eight retinal layers and the choroid. For the rest of this discussion, we will group the choroid with the retina and refer to nine retinal layers. The top image in the figure was segmented in two iterations of the algorithm as the inner segment and outer segment (IS/OS) were merged on the first iteration. On the other hand, the bottom image was segmented in a single algorithm iteration. Note that although the segmentation algorithm does not rely on a prespecification of the number of retinal layer boundaries, it was able to find the nine layers in all four images. These results are in good agreement with other state-of-the-art segmentation algorithms that search for nine layer boundaries (e.g. [26]).

3.4.3 Parameter Determination using Synthetic Data

One of the primary uses of generating synthetic data is to evaluate the parameter choices for various image processing algorithms. Three parameters from the segmentation algorithm presented in Section 3.2 are sequentially examined in order to determine a preferred setting

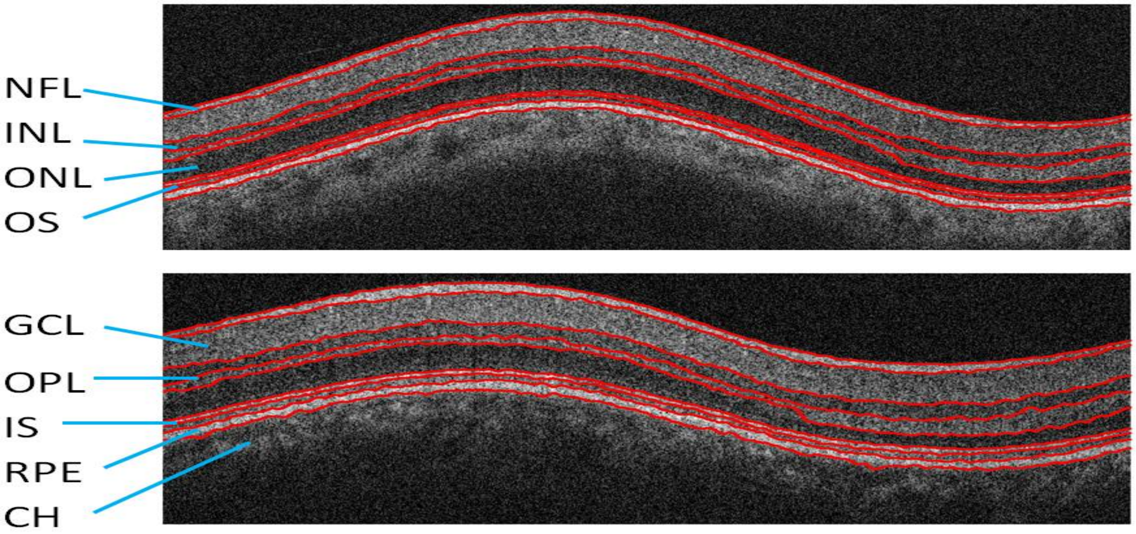


Figure 3.35: Segmentation of retinal images. Boundaries are labeled. NFL: nerve fiber layer, GCL: ganglion cell layer, IPL/OPL: inner/outer plexiform layer, INL/ONL: inner/outer nuclear layer, ELM: external limiting membraneIS/OS: inner/outer segment, RPE: retinal pigment epithelium, CH: choroid.

for each of them. A parameter setting is considered valid only if that setting can be used to segment the correct number of boundaries in the synthetic images. For the synthetic image used in these experiments, the number of required boundaries is nine. For parameter determination, the signal-to-noise ratio (SNR) of the generated image was set to the SNR of the *in vivo* image and was approximately 15 dB.

Note that the contributions presented in this section are not meant to establish the exact optimal parameters. Rather, our intention is to illustrate that the synthetic image generation technique can be used to validate and tune parameter choices guided by intuition about the physical parameter interpretations.

3.4.3.1 Smoothness Threshold

The hard smoothness threshold in the segmentation algorithm presented in Section 3.2 is used to prune boundaries that are not smooth enough. This threshold was implemented as an ad-hoc pruning technique to remove false positives from the segmentation after each algorithm iteration. The smoothness threshold is compared to the mean-squared error (MSE) between an identified boundary and a smoothed version of it, and the boundary is pruned if the MSE exceeds the threshold. If this threshold is too low, then some real boundaries will be pruned resulting in false negatives. On the other hand, if this threshold is too high, then the algorithm will find false positives that it deems are real boundaries even though they are not. Thus, we expect that the segmentation algorithm will effectively segment all nine layer boundaries only if the smoothness threshold is within a certain range.

Physical intuition can help guide the selection of the smoothness threshold. The MSE is measured in squared pixels and represents the average squared deviation of an identified boundary from its smoothed version. Since the retinal layers are smooth continuous structures, we expect that a boundary between two layers does not deviate from its smoothed version too much. On the other hand, we do expect a nonzero MSE due to the presence of speckle and other imaging noise. Intuitively, then, we expect that the real boundaries between retinal layers will have a nonzero MSE that is no larger than approximately 10 squared pixels, which, using the fact that the axial distance between adjacent pixels is 2.5 μm, maps to 8 μm. This gives a rough order of magnitude estimate that helps guide parameter selection. In order to verify our physical intuition, we varied the threshold on this MSE from 1.5 to 5.5 in steps of 1. The experiment showed that 1.5 was too low and resulted in missing boundaries. Conversely, 5.5 was found to be too high and resulted in extra boundaries being segmented. When the smoothness constraint was set between 2.5 and 4.5, however, the number of identified boundaries equals nine. Thus, the valid range for this smoothness threshold is between 2.5 and 4.5, which is in good agreement with our physical intuition. Furthermore, the percent difference over the valid range is 66%, which indicates that our segmentation algorithm is robust to the MSE threshold parameter. An intermediate value of 3.5 is used as the smoothness constraint from here on. Some results of this test are shown in Figure 3.36.

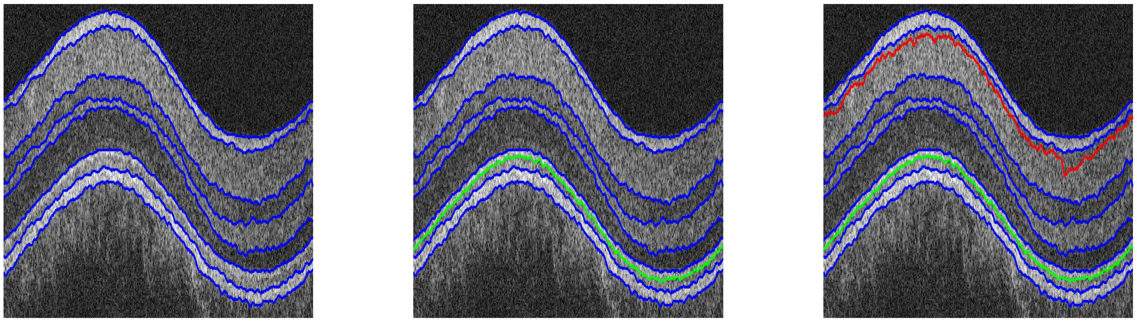


Figure 3.36: Effect of smoothness threshold on segmentation algorithm. Left panel: Eight boundaries are found when the smoothness threshold is set to 1.5. The outer segment boundary is missing. Middle panel: The correct nine boundaries are found when the smoothness threshold is set to 3.5. The outer segment boundary is illustrated in green. Right panel: An additional false boundary is found when the smoothness threshold is set to 5.5. This false positive is shown in red.

3.4.3.2 Number of Sparse Bayesian Learning Algorithm Iterations

The sparse Bayesian learning (SBL) algorithm is used to compute a sparse approximation for each column in an image by inserting breakpoints at discontinuities between piecewise constant segments. As discussed in Section 3.2, the SBL algorithm runs for a pre-specified number of iterations. In general, the number of identified breakpoints increases with the number of SBL iterations up to a point. If the number of iterations is too low then the segmentation algorithm may not find enough breakpoints which can result in missing boundaries in the final segmentation. On the other hand, the computational time required to run the SBL algorithm scales with the number of iterations, and so if the number of iterations is too high then SBL will be very slow. Thus, we want to find the minimum number of iterations needed for the algorithm to locate the correct number of boundaries.

Again, physical intuition can help guide the selection of the number of iterations for which to run the SBL algorithm. We know that there are at least nine identifiable retinal layer boundaries which means that the minimum number of SBL iterations required to locate the boundaries is nine. Ideally, a single breakpoint would be identified for each retinal layer boundary. However, due to the presence of speckle noise, some breakpoints correspond to discontinuities in the speckle pattern and not to real retinal layer boundaries. Note that SBL is relatively insensitive to the additive Gaussian noise present in OCT retinal images (shown in Section 3.4.4) and thus the primary noise source of error is speckle noise. Furthermore, due to the high level of smoothing applied to an image prior to applying SBL, some retinal boundary edges stretch across multiple breakpoints. Thus, we expect to have to run the SBL algorithm for more than nine iterations to locate the breakpoints that correspond to all of the real retinal layer boundaries.

In this experiment, we vary the number of SBL iterations in steps of 5. As can be seen in Figure 3.37, 25 SBL iterations are not enough to capture all of the retinal layer boundaries, as the inner nuclear layer boundary is not identified by our segmentation algorithm. On the other hand, 30 iterations is high enough to locate all nine of the retinal layer boundaries. Increasing the number of iterations up to 80 does not result in a change in the number of identified retinal layer boundaries. Since the minimum number of iterations that is able to find all of the retinal layer boundaries is desired in order to minimize computation time, the number of iterations used from here on is 30. This experiment can also be used to determine a good rule of thumb for initial guesses of the best number of SBL iterations. Since we showed that locating the nine retinal layer boundaries requires at least 30 SBL iterations, we propose that a good initial guess of the number of SBL iterations is three times the number of desired breakpoints. The ideal number of SBL iterations, which should be chosen to balance the ability to find all of the desired breakpoints with the amount of computation time, should be close to this value.

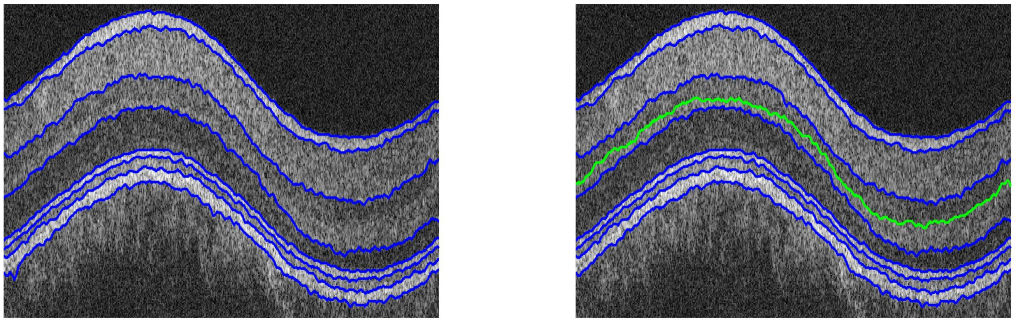


Figure 3.37: Effect of number of Sparse Bayesian Learning algorithm iterations on segmentation algorithm. Left panel: When the number of iterations is 25, the inner nuclear layer boundary is missed. Right panel: When the number of iterations is 30 or above, all nine retinal boundaries are found. The inner nuclear layer boundary is illustrated in green.

3.4.3.3 Merge Penalty

After determining the valid ranges for the previous two hard constraints/thresholds, the final segmentation algorithm parameter that is tested is the merge penalty parameter. Recall that the merge penalty is used as a regularization term in the build new layers step of the segmentation algorithm in Section 3.2 in order to remove the bias towards merging layers. If this parameter is too low, then the segmentation algorithm may miss boundaries because the corresponding layers may merge too easily. On the other hand, if this parameter is too high then layers that should merge may not. As is the case for the number of initial SBL iterations, it is desired to minimize the merge penalty because of computation time considerations. Specifically, the size of the graph used in the identify new layers module scales with the number of layers at the current column of the image. As the layers merge, the total number of layers is reduced and thus there are fewer nodes in the graph. If two or more layers are going to merge, it is preferable that they do so early in the algorithm.

Unlike the smoothness threshold parameter and the number of SBL iterations, the merge penalty parameter does not have a simple physical interpretation. However, we can still estimate its rough order of magnitude by considering the cost functions in Equation 3.13

and Equation 3.14. Recall from Section 3.2.2.4 that the merging of two layers will cause the shortest path in a graph traversal to visit two fewer nodes compared to the case for which there is no merging of layers. In this case there will be one less beginning node to end node transition and one less end node to beginning node transition. Because the maximum beginning node to end node transition cost without a merge is 2 and the maximum end node to beginning node transition cost is 1, the reduction in cost for a single merge between two layers would be 3 if no regularization term were applied. Thus, a good initial guess for the merge penalty parameter is 3. In order to verify that our understanding of the merge penalty is correct, we vary the parameter between 3 and 6 in steps of 1. Figure 3.38 shows that fewer than the desired nine boundaries are found when this parameter is set to 3. On the other hand, for values between 4 and 6, all nine boundaries are found. Thus, the preferred parameter setting for the merge penalty is 4.

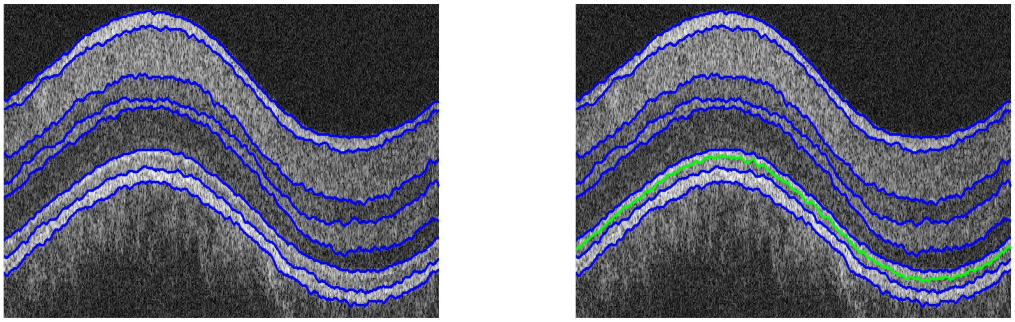


Figure 3.38: Effect of the merge penalty on segmentation algorithm. Left panel: When the merge penalty is 3, the outer segment boundary is missed. Right panel: When the merge penalty is 4 or above, all nine retinal layer boundaries are found. The outer segment boundary is shown in green.

3.4.4 Segmentation Algorithm Noise Sensitivity and Accuracy

Generating synthetic OCT images enables a quantitative evaluation of the segmentation algorithm introduced in Section 3.2. Using the parameter settings found in Section 3.4.3, the segmentation algorithm was run on a total of 30 synthetic images generated from two eyes. A single *in vivo* retinal image from each eye was used to generate 3 instances of noise-free images. For each of these instances, 5 different additive Gaussian noise settings were added that resulted in SNRs ranging from 2 to 22 dB. After running the segmentation algorithm, the number of boundaries identified was calculated. The algorithm was deemed to have failed if less than nine boundaries were found. For SNRs above 10 dB, the segmentation algorithm successfully identified all nine boundaries. For SNRs below 7.5 dB, the algorithm tends to miss the IS/OS segment, which is very thin boundary with low contrast. These results indicate that the algorithm is robust to noise for SNRs above 10 dB, but breaks down for SNRs below 7.5 dB. Two examples of segmented images are shown in Figure 3.39.

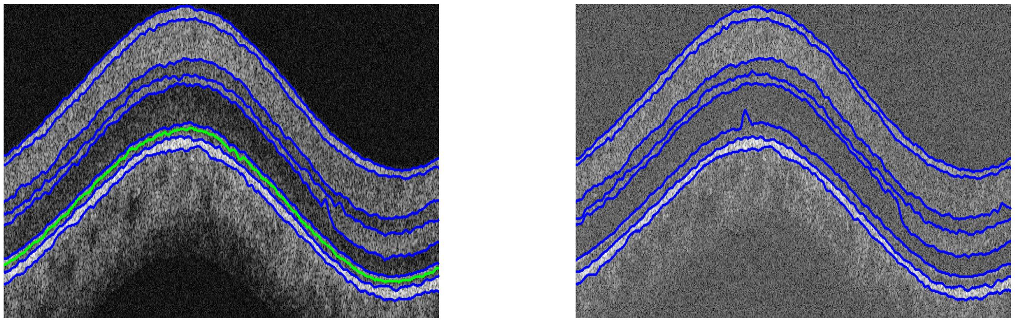


Figure 3.39: Effect of the signal to noise ratio (SNR) on segmentation algorithm. Left panel: All boundaries are correctly identified when SNR = 18.1 dB. Right panel: Outer segment boundary is not identified when SNR = 3.5 dB.

In order to quantitatively evaluate the accuracy of the segmentation algorithm, the identified boundaries were compared to the ground truth boundaries. The average mean-squared error (MSE) was used as a metric to determine the boundary accuracy. Note that only the cases for which all nine boundaries were identified are included in this quantitative evaluation. Figure 3.40 illustrates that the MSE lies between 1.4 and 2.1 squared pixels for all SNRs tested, indicating a high level of accuracy. One counterintuitive observation is that it seems that the MSE actually decreases for decreasing SNR from 22 dB down to 7.5 dB. This suggests that the segmentation algorithm is more sensitive to the speckle noise pattern than to the additive complex Gaussian noise and that adding Gaussian noise breaks up the speckle pattern in a way that aids the segmentation algorithm. This also suggests that incorporating speckle statistics into the segmentation algorithm could increase its robustness.

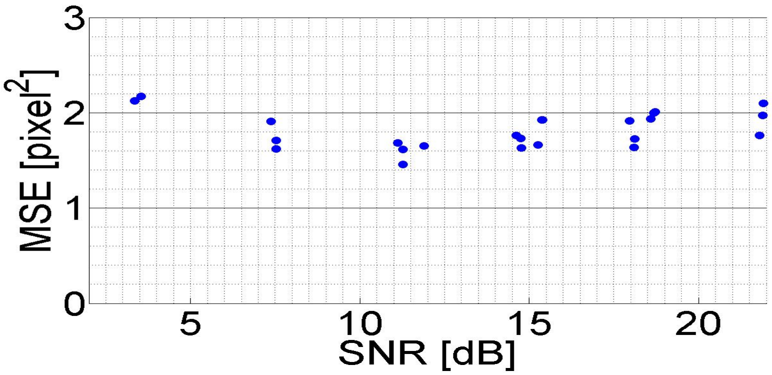


Figure 3.40: Image regions used to compare distributions and speckle sizes of *in vivo* and synthetic OCT data.

3.5 Conclusions

In this chapter we studied the segmentation of OCT retinal images. In Section 3.2, we developed a segmentation algorithm for OCT retinal images. The two primary novelties of our algorithm are (a) the sparse transformation of the input image from the pixel space to a much smaller space while still capturing layer properties that are useful for image segmentation and (b) the relaxation of anatomical assumptions about the number of retinal layers as well as the relative positions of the layers made by most other segmentation algorithms. There are a number of limitations to our novel segmentation algorithm. First, building the layers using the graph-based approach relies on the assumption that the layers extend across the entire image. Blood vessel shadowing [46] adds artificial vertical discontinuities in the retinal layers. If the SBL algorithm misses a breakpoint in even one of the columns with shadowing, the corresponding layer will not be found in the build layers step. This problem could be alleviated by adding a column-look-ahead step, where the build layers step could be run for multiple column steps (eg 1 and 2 and 3) and the extension with the lowest cost could be chosen. Another limitation is that the framework of the graph-based approach does not allow layers to have zero width in any column. However, near the fovea, a few of the layers do in fact have zero width. Thus, the algorithm cannot successfully segment images that extend across the fovea centralis without modification.

In order to help quantitatively evaluate our segmentation algorithm, we introduced a novel technique for generating synthetic OCT retinal B-scan intensities with known ground truth segmentations in Section 3.3. Our method improves upon an existing synthetic image generation method by (a) computing a nonparametric smooth approximation to the input image that can represent unexpected changes in retinal layer structure and (b) modeling the speckle noise formation process accurately. The smooth approximation and complex additive Gaussian noise models used for synthetic data generation were validated by showing that the intensity distributions in the vitreous and ganglion cell layer are in good agreement with the respective distributions in *in vivo* images. The synthetic speckle noise pattern generated was validated by showing that the axial and transverse speckle patterns are similar to the speckle patterns in *in vivo* retinal images. One limitation of our synthetic image generation method is that the synthetic images appear similar to the *in vivo* images. One way to generalize this technique would be to allow deviations from the input segmentation boundaries. Other modifications can be made to allow for simulated ocular pathologies such as cysts by adding regions of low intensity at random positions in the retina.

In Section 3.4 we evaluated our segmentation algorithm. First, we used *in vivo* retinal images to show that our algorithm is capable of segmenting all nine retinal layers. Then, we described how our synthetic image generation technique can be combined with physical intuition to guide the choices of the parameters for our segmentation algorithm. This methodology can be applied to any segmentation algorithm to determine good parameter settings. Finally, we tested the accuracy and noise sensitivity of our segmentation algorithm by using synthetic images generated with our technique. Specifically, we varied the signal to noise ratio (SNR) in the synthetic images and computed the mean-squared error (MSE) between the ground truth boundaries and those identified by our segmentation algorithm. We showed that for SNRs between 10 dB and 22 dB, the segmentation algorithm identifies all nine retinal layers, which indicates that our algorithm is fairly robust to noise. Furthermore, the average MSE between the known boundaries and the boundaries identified by our segmentation algorithm is less than 2 squared pixels, which indicates that our algorithm is able to segment the retinal layers with high accuracy.

Chapter 4

Blood Flow Velocity Measurement with Doppler OCT

4.1 Introduction

Accurate knowledge of ocular perfusion is important for the understanding of pathophysiology of many retinal diseases. For example, both glaucoma and diabetic retinopathy are linked to decreases in retinal blood flow rates [47, 48]. Thus, there is considerable interest in methods for the assessment of retinal blood flow for the diagnosis of eye diseases. Fluorescein angiography, in which dye is injected and its velocity is determined using physical displacement, can be used for qualitative flow analysis [49], but quantitative measurements are not feasible with this modality. Doppler methods, based on the frequency shift imposed on an electromagnetic wave by moving particles, have been applied to retinal circulation analysis. Laser Doppler velocimetry can be used to measure blood flow in absolute units [50] by directly measuring the Doppler shift induced by moving particles. However, since it operates in the frequency domain, temporal (and consequently spatial via $v = ct$) resolution is compromised and speed distribution across a vessel cannot be calculated [12]. Furthermore, the inability to resolve flow at different depths reduces its utility, especially for *in vivo* measurements.

Doppler OCT is able to compute the blood flow velocity distribution inside a vessel as well as depth resolved images of retinal blood flow. Enface images of vascular networks can also be computed. Doppler OCT methods are sensitive to the axial flow velocity component only. As noted in Section 2.3, blood flow in a retinal vessel can be computed with Doppler OCT only if the Doppler angle, or angle between the incident light beam and the blood vessel, is known. The standard method for computing blood flow velocities in retinal vessels is phase-resolved (PR) Doppler OCT. PR is able to calculate blood flow velocity by evaluating the phase differences between adjacent A-lines [13–17]. Ideally, the two A-lines used to compute the Doppler phase shifts in a blood vessel would be acquired at the same transverse beam position. In practice, however, the beam must be scanned transversally in order to reduce the total amount of time a patient's eye is imaged. This transverse scanning, along with other unwanted sources of noise such as patient movement, induces nonzero Doppler shifts even in stationary background tissue. An example PRDOCT image is shown in Figure 4.1.

The PR algorithm has been used to measure total retinal blood flow near the optic nerve region where the axial component of blood flow velocities is typically high [19]. However, a number of limitations to accurately measuring flow velocities in a blood vessel with PR Doppler OCT have been noted. For example, in the macular region, the axial component of blood flow velocities is typically low due to both near-perpendicular incidence angles and very slow flow in small vessels. The small axial flow component makes vessels in the macula difficult to detect with PR Doppler OCT because the associated Doppler phase/frequency bandwidth mixes with the bandwidth of the surrounding tissue. Although spatial frequency filtering Doppler OCT methods have been proposed to alleviate this difficulty, a quantitative analysis of the effect vessel size on Doppler OCT measurements is still lacking.

In order to overcome the difficulty in assessing the ability of Doppler OCT techniques to measure blood flow velocities accurately, we introduce a novel simulation of a blood vessel immersed in static tissue in Section 4.2. For the first time, a full two dimensional retinal

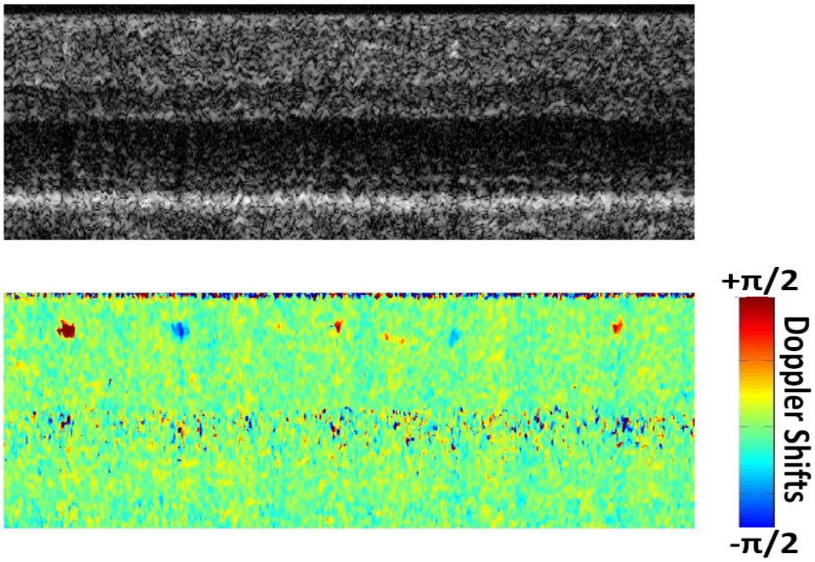


Figure 4.1: Example of a phase resolved (PR) Doppler OCT image in the human retina. Top panel: OCT structural image. Bottom panel: PR Doppler OCT image of the measured Doppler phase shifts. The green and yellow indicate the small (nearly zero) Doppler shifts in the static background tissue that are due to beam motion, patient motion and system noise. The red and blue circular regions indicate retinal blood vessels. The sign of the Doppler phase shift indicates the direction of the flow velocity.

vessel simulation is implemented that accurately represents the way Doppler OCT is applied in clinical studies. In practice, Doppler OCT techniques are applied to two dimensional B-scans and flow is computed by averaging the Doppler phase shifts inside a blood vessel. Other Doppler OCT simulation techniques have primarily been one dimensional in nature so that effects on Doppler flow measurements cannot be accurately assessed. In contrast, our simulation can be used to objectively and quantitatively evaluate the flow accuracy of various Doppler OCT algorithms by providing ground truth knowledge of the real flow velocities.

The utility of our retinal vessel simulation in quantitatively evaluating Doppler OCT algorithms is then shown with two case studies. In Section 4.3, we compare the accuracy of the PR algorithm and a spatial frequency filtering technique called moving-scatterer-sensitive (MSS) Doppler OCT as blood vessels and axial flow components become increasingly small. For this case study, we first validate the simulation with *in vivo* Doppler OCT retinal images and then use the simulation to extrapolate the results in a controlled setting. The material presented in Section 4.3 was published in [51]. In Section 4.4, we study the effect of transverse step size on the PR algorithm to show that it produces inaccurate blood flow velocity measurements as the transverse step size between adjacent axial scans increases. For this case study, we first validate the simulation with *in vitro* flow phantom experiments and again use the simulation to extrapolate the results.

4.2 Retinal Vessel Simulation

4.2.1 Introduction

In this section, we develop a novel simulation of a blood vessel immersed in tissue. The primary motivation for the development of this simulation is the need for objective quantitative accuracy analysis of Doppler OCT algorithms in a controlled environment. The synthetic data provides ground truth knowledge of blood flow velocities that can be used to assess the accuracy of Doppler OCT algorithms.

The primary novelty of our technique is that it simulates an entire two dimensional OCT B-scan of flowing particles that model moving red blood cells surrounded by stationary particles that model static background retinal tissue. Our simulation can be used to assess blood flow measurements in a blood vessel. Other simulations have been developed in order to evaluate Doppler OCT algorithms [18, 52]. However, our simulation is more realistic in terms of mimicking clinical Doppler measurements because (a) B-scans are used to assess velocity measurements and (b) we model both stationary and flowing particles. For example, the authors in [52] developed a one dimensional simulation to study the effect of transverse step size on phase-resolved (PR) Doppler OCT measurements. Because their method is one dimensional, it cannot be used to assess the effect on Doppler flow measurements that are computed by integrating the Doppler phase shifts in a circular cross section of a blood vessel. Furthermore, their simulation did not include flowing particles so that the results are limited to the effects on Doppler phase shifts in background tissue. On the other hand, the authors in [18] developed a simulation to study the effect of signal to noise ratio on Doppler velocity measurements in flowing particles. This simulation, however, is zero dimensional in the sense that the authors simply sampled from a theoretical phase distribution. The lack of geometry in this simulation as well as the exclusion of stationary particles limit its general application to the assessment of clinical Doppler OCT measurements. Each of these simulations were developed for particular studies. On the other hand, our novel simulation can be used for a variety of Doppler OCT studies as discussed in Section 4.3 and Section 4.4.

The organization of this section is as follows. First, the geometry used to model a blood vessel is introduced. Then, we provide an overview of the simulation procedure and then describe each of the simulation steps in detail. A table summarizing the simulation parameters is then provided. Finally, a visual comparison between a blood vessel detected in an *in vivo* image and a simulated blood vessel is shown. The simulation is validated in subsequent sections with two case studies.

4.2.2 Retinal Blood Vessel Geometry

The geometry of a blood vessel is approximated with a cylinder with arbitrary orientation. Then the blood vessel is completely described by the vessel radius r, a unit vector $\hat{n}$ that is normal to the vessel cross section and a point $P_0 = (x_0, y_0, z_0)$ that is centered on its circular cross section or core. The simulation geometry is illustrated in Figure 4.2.

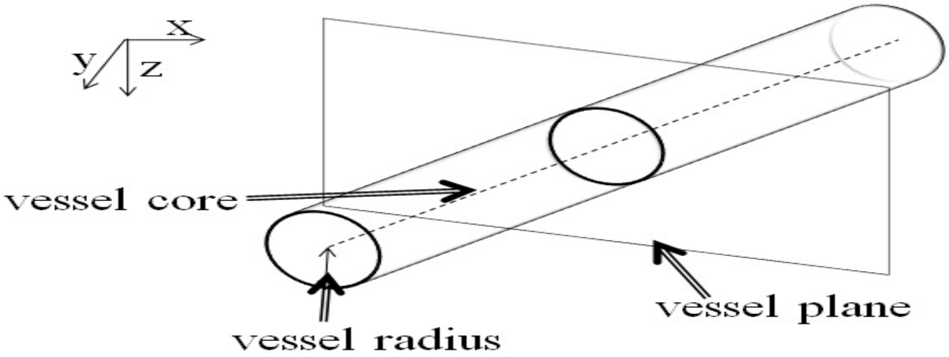


Figure 4.2: Vessel geometry for the Doppler OCT simulation.

Let P be any point in three-dimensional space and let $\hat{p}$ denote the unit vector from the point P_0 on the vessel core to the point P. Then, the point P_c on the vessel core that is closest to P can be found using the projection:

$$P_c(P) = P_0 + \langle (P - P_0)\hat{p}, \hat{n} \rangle \, \hat{n}, \tag{4.1}$$

where the angled brackets denote a vector inner product. Note that P_C is a function of the arbitrary point P and that the explicit dependence will be dropped from here on. Then, the equation of the cylinder is given by:

$$||(P - P_C)\hat{p}||_2^2 = r^2. \tag{4.2}$$

The vessel direction is given in spherical coordinates, with θ_a being the azimuth angle, or rotation from the x-axis in the xz plane, and θ_z being the zenith or polar angle from the z-axis. Setting the beam direction be to the $+z$ axis means that the zenith angle is equal to the Doppler angle θ_D. The vessel normal can then be written as:

$$\hat{n} = [n_x, n_y, n_z] = [\sin\theta_D \cos\theta_a, \sin\theta_D \sin\theta_a, \cos\theta_D]. \tag{4.3}$$

A two dimensional OCT B-scan produces a cross sectional image in the xz plane. The intersection of the vessel with this plane can be found as follows. Let $P = (x, 0, z)$ denote any point in the xz plane. By convention, the vector describing the vessel core runs through the origin so that $P_0 = (0, 0, 0)$. Plugging these values and the normal vector $\hat{n}$ into Equation 4.1 and the result into Equation 4.2 gives the equation for the intersection of the cylinder $\mathrm{Cyl}(x, z)$ with the xz plane as

$$\begin{aligned} ||(P - P_C)\hat{p}||_2^2 &= r^2 = Ax^2 + Bxz + Cz^2, \\ A &= (1 - n_x^2)^2 + n_x^2 n_y^2 + n_x^2 n_z^2, \\ B &= 2[n_x n_y^2 n_x - (1 - n_x^2) n_x n_z - (1 - n_z^2) n_x n_z], \\ C &= (1 - n_z^2)^2 + n_x^2 n_z^2 + n_y^2 n_z^2. \end{aligned} \tag{4.4}$$

Note that this equation describes an ellipse in the xz plane, as shown in Figure 4.3.

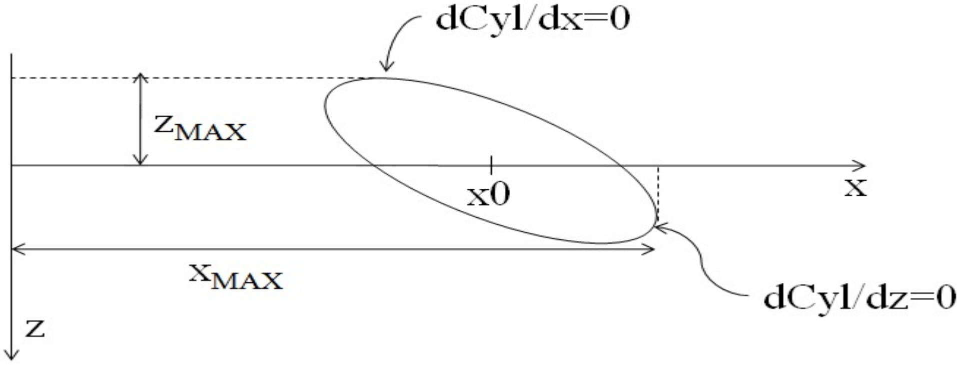


Figure 4.3: Intersection of cylindrical vessel with scan plane. This intersection defines the vessel boundary that can be used in subsequent calculations of flow.

As shown in Figure 4.3, the maximum extent z_{MAX} (in the z direction) of the vessel in the xz plane can be found by setting the partial derivative of the cylinder $\mathrm{Cyl}(x, z)$ with respect to x equal to zero. Similarly, the maximum extent x_{MAX} in the x direction can be found by setting the partial derivative of the cylinder Cyl(x, z) with respect to z equal to zero. Performing this both for z and for x gives:

$$\begin{aligned} z_{MAX} &= \frac{r}{\sqrt{C - B/4A^2}}, \\ x_{MAX} &= \frac{r}{\sqrt{A - B/4C^2}}. \end{aligned} \tag{4.5}$$

4.2.3 Simulation Procedure

Our novel retinal vessel simulation generates a synthetic OCT B-scan of a blood vessel immersed in background tissue. However, a complete model of red blood cell flow through a blood vessel immersed in retinal tissue would be very difficult to implement in practice. We thus use a number of simplifying techniques to make the problem tractable, including the cylindrical model of a blood vessel previously discussed. Because the geometry of moving red blood cells is fairly complex, we model them with discrete point scatterers. Similarly, the geometry of static retinal tissue that consists of different types of cells with different geometries is modeled with discrete point scatterers. To initialize the simulation, we generate random positions according to a uniform distribution for a fixed number of particles within a rectangular scan volume. The number of generated particles is determined by a simulation parameter and is kept fixed for computational simplicity. Each particle is labeled as a stationary or flow particle depending on whether it lies inside the cylindrical blood vessel. Then, the beam position is initialized and the iterative part of the simulation proceeds until an entire complex OCT B-scan is computed.

The iterative part of the simulation consists of a number of steps. First, the OCT signal is computed using a discrete version of Equation 2.13 for all of the particles in the current coherence volume. Note the center of the coherence volume coincides with the center of the input beam. Then, the beam is moved axially and the OCT signal is recomputed for the new coherence volume. These two steps are repeated until an entire axial scan, or A-line, is acquired, and then the axial beam position is reset to its initial value. After each axial scan, the flow particles are moved according to their respective preset velocities. By convention, the velocity of a particle is computed using a parabolic flow profile and the speed of a particular particle is determined by its distance to the vessel core. Although not entirely realistic, parabolic flow is typically assumed when using Doppler OCT techniques. Some flow particles leave the rectangular scan volume after they are moved. If this were not compensated by bringing new flow particles into the scan region, then eventually there would be no flow particles remaining. This necessitates the need for flow particle replacement. In practice, red blood cells enter the scan volume at random times and in random positions. A more realistic model would account for this by sampling from a probability distribution that describes the entry of particles into the scan region. However, this would lead to a variable number of flow particles in the scan region and make the particles computationally difficult to track. Because we fix the number of particles in the simulation, a flow particle is replaced upon exit from the scan region with another flow particle. Furthermore, because flow particles near the center of the vessel exit the scan region faster than particles near the vessel walls, allowing the new flow particle to have any speed would result in a clustering of flow particles near the vessel walls. In order to maintain the uniform spatial distribution of the flow particles, the speed of the replaced flow particle is preserved in the new flow particle.

After all of the flow particles are replaced, the transverse position of the beam is updated. The iterative part of the simulation is then repeated until the OCT signal at all axial and transverse beam positions is computed. When the entire B-scan simulation is complete, the complex OCT signal data is saved to a file. Figure 4.4 depicts a flowchart for the simulation procedure. The individual modules are explained in detail in the following subsections.

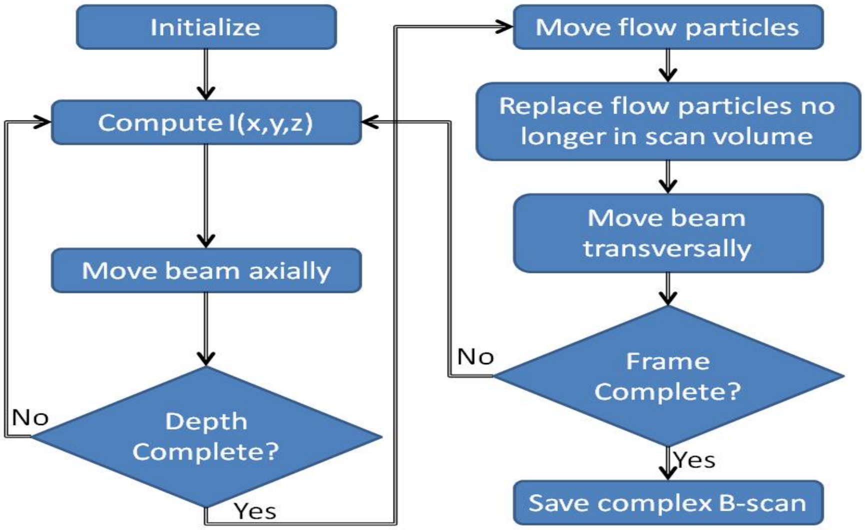


Figure 4.4: Retinal vessel simulation flowchart.

4.2.3.1 Initialize

The scan region is defined by the scan length Δx and scan depth Δz. The transverse step size between adjacent A-lines is δx and the axial step size between adjacent depth pixels is δz. The total number of transverse and axial pixels are then calculated using:

$$\begin{aligned} N_x &= \lceil \Delta x / \delta x \rceil + 1, \\ N_z &= \lceil \Delta z / \delta z \rceil + 1, \end{aligned} \tag{4.6}$$

where the $\lceil a \rceil$ denotes the ceiling function defined by the smallest integer greater than or equal to a.

The full-width at half-maximum (FWHM) transverse resolution ω_0 and the axial resolution ω_z are simulation parameters. Each coherence volume is a 1 x 3 vector $V^C = (V_x^C, V_y^C, V_z^C)$ defined by a width in the x, y and z dimensions. This width is determined another simulation parameter $cVolPw$ that specifies the $1/e^{cVolPw}$ width of the assumed

Gaussian beam and Gaussian source spectrum. Then, the coherence volume boundary in the transverse dimension is given by:

$$V_x^C = \omega_0 \sqrt{\frac{cVolPw}{\ln 2}}, \tag{4.7}$$

and similarly for the axial converssion. Let

$$\begin{array}{rcl} X_{max} & = & \Delta x/2 + V_x^C, \\ Y_{max} & = & V_y^C, \\ Z_{max} & = & \Delta z/2 - V_z^C, \end{array} \tag{4.8}$$

denote the maximum extent of the simulation with respect to the origin. Then, the particles are randomly placed within the 3 dimensional scan volume V^S defined by $[-X_{max}, X_{max}]$ x $[-Y_{max}, Y_{max}]$ x $[-Z_{max}, Z_{max}]$.

The density of the particles ρ_p, defined by the number of particles per cubic micron, is a simulation parameter. The total number of particles generated is given by $N_p = \lceil \rho_p/(V_x^S \cdot V_y^S V_z^S) \rceil$. The particles are generated with random positions inside V^S. Each particle is also assigned a random phase between $-\pi$ and π.

The parameters that describe the vessel are its diameter D_{VES}, Doppler angle θ_D and azimuthal angle θ_a. The particles that lie inside the vessel are deemed to be flow particles, and the particles that lie outside of the vessel are stationary particles. Whether a particle lies inside or outside of the vessel is determined by Equation 4.1 and Equation 4.2. Flow particles are assigned reflectivities r_{FLOW} and stationary particles are assigned reflectivities r_{STAT}. By convention, $r_{FLOW} + r_{STAT} = 1$.

After the all of the particles are set, the initialization is completed by setting the beam position to $P_{beam} = (-\Delta x/2, 0, -\Delta z/2)$. The initialized simulation is illustrated in Figure 4.5.

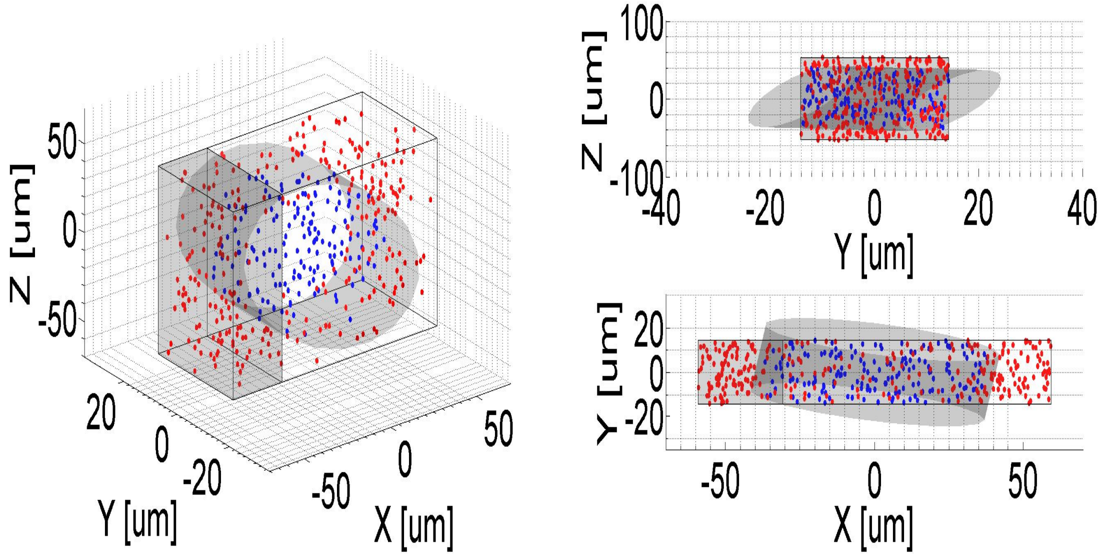


Figure 4.5: Initialized simulation. Red dots denote stationary particles. Blue dots denote flow particles that lie inside the vessel. The vessel is shown as a gray cylinder. The transparent box defines the region V^S. The gray box denotes the region of the first A-line.

4.2.3.2 OCT Signal Calculation

The noiseless complex OCT signal emanating from a coherence volume V_0^C centered at a point $(x_0, 0, z_0)$ within the sample can be calculated using:

$$A = \sum_{p \in V_0^C} r_p \exp(-i2k_0(z_p - z_0))G_p, \tag{4.9}$$

where p is the pth particle index, r_p is its reflectivity and (x_p, y_p, z_p) is its position in the sample, and G_p is a Gaussian function defined by:

$$G_p = \exp\left(-4\ln 2\left(\frac{(x_p - x_0)^2 + (y_p)^2}{\omega_0^2} + \frac{(z_p - z_0)^2}{\omega_z^2}\right)\right). \tag{4.10}$$

An A-line is computed by moving z_0 from $-\Delta z/2$ to $\Delta z/2$ in increments of δz and computing Equation 4.9 at each step. After each A-line is computed, the axial beam position is reset to $-\Delta z/2$ and the transverse beam position increments by δx.

4.2.3.3 Move Flow Particles

After each A-line, the flow particles are moved. The velocities inside the vessel are determined by a parabolic flow profile in a vessel cross section. First, the distance d_p from each flow particle to the vessel core is computed using Equation 4.2 with the distance d_p replacing the vessel diameter D_{VES}. Then, the velocity of the particle is computed using:

$$V(d_p) = V_{MAX}\left(1 - \frac{d_p^2}{(D_{VES}/2)^2}\right). \tag{4.11}$$

The maximum velocity V_{MAX} in the vessel is determined using the vessel diameter and the preset flow rate R according to:

$$V_{MAX} = \frac{8R}{\pi D_{VES}^2}. \tag{4.12}$$

The flow rate R is a simulation parameter. Note that Equation 4.12 is derived by noting that the flow rate through a vessel is equal to the average velocity in the vessel cross section divided by the cross sectional area and that the maximum velocity is equal to two times the average velocity when a parabolic flow profile is assumed. The distance that each particle moves can be calculated by multiplying Equation 4.11 by the time between A-lines τ. τ is another simulation parameter. This distance is added to the current particle position to complete flow particle movement.

4.2.3.4 Replace Flow Particles

As the beam moves along the transverse dimension, the flow particle positions are sequentially updated. Because the scan volume has finite dimension bounded by a box V^S, some particles may move out of it. Since the number of particles in the simulation is fixed, particles that move out of the scan region must be replaced by new particles. As discussed in the beginning of Section 4.2.3, a new flow particle replaces an exiting with the same speed.

Also, the angle of rotation around the vessel core of the new flow particle position is set to the angle of rotation of the particle that it is replacing. Physically, this means that the new particle position is forced to lie on a straight line in the opposite direction of the vessel $-\hat{n}_{FLOW}$ from the position of the particle that is being replaced.

Marching along a line in the opposite direction of the vessel will be referred to as backtracking. The vessel direction will now be subscripted with $FLOW$ in order to avoid confusion.

A simple procedure can be followed in order to implement the replacement protocol. Suppose that the particles are randomly distributed in the region $V^S = [-X_{max}, X_{max}] \times [-Y_{max}, Y_{max}] \times [-Z_{max}, Z_{max}]$. This region describes a 3-dimensional cube that has 6 faces. Each face lies in a plane that is described by a point in the plane and a unit vector that is normal to the plane. For the following discussion, $\vec{p}$ will be used to denote both a point in three dimensional space as well as a vector from the origin to that point. The definitions of each face of a plane are given in Equation 4.13.

$$\begin{array}{rrcccc}
F_1\colon & \vec{p}_{F_1} & = & [-X_{max}, 0, 0]; & \hat{n}_1 & = & [1, 0, 0] \\
F_2\colon & \vec{p}_{F_2} & = & [X_{max}, 0, 0]; & \hat{n}_2 & = & [-1, 0, 0] \\
F_3\colon & \vec{p}_{F_3} & = & [0, -Y_{max}, 0]; & \hat{n}_3 & = & [0, 1, 0] \\
F_4\colon & \vec{p}_{F_4} & = & [0, Y_{max}, 0]; & \hat{n}_4 & = & [0, -1, 0] \\
F_5\colon & \vec{p}_{F_5} & = & [0, 0, -Z_{max}]; & \hat{n}_5 & = & [0, 0, 1] \\
F_6\colon & \vec{p}_{F_6} & = & [0, 0, Z_{max}]; & \hat{n}_6 & = & [0, 0, -1]
\end{array} \tag{4.13}$$

If a flow particle has moved out of the region, then it must have crossed one of the faces F_i. A backtracked vector $\vec{l}_{BT}$ from the position of the flow particle $\vec{p}_{FLOW}$ along the opposite direction of the vessel $\hat{n}_{FLOW}$ will thus intersect F_i. The position of this intersection will be called the particle exit position $\vec{p}_{EXIT}$, because it is the position through which the particle exited the scan region. The backtracked flow vector can be described by:

$$\vec{l}_{BT} = \alpha \hat{n}_{FLOW} + \vec{p}_{FLOW}, \tag{4.14}$$

where $\alpha < 0$ for backtracking. $\vec{l}_{BT}$ will also intersect other faces of the box at different positions. The intersection point that is furthest from the particle position is called the particle entry position $\vec{p}_{ENTRY}$ because it is the position through which the particle would have entered the scan region. The conventions are shown in two dimensions in Figure 4.6. The distance between $\vec{p}_{FLOW}$ and $\vec{p}_{EXIT}$ will be denoted by $|\alpha_{EXIT}|$ and the distance between $\vec{p}_{FLOW}$ and $\vec{p}_{ENTRY}$ will be denoted by $|\alpha_{ENTRY}|$.

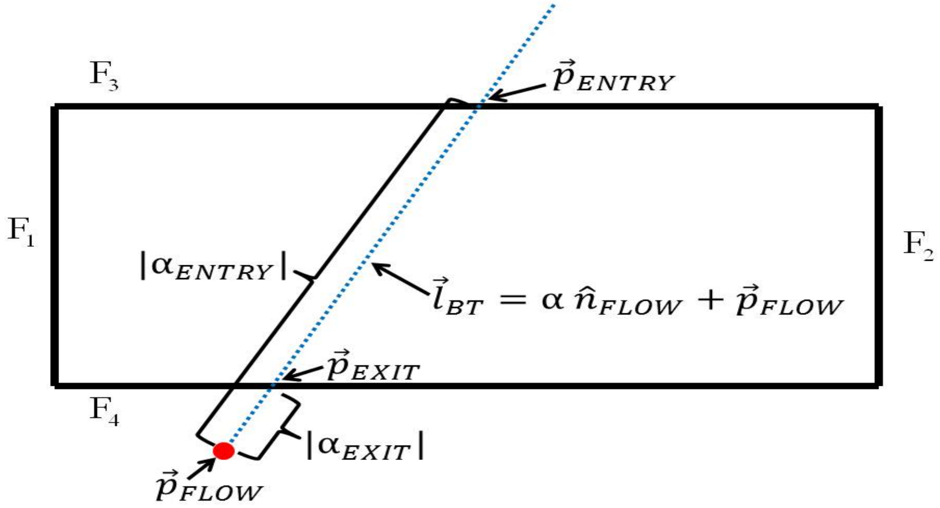


Figure 4.6: Notation conventions for flow particle replacement illustrated in 2 dimensions for simplicity. F_j denote the faces of the bounding box V^S of the scan volume. p_{FLOW} is the position of the particle that moved outside of the scan volume and is being replaced. The particle position is backtracked along the vector $\vec{l}_{BT}$ which is in the direction opposite to the flow vector $\hat{n}_{FLOW}$. $\vec{p}_{EXIT}$ denotes the position at which the particle crossed one of the faces to exit the scan volume and is a distance $|\alpha_{EXIT}|$ from p_{FLOW}. $\vec{p}_{ENTRY}$ denotes the position at which the particle would have entered the scan volume and is a distance $|\alpha_{EXIT}|$ from p_{FLOW}.

The particle entry and exit positions are calculated as follows. First, note that the face F_j lies in a plane that is defined by all of the points $\vec{p}$ for which $\langle \vec{p} - \vec{p}_{F_j}, \hat{n}_j \rangle = 0$, where the angle brackets denote the inner product. Then, the backtracked line intersects this plane only if $\langle \vec{l}_{BT} - \vec{p}_{F_j}, \hat{n}_j \rangle = \langle \alpha_j \hat{n}_{FLOW} + \vec{p}_{FLOW} - \vec{p}_{F_j}, \hat{n}_j \rangle = 0$. Distributing $\hat{n}_j$ and solving for α_j gives the distance from the point $\vec{p}_{FLOW}$ to the plane and is given by:

$$\alpha_j = \frac{\langle \vec{p}_{F_j} - \vec{p}_{FLOW}, \hat{n}_j \rangle}{\langle \hat{n}_{FLOW}, \hat{n}_j \rangle}. \tag{4.15}$$

Note that the application of Equation 4.15 requires that $\hat{n}_{FLOW}$ and $\hat{n}_j$ cannot be parallel. If the two vectors are parallel so that the flow direction is along one of the faces, the computation is skipped and the jth face is not considered for the entry and exit positions.

After the α_j's are computed for each j, Equation 4.14 is applied to compute the intersection points $\vec{i}_j$ of the line $\vec{l}_{BT}$ with each of the faces. If an intersection point lies on the original 3 dimensional cube boundaries, then it is a candidate for the entry and exit positions. Once all of the candidates have been identified, the magnitudes of the α's are compared; the closest candidate intersection point is defined as the exit position and the farthest candidate intersection point is defined as the entry position.

One additional step is needed to complete the calculation of the position of the new particle. The entry point $\vec{p}_{ENTRY}$ is one possible choice for the new position. However, in the case of very fast flow, particles would tend to cluster at the entry face if replaced at $\vec{p}_{ENTRY}$. In order to prevent this inhomogeneity, a random offset is added to the entry position. First, the distance that a replaced particle moves in between scans is computed. This distance is the maximum sensible offset and is denoted by d_{MAX}. Then, a random number r is generated in the interval $[0, d_{MAX}]$ and used as the distance between $\vec{p}_{ENTRY}$ and the new particle position. Finally, the new particle position $\vec{p'}_{FLOW}$ is computed according to:

$$\vec{p'}_{FLOW} = r \cdot \hat{n}_{FLOW} + \vec{p}_{ENTRY}. \tag{4.16}$$

Flow particle backtracking is shown in Figure 4.7.

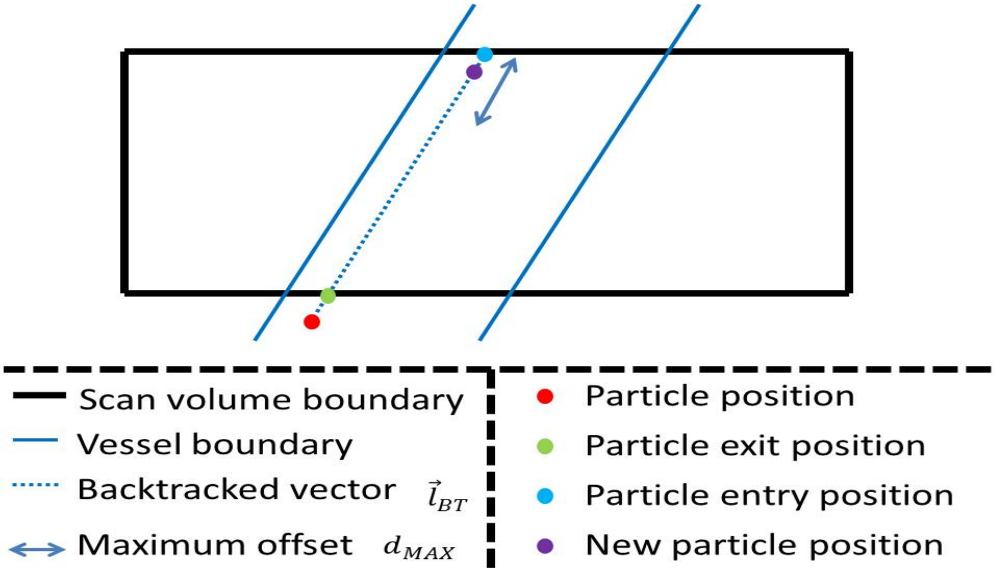


Figure 4.7: Flow particle backtracking and replacement. The offset from the entry position $\vec{p}_{ENTRY}$ is determined by a random number in between 0 and d_{MAX}.

4.2.4 Summary of Simulation Parameters

A number of parameters are used in the retinal vessel simulation. They are summarized in Table 4.1.

4.2.5 Summary

In this section, we proposed a novel two dimensional simulation of a retinal vessel immersed in background tissue. Our simulation is more realistic than others previously reported because it is two dimensional and because we model both stationary and flow scatterers. After

Parameter	Description
$\Delta x, \Delta z$	Scan length/depth in μm
$\delta x, \delta z$	Axial/transverse step size between scans in μm
ω_0, ω_z	$1/e^2$ beam width/coherence length in μm
$cVolPw$	$1/e^{cVolPw}$ is the coherence volume bounds
ρ_p	Density of particles in #particles/μm^3
D_{VES}	Vessel diameter in μm
θ_D	Doppler angle in degrees
θ_a	Azimuthal angle in degrees
R	Flow rate in microliters per minute
τ	A-line rate in μs/A-line
r_{STAT}	Stationary particle reflectivity ($r_{STAT} + r_{FLOW} = 1$)

Table 4.1: Summary of parameters for retinal vessel simulation.

computing the simulated B-scans, the Doppler phase shifts can be calculated using Equation 2.22. Figure 4.8 shows a visual comparison between a real retinal vessel from an *in vivo* image and a simulated retinal vessel that provides qualitative validation of the synthetic OCT data. The OCT imaging system used to acquire the *in vivo* Doppler image was described in 3.4.2. The parameters of the simulation were chosen to mimic those of the clinical imaging system.

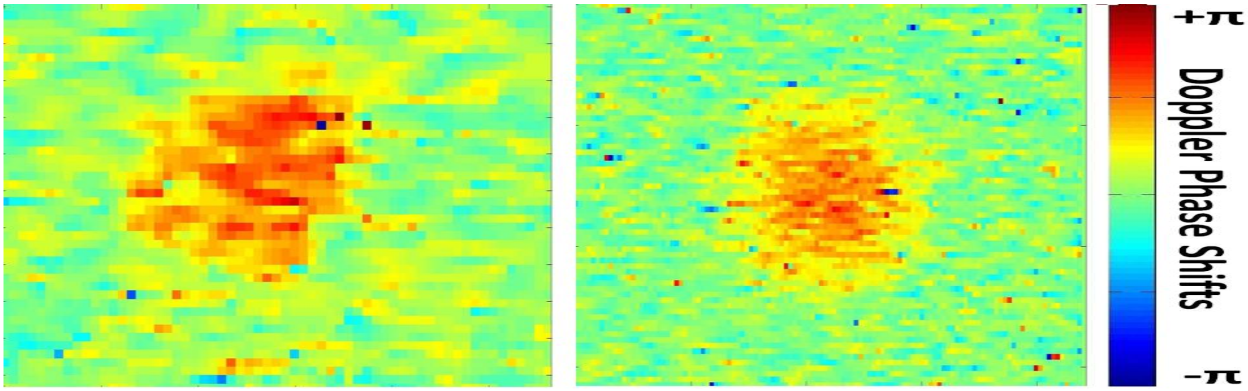


Figure 4.8: Visual comparison between *in vivo* retinal vessel and simulated vessel computing using vector average phase-resolved Doppler OCT.

The purpose for developing this simulation was to provide ground truth blood flow velocity data in order to assess the accuracy of Doppler OCT algorithms. Two case studies are presented in the following sections. For each case study, we first validate that the simulation produces realistic Doppler OCT results using either *in vivo* retinal data or *in vitro* experimental data. We then use the simulation to extrapolate the results in a controlled setting in order to assess the accuracy of Doppler OCT algorithms.

4.3 Case Study I: Phase-Resolved vs Moving-Scatterer-Sensitive Doppler OCT Algorithm Comparison

4.3.1 Introduction

The goal of the first case study that illustrates the utility of our retinal vessel simulation introduced in Section 4.2 is to examine the effect of vessel diameter on the relative blood flow estimates in two Doppler OCT algorithms. Doppler OCT was developed to measure axial phase/frequency shifts in moving red blood cells. The phase-resolved (PR) Doppler OCT algorithm is the most commonly used Doppler technique for computing blood flow velocities. When the incident beam is focused near the edge of a blood vessel, the coherence or probe volume within the sample contains both moving blood cell scatterers as well as scatterers from the static tissue background. For large blood vessels, the number of scans that contain both a stationary tissue component and a flow component is far less than the number of scans that contain only a flow component. On the other hand, for blood vessels that are small relative to the size of the coherence volume, scans that contain both a stationary component and a flow component dominate. An illustration of the size of the a blood vessel relative to the size of the coherence volume is shown in Figure 4.9.

Because *in vivo* measurements are acquired while transversally scanning the input beam, the static background tissue will have a nonzero Doppler bandwidth that increases with increased scanning speed [53]. An increase in the static tissue bandwidth will decrease the flow sensitivity of the PR algorithm because as the bandwidth of the stationary scatterers

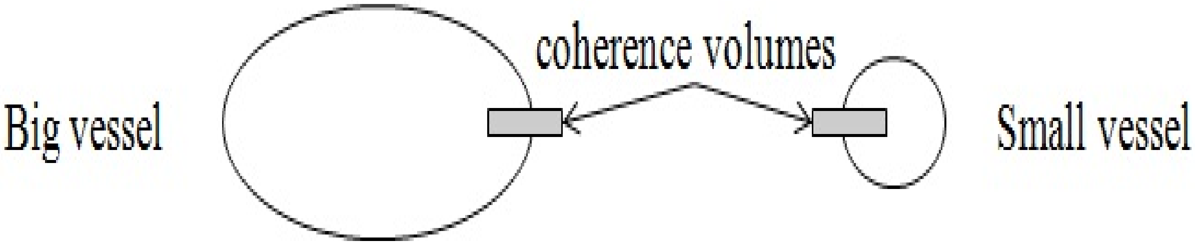


Figure 4.9: Vessel size compared with coherence volume size.

increases it will begin to overlap with the bandwidth of the flow scatterers. This is especially problematic when the axial Doppler frequency shifts are small. Because the flow rate in retinal blood vessels approximately scales with the cube of the vessel diameter, smaller blood vessels exhibit smaller Doppler frequency shifts [54]. Combined with the fact that scans of small blood vessels contain a significant stationary component, the reduced flow sensitivity of the PR algorithm is more pronounced in small blood vessels. The PR algorithm flow velocity estimate, which approximates the centroid of the combined distribution of stationary and flow scatterers, will be biased towards zero because of the nonzero static tissue bandwidth. Figure 4.10 illustrates the reduced sensitivity of Doppler OCT measurements in the case of nonzero static tissue bandwidth.

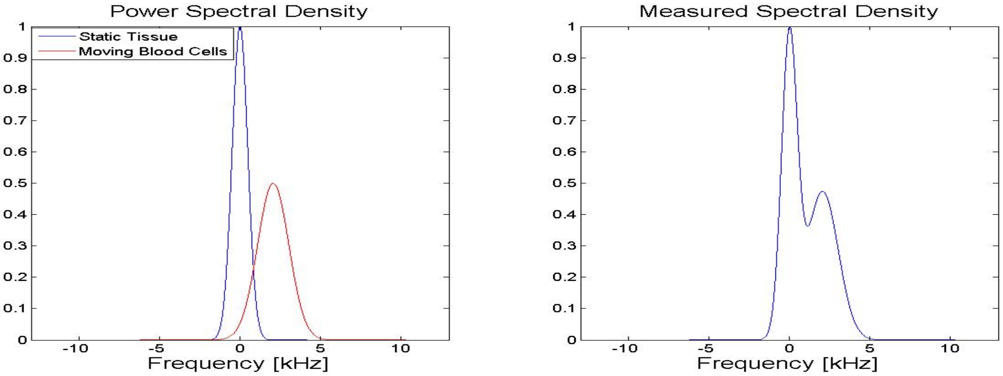


Figure 4.10: Illustration of the reduced sensitivity of the phase-resolved algorithm due to nonzero static tissue bandwidth. The PR algorithm flow estimate, which approximates the centroid of the combined distribution, will be biased towards zero because of the nonzero static tissue bandwidth.

Spatial frequency filtering techniques have been proposed to overcome the limitations of the PR algorithm for small Doppler frequency shifts. One example of a computationally efficient spatial frequency filtering technique is the moving-scatterer-sensitive (MSS) algorithm [55]. Although MSS has been shown to help visualize small blood vessels that are masked in the PR algorithm, a quantitative analysis of the accuracy of the MSS algorithm relative to PR is lacking. The purpose of this study is to utilize our novel retinal vessel simulation to compare the effectiveness of the PR and MSS algorithms. As mentioned in Section 4.2, the simulations developed in [52] and [18] cannot be used to quantitatively assess the effect of the overlap between the static tissue Doppler bandwidth and the flowing particle bandwidth. However, a quantitative comparison is necessary to determine which algorithm can measure blood flow velocities more accurately so that the better one can be used in clinical studies. Thus, our novel retinal vessel simulation helps determine which algorithm produces more accurate blood flow velocities in practice. We next describe the two algorithms that will be compared in this study.

4.3.2 Methods

The vector averaging phase-resolved (PR) algorithm [18] computes Doppler frequency shifts according to:

$$\Delta\phi(m,n) = \arctan\left(\frac{\Im(\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} A(m,n)A^*(m,n+1))}{\Re(\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} A(m,n)A^*(m,n+1))}\right), \tag{4.17}$$

as previously stated in Equation 2.22.

The moving-scatterer-sensitive (MSS) algorithm [55] was designed to overcome the reduced sensitivity of the PR algorithm in the presence of stationary scatterers. The MSS algorithm first subtracts adjacent A-scans:

$$D(m,n) = A(m,n+1) - A(m,n). \tag{4.18}$$

MSS then applies Equation 4.17 with A replaced by D. To demonstrate how this method helps to separate remove the stationary signal component, we examine the frequency response of 4.18. Specifically, after taking the Fourier transform over n we can write:

$$D(m,f) = A(m,f)e^{i2\pi fT} - A(m,f) = 2A(m,f)e^{i\pi fT} sin(\pi fT), \tag{4.19}$$

which gives a transfer function $|H(m,f)|^2 = 4sin^2(\pi fT)$. This transfer function is that of a high-pass filter. When the Doppler frequencies of the background tissue are within the stop band of this filter, its influence on estimating the Doppler frequency shift of moving scatterers is greatly reduced.

In this study we compute the relative blood flow estimates of the PR and MSS algorithms. As shown in Equation 2.24, the absolute volumetric flow rate is computed by integrating the estimated Doppler total velocity estimates over the vessel cross sectional area. Computation of the total flow velocity requires the measurement of the Doppler angle, or the angle between the incident beam and the blood vessel. However, comparing the ratio of two velocity estimation techniques does not require the Doppler angle because the angle in the numberator and denominator of the ratio cancel. Thus, in this study we do not compute the Doppler angle. Rather, the blood flow estimates are simply computed by summing the axial Doppler phase shifts inside a blood vessel.

The simulation was described in Chapter 4.2. The simulation beam width and coherence length were set to 22 μm and 5 μm, respectively. The axial adjacent scans was set to 2.3 μm. The transverse step size between adjacent A-lines was varied as described in Section 4.3.3. The A-line rate of the simulation was set to 37 kHz. The particle density was set to 0.15 particles/μm^3. The vessel diameters were varied from 15 to 180 μm and the flow rates varied from 0.05 to 15 μL/min. The flow rate for each vessel diameter was set according the the power curve fit of flow versus diameter as given in [54]. The scan length and scan width scaled with the size of each vessel diameter. After the complex OCT B-scans are simulated, the PR and MSS algorithms were applied to compute the Doppler phase shifts. The blood vessels were identified either manually or automatically, as detailed in Section 4.3.3, and the ratio of the summation of the PR Doppler shifts to the summation of the MSS Doppler shifts inside each blood vessel was computed.

The simulation results were validated using *in vivo* retinal scans obtained with the RT-vue spectral domain OCT system described in 3.4.2. All of the simulation parameters were chosen to mimic this OCT imaging system. For the *in vivo* Doppler images, double circular scans were taken both in the peripapillary and macular regions. There were 4000 axial scans in each B-scan. For the peripapillary scans the inner and outer circle diameters were 3.4 and 3.75 millimeters, resulting in transverse step sizes of 2.7 and 2.9 microns, respectively. For the macular scans the inner and outer diameters were 1.9 and 2.2 millimeters, resulting in transverse step sizes of 1.5 and 1.7 microns, respectively. Each position was scanned 6 times resulting in 12 frames for each scan. Both the PR and MSS algorithms are applied to all frames. Each blood vessel was manually segmented as described in Section 4.3.3, and the ratio of the summation of the PR Doppler shifts to the summation of the MSS Doppler shifts inside each blood vessel was computed.

4.3.3 Results

4.3.3.1 Retinal and simulated PR-to-MSS flow ratio

The simulation is first validated using *in vivo* measurements. The PR-to-MSS flow ratio as a function of vessel diameter as measured by the PR algorithm was calculated after manually segmenting each blood vessel with a bounding rectangle. For this particular study the transverse resolution was set to 22 microns for the simulation in order to match the RTVue specifications. The results of this experiment are shown in Figure 4.11.

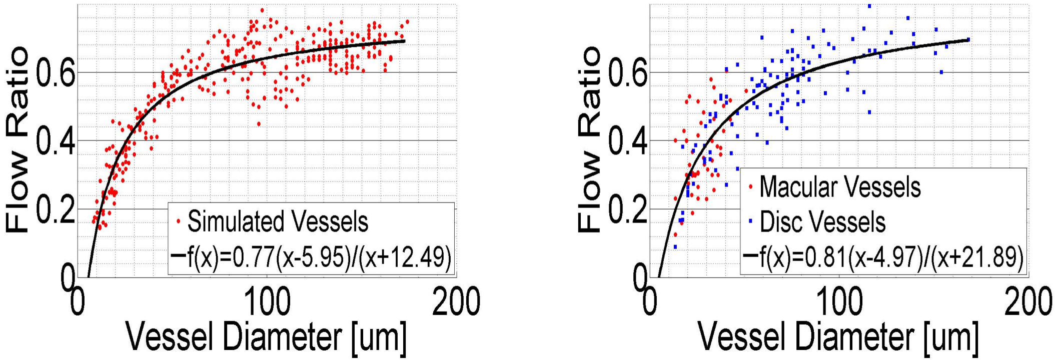


Figure 4.11: Left Panel: Simulated flow ratio. Right Panel: Retinal PR-to-MSS flow ratio.

The following curve was used for analysis of Figure 4.11:

$$f(x) = \frac{a(x-b)}{x+c}. \tag{4.20}$$

This fit is used because of its simple interpretation. First, the parameter a represents the asymptotic value or the ratio as the diameter gets very large. When this parameter is less than one it means that even for very large diameters there is a discrepancy between the PR and MSS flow estimates. Second, the parameter b represents the diameter at which the ratio goes to zero. This value is not physically relevant as the Doppler shifts can only be zero in the absence of noise. In practice this ratio never falls to zero. Finally, the parameters b and c are related to the value at which the ratio decreases significantly from a. Specifically, when $x = c + 2b$, the ratio falls to $a/2$.

The results of the retinal measurement and simulated measurement fits are in close agreement as shown in Figure 4.11. The significant difference lies in the parameter c. For the retinal flow ratio, the vessel diameter at which the ratio falls to $a/2$ is 31.8 microns. For the simulated flow ratio this value is 24.4 microns. We propose that the reason for the large discrepancy in the parameter c is that although the ideal transverse resolution of the RTVue system is 22 microns, the real transverse resolution is likely higher due to aberration effects. Thus the value at which the flow ratio is significantly reduced is shifted towards a larger vessel diameter when compared to the simulation, which uses a transverse resolution of exactly 22 microns.

In order to test this hypothesis a second simulation was implemented with a variable beam width in the range between 22 and 40 microns. In this experiment, the flow ratio is plotted as a function of the ratio of the vessel diameter to the beam width. In this case, since many more simulations were analyzed, the vessel boundary was detected automatically using a parabolic fit as described next. In order to compute the vessel boundary the known vessel center is used. Parabolas are fit to the centerline Doppler shifts in both the axial and transverse directions using a linear least-squares fit with the constraint that the shifts must be zero both on and outside of the boundary. The fitting function can be written as:

$$f(x) = \begin{cases} ax^2 + bx + c & \text{if } |x| < B \\ 0 & \text{otherwise,} \end{cases} \tag{4.21}$$

where B defines the vessel boundary. The least-squares fit seeks to minimize the sum squared error between $\hat{f}$ and the measured Doppler shifts, with B included as a parameter. This is done in both the axial and transverse directions to obtain B_{AXIAL} and $B_{TRANSVERSE}$ and the smaller of the two is taken to be the vessel diameter. Then, to compute the flow ratios Doppler shifts in an ellipse with major and minor axes defined by B_{AXIAL} and $B_{TRANSVERSE}$ are summed. Any case for which boundary detection is grossly in error is excluded from further analysis. It is important to note that fitting a parabola to the profile tends to produce a larger diameter estimate than does thresholding at the boundaries based on the noise level. Thus some noise pixels are included in the diameter and flow calculations. While this may have a significant effect on the diameter calculation, it has little effect on the flow calculation since the noisy Doppler shifts tend to sum to zero.

After implementing the boundary finding technique, the PR to MSS flow ratio is computed as a function of the beam diameter (as measured by PR) to the beam width. The results are shown in Figure 4.12. From the figure we determine that the PR diameter-to-beam width ratio for which the flow ratio decreases to one-half the maximum value is 0.93. This value can be used to obtain an estimate for the true beam spot size on the retina. Specifically, we can divide the measured retinal vessel diameters by possible beam spot sizes, find the new curve fit and compute the vessel diameter-to-beam width ratio for which the flow ratio decreases to one-half of its maximum value. The beam spot size that produces the result that is closest to 0.96 can be used as an estimate for the true spot size on the retina. Using 0.96 and the retinal flow ratio data calculated above we estimate that the real beam spot size on the retina is 33.1 microns. It should be noted that using this method we estimate that the beam spot size in the simulation is really 25.4 microns, not the 22 microns to which this value is set. This discrepancy is likely due to the fact that for the original simulation, the vessels were manually detected whereas for the second simulation the automatic boundary detection scheme was used.

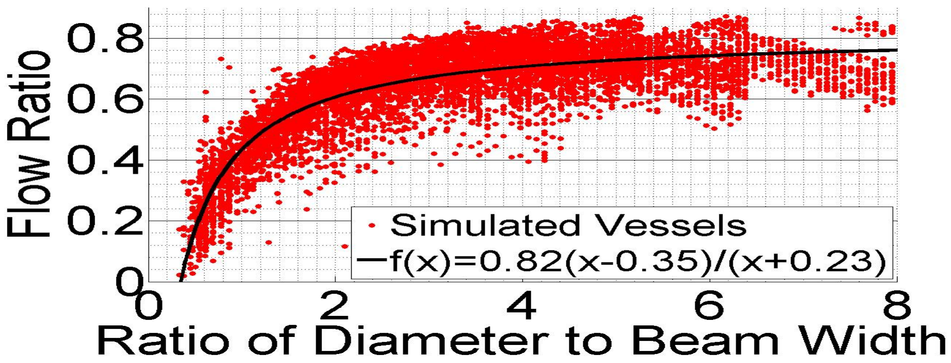


Figure 4.12: Simulated PR-to-MSS flow ratios as a function of diameter-to-beam width ratio.

4.3.3.2 Simulated Flow and Simulated Diameter Accuracy

The simulation is next extended and used to evaluate the accuracy, in terms of flow and vessel diameter estimates, of the PR and MSS algorithms. In these experiments the vessel diameters were computed automatically using the parabolic fitting technique described previously.

The simulation enables the evaluation of the flow accuracy of both the PR and MSS algorithms. Figure 4.13 demonstrates that both PR and MSS produce widely varying flow estimates when the vessel diameter is less than the beam width. For vessel diameters between one and three times the beam width the PR method produces a significant but systematic underestimate of the flow. For vessel diameters above three times the beam width the PR method produces stable and accurate flow estimates. On the other hand, the MSS method does not produce highly stable flow estimates for any vessel diameter as evidenced by its high variability for all diameters. Furthermore, even for very large vessels, the MSS method overestimates the flow. Both of these effects are caused by the high-pass nature of the MSS algorithm. Such high-pass filtering has been shown [53] to distort the power spectrum of the Doppler signal. This is especially evident when the Doppler frequency shifts are low, as is typically the case for small blood vessels.

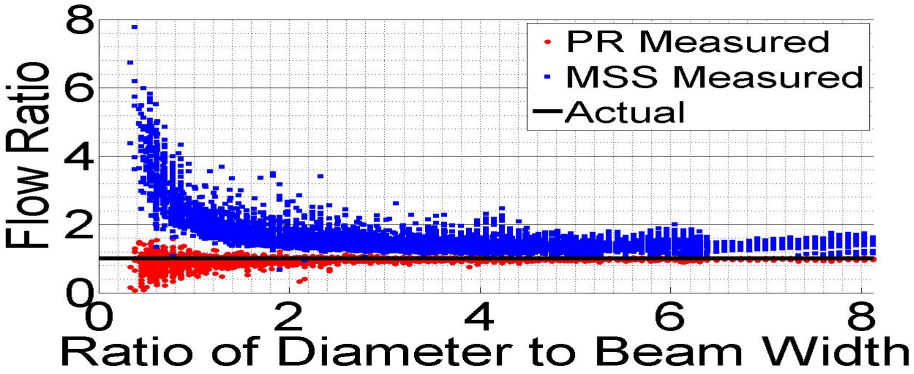


Figure 4.13: Simulated PR-to-actual and MSS-to-actual flow ratios as a function of diameter-to-beam width ratio.

The simulated vessel diameter measurements are plotted in Figure 4.14. These relationships are linear for both the PR and MSS methods, with a slight increase in the difference between the two methods as the actual diameter increases. Based on the linear regression equations the PR method slightly underestimates the vessel diameter whereas the MSS algorithm overestimates it. It is clear that the PR algorithm produces more precise diameter estimates than does the MSS algorithm. It should again be noted that these diameter measurements were made automatically using the parabolic fitting scheme described above. Fitting a parabola to the profile tends to produce a larger diameter estimate than does thresholding at the boundaries based on the noise level. Thus the intercepts should be slightly lower than reported here if thresholding at the boundary is used to determine the vessel diameter.

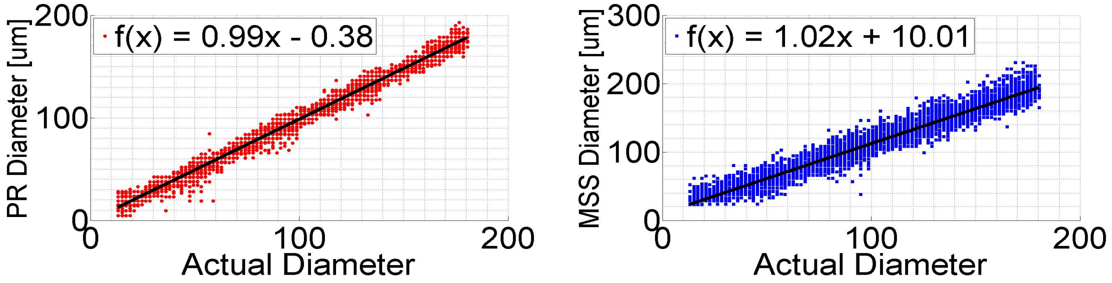


Figure 4.14: Simulated diameter measurement accuracy

4.3.4 Summary

In summary, we have shown that the PR-to-MSS flow ratio decreases significantly as the vessel diameter decreases. Similar relationships have been shown in both retinal and simulation measurements. For all vessel diameters examined the PR algorithm produces more precise flow and diameter estimates than MSS due to the high-pass nature of the MSS algorithm. The results indicate that although the MSS algorithm can be used to visualize small blood vessels that are masked by the PR estimates, it cannot be used reliably to measure the flow in these vessels. The PR algorithm, however, underestimates the flow for vessel diameters less than three times the beam width. But, the simulation predicts a systematic decrease for diameters between one and three times and beam width. Thus, it seems plausible that a correction factor can be used for the flow estimates in these cases. Further testing and analysis is required to determine whether these observations may help to correct for algorithm bias in *in vivo* retinal flow estimates.

4.4 Case Study II: Effect of Transverse Step Size on Flow Velocity Measurements in Phase-Resolved Doppler OCT

4.4.1 Introduction

The goal of the second case study that illustrates the utility of our retinal vessel simulation introduced in Section 4.2 is to study the effect of transverse step size on Doppler OCT measurements. In Section 4.3 we showed that the phase-resolved algorithm is more accurate than the moving-scatterer-sensitive algorithm in terms of estimating retinal blood flow. Thus, we limit this study to the phase-resolved Doppler OCT algorithm (PRDOCT).

The precision and accuracy of PRDOCT measurements relies on both a large overlap between adjacent A-scans [52] as well as high signal-to-noise (SNR) ratios [56]. The overlap between scans is related to the transverse step size. The authors in [52] derived a theoretical expression for the phase difference between adjacent A-lines as a function of transverse step size. Using this model as well as a one-dimensional simulation, they showed that increasing the transverse step size leads to a broadening of the Doppler spectrum and a resulting increase in the measured root-mean-squared phase noise. However, this study focused only on bulk or stationary scatterers and did not analyze the effect on moving particles such as red blood cells (RBCs). The authors in [56] showed that a decrease in the SNR also leads to a broadening of the Doppler spectrum. Because of the cyclic nature of phase measurements and subsequent phase wrapping, the authors showed that this spectrum broadening leads to a decrease in measured Doppler velocities, and that the effect was more pronounced at higher flow speeds [18]. The authors then proposed a modified complex vector averaging technique that mitigated this effect. However, this study did not analyze the effect of transverse step size on PRDOCT measurements.

The purpose of this study is to examine the dependence of average velocity measurements inside a vessel on the transverse step size in spectral domain PRDOCT using the vector averaging technique. The primary contributions of this study are (1) the derivation of a mathematical expression for the effect of transverse step size on PRDOCT and (2) the proposal of two techniques to correct for this effect. This is the first time that these correction techniques are proposed, and they can be used to obtain more accurate blood flow velocity estimates for *in vivo* clinical studies.

We first validate the simulation with *in vitro* blood flow phantom experiments using small transverse step sizes that are clinically relevant. We then use the simulation to extrapolate the results over a larger range of transverse step sizes and flow rates. We fit parameterized logistic functions to the curves in order to compare the effect at different flow rates. We then present lookup table and phase unwrapping techniques to correct for the effect of increased transverse step size on measured velocity data. We also show that increasing the beam width mitigates the effect without the need to correct the measured velocities.

4.4.2 Theory and Simulation

The simulation was described in Section 4.2. The constant parameters in this simulation include the scan length (140 μm), scan depth (140 μm), full width $1/e^2$ intensity beam width (20 μm), coherence length in tissue (6.5 μm), scatterer density (0.08 particles/μm^3), source wavelength (840 μm) and A-line rate (17 kHz). No additive noise is implemented, unless otherwise stated. Scatterers are randomly placed within the scan volume. An 80 μm diameter vessel with a parabolic flow profile is placed at the scan center, and all scatterers inside the vessel simulate flowing blood particles, while all scatterers outside of the vessel simulate particles in the surrounding tissue. The Doppler angle, defined as the angular deviation from perpendicularity between the incident beam and the blood vessel, is set to 100°. The transverse step size is a variable parameter.

The simulated complex OCT Bscan was used to compute Doppler phase shifts and corresponding axial velocities using Equation 2.22 and Equation 2.21. Because of the cyclic nature of the arctangent function, the maximum detectable axial velocity shift without phase unwrapping is $V_{max} = \lambda_0/4n\Delta t$. Throughout this work, the Doppler velocities are examined over the entire vessel area in the xz (transverse-axial) plane. The illustrated distributions and the computed Doppler shifts and velocities are aggregated using all of the pixels inside the vessel. Because the average velocity in a circular cross section of a parabolic flow profile is half that of the maximum velocity [57], the relative average velocity V_{rel} is defined in terms of the measured velocity V as:

$$V_{rel} = \frac{V}{V_{max}/2} = \frac{8n\Delta t}{\lambda_0} \cdot V. \tag{4.22}$$

V_{rel} ranges between zero and one and is proportional to the volumetric flow through a vessel [57]. The effect of transverse step size on V_{rel} is the primary focus of this case study.

The authors in [52] showed that increasing the transverse step size between adjacent A-lines leads to a broadening of the Doppler spectrum. Due to the cyclic nature of phase measurements and consequent phase wrapping, this spectrum broadening introduces a systematic decrease in measured average Doppler phase shifts and velocities in a vessel. To illustrate this effect, a simulation of an OCT B-scan was used to calculate the phase distribution inside a vessel. Figure 4.15 shows the Doppler broadening and the resultant phase wrapping with increasing transverse step size for two different settings for V_{rel}. The inclusion of the wrapped pixels in the averaging calculation leads to a systematic decrease in and subsequent underestimation of the average Doppler phase shift inside a vessel. This underestimation is more pronounced at high velocities because of the proximity to the phase wrapping limit. Specifically, for $V_{rel} = 0.36$, the average Doppler phase shift inside the vessel falls by a factor of 0.84 when the transverse step size is increased from 0.5 μm to 8.5 μm. On the other hand, for $V_{rel} = 0.85$, the average Doppler shift falls by a factor 0.65 over the same range.

4.4.3 Results and Discussion

4.4.3.1 Validation Using *In Vitro* Experiments

As a first experiment, the simulation is validated using an *in vitro* flow phantom experiment with a spectral domain OCT system. A detailed description of the experimental setup was given in Section 2.2.4. The parameters of the system, such as source center wavelength, beam width, coherence length and A-line rate, match those used in the simulation. A

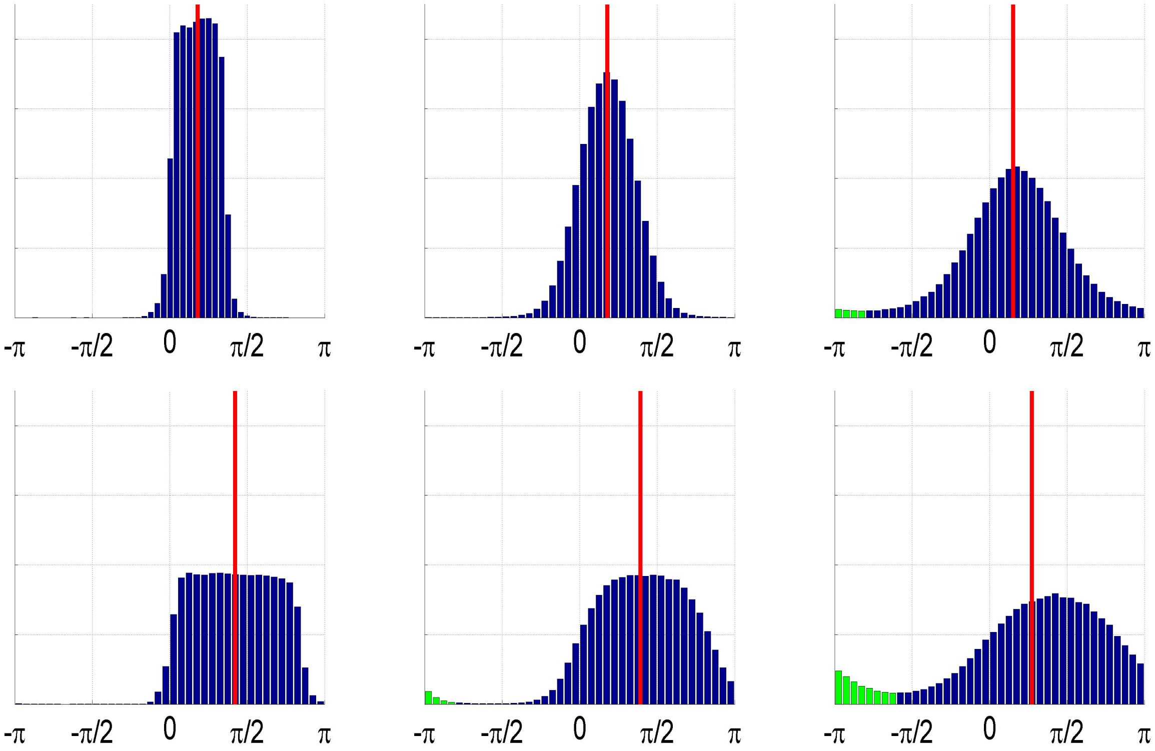


Figure 4.15: Simulated Doppler broadening, consequent phase wrapping and mean Doppler phase shift underestimation with increasing transverse step size for two different velocity settings. (a,b,c): V_{rel} set to 0.36. (d,e,f): V_{rel} set to 0.85. The transverse step sizes are 0.5 μm (a,d), 4.5 μm (b,e) and 8.5 μm (c,f), respectively. The distributions are computed using all of the pixels that are inside of the vessel. Note that the distribution of velocities is significantly wider for a higher velocity even for a small step size (a,d). This is because the range of velocities in a vessel with a higher average velocity is significantly larger than the range for a lower average velocity. The green bins indicate the phase wrapping. The vertical red lines indicate the average Doppler shifts in the vessel and are located at 0.57 radians, 0.56 radians and 0.48 radians, respectively, for $V_{rel} = 0.36$ and 1.32 radians, 1.23 radians and 0.86 radians, respectively, for $V_{rel} = 0.85$. The underestimation caused by increasing transverse steps size has a significantly larger impact for faster velocities.

syringe pump (Harvard Apparatus) was used to control the flow of human blood through a glass capillary tube (Wale Apparatus) with an inner diameter of 200 μm. A dual-plane scan protocol [12] was used to compute Doppler velocities in the tube. A visual comparison of the simulated and phantom Doppler velocities is show in Figure 4.16. For the simulated velocity calculation, the known vessel diameter was used to compute the average velocity inside the vessel. For the phantom experiments, the glass capillary was manually segmented with an ellipse, and the average velocity inside of the segmented capillary is reported.

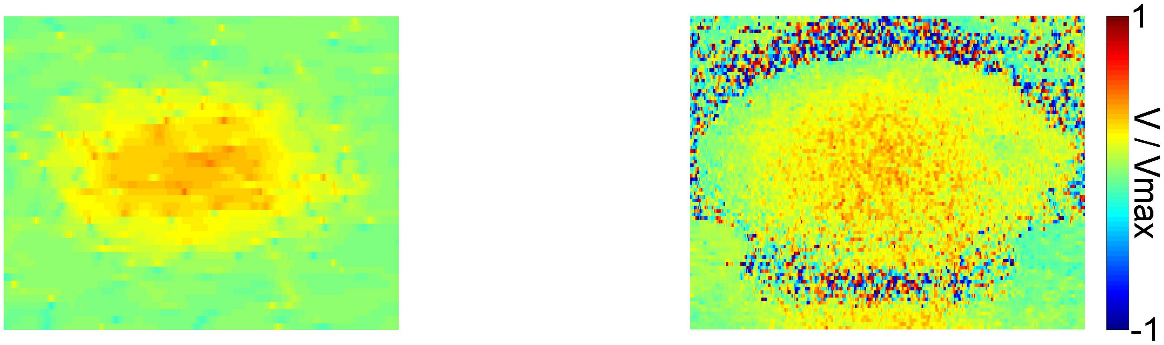


Figure 4.16: Sample Doppler velocity shifts computed using PRDOCT. Left panel illustrates results of simulated 80 μm diameter vessel in tissue. Right panel illustrates results of *in vitro* experiment using a 200 μm diameter tube. Vmax denotes the maximum measurable velocity without phase unwrapping and is equal to $\lambda_0/2n\Delta t$. The capillary tube was manually segmented with an ellipse.

In the first experiment, the transverse step size Δx was varied between 0.9 μm and 4.2 μm in increments of 0.3 μm, which covers the range typically used in in vitro Doppler experiments [18,19]. Two flow speeds were used to study how well the simulation mimics the phantom studies. The simulation flow speeds were chosen so that the average velocity in the simulated vessel was approximately equal to the average velocity inside the glass capillary. These two velocities are $V_{rel} = 0.49$ and $V_{rel} = 0.875$. Figure 4.17 qualitatively shows that the transverse step size affects the simulated and in vitro PRDOCT measurements in a similar manner. The Doppler phase shift broadening for both the in vitro phantom experiments and the simulation is illustrated in the top panel of Figure 4.17 (a,b) for the two flow speeds. This broadening is defined as the change in the standard deviation σ of the Doppler phase shifts inside the capillary/vessel relative to the standard deviation at 0.9 μm. Formally, the broadening is defined by:

$$\Delta\sigma = \sigma(\Delta x) - \sigma(\Delta x = 0.9). \tag{4.23}$$

The broadening for a higher velocity is greater than the broadening for a lower velocity because the standard deviation calculation is affected by phase wrapping as previously described. The phase wrapping subsequently leads to a decrease in the average velocity in the vessel, as illustrated in Figure 4.17 (c,d). Qualitatively, both the simulation and phantom experiments show almost no falloff up to transverse step sizes of 4.2 μm for V_{rel} = 0.49, but both show a falloff for V_{rel} = 0.875 over the same transverse step size range.

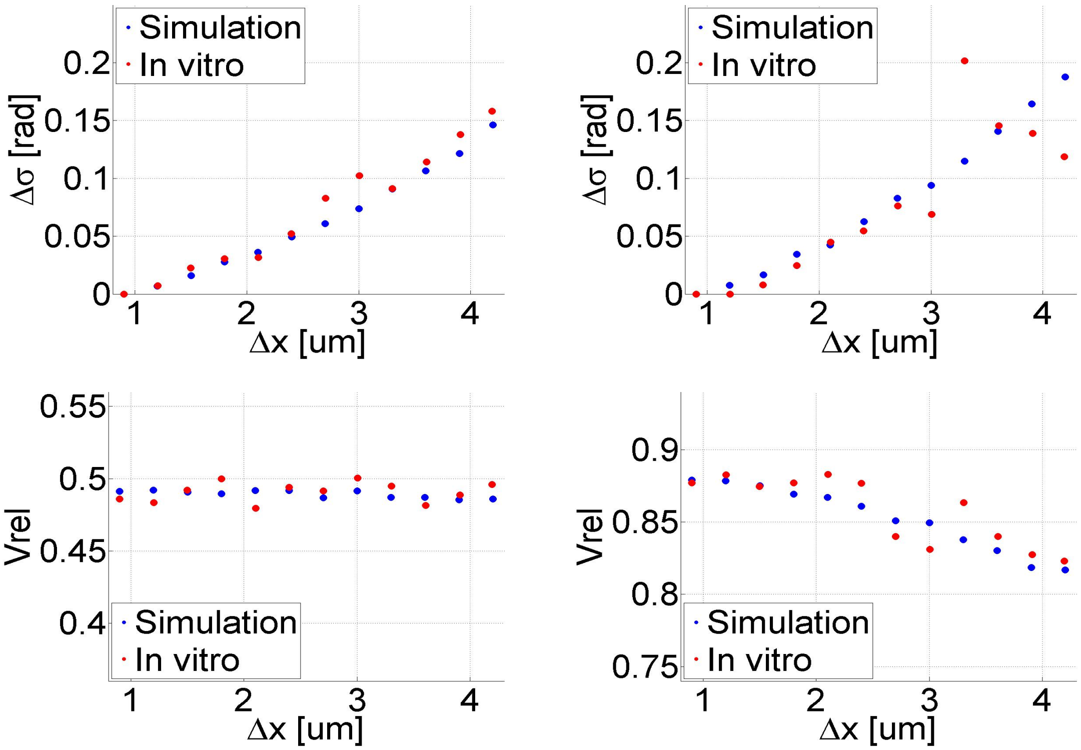


Figure 4.17: Comparison of broadening and subsequent velocity underestimation for simulation and in vitro experiments for two different flow rates. The broadening $\Delta\sigma$ (a,b) is defined in Equation 4.23 and is the relative broadening over the range of transverse step sizes. The velocity V_{rel} (c,d) is defined in Equation 4.22. (a,c): flow speed set so that $V_{rel} = 0.49$ at Δx = 0 μm. (b,d): flow speed set so that $V_{rel} = 0.875$ at $\Delta x = 0$ μm. (a,b) illustrate that the broadening effect is similar for the simulation and phantom experiments. (c,d) illustrate the falloff on average vessel velocity is similar for the simulation and phantom experiments.

For a quantitative comparison of the velocity falloff with transverse step size, the coefficient of variation of the root-mean-square-deviation between the simulated measurements

$V_{rel,sim}$ and the phantom measurements $V_{rel,phtm}$ is computed from Figure 4.17 (c,d). This quantity is defined as:

$$CVRMSD = \frac{\sqrt{\sum_{m=1}^{M}(V_{rel,sim} - V_{rel,phtm})^2}}{\frac{1}{M}\sum_{m=1}^{M} V_{rel,phtm}} \cdot 100, \tag{4.24}$$

and is a normalized measure of the difference between the two measurements. For the average velocity falloff curves for the two flow speeds studied, the CVRMSD values are 1.53% and 1.47%, thus validating the use of the simulation to study the effect of transverse step size on PRDOCT measurements.

4.4.3.2 Phase Resolved Doppler OCT Falloff with Increasing Transverse Step Size

As a second experiment, seven different flow rates ranging from $V_{rel} = 0.12$ to $V_{rel} = 0.84$ are simulated using transverse step sizes ranging from 0.5 μm to 24.5 μm in steps of 2.0 μm. It can be seen in Figure 4.18 that the measured velocities fall off in a sigmoidal fashion. Initially, a curve fit was performed for all velocities using the parametric logistic function defined by:

$$V_{rel} = V_{rel,0}\frac{1 + r^M}{1 + r^{M-\Delta x}}, \tag{4.25}$$

where $V_{rel,0}$ is the relative velocity at $\Delta x = 0$, r is the falloff rate and M is the step size at which the derivative of V_{rel} with respect to Δx reaches its maximum. More intuitively, M is the step size at which the maximum falloff in V_{rel} occurs. Note that $V_{rel,0}$ is the real average relative velocity inside the vessel and V_{rel} is the measured relative velocity as defined by Equation 4.22. It was found that the falloff rate was approximately constant for all of the curve fits and so this parameter was fixed at $r = 0.655$ and the curves were fit again. The fitting results are shown in the left panel Figure 4.18 in black.

The right panel in Figure 4.18 illustrates that the step size at which the falloff rate reaches its maximum value increases with increasing velocity. This indicates that the shapes of the falloff curves are approximately the same but they are shifted based on the value of M. The velocity underestimation in PRDOCT results from the broadening of the Doppler spectrum due to high transverse step sizes and the limited range of possible phase shifts spanning from -π to π. The broadening for higher velocities reaches the phase shift limit earlier than for lower velocities. However, once the broadening causes the Doppler spectrum to reach this limit, it affects the PRDOCT measurements similarly regardless of the flow velocity.

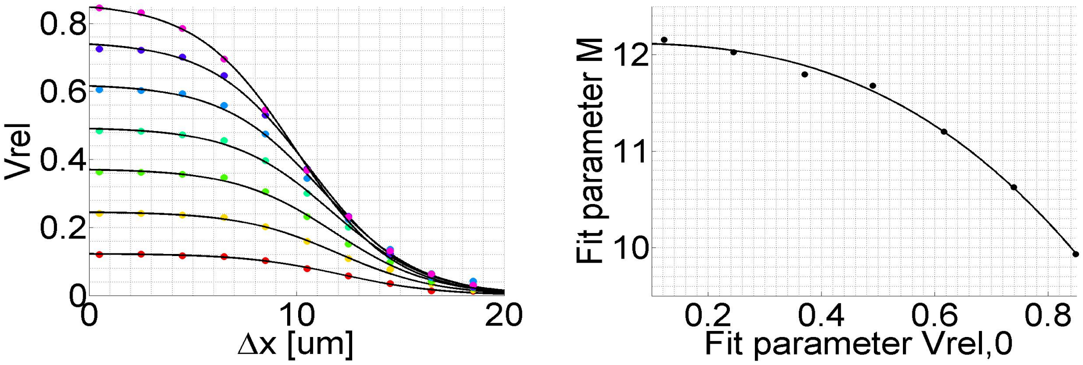


Figure 4.18: Simulation results for falloff of PRDOCT measurements for various flow speeds. Left panel: logistic curve fit for various flow speeds as given by Equation 4.25 with r = 0.655. Right panel: variation of the parameter M (Equation 4.26), which is the step size at which the derivative of V_{rel} with respect to Δx reaches its maximum, with the nominal velocity $V_{rel,0}$ at $\Delta x = 0$.

A power curve was used to fit the parameter M versus the real velocity $V_{rel,0}$ (Figure 4.18 (right)). The equation for this curve is:

$$M(V_{rel,0}) = -3.39 \cdot V_{rel,0}^{2.69} + 12.12. \tag{4.26}$$

The falloff factor for average velocity measurements related to increasing transverse step size can be defined by the right most term in Equation 4.25, or $(1+0.655^{M})/(1+0.655^{M-\Delta x})$. As illustrated in Figure 4.18, this falloff factor depends on the real velocity $V_{rel,0}$. It is possible to bound the falloff for various transverse step sizes based on Equation 4.25 and Equation 4.26. For example, since the maximum real velocity is $V_{rel,0} = 1.0$, the minimum value for M as given by Equation 4.26 is $M = 8.73$. For $\Delta x \leq 5$ μm, the falloff factor is bounded above by 0.85. Thus, even for velocities up to the phase wrapping limit, the velocity measured using vector averaging technique for PRDOCT falls off by at most 15%. Although this bounding procedure is helpful in understanding the extent to which velocity/flow is underestimated, it does not help to correct for it.

It is clear from Equation 4.25 that $V_{rel,0}$ is the desired velocity. In practice, velocity is measured using scans taken at some transverse step size larger than zero and will exhibit some falloff. It is desirable to devise a method that mitigates the falloff effect for increasing transverse step size that can be applied to measured velocity data to increase the accuracy of flow estimation. Two techniques are discussed next.

Velocity Correction Using a Lookup Table

One aim of this experiment was to derive a correction factor for a particular Δx that could be applied to a measured velocity V_{rel} to compute the real velocity $V_{rel,0}$. This correction factor is precisely the inverse of the falloff factor previously defined and is equal to $(1 + 0.655^{M-\Delta x})/(1 + 0.655^{M})$. However, due to the complicated nature of the relationship between M and $V_{rel,0}$ as given by Equation 4.26, it is not possible to find this solution in closed form. On the other hand, it is possible to compute $V_{rel,0}$ using a lookup table and a numerical solver. Specifically, Equation 4.25 and Equation 4.26 can be combined into:

$$V_{rel} = V_{rel,0} \frac{1 + 0.655^{-3.39 \cdot V_{rel,0}^{2.69} + 12.12}}{1 + 0.655^{-3.39 \cdot V_{rel,0}^{2.69} + 12.12 - \Delta x}}. \tag{4.27}$$

A lookup table can be generated by solving Equation 4.27 for various transverse step sizes and relative average velocity measurements using numerical techniques. An example of a one such lookup table is given in Table 4.2. The gradation of this lookup table is limited only by computer memory considerations.

$\Delta x \backslash V_{rel}$	0.1	0.3	0.5	0.7	0.9
2.6 μm	0.101	0.304	0.507	0.715	0.935
3.3 μm	0.102	0.306	0.511	0.723	0.957
4.0 μm	0.103	0.308	0.516	0.734	0.995

Table 4.2: Example lookup table for computing velocity $V_{rel,0}$ from measured average relative velocity V_{rel} and transverse step size Δx.

Although a lookup table could be used to compute $V_{rel,0}$, its application to practical Doppler OCT measurements is limited by the scope of this work. The focus of this work was to study the effect of transverse step size on PRDOCT average velocity measurements. Increasing the transverse step size leads to a broadening of the Doppler spectrum which decreases average velocity measurements because of phase wrapping. However, increasing the transverse step size is not the only factor that can lead to Doppler broadening. As the authors in [18] point out, decreasing the signal-to-noise ratio (SNR) also leads to Doppler broadening and subsequent velocity underestimation. In order to examine the effect of SNR on the falloff, a complex additive Gaussian noise term [42] is added to simulation measurements in Equation 4.9 at each pixel. The SNR is defined by

$$\mathrm{SNR} = 20 \cdot \log_{10}\left(\frac{\mathrm{mean}(|A|) - \mathrm{mean}(|N|)}{\mathrm{std}(|N|)}\right), \tag{4.28}$$

where A is the amplitude computed in Equation 4.9, N is the complex additive Gaussian noise term and the mean and standard deviation are computed using the entire simulation region that is 140 μm by 140 μm. The SNR is measured in decibels (dB). The noise characteristics are measured by implementing a simulation with the complex reflectivity r_p in Equation 4.9 set to zero. Figure 4.19 shows that an SNR above 21 dB is required to maintain the accuracy of the lookup table. For SNR values lower than 14 dB there is a significant velocity underestimation even at a step size of $\Delta x = 0.5$ μm. In practice, then, our lookup table correction method can be applied only when the SNR level are acceptable. In addition to SNR, other factors, such as transverse flow component [58] and phase noise caused by system shot noise and galvanometer jitter [53] can lead to a broadening of the Doppler spectrum and can result in velocity underestimation. A complete analysis of all of these factors and how the combination of factors leads to Doppler broadening and subsequent velocity underestimation is out of the scope of this work. One approach to combining these effects, however, is to model each effect as the convolution of the Doppler spectrum with a Gaussian function with variance equal to the broadening effect. The effects could be combined by summing the variances of each Gaussian to calculate the cumulative broadening.

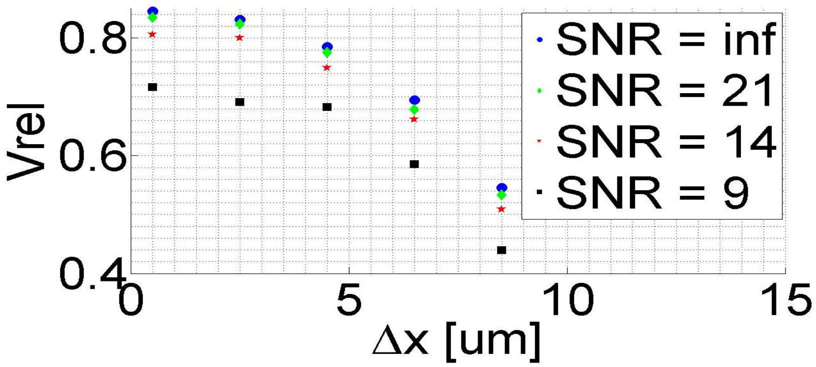


Figure 4.19: Simulation results for falloff of PRDOCT measurements for various signal-to-noise ratios. SNR is measured in decibels (dB). The blue markers have been generated by simulating without noise so that the SNR in this case is equal to infinity.

Phase Unwrapping

Another method that can help correct for velocity underestimation for increasing transverse step size is robust phase unwrapping. In this work, a publicly available implementation [59] of a two-dimensional unwrapping algorithm using branch cuts [60] is adapted for phase unwrapping in vessels with a circular cross section. In one-dimensional phase unwrapping, phase discontinuities greater than π are identified and corrected by adding or subtracting

2π so that the discontinuity is removed. It has been shown that the real unwrapped phase can be recovered if there are no real discontinuities greater than π [61]. Applying this technique to two-dimensional unwrapping, however, introduces inconsistencies in that the unwrapping procedure can become path-dependent [62]. In order to combat this difficulty, Goldstein [62] and others introduced techniques for identifying branch cuts in the phase image that isolate regions that can be unwrapped in a path-independent manner. By operating on each region separately, this technique can be used to compute the unique unwrapped phases. The algorithm used here is based on this idea. Figure 4.20 shows the results of two-dimensional phase unwrapping applied to simulated vessels. After unwrapping, there is no longer a bias in the average Doppler phase shifts inside the vessel.

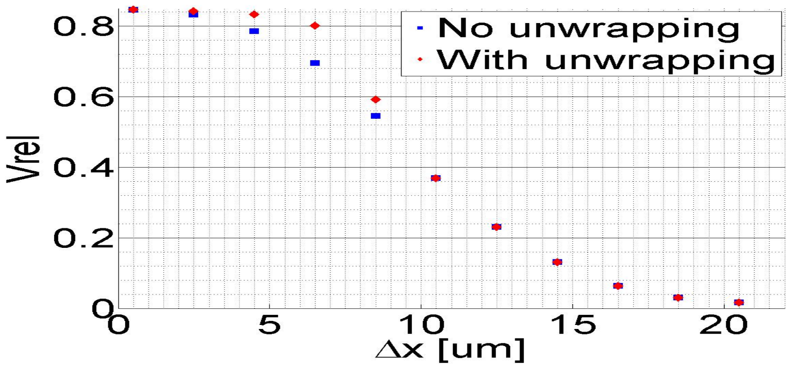


Figure 4.20: Simulation results illustrating the reduced falloff in PRDOCT measurements when phase unwrapping is implemented. Phase unwrapping corrects for the falloff for step sizes less than 4.5 μm. For step sizes above 10 μm, phase unwrapping is too difficult and the result reduces to the no unwrapping case.

Phase unwrapping can also fail under low SNR conditions. Figure 4.21 illustrates the effect of SNR on the measured average velocities after phase unwrapping. For transverse step sizes less than 4.5 μm, the SNR can be reduced to 14 dB without a significant change in the measured velocities. This is in direct contrast to the computed average velocities without phase unwrapping (Figure 4.19) which exhibit a significant decrease for an SNR equal to 14 dB. Thus, the phase unwrapping method for correcting for the falloff with transverse step size is more robust to noise than the lookup table method.

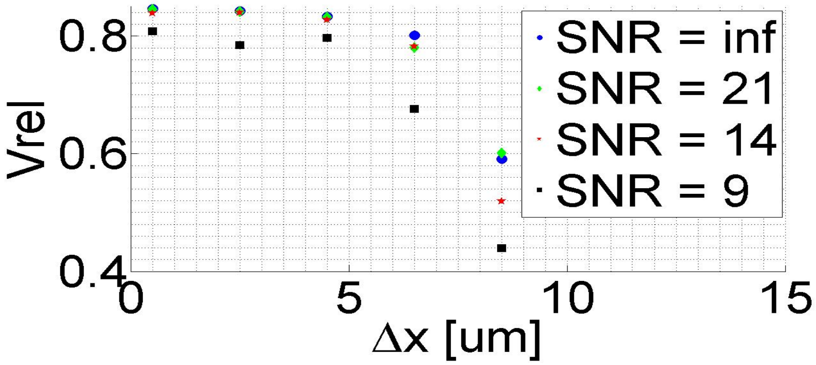


Figure 4.21: Simulation results for falloff of phase unwrapped PRDOCT measurements for various signal-to-noise ratios. SNR is measured in decibels (dB). The blue markers have been generated by simulating without noise so that the SNR in this case is equal to infinity.

4.4.3.3 Increasing the Beam Width

Both the lookup table and phase unwrapping methods are techniques to correct measured velocities for the falloff effect caused by increasing the distance between adjacent A-lines. One method that could mitigate the effect of velocity underestimation without correcting measured velocities is increasing the beam width. Using a larger beam width leads to an increase in the amount of overlap between adjacent A-lines relative to a smaller beam width and should thus reduce the falloff in velocity estimation. In order to test this hypothesis, an experiment was performed to compare the falloff of PRDOCT using two different beam widths equal to 20 μm and 34 μm. Figure 4.22 illustrates that the falloff is slower for the higher beam width. The falloff factor at a transverse step size of $\Delta x = 4.5$ μm is 0.94 for the 20 μm beam width and 0.98 for the 34 μm beam width. For a quantitative comparison, Equation 4.25 was fit to both curves. The parameter M was found to be 10.64 for the 20 μm beam width and 17.15 for the 34 μm beam width. The ratio of the M parameter is 0.62, approximately equal to the ratio of the beam widths which is equal to 0.59. Thus, the effect of increasing the transverse step size on the average velocity measurements in a vessel is approximately proportional to the beam width. Although increasing the beam width can reduce the falloff effect, the tradeoff is a decrease in transverse resolution as well as some loss of signal. This should work well for larger vessels with high velocities but may obscure smaller blood vessels.

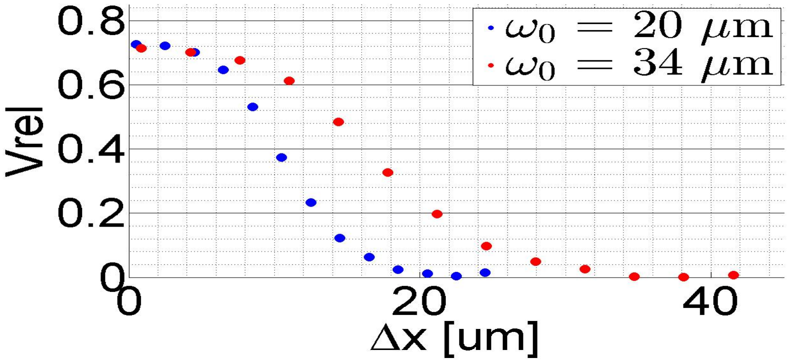


Figure 4.22: Simulation comparison of falloff for different beam widths.

4.4.3.4 Comparison of Phase Averaging and Complex Vector Averaging

In a paper previously published by our research group, the authors measured a falloff in flow measurements that was significantly higher than that found here [19]. Specifically, the ratio of the flow measured at a step size of 4.0 μm relative to the measured flow at 0.7 μm was 0.683. This value was then used as a correction factor for all measurements to correct for the falloff in flow. It is clear from these experiments, however, that the falloff depends on the actual Doppler velocities and thus no single correction factor can be used for all flow measurements. Also, as previously discussed, our experiments show a significantly smaller falloff than 0.683 for all flow velocities up to 0.85 x V_{max}. In order to understand this discrepancy, another experiment was performed to study the difference between phase averaging PRDOCT (used in [19]), in which the phase differences between adjacent A-scans are computed first computed and then averaged, and complex vector averaging PRDOCT (used here). As the authors in [18] explain, the vector averaging procedure helps mitigate the effect of phase wrapping caused by the broadening of the Doppler spectrum due to either low SNR or, in this case, high transverse step size. The phase averaging and vector averaging PRDOCT techniques are compared via histograms in Figure 4.23. It can be seen that even for a small step size of $\Delta x = 0.5\ \mu$m, the phase averaging technique suffers from phase wrapping-induced average phase shift underestimation whereas the vector averaging technique does not. This effect is more pronounced at larger step sizes.

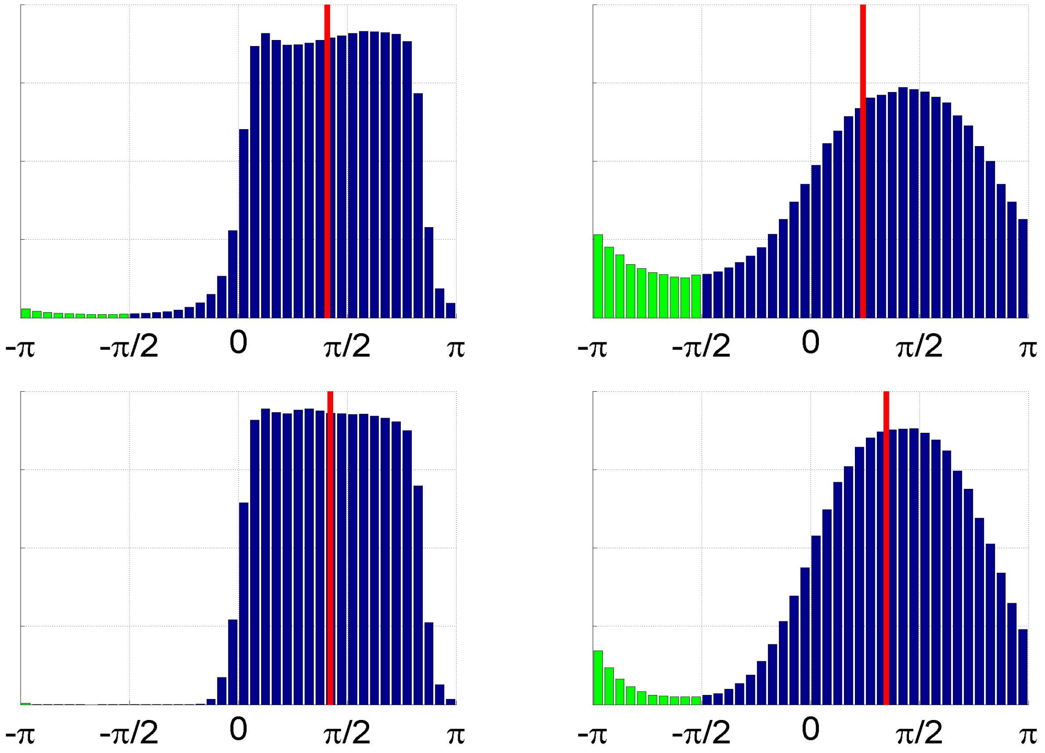


Figure 4.23: Simulated phase distributions for PRDOCT techniques. (a,b): phase averaging PRDOCT. (c,d): vector averaging PRDOCT. Transverse step sizes were set to 0.5 μm (a,c) and 6.5 μm (b,d). Green bins indicate wrapped phases. Vertical red lines indicate the average Doppler shifts and are located at 1.28 (a) radians, 0.75 (b) radians, 1.33 (c) radians and 1.09 (d) radians, respectively. Phase averaging technique is highly susceptible to phase wrapping induced phase shift underestimation.

Figure 4.24 shows the complete falloff curves for phase averaging and vector averaging for a single flow speed. At the high velocity used here, there is a significant difference between the two techniques. At a 4.5 μm step size, the standard averaging technique falls by a factor of approximately 0.75. This value is still slightly larger than that reported in [19]. The reason for this may be because a higher velocity was used, or it may be a result of other phenomena exhibited by blood, such as tumbling, that are not modeled in the simulation. It is clear, however, that vector averaging is significantly more robust at estimating average velocities inside a vessel.

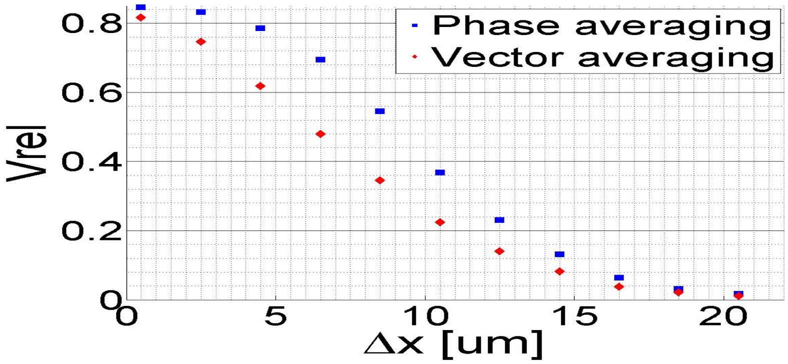


Figure 4.24: Simulation results for comparison of falloff for phase averaging and vector averaging for simulated average velocities in a vessel. Phase averaging suffers from a significant falloff even for small transverse step sizes.

4.4.4 Summary

In summary, we have shown that PR Doppler OCT blood flow estimates fall of significantly with increasing transverse step size. After validating our retinal vessel simulation with *in vitro* flow phantom experiments, we used the simulation to extrapolate the experimental results. We applied logistic curve fits to describe the falloff in the velocity estimates for increasing transverse step sizes which indicated that for the same transverse step size, higher velocities cause a larger falloff in the PRDOCT measurements than lower velocities. This means that it is not possible to correct clinical PR Doppler OCT measurements with a single correction factor. We proposed lookup table a phase unwrapping techniques that can be used in clinical studies to correct for the velocity underestimation. We also showed that increasing the beam width can mitigate the effect of transverse step size on clinical OCT blood flow velocity measurements.

4.5 Conclusions

In this chapter we studied blood flow velocity measurements with Doppler OCT. In Section 4.2, we introduced a simulation of a retinal vessel immersed in background tissue. The synthetic data provides ground truth knowledge of blood flow velocities that can be used to assess the accuracy of Doppler OCT algorithms. The primary novelty of our technique is that it simulates an entire two dimensional OCT B-scan of flowing particles that model moving red blood cells surrounded by stationary particles that model static background retinal tissue. Our simulation mimics clinical Doppler OCT measurements more closely than other related simulations. However, our simulation does not provide a complete picture of the scattering from red blood cells in a retinal blood vessel. Specifically, we model both static tissue scatterers and moving red blood cells with discrete point scatterers. In reality, the geometries of these scatterers are far more complex than this. Moreover, for computational considerations, we fix the number of flow particles in each simulation. In practice, red blood cells enter the scan region at random times and in random positions. A more realistic model would account for this by sampling from a probability distribution that describes the entry of flow particles into the scan region. However, our implementation replaces each particle that leaves the scan region with a new particle deterministically both in time and in space.

Although our simplified model of blood flow through a vessel immersed in retinal tissue does not entirely capture reality, it is still useful in studying various Doppler OCT algorithms. We illustrated its effectiveness with two case studies. First, we examined the effect of vessel diameter on the relative blood flow estimates of the phase-resolved (PR) and moving-scatterer-sensitive (MSS) Doppler OCT algorithms in Section 4.3. After validating our simulation with *in vivo* OCT retinal images, we showed that while PR underestimates flow velocity measurements for very small vessels, MSS wildly overestimates the flow. Furthermore, we showed that for large vessel diameters, the PR algorithm is more accurate for measuring both flow rates and vessel diameters. Thus, although MSS can be used for visualizing flow in small vessels, the PR algorithm is the method of choice for measuring flow in clinical Doppler OCT studies. As a result, our next case study focused only on the PR Doppler OCT algorithm.

A second case study that illustrates the utility of our Doppler OCT simulation was presented in Section 4.4. In this study, we examined the dependence of complex vector averaging PR measurements of average velocity in a vessel on transverse step size. We showed that increasing the transverse step size leads to a broadening of the Doppler spectrum and subsequent phase wrapping. The phase wrapping causes a systematic underestimation of the average velocity in a vessel, and is more pronounced at higher velocities. After validating the simulation with an in vitro flow phantom experiment, we derived a parameterized logistic expression describing the velocity underestimation of PR Doppler OCT with increasing transverse step size and that the logistic function parameters depend on velocity. This indicates that for the same transverse step size, higher velocities cause a larger falloff in the PRDOCT measurements so that correcting for this falloff using a single correction factor for all velocities is not possible. We then presented lookup table and phase unwrapping techniques for correcting the measured velocities. We showed that an SNR above 21 dB is required to maintain the accuracy of the lookup table method, whereas an SNR above only 14 dB is required for the phase unwrapping method. Thus, the phase unwrapping technique appears to be the more precise method due to the relative insensitivity to SNR. Additionally, we showed that increasing the beam width can mitigate the falloff effect for transverse step sizes lower than 4.5 μm without the need to correct measured velocities. All of these proposed techniques can be used to improve the accuracy of clinical Doppler OCT measurements.

Chapter 5

Split-Spectrum Amplitude-Decorrelation Angiography (SSADA)

5.1 Introduction

As discussed in Chapter 4, Doppler OCT is able to calculate absolute blood flow velocity by evaluating the phase differences between adjacent A-scans [13–17]. However, Doppler OCT requires good phase stability for quantifying slow flow velocities [63–65] and suffers from reduced sensitivity for small blood vessels [51, 66]. Furthermore, the measured Doppler frequency shift in a blood vessel varies inversely with the cosine of the Doppler angle, or the angle between the incident beam and blood vessel [19]. Yet many blood vessels in the human retina are nearly perpendicular to the incident beam, so that the Doppler frequency shifts in these vessels are small and difficult to detect. Several techniques, including joint spectral and time domain OCT [56], use of a modified Hilbert transform [67] and smart scanning protocols [68], have been introduced to overcome some of the limitations of Doppler OCT. However, techniques that do not depend critically on the Doppler angle may be particularly useful for visualizing blood vessels in the human retina.

Several angiographic techniques have been introduced to overcome some of the limitations of Doppler OCT algorithms for visualizing retinal blood flow velocities. Some of these techniques are phase-based, while others are amplitude-based. Some of the phase-based techniques include optical coherence angiography [69], optical micro-angiography [70,71], Doppler variance [72,73] and phase variance [74,75]. Some of the amplitude-based techniques include scattering optical coherence angiography [76, 77], speckle variance [78, 79] and correlation mapping [80]. These methods have be have been implemented using both spectral-domain and swept-source OCT imaging systems and have been used to image microvascular networks in human eyes *in vivo*. However, absolute blood flow velocity quantification has yet to be demonstrated.

For a typical Fourier domain OCT setup, the axial resolution, determined by the source central wavelength and its spectral bandwidth, is higher than the lateral resolution determined by the laser beam profile in both X and Y directions. This anisotropic resolution, with higher axial than transverse resolution, will result in higher flow sensitivity for axial motion. In the fundus, ocular pulsation related to heart beat, driven by the retrobulbar orbital tissue, mainly occurs along the axial direction. The anisotropic resolution cell of retinal OCT imaging is very sensitive to this axial motion noise. On the other hand, retinal and choroidal blood flow vectors are primarily transverse to the OCT beam, along the wider (less sensitive) dimensions of the OCT resolution cell. Therefore, to improve the signal-to-noise ratio (SNR) of flow detection, it is desirable to lower the axial resolution and dampen the axial flow sensitivity. This reduces the axial motion noise without sacrificing the transverse flow signal.

To overcome this limitation, we develop split-spectrum amplitude-decorrelation angiography (SSADA) to improve the signal-to-noise ratio (SNR) of flow detection that mitigates the bulk motion decorrelation effect suffered by other amplitude-based angiographic techniques. The full OCT spectrum is split into several narrower bands. Inter-B-scan decorrelation was computed using the spectral bands separately and then averaged. We applied the SSADA algorithm to *in vivo* images of the human macula to visualize the capillary network.

After demonstrating the ability of our novel algorithm to detect flow in retinal blood vessels, we extend SSADA to enable blood velocity quantification. The ability to measure blood flow is critical to the diagnosis of many ocular pathologies. In order to use SSADA for velocity quantification we perform *in vitro* flow phantom experiments. We correlate the SSADA signal at multiple time scales with various preset velocities and derive a linear model relating SSADA measurements to absolute flow velocities using the phantom data. We then discuss the operating range for the linear model as well as its implication for velocity quantification with SSADA in a clinical setting.

The organization of this chapter is as follows. In Section 5.2, we introduce the theory behind the SSADA algorithm. In Section 5.3, we demonstrate macular angiography with the SSADA algorithm. The material presented in Section 5.3 was published in [81]. Then, in Section 5.4, we perform flow phantom experiments that enable flow velocity quantification with SSADA. The material presented in Section 5.4 was published in [82].

5.2 SSADA Theory

Speckle decorrelation has long been used in ultrasound imaging [83, 84] to detect optical scattering from moving particles such as red blood cells. This phenomenon is also clearly exhibited by the real-time OCT reflectance images. The scattering pattern of blood flow varies rapidly over time. This is caused by the fact that the flow stream drives randomly distributed blood cells through the imaging volume (voxel), resulting in decorrelation of the received backscattered signals that are a function of scatterer displacement over time. The contrast between the decorrelation of blood flow and static tissue may be used to extract flow signals for angiography.

Each pixel in a B-scan OCT image is formed from backscattered signals of a 3D volume in space, referred to as a resolution cell (Figure 5.1(A)). The statistical changes in the

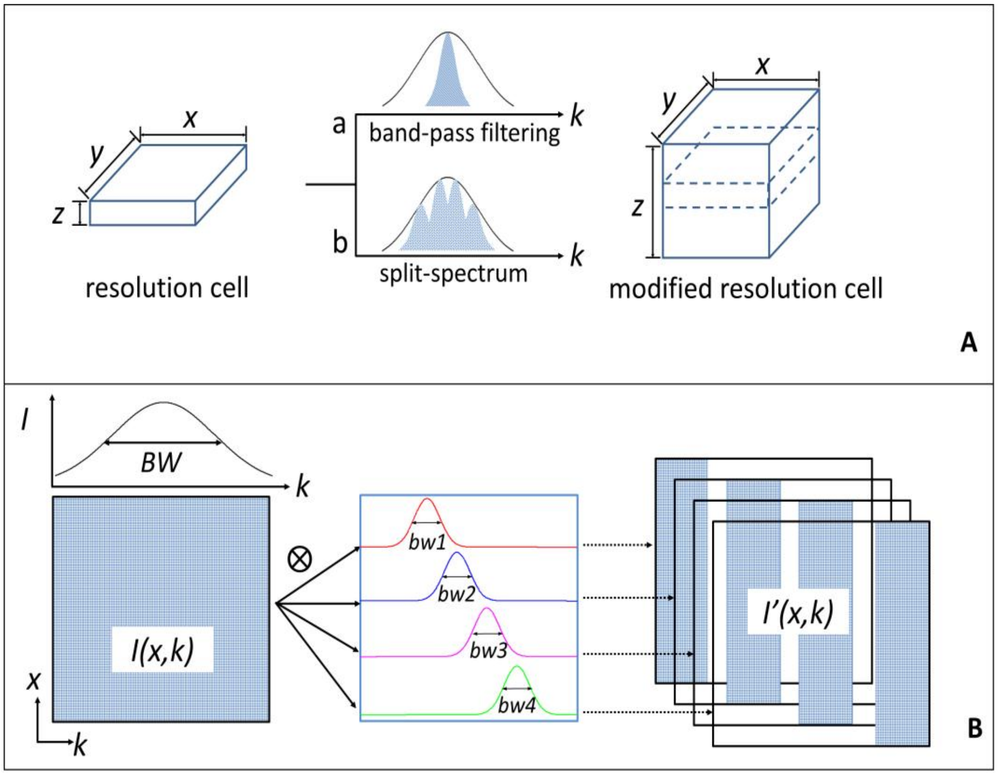


Figure 5.1: Diagrams of the modification of the OCT imaging resolution cell and the split-spectrum method used for this purpose. (A) The resolution cell in the current configuration can be modified into a new resolution cell by using band-pass filtering and split-spectrum methods. (B) Steps showing how the original 2D spectral interferogram $I(k)$ was split into four new spectra $I(k)$ with smaller k bandwidth. BW and bw indicate the bandwidth of full-spectrum and Gaussian filters, respectively. The regions with non-zero values in the data block are indicated by the blue pattern.

envelope intensity are related to the motion of scatterers through the OCT resolution cell. As previously mentioned, the axial resolution (Z) is typically higher than the transverse (X,Y) resolution in an OCT imaging system. However, since retinal blood flow is primarily in the transverse direction, whereas bulk motion noise sources, such as pulsation related to heartbeat, occur primarily in the axial direction [81] it is desirable to dampen the axial flow sensitivity. In order to decrease the sensitivity to the axial motion noise and thus improve the signal-to-noise ratio of flow detection, the SSADA algorithm digitally broadens the axial resolution prior to computing speckle decorrelation.

One straightforward way to achieve this resolution modification is band-pass filtering of the spectral interferogram (shown in Figure 5.1(A)). Unfortunately, this also sacrifices most of the speckle information in the spectral interferogram and decreases the flow signal. Thus this is not an effective way to increase the SNR of flow (decorrelation) detection. A better way to decrease axial resolution without losing any speckle information is to split the spectrum into different frequency bands (shown in Figure 5.1(A)) and calculate decorrelation in each band separately. The decorrelation (flow) images from the multiple spectral bands can then be averaged together to make full use of the speckle information in the entire OCT spectrum. The details of the split-spectrum procedure (Figure 5.1(B)) are explained below.

5.2.1 Split-spectrum

The OCT spectral interferogram is given in Equation 2.12 and is slightly modified in Equation 5.1 to fit the current discussion.

$$I(k) \propto \Re\left\{\iint S(k)r(x,z)\mathrm{exp}\left(4\mathrm{ln}2\left(\frac{x^2}{\omega_0^2}\right)\right)\mathrm{exp}(-i2kz)dxdz\right\} \tag{5.1}$$

The transverse coordinate is denoted by x and the beam is assumed to be at transverse position $x_b = 0$.

Each pixel in an OCT image is formed by a coherent sum of all of the light backscattered from a 3D coherence volume within a sample. The dimensions of the coherence volume are determined by the full-width at half-maximum (FWHM) transverse beam width ω_0 and FWHM coherence length δz_0. In a typical OCT system the axial resolution is higher than the transverse resolution, which leads to higher decorrelation sensitivity for axial motion [85]. In the fundus, blood flow is primarily in the transverse direction, whereas bulk motion noise sources, such as pulsation related to heartbeat, occur primarily in the axial direction [81].

In order to decrease the sensitivity to this axial motion noise and thus improve the signal-to-noise ratio of flow detection, the SSADA algorithm digitally broadens the axial resolution prior to computing speckle decorrelation.

In the SSADA implementation, a desired axial resolution $\delta z_0'$ is first chosen that is at least as large as the nominal axial resolution δz_0. In order to create an isotropic coherence volume that is equally sensitive to axial and transverse flow, the desired axial resolution is set to the transverse resolution, or $\delta z_0' = \omega_0$. A set of Gaussian filters in wavenumber is then created that, when multiplied with the interferometric signal in Equation 5.1, increases the axial resolution from δz_0 to $\delta z_0'$. The jth filter in the filterbank can be expressed as:

$$G_j(k) = \exp(-4\ln2(k - k_{0j}^2)/\Delta k_G^2), \tag{5.2}$$

where k_{0j} is the center wavenumber and Δk_G is its FWHM bandwidth. In order to determine the necessary Δk_G Fourier transform properties as well of properties of Gaussian functions are leveraged [7]. The source spectrum is approximated with the equivalent Gaussian spectrum given in Equation 2.7 to simplify the subsequent mathematical analysis. Note that the product of Equation 2.7 (FWHM = Δk) and Equation 5.2 (FWHM = Δk_G) is a product of two Gaussians in wavenumber space and a corresponding convolution of Gaussians in $z-$space. The convolution results in another Gaussian in z-space with FWHM equal to $\sqrt{(4\ln2/\Delta k)^2 + (4\ln2/\Delta k_G)^2}$ [7]. This quantity is the desired new axial resolution, as it defines the modified FWHM axial resolution resulting from a spectrum split. Note that the first term in the square root is equal to the original axial resolution . Thus, in order to increase the axial resolution from δz_0 to $\delta z_0'$ the FWHM bandwidth should be set to:

$$\Delta k_G = 4\ln2/\sqrt{\delta z_0'^2 - \delta z_0^2}. \tag{5.3}$$

In order to complete the filterbank specification, the distance in wavenumber between adjacent filters is defined as $0.5 \cdot \Delta k_G$. The center wavenumber of each filter is then determined with the constraint that all of the respective center wavenumbers k_{0j} fit within the range of acquired wavenumbers between k_{min} and k_{max}.

Each Gaussian filter is multiplied by the interferogram in Equation 5.1 to create a set of signals $I_j(k) = G_j(k) \cdot I(k)$. Performing Fourier transforms produces a set of $z-$space signals $I_j(z) = G_j(z) * I(z)$. The SSADA algorithm operates only on the OCT magnitude, or $A_j(z) = |I_j(z)|$.

5.2.2 Decorrelation

The SSADA algorithm computes decorrelation at a single voxel between OCT magnitudes separated in time. Let $A_{j,m\Delta t}(x,z)$ denote the jth split spectrum OCT magnitude at voxel (x,z) $(j \in 0,1,...,J-1)$ taken at time $t = m\Delta t$ $(m \in 0,1,...,M-1)$. The split-spectrum amplitude-decorrelation angiography signal at a voxel (x,z) at time lag Δt is defined by:

$$\bar{D}(x,z,\Delta t) = 1 - \frac{2}{M-1}\frac{1}{J}\sum_{j=0}^{J-1}\sum_{m=0}^{M-2}\frac{A_{j,m\Delta t}(x,z) \cdot A_{j,(m+1)\Delta t}(x,z)}{A_{j,m\Delta t}(x,z)^2 + A_{j,(m+1)\Delta t}(x,z)^2}. \tag{5.4}$$

The average decorrelation value $\bar{D}(x,z)$ is bounded below by zero (no decorrelation) and above by one.

For the *in vivo* demonstration presented in the next section, decorrelation is computed between repeated B-scans at the same slow axis scan position (M-B mode imaging). For the phantom experiments in the subsequent experiment, decorrelation is computed between repeated A-lines at the same transverse scan position (M-mode imaging).

5.3 Macular Angiography

To demonstrate the performance of the SSADA algorithm, macular imaging was performed on three normal volunteers using the swept-source OCT system described in [81]. The axial resolution of this system is 5 μm, while the transverse resolution is approximately 11 μm. The axial line rate is 100 kHz. In this demonstration, the system captured 200 A-scans to cover 3 mm for each B-scan. For 3D data acquisition, the entire scan volume was evenly divided into 200 steps, with eight repeated B-scans in each step. In doing so, it required 3.2 seconds to complete one 3D volumetric scan. Under this scanning protocol, the SSADA algorithm was applied to the repeated frame sequences at each step. This implementation computes decorrelation using M-B mode imaging, and the time between measurements at the same voxel is $\Delta t = 2$ ms. Maximum projection images over depth of the retina were used to compute enface decorrelation images.

The macular region of the fundus is responsible for central vision. Capillary dropout in the macular region due to diabetic retinopathy is a major cause of vision loss [86]. Focal loss of the choriocapillary is a likely causative factor in the pathogenesis of both dry and wet age-related macular degeneration [87], the leading cause of blindness in industrialized nations [88]. Thus macular angiography is important. The SSADA algorithm was used to demonstrate macular angiography of both the retinal and choroidal circulations in a normal eye (Figure 5.2). The vascular pattern and capillary networks visualized by SSADA were similar to those previously reported using phase-based OCT angiography techniques [89, 90]. The flow pixels formed a continuous microcirculatory network in the retina. There was an absence of vascular network in the foveal avascular zone (Figure 5.2(A)) of approximately 600 μm diameter, in agreement with known anatomy [91]. There were some disconnected apparent flow pixels within the foveal avascular zone (Figure 5.2(A)) due to noise. Inspection of Figure 5.2(C) shows these false flow pixels to be decorrelation noise in the high reflectance layers of the RPE and photoreceptors. The choriocapillaris layer forms a confluent overlapping plexus [92], so it is to be expected that the projection image of the choroid circulation shows confluent flow (Figure 5.2(B)).

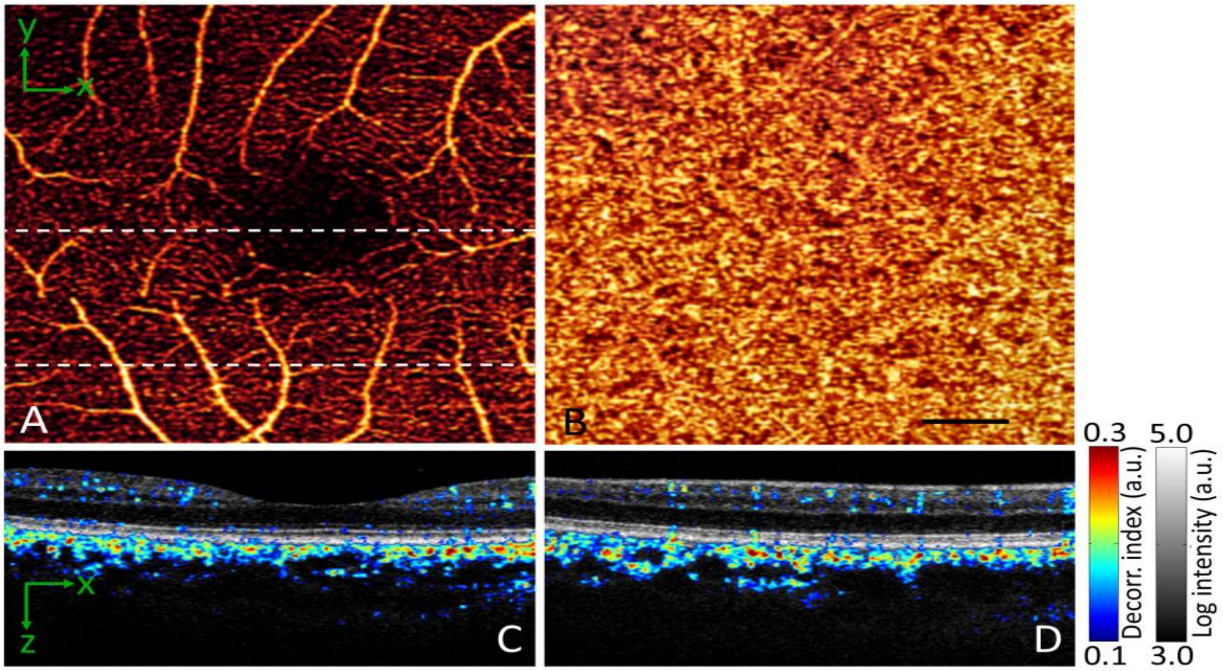


Figure 5.2: *In vivo* 3D volumetric [3.0 (x) 3.0 (y) 2.9 (z) mm] OCT of the macula processed with the SSADA algorithm. The images in the bottom panels have been cropped from 2.9 mm to 1.5 mm axially. (A) En face maximum decorrelation projection angiogram of the retinal circulation. (B) En face maximum decorrelation projection angiogram of the choroidal circulation. Black bar, 500 μm. (C) Horizontal OCT cross section through the foveal center (upper dashed line in A) with merged flow (decorrelation represented in color scale) and structure (reflectance intensity represented in gray scale) information. (D) Merged horizontal cross section of the inferior macula (lower dashed line in A).

The cross sections (Figure 5.2(C, D)) showed retinal vessels from the NFL to the outer plexiform layer, in agreement with known anatomy [93]. The flow in the inner choroid had higher velocity as based on decorrelation seen in the color scale. The volume was also greater than the retinal circulation (Figure 5.2(C, D)), again consistent with known physiology that the choroidal circulation has much higher flow than the retinal circulation [92]. There were signal voids in the outer choroid which may be due to fringe washout from high flow velocity and the shadowing effect of overlying tissue. The cross sections (Figure 5.2(C, D)) also showed a few spots of decorrelation in the RPE layer. These must be artifacts because the RPE is known to be avascular. This is likely due to the projection of decorrelation of flow in a proximal layer (i.e., inner retinal layers) onto distal layers with a strong reflected signal (i.e., RPE). There was also a tendency for vessels to form vertical arrays in the inner retina, which may in some instances be due to the projection artifact as well.

5.4 Velocity Quantification With SSADA

As previously mentioned, several angiographic techniques have been introduced for imaging retinal blood flow. While good qualitative imaging results have been shown for all of these methods, quantitative results that map angiograms to flow velocities are lacking. Decorrelation has been used for quantitative flow measurements in ultrasound [83,84] and thus has the potential to be useful for measuring flow velocities in OCT. Recently, a decorrelation-based method termed intensity-based Doppler variance (IBDV) was introduced and its relationship with velocity was established [94–96]. This method computes decorrelation using only the amplitude signal. One potential advantage of SSADA is that the algorithm first digitally creates an isotropic coherence volume, or resolution cell, prior to computing decorrelation. This should make the algorithm equally sensitive to axial and transverse motion so that SSADA may be used to quantify flow independent of Doppler angle.

The purpose of the work presented in this section is to determine a relationship between SSADA measurements and flow velocity. We hypothesize that this relationship is linear within a certain range of velocities and use in vitro blood flow phantom experiments to test this hypothesis. Whole blood was used in order to to closely mimic in vivo retinal imaging. The SSADA algorithm computes decorrelation between two OCT amplitudes that are taken at the same scanning position but are separated in time. We first examine the dependence of SSADA on Doppler angle after splitting the source spectrum in wavenumber so that the axial resolution of the OCT imaging system is equal to its transverse resolution. The concept of multi-timescale SSADA is introduced, where the time separation between amplitude measurements is varied, and used to examine the time-dependence of decorrelation. We derive an equation that can be used to calculate the absolute flow velocity from a measured decorrelation value and particular time separation between OCT amplitude measurements. We then define the saturation velocity for which the decorrelation values saturate and the linear relationship is no longer valid.

5.4.1 Materials and Methods

The system used in the following experiments has been described in Section 2.2. The axial resolution in blood is $\delta z_0 = 6.5$ μm and the transverse resolution is 11.8 μm, both defined by the full-width at half-maximum values.

5.4.1.1 Scan Protocol

Data was collected using both dual-plane [12] and M-mode protocols. The dual-plane protocol was implemented with two parallel B-scans that were separated by 100 μm and repeated eight times. The B-scans each consisted of 700 A-lines and covered 700 μm in the transverse dimension, resulting in a transverse step size of 1 μm. M-mode scans consisted of 2600 A-lines at the same transverse position inside the capillary. Measurements were taken for five different preset flow rates and five different Doppler angles. The Doppler angle was controlled using a ball and socket mount.

5.4.1.2 Data Processing

The acquisition software was provided by Bioptigen Inc. (Durham, NC), and the custom processing and analysis software was coded in MATLAB. After background subtraction, resampling to k-space, dispersion compensation and Fourier transformation we obtain complex OCT data. Sample OCT intensity images are shown in Figure 5.3.

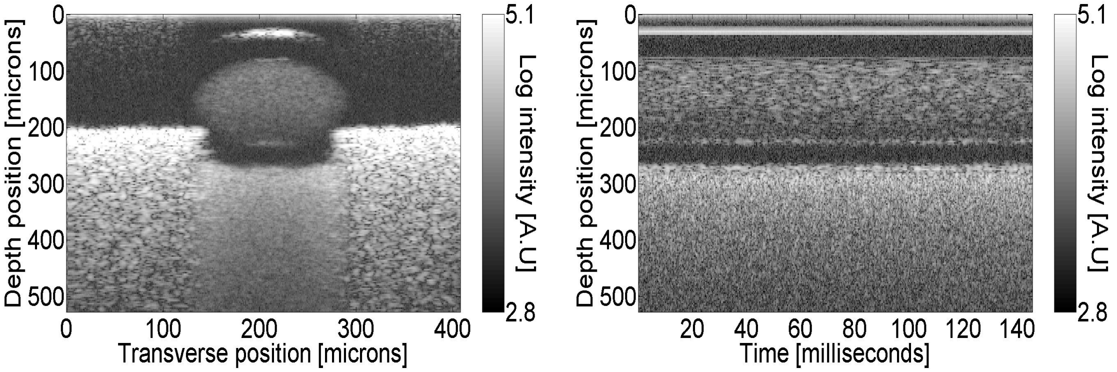


Figure 5.3: Sample intensity images. Capillary was placed on top of a piece of paper. Left panel: B-scan; Right panel: M-mode scan.

The dual-plane B-scans were used to compute the Doppler angle between the probe beam and the glass capillary, while the M-mode scans were used for velocity and decorrelation calculations. The velocities were calculated by first computing Doppler frequency shifts using the phase-resolved Doppler OCT algorithm [18] and then converting to absolute velocity using the measured Doppler angle. The capillary tube was automatically segmented in the M-mode scans using intensity thresholding and morphological operations. Figure 5.4 illustrates that there are significant projection artifacts in the paper underneath the glass capillary, which is likely due to multiple scattering by the moving blood cells. To avoid the influence of these artifacts on the measured Doppler frequency shifts, we chose a subregion in the upper half of the capillary to use for all subsequent calculations.

In order to apply the SSADA algorithm, a Gaussian filterbank is defined as previously described. In this case, the axial resolution if 6.5 μm, while the desired axial resolution

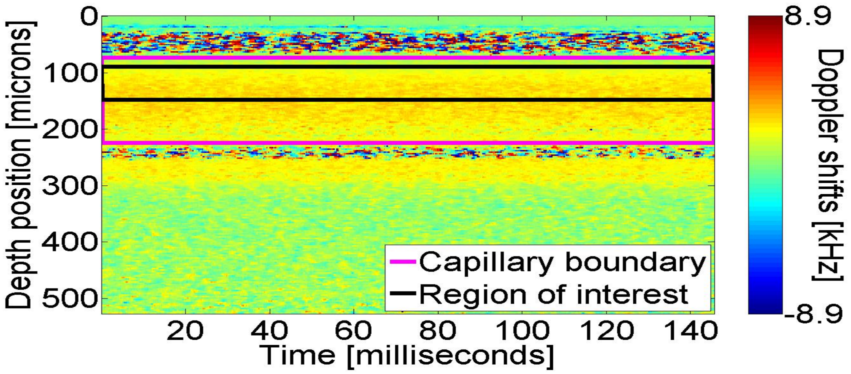


Figure 5.4: Sample M-mode Doppler Frequency shifts from phantom study. The black box denotes the entire capillary, while the purple box denotes the region of interest used in the rest of this study.

is 11.8 μm. Thus, the FWHM bandwidth of each Gaussian filter in wavenumber space is 285 radians/mm, as determined by Equation 5.3. The distance between the Gaussian filters is one-half of this, or 142.5 radians/mm. The wavenumbers acquired by the spectrometer ranges from 717 to 779 radians/mm, and so five Gaussian filters with centers within this range are used. The respective filter centers are located at 719, 733, 747, 761 and 776 radians/mm, respectively.

In this set of experiments, M-mode scans were used to compute the decorrelation of a single A-line. The main purpose of the switch to M-mode imaging is that it enables the study of decorrelation at very fine time scales. The minimum time lag available for this M-mode protocol is $\Delta t = \tau = 56$ μs. In order to study the time course of decorrelation, multi-timescale SSADA (MSSADA) is used by applying Equation 5.4 for multiple time scales $\Delta t = k\tau$ for $k \in \{1, 2, ..., 600\}$. The other parameters used in applying Equation 5.4 include M = 2000 and J=5. Note that because M-mode imaging is used there is no dependence on the transverse coordinate. Applying Equation 5.4 for the time-scales listed gives a MSSADA image as shown in Figure 5.5.

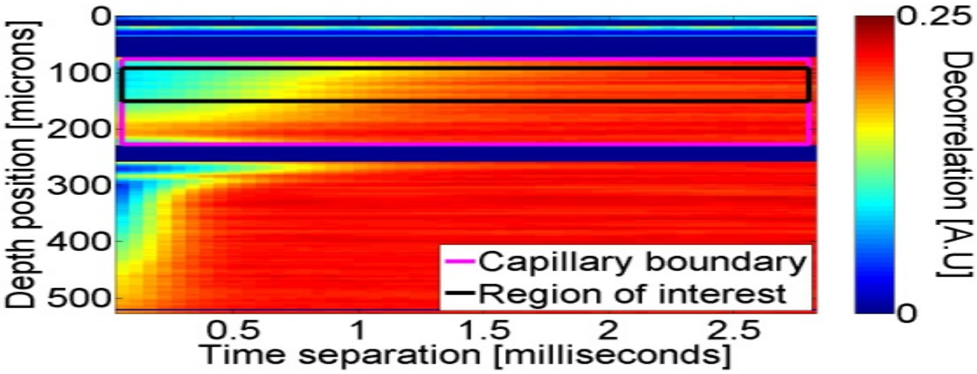


Figure 5.5: Multi-timescale SSADA M-mode image of blood flow through capillary tube. Time separation indicates the time between the two A-lines used to compute the decorrelation.

5.4.2 Results

5.4.2.1 Doppler Angle Dependence

Since SSADA is an amplitude-based method and we create an isotropic voxel by splitting the source spectrum, we expect that the measured SSADA signal will exhibit similar sensitivities in the axial and transverse directions. In order to test this claim, we analyze the variation of the decorrelation signal with Doppler angle. To avoid both noise and saturation artifacts, we choose time separations Δt for which decorrelation values lie in a central region around 0.145 at zero Doppler angle. Specifically, time separations of $\Delta t = 784$, 280 and 168 μs for respective flow speeds of 0.37, 1.29 and 2.00 mm/s were used to plot SSADA measurements versus Doppler angle. Figure 5.6 shows that while the decorrelation signal remains relatively constant for Doppler angles less than 10°, it increases above 10° and then appears to plateau above 20°. This indicates that while SSADA measurements are equally sensitive to axial and transverse flow for Doppler angles less than 10°, the measurements are more sensitive to axial flow for Doppler angles above 10°.

The solid line in Figure 5.6 indicates a linear fit of the relationship between decorrelation and Doppler angle. Although there appears to be a jump in the decorrelation values above

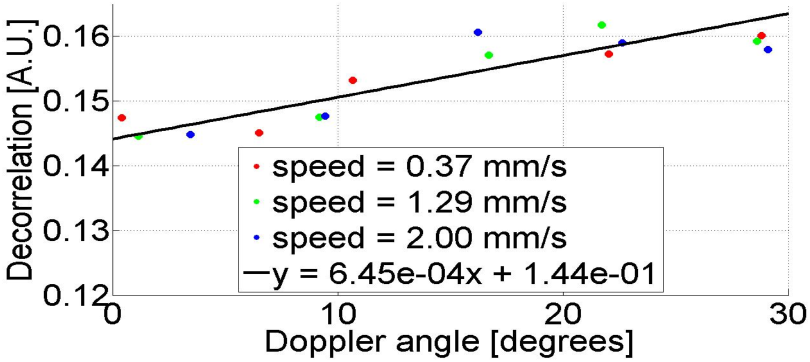


Figure 5.6: Dependence of measured SSADA signal on Doppler angle. Dark line indicates linear fit with Pearsons $R = 0.89$ (p < 0.005). Spearmans correlation coefficient $\rho = 0.87$, indicating a monotonic relationship.

10°, the linear fit is useful in determining how sensitive decorrelation is to changes in Doppler angle. The change in decorrelation due to an angular variation can be used as an indicator of this sensitivity. Specifically, we calculate the difference between the decorrelation values at 0° and 30° that is predicted by the linear fit. We then divide that number by the average decorrelation value over that range and multiply by 100 to convert to a percentage. This deviation is calculated to be 12.8% and indicates a small but significant dependence of decorrelation measurements on Doppler angle.

5.4.2.2 Saturation

SSADA effectively enumerates the dissimilarity between a pixels amplitude at two different time instances. If the interval between the two measurements is long enough, the respective amplitudes will be independent, and the decorrelation signal will be fully saturated. This defines a state of complete decorrelation, and any increase in the time interval will not alter the SSADA measurement. Thus, only decorrelation values that are below the saturation level are useful for distinguishing between varying flow speeds. By visually inspecting Figure 5.7, we can see that complete decorrelation occurs in approximately 500 μs for a flow speed equal to 2 mm/s. At this speed and time separation the red blood cells (RBCs) are displaced by only 1.0 μm, less than one-tenth of the coherence volume size. This suggests that the decorrelation reaches full saturation well before the RBCs move completely through the coherence volume, which indicates a high sensitivity to speckle variation caused by the RBCs moving through the coherence volume. Note that the curves in Figure 5.7 asymptotically approach the complete decorrelation value of 0.21. This motivates us to define a threshold over which the decorrelation rate slows down considerably and the curves in Figure 5.7. begin to flatten. We set this decorrelation saturation threshold to 85% of the asymptotic decorrelation value. The resulting threshold is 0.18, and all decorrelation values above this threshold are referred to as saturated.

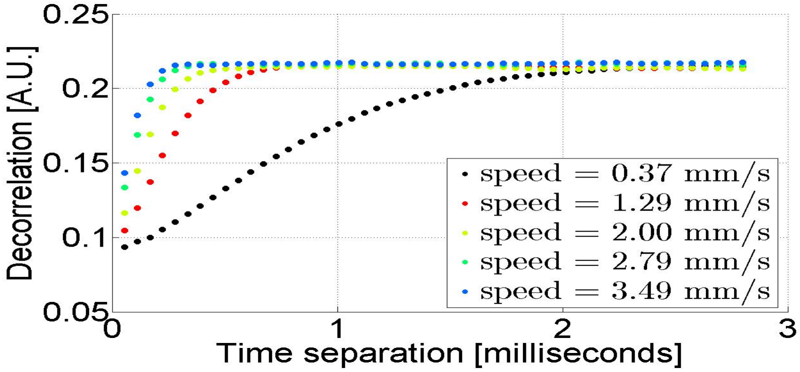


Figure 5.7: Multi-timescale decorrelation averaged over capillary for various flow speeds. The asymptotic decorrelation value for our experiments is 0.21.

5.4.2.3 Relationship between decorrelation and velocity

The goal of this work is to be able to relate the velocity of blood flow to a measured decorrelation value. In order to do so, we examine the velocity dependence of the decorrelation signal at various time separations/intervals. We expect that, for a fixed time separation, the decorrelation will increase with increasing velocity. However, there should be some minimum velocity beneath which measured decorrelations are indistinguishable from noise, and that this velocity should be related to various system noise sources. Furthermore, there should also be a maximum velocity above which measured decorrelations are saturated. As discussed in the section on saturation, the rate of decorrelation begins to slow before decorrelation values are fully saturated. This means that the relationship between decorrelation and velocity changes as the decorrelation approaches full saturation. Therefore, in order to determine a relationship between decorrelation and velocity that does not change with time, we exclude all decorrelation values above the previously defined decorrelation saturation threshold of 0.18 from further analysis. With the saturated points removed, we fit a linear model to the decorrelation versus velocity relationship for a given time separation. The data and fits are shown in Figure 5.8.

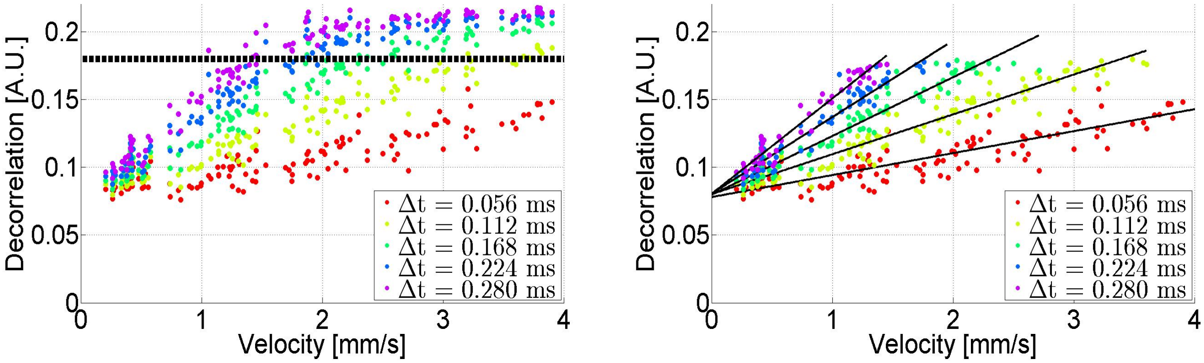


Figure 5.8: Decorrelation vs velocity for various time separations (Δ t) between A-lines. Left panel: black dotted line indicates the saturation cutoff value of 0.18. Right Panel: Linear fits of decorrelation vs velocity after saturated points have been removed.

We summarize the fitting parameters in Table 5.1. From this table we see that the slope changes with time separation. On the other hand, the intercept values remain relatively constant with changing time separations. We thus establish the linear relationship

$$D_{\Delta t} = m_{\Delta t}^{(1)} \cdot v + b^{(1)}, \tag{5.5}$$

where $D_{\Delta t}$ is the measured decorrelation value at a particular time separation Δt, v is the flow velocity, $m_{\Delta t}^{(1)}$ is the slope parameter that is a function of Δt and $b^{(1)}$ is the intercept parameter. The significance of this intercept parameter, which is equal to the decorrelation when the velocity is zero, is treated in Section 5.4.3.1.

Time separation [ms]	**Slope [s/mm]**	**Intercept [a.u.]**	**R-squared**
0.056	0.0162	0.0780	0.76 (p<0.001)
0.122	0.0294	0.0800	0.91 (p<0.001)
0.168	0.0434	0.0796	0.94 (p<0.001)
0.244	0.0564	0.0804	0.94 (p<0.001)
0.280	0.0704	0.0802	0.95 (p<0.001)

Table 5.1: Summary of linear fit of decorrelation versus velocity.

We next examined the dependence of the slopes $m_{\Delta t}^{(1)}$ on time separation Δt and found it to be linear, as described by the equation $m_{\Delta t}^{(1)} = 0.24 \cdot \Delta t + 0.0025$ (Figure 5.9). Since the intercept was very small, the equation could be simplified to $m_{\Delta t}^{(1)} \approx m^{(2)} \cdot \Delta t = 0.24 \cdot \Delta t$. The effect of this approximation is discussed in Section 5.4.3.2.

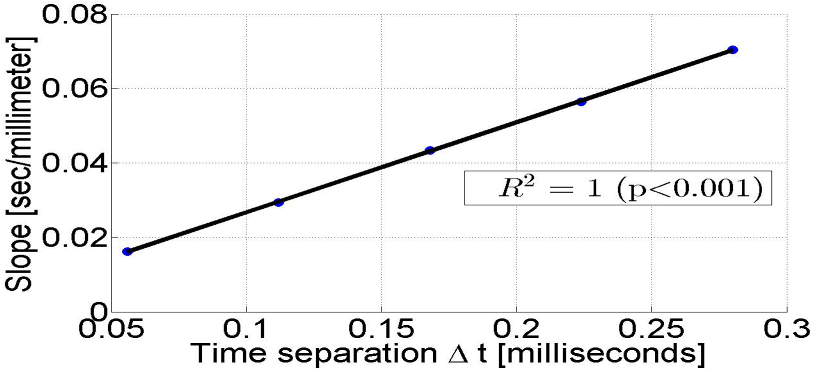


Figure 5.9: Linear fit of slope $m_{\Delta t}^{(1)}$ versus time Δt.

Plugging this relationship into Equation 5.5 gives the decorrelation as a function velocity and time separation as:

$$D(v, \Delta t) = m^{(2)} \cdot \Delta t \cdot v + b^{(1)} = 0.24 \cdot \Delta t \cdot v + 0.08. \tag{5.6}$$

In practice, we measure the decorrelation at a particular time separation and wish to find the flow velocity. Thus, we can invert Equation 5.6 to solve for velocity. Substituting m for $1/m^{(2)}$ and b for $b^{(1)}$ we can write:

$$v(D, \Delta t) = \frac{m \cdot (D - b)}{\Delta t} = \frac{4.17 \cdot (D - 0.08)}{\Delta t}. \tag{5.7}$$

This model is only valid for a specific range of velocities and time separations, which define an operating range for our model. Using Equation 5.7, we can compute the saturation velocity v_{SAT}, which is defined as the velocity at which the decorrelation reaches the saturation cutoff value of 0.18, for various time separations . The linear model in Equation 5.7 does not hold for velocities above v_{SAT}. Some time separation-saturation velocity pairs are illustrated in Table 5.2.

Time separation [ms]	**Saturation velocity [mm/s]**
0.056	7.4
0.122	3.7
0.168	2.5
0.244	1.8
0.280	1.5

Table 5.2: Operating range for linear model.

5.4.3 Discussion

5.4.3.1 Model Parameters

In order to study the parameters of our model, we first rewrite Equation 5.7 as:

$$v \cdot \Delta t = \Delta x = m \cdot (D - b) = 4.17 \cdot (D - 0.08), \tag{5.8}$$

where Δx is the distance that the RBCs move between scans separated by time interval Δt and all other terms have been previously defined. The two parameters in this model are the slope m and the decorrelation intercept b. Since decorrelation is dimensionless, the parameter b must be dimensionless as well. Furthermore, we see from Equation 5.8 that when the RBC displacement equals zero, the decorrelation equals 0.08. Thus the parameter b is equal to the decorrelation in non-flow pixels, or the bulk decorrelation. It equals the minimum measurable decorrelation value and can be defined as the decorrelation noise floor. We expect that this parameter will vary inversely with the system signal-to-noise ratio, similar to the way that the phase noise floor does for Doppler OCT [97]. Further experiments are required to verify if and how this parameter relates to the signal-to-noise ratio of a particular OCT imaging system.

The slope parameter m must have spatial units (eg μm) in order to ensure consistency. We can understand its significance by examining decorrelation saturation. Specifically, after applying the decorrelation saturation cutoff value ($D_{SAT} = 0.08$) to Equation 5.8 and solving for m we have:

$$m = 4.17 = \Delta x_{SAT}/0.1, \tag{5.9}$$

where $\Delta x_{SAT} = 0.417$ μm indicates the distance the RBCs must move in order to saturate the decorrelation measurement. Δx_{SAT} is proportional to the saturation velocity v_{SAT} for a constant Δt and should depend on a particular OCT imaging system. We expect that Δx_{SAT}, and thus m, should be proportional to a particular spatial scale related to the physical imaging parameters. It has yet to be determined, however, what the relevant spatial scale might be. One hypothesis is that Δx_{SAT} increases with increasing beam width ω_0 so that the ratio $\Delta x_{SAT}/\omega_0 \approx 0.0353$ is a constant. It may also be the case that Δx_{SAT}, and consequently v_{SAT} scales with wavelength, so that the saturation velocities given in Table

5.2 increase by a factor of 1.24 for a 1050 nm OCT imaging system. Further experiments using different wavelengths and beam widths are required to test these hypotheses.

5.4.3.2 Model Limitations

There are a number of limitations of the linear model in Equation 5.7. First, in order to establish the linear relationship between decorrelation and velocity we had to exclude saturated points from our analysis. In practice, this establishes an upper limit on a velocity-time separation pairing for quantitative SSADA using our linear model. Furthermore, as we can see in Figure 5.7, for very slow flow speeds (e.g. 0.37 mm/s) the curve looks more s-shaped than linear. We expect that for slower speeds the curve will look even more s-shaped. It seems then that a more accurate model for the decorrelation to velocity relationship might be sigmoidal. Both of these models would also naturally handle the saturated data points.

Another limitation of our model is the assumption that the parameter b, related to the decorrelation of stationary scatterers, is independent of the time separation Δt. For the time separations examined in our analysis, this parameter was indeed constant. However, we expect that the bulk decorrelation should increase with time and reach a saturation point when the time separation is very large. Furthermore, we treated the intercept as a small constant and consequently left it out of the derivation of Equation 5.6. This parameter may be related to the decorrelation caused by shot noise, system vibrations or Brownian motion. Accounting for this parameter would cause a slight increase in the calculated saturation velocities. Lastly, we expect that the decorrelation should depend on scatterer size and possibly blood cell concentration. We used blood in our experiments to mimic *in vivo* measurements as closely as possible. However, we might gain some additional insight by varying scatterer size and hematocrit levels. Additional experiments are needed to test these hypotheses.

5.4.3.3 Comparison with previous work on intensity-based Doppler variance angiography

Liu and colleagues also performed similar flow phantom experiments [95, 96] and found similar results for both intensity-based Doppler variance (IBDV) and Doppler phase variance. Note that they used IBDV which is similar to SSADA but does not apply spectrum splitting and uses a difference averaging procedure. Many of the results of the two works are similar, including establishing decorrelation saturation values and a linear range relating the calculated signal to velocity that depends on the time separation between measurements. However, there are a couple of important differences between the results in that work and those shown here. First, there is a significant difference regarding the dependence on Doppler angle. In order to compare the variation with Doppler angle, we first normalize our SSADA measurements by subtracting the background decorrelation of 0.08 found at zero flow. Because the IBDV background values were negligibly small compared to flow signal in [96] background subtraction was not necessary for IBDV. We compare the variation in the measurements over a Doppler angular range of approximately 18° (the largest angle tested by Liu et al.). Specifically, after background subtraction, the SSADA signal increases from 0.065 at perpendicular incidence to approximately 0.080 at a Doppler angle of 18° for the data presented in 5.6. Thus, our results in this work indicate that the SSADA signal increases by approximately 23% over an angular range of 18°. On the other hand, the IBDV signal in [96] increases from 80 at perpendicular incidence to approximately 150 at 18°, an 87.5% increase over the same angular range. Thus, the Doppler angle dependence of IBDV [96] was significantly higher than the angle dependence of SSADA reported here. Another important difference between this work and that found in [95] is that in [95] the authors showed a saturation velocity over 100 mm/s for an inter-B-frame time separation of 0.02 ms, whereas our model predicts a saturation velocity of approximately 20 mm/s for that time scale.

We hypothesize that these significant differences are likely caused by the choice of the algorithms and the difference in the flowing phantom. Specifically, by creating an isotropic voxel after splitting the source spectrum, SSADA aims to reduce flow directional sensitivity over IBDV. This could explain the reduced directional dependence of SSADA over IBDV, which did not split the OCT signal spectrum and therefore had much finer axial resolution compared to transverse spot size. Additionally, intralipid solution composed of spherical particles of 0.356 μm diameter was used as the flowing phantom in [95, 96]. In contrast our flow phantom used whole blood where the predominant scattering particles are red blood cells that have an average diameter of 7.2 μm and a disc-like shape [98]. Because Doppler variance decreases with increasing particle size [99], we expect that the saturation velocity decreases as well. Additional effects of blood such as tumbling and high viscosity could cause these observed differences as well.

5.4.3.4 Clinical SSADA

The flow phantom experiments presented in this work have a number of implications for clinical SSADA. We have shown that SSADA measurements have little dependence on Doppler angle for Doppler angles less than 10°. So for retinal imaging where the OCT beam is nearly perpendicular to retinal vessels, clinical SSADA may be effectively angle independent. The clinical SSADA scans previously published by our research group [81, 100, 101] used a inter-frame time separation of 2.0 ms, which is on the long side of the time scale investigated in this article. Referring back to Figure 5.5 we see that for this time scale the saturation velocity is 0.2 mm/s (0.3 mm/sec if adjusted for the longer wavelength of 1050 nm). Since human retinal capillary flow speeds have been estimated to be in the range of 0.4-3.0 mm/s [102–104], this suggests that SSADA is well suited for detailed angiography down to the capillary level. However, decorrelation signal should be saturated even at the capillary level according to our phantom calibration. This does not entirely agree with the clinical retinal angiograms that we have observed, where there is a gradation of flow signals at the smallest retinal vessels as visualized by a false color scale [81, 100], and this graduated flow signal increased with visual stimulation [101]. We hypothesize that this gradation could be due to the fact that capillaries are smaller than the diameter of the OCT probe beam. In this case, the static background tissue may lower the overall decorrelation value and thus increase the saturation velocity. Our flow phantom used a capillary tube that is much larger than the OCT beam diameter. This is one aspect in which our phantom setup differs significantly from real human capillaries. In other aspects such as the beam diameter, the use of whole blood, and the SSADA algorithm, the phantom results should simulate the clinical parameters well.

For the purpose of measuring flow velocity, a faster OCT system with a shorter interframe time scale would be better suited. Specifically, in order to bring the saturation velocity above 3.0 mm/s and thus enable capillary velocity quantification within the linear range, a time separation less than 139 μs is needed, according to 5.5. If our M-B mode imaging protocol calls for 200 A-lines per B-scan, then an imaging speed of 1.4 million A-lines per second (1.4 MHz) is needed. Thus, megahertz OCT systems [105, 106] may be useful for blood flow velocity quantification within the linear range of SSADA.

5.4.4 Summary

We performed *in vitro* flow phantom experiments in order to derive a linear relationship between SSADA measurements and absolute flow velocity. We hypothesized that SSADA measurements are independent of Doppler angle after splitting the spectrum to create an isotropic voxel. Contrary to our hypothesis, an angle dependence was found, but the variation due to Doppler angle was small and much reduced compared to previous work using non-split-spectrum intensity-based angiography [95]. The phantom experiments established that the decorrelation signal is linearly related to velocity over a limited range, and that this range is dependent on the time scale of SSADA measurement. The saturation velocities were also proportional to the SSADA time scale. Extrapolation to OCT systems used for clinical retinal imaging with SSADA [81, 100] indicate that the systems should be detecting flow down to the slowest capillary velocity.

5.5 Conclusions

In this chapter we introduced a novel method for visualizing and quantifying retinal blood flow velocities called split-spectrum amplitude-decorrelation angiography (SSADA). Unlike Doppler OCT techniques, our amplitude-based SSADA algorithm is sensitive to transverse flow and is thus suitable for flow detection in the human macular region in which the blood vessel are nearly perpendicular to the input probe beam. One advantage that our SSADA algorithm has over other amplitude-based angiography algorithms is that SSADA computes decorrelation at a pixel after damping the axial flow sensitivity, which reduces its susceptibility to pulsation noise that is primarily manifested in the axial direction. In Section 5.2, we described the theory behind the SSADA algorithm and described how the axial resolution could be digitally increased by splitting the source power spectrum. Then, in Section 5.3, we demonstrated that SSADA can be used to detect transverse flow in the human macular region.

After demonstrating the ability of our novel algorithm to detect flow in retinal blood vessels, we extended SSADA to enable blood velocity quantification in Section 5.4 using *in vitro* flow phantom experiments. We first showed that SSADA is insensitive to Doppler angles less than 10°, which means that SSADA is effectively angle independent in the human macular region. This is important for clinical SSADA implementations because it means that measurement of the Doppler angle is not necessary to compute absolute blood flow velocities. Then, we derived a linear model that relates the SSADA signal to total blood flow velocity measurements. We identified an operating range over which our linear model is valid and established that our clinical SSADA implementation should be detecting flow down to the lower capillary velocities. Thus, SSADA is a novel technique with reduced axial noise sensitivity that can be used to quantify absolute retinal blood flow velocities.

Chapter 6

Summary and Possible Extensions

In this thesis, we presented a number of novel contributions to Fourier domain OCT imaging of the human retina. One common theme throughout this work was the generation of synthetic data to provide a ground truth for quantitative evaluation of various OCT image processing algorithms. Synthetic data is always first validated against either *in vivo* or *in vitro* data and then used to extrapolate the results in a variety of conditions.

For structural OCT imaging, we first discussed the importance of retinal layer segmentation for diagnosing ocular pathologies. Then, we introduced a novel retinal layer segmentation algorithm that is capable of segmenting the nine retinal layer boundaries. The segmentation algorithm utilized a sparse learning approach to calculate a sparse approximation to an *in vivo* retinal image. A graph theoretic approach is then taken to segment the retinal boundaries using the sparse approximation. To enable quantitative evaluation of the segmentation algorithm, we developed a novel technique for generating synthetic OCT intensity images and validated it by comparing to *in vivo* retinal scans. We then used our technique to provide an objective and quantitative evaluation of our segmentation algorithm, which was shown to be fairly robust to signal-to-noise ratios down to 10 dB. The synthetic OCT intensity image generation technique could be very useful for the evaluation of other image processing algorithms.

For functional OCT imaging, we first discussed the importance of measuring blood flow velocities in the retina for diagnosing ocular pathologies. We then implemented a novel simulation of a retinal vessel immersed in tissue and subsequently used it to generate synthetic Doppler OCT retinal data. We presented the utility of our simulation with two case studies. First, we studied the accuracy of two Doppler OCT algorithms. We showed that the phase-resolved (PR) algorithm is more accurate than the moving-scatterer-sensitive (MSS) algorithm in terms of measuring blood flow and vessel diameter. However, for vessel diameters less than 3 times the beam width, we showed that PR underestimates flow velocity measurements. Second, we used the simulation to analyze the effect of transverse beam step size on the velocities calculated by PR Doppler OCT. We demonstrated that increasing the transverse step size produces a systematic reduction in measured Doppler velocities. We then proposed two techniques, including a table lookup method and careful phase unwrapping, for correcting the measured velocities and discussed the practical limitations of each method.

We also presented an amplitude-based flow velocity imaging algorithm called split-spectrum amplitude-decorrelation angiography (SSADA). In contrast to Doppler based methods, we showed that SSADA is sensitive to flow that is transverse to the input beam, which has important practical implications because most of the blood vessels in the human macula are nearly perpendicular to the OCT illumination beam. We then used *in vitro* flow phantom experiments in order to derive a linear relationship between SSADA measurements and absolute flow velocity to enable flow quantification with SSADA. The phantom experiments established that the decorrelation signal is linearly related to velocity over a limited range, and that this range is dependent on the time scale of SSADA measurement. The saturation velocities were also proportional to the SSADA time scale. Extrapolation to OCT systems used for clinical retinal imaging with SSADA indicate that the systems should be detecting flow down to the lowest range capillary velocity.

Each topic presented in this thesis could be extended in future work. Our novel segmentation algorithm could be extended to relax the constraint that retinal layers must extend across the entire image. Doing so might allow pathological cases to be handled. Also, speckle statistics could be incorporated into the graph cut algorithm to increase robustness. The synthetic image generation method could be extended to allow for the simulation of different types of retinal structures, such as pathologies.

Our retinal vessel simulation was shown to be useful for studying Doppler OCT measurements and generating speckle patterns. In order to enable its application to SSADA measurements, the geometry of the scatterers should be more realistic. For example, the blood cells could be modeled by spheres with diameters equal to the approximate red blood cell diameter instead of using a point scatterer model. Also a more realistic model could be used for particle entry into the scan region. Additionally, the vessel simulation method could be used to develop an improved method for estimating blood vessel diameters.

The SSADA algorithm could be improved by incorporating image registration to reduce the effect of bulk motion noise on decorrelation measurements. For quantitative SSADA, a more accurate nonlinear model (e.g. sigmoidal) could be examined to extend the operating range and enable absolute flow velocity measurements for fast flow speeds. The model could also be tested on ultra-high speed systems for *in vivo* velocity quantification.

Reference List

[1] D. Huang, E.A. Swanson, C.P. Lin, J.S. Schuman, W.G. Stinson, W. Chang, M.R. Hee, T. Flotte, K. Gregory, C.A. Puliafito, and J.G. Fujimoto. Optical coherence tomography. *Science*, 254(5035):1178–1181, 1991.

[2] W. Drexler and J. G. Fujimoto (eds.). *Optical Coherence Tomography: Technology and Applications*. Springer, Berlin, Germany, 2008.

[3] Rhcastilhos. Schematic diagram of the human eye. `https://commons.wikimedia.org/wiki/File:Schematic_diagram_of_the_human_eye_en.svg`, 2007. [Retrieved April 5, 2011].

[4] A. Chan, J. S. Duker, T. H. Ko, J. G. Fujimoto, and J. S. Schuman. Normal macular thickness measurements in healthy eyes using stratus optical coherence tomography. *Arch. Ophthalmol*, 124(2):193–198, 2006.

[5] J. S. Schuman, M. R. Hee, C. A. Puliafito, C. Wong, T. Pedut-Kloitzman, C. P. Lin, E. Hertzmark, J. A. Izatt, A. Swanson, and J. G. Fujimoto. Quantification of nerve fiber layer thickness in normal and glaucomatous eyes using optical coherence tomography. *Arch. Ophthalmol. (Chicago)*, 113:586–596, 1995.

[6] M. R. Hee, C. A. Puliafito, J. S. Duker, E. Reichel, J. G. Coker, J. R. Wilkins, J. S. Schuman, E. A. Swanson, and J. G. Fujimoto. Topography of diabetic macular edema with optical coherence tomography. *Ophthalmology (Philadelphia)*, 105(2):360–370, 1998.

[7] A. D. Poularikas and S. Seely. *Signals and Systems*. Krieger Publishing Co., New York, NY, 1991.

[8] S. Nemoto. Determination of waist parameters of a gaussian beam. *Applied Optics*, 25(21):3859–3863, 1986.

[9] S.H. Yun, G.J. Tearney, J.F. de Boer, and B.E. Bouma. Motion artifacts in optical coherence tomography with frequency-domain imaging. *Optics Express*, 12(13):2977–2998, 2004.

[10] T. J. Eon, Y. C. Ahn, C. S. Kim, and Z. Chen. Calibration and characterization protocol for spectral-domain optical coherence tomography using fiber bragg gratings. *Journal of Biomedical Optics Letters*, 16(3):030501, 2011.

[11] X.J. Wang, T.E. Milner, and J.S. Nelson. Characterization of fluid flow velocity by optical doppler tomography. *Optics Letters*, 20(11):1337–1339, 1995.

[12] Y. Wang, B. Bower, J.A. Izatt, O. Tan, and D. Huang. In vivo total retinal blood flow measurement by fourier domain doppler optical coherence tomography. *Journal of Biomedical Optics*, 12(4):041215, 2007.

[13] Y. Zhao, Z. Chen, C. Saxer, S. Xiang, J.F. de Boer, and J.S. Nelson. Phase-resolved optical coherence tomography and optical doppler tomography for imaging blood flow in human skin with fast scanning speed and high velocity sensitivity. *Optics Letters*, 25(2):114–116, 2000.

[14] V. X. D. Yang, M. L. Gordon, A. Mok, Y. Zhao, Z. Chen, R. S. C. Cobbold, Brian C. Wilson, and I. Alex Vitkin. Improved phase-resolved optical doppler tomography using the kasai velocity estimator and histogram segmentation. *Optics Communications*, 208(4):209–214, 2002.

[15] R. Leitgeb, L. Schmetterer, W. Drexler, A. Fercher, R. Zawadzki, and T. Bajraszewski. Real-time assessment of retinal blood flow with ultrafast acquisition by color doppler fourier domain optical coherence tomography. *Optics Express*, 11(23):3116–3121, 2003.

[16] M. A. Choma, A. K. Ellerbee, S. Yazdanfar, and J. A. Izatt. Doppler flow imaging of cytoplasmic streaming using spectral domain phase microscopy. *Journal of Biomedical Optics*, 11(2):024014, 2006.

[17] B. Baumann, B. Potsaid, M. F. Kraus, J. J. Liu, D. Huang, J. Hornegger, A. E. Cable, J. S. Duker, and J. G. Fujimoto. Total retinal blood flow measurement with ultrahigh speed swept source/fourier domain OCT. *Biomedical Optics Express*, 2(6):1539–1552, 2011.

[18] A. Szkulmowska, M Szkulmowski, A. Kowalczyk, and M. Wojtkowski. Phase-resolved doppler optical coherence tomography - limitations and improvements. *Optics Letters*, 33(13):1425–1427, 2008.

[19] Y. Wang, B. Bower, J.A. Izatt, O. Tan, and D. Huang. Retinal blood flow measurement by circumpapillary fourier domain doppler optical coherence tomography. *Journal of Biomedical Optics*, 13(6):064003 1–9, 2008.

[20] C.A. Puliafito, M.R. Hee, C.P. Lin, 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*, 102(2):217–229, 1995.

[21] U. Schmidt-Erfurth, R. A. Leitgeb, S. Michels, B. Povazay, S. Sacu, B. Hermann, C. Ahlers, H. Sattmann, C. Scholdaand A. F. Fercher, and W. Drexler. Three-dimensional ultrahigh-resolution optical coherence tomography of macular diseases. *Invest. Ophth. and Vis. Sci.*, 46(9):3393–3402, 2005.

[22] M. Mujat, R. Chan, B. Cense, B.H. Park, T. Akkin C. Joo, T.C. Chen, and J.F. de Boer. Retinal nerve fiber layer thickness map determined from optical coherence tomography images. *Optics Express*, 13(23):9480–9491, 2005.

[23] A. Mishra, A. Wong, K. Bizheva, and D. Clausi. Intra-retinal layer segmentation in optical coherence tomography images. *Optics Express*, 17(26):23719–23728, 2009.

[24] J.M. Schmitt, S.H. Xiang, and K.M. Yung. Speckle in optical coherence tomography. *J. of Biomedical Optics*, 4(1):95–105, 1999.

[25] J. Tokayer, A. Ortega, and D. Huang. Sparsity-based retinal layer segmentation of optical coherence tomography images. In *IEEE International Conference on Image Processing*, 2011.

[26] 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. *Optics Express*, 18(18):19413–19428, 2010.

[27] M.K. Garvin, M.D. Abramoff, R. Kardon, S.R. Russel, X. Wu, and M. Sonka. Intraretinal layer segmentation of macular optical coherence tomography images using optimal 3d graph search. *Medical Imaging, IEEE Transactions on*, 27(10):1495–1505, 2008.

[28] 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. *Optics Express*, 18(20):21293–21307, 2010.

[29] H. Ishikawa, D. Stein, G. Wollstein, S. Beaton, J.G. Fujimoto, and J.S. Schuman. Macular segmentation with optical coherence tomography. *Investigative Ophthalmology and Visual Science*, 46(6):2012–2017, 2005.

[30] D.C. Fernandez, H.M. Salinas, and C.A. Puliafito. Automated detection of retinal layer structures on optical coherence tomography images. *Optics Express*, 13(25):10200–10216, 2005.

[31] G. Wang, Y. Bresler, and V. Ntziachristos. Guest editorial: Compressive sensing for biomedical imaging. *IEEE Transactions on Medical Imaging*, 30(5):1013–1016, 2011.

[32] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Sys., Man., Cyber*, 9(1):62–66, 1979.

[33] R. Pique-Regi, J. Monso-Varona, A. Ortega, R.C. Seeger, T.J. Triche, and S. Asgharzadeh. Sparse representation and bayesian detection of genome copy number alterations from microarray data. *Bioinformatics*, 24(3):309–318, 2008.

[34] M.E. Tipping and A.C. Faul. Fast marginal likelihood maximisation for sparse bayesian models. In C.M. Bishop and B.J. Frey, editors, *Proc. of the Ninth International Workshop on Artificial Intelligence and Statistics*, Key West, FL., 2003.

[35] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[36] T. H. Ko, J. G. Fujijmoto, J. S. Schuman, L. A. Paunescu, A. M. Kowalevicz, I. Hartl, W. Drexler, G. Wollstein, H. Ishikawa, and J. S. Duker. Comparison of ultrahigh- and standard-resolution optical coherence tomography for imaging macular pathology. *Ophthalmology*, 112:1922–1935, 2005.

[37] P. Serranho, C. Maduro, T. Santos, J. Cunha-Vaz, and R. Bernardes. Synthetic oct data for image processing performance testing. In *IEEE International Conference on Image Processing*, 2011.

[38] B. Delaunay. Sur la sphere vide. *Bull. Acad. Science USSR VII: Class. Sci. Mat. Nat.*, pages 793–800, 1934.

[39] R. Franke and G. Nielson. Smooth interpolation of large sets of scattered data. *Int. J. Numer. Meths. Eng.*, 15:1691–1704, 1980.

[40] O. R. Musin. Properties of the delaunay triangulation. In *Proceedings of the thirteenth annual symposium on Computational geometry*, pages 424–427, 1997.

[41] R. Stafford. How to get pixels from a triangle. `http://www.mathworks.com/matlabcentral/newsreader/view_thread/170977`, 2008. [Retrieved March 23, 2012].

[42] S. M. Motaghian Nezam, C. Joo, G. J. Tearney, and J. F. de Boer. Application of maximum likelihood estimator in nano-scale optical path length measurement using spectral-domain optical coherence phase microscopy. *Optics Express*, 16(22):17186–17195, 2008.

[43] J. Sijbers, A. A. den Dekker, E. Raman, and D. Van Dyck. Parameter estimation from magnitude mr images. *International Journal of Imaging Systems and Technology*, 10:109–114, 1999.

[44] B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, New York, NY, 1986.

[45] G. Lamouche, C. E. Bisaillon, S. Vergnole, and J. P. Monchalin. On the speckle size in optical coherence tomography. In *Proc. SPIE 6847, Optical Coherence Tomography and Coherence Domain Optical Methods in Biomedicine XII, 684724M*, 2008.

[46] S. Lu, C.Y. Cheung, J. Liu, J.H. Lim, C.K. Leung, and T.Y. Wong. Automated layer segmentation of optical coherence tomography images. *IEEE Transactions on Biomedical Engineering*, 57(10):2605–2608, 2010.

[47] J. Flammer, S. Orgl, V. P. Costa, N. Orzalesi, G. K. Krieglstein, L. M. Serra, J. P. Renard, , and E. Stefnsson. The impact of ocular blood flow in glaucoma. *Progress in Retinal and Eye Research*, 21(4):359–393, 2002.

[48] L. Schmetterer and M. Wolzt. Ocular blood flow and associated functional deviations in diabetic retinopathy. *Diabetologia*, 42(4):387–405, 1999.

[49] R.W. Flower. Extraction of choriocapillaris hemodynamic data from icg fluorescence angiograms. *Investigative Ophthalmology and Visual Science*, 34:2720–2729, 1993.

[50] G.T. Feke, D.G. Goger, H. Tagawa, and F.C. Delori. Laser doppler technique for absolute measurement of blood speed in retinal vessels. *IEEE Transactions on Biomedical Engineering*, 34(9):673–800, 1987.

[51] J. Tokayer, O. Tan, and D. Huang. Effect of blood vessel diameter on relative blood flow estimates in doppler optical coherence tomography algorithms. In *Proc. SPIE 7889, Optical Coherence Tomography and Coherence Domain Optical Methods in Biomedicine XV, 78892X*, 2011.

[52] B. J. Vakoc, G.J. Tearney, and B.E. Bouma. Statistical prperties of phase-decorrelation in phase-resolved doppler optical coherence tomography. *IEEE Transactions on Medical Engineering*, 28(6):814–821, 2009.

[53] V.J. Srinivasan, S. Sakadzic, I. Gorczynska, S. Ruvinskaya, W. Wu, J.G. Fujimoto, and D.A. Boas. Quantitative cerebral blood flow with optical coherence tomography. *Optics Express*, 18(3):2477–2494, 2010.

[54] C.E. Riva, J.E. Grunwald, S.H. Sinclair, and B.L. Petrig. Blood velocity and volumetric flow rate in human retinal vessels. *Investigative Ophthamology and Visual Science*, 26:1124–1132, 1985.

[55] H. Ren and X. Li. Clutter rejection filters for optical doppler tomography. *Optics Express*, 14(13):6103–6112, 2006.

[56] M. Szkulmowski, A. Szkulmowska, T. Bajraszewski, A. Kowalczyk, and M. Wojtkowski. High-penetration swept source doppler optical coherence angiography by fully numerical phase stabilization. *Optics Express*, 16(9):6008–6025, 2008.

[57] H. Wehbe, M. Ruggeri, G. Gregori S. Jiao, C. A. Puliafito, and W. Zhao. Automatic retinal blood flow calculation using spectral domain optical coherence tomography. *Optics Express*, 15(23):15193–15206, 2007.

[58] S. G. Proskurin, Y. He, and R. K. Wang. Determination of flow velocity vector based on doppler shift and spectrum broadening with optical coherence tomography. *Optics Letters*, 28(14):1227–1229, 2003.

[59] C. Smith. GoldsteinUnwrap2D_r1. `http://www.mathworks.com/matlabcentral/fileexchange/29497`, 2010. [Retrieved July 1, 2013].

[60] D. C. Ghiglia and M. D. Pritt. *Two-Dimensional Phase Unwrapping: Theory, Algorithms and Software*. Wiley-Interscience, Malabar, Fl, 1991.

[61] K. Itoh. Analysis of the phase unwrapping problem. *Applied Optics*, 21(14):2470, 1982.

[62] R. Goldstein, H. Zebker, and C. Werner. Satellite radar interferometry: Two-dimensional phase unwrapping. *Radio Science*, 23:713–720, 1988.

[63] H. C. Hendargo, R. P. McNabb, A. H. Dhalla, N. Shepherd, and J. A. Izatt. Doppler velocity detection limits in spectrometer-based versus swept-source optical coherence tomography. *Biomedical Optics Express*, 2(8):2175–2188, 2011.

[64] R. K. Wang and Z. Ma. Real-time flow imaging by removing texture pattern artifacts in spectral-domain optical doppler tomography. *Optics Letters*, 31(20):3001–3003, 2006.

[65] Y. J. Hong, S. Makita, F. Jaillon, M. J. Ju, J. Min, B. H. Lee, M. Itoh, M. Miura, and Y. Yasuno. High-penetration swept source doppler optical coherence angiography by fully numerical phase stabilization. *Optics Express*, 20(3):2740–2760, 2012.

[66] H. Ren, T. Sun, D.J. MacDonald, M.J. Cobb, and X. Li. Real-time in vivo blood-flow imaging by moving-scatterer-sensitive spectral-domain optical doppler tomography. *Optics Letters*, 31(7):927–929, 2006.

[67] Y. K. Tao, A. M. Davis, and J. A. Izatt. Single-pass volumetric bidirectional blood flow imaging spectral domain optical coherence tomography using a modified hilbert transform. *Optics Express*, 16(16):12350–12361, 2008.

[68] I. Grulkowski, I. Gorczynska, M. Szkulmowski, D. Szlag, A. Szkulmowska, R. A. Leitgeb, A. Kowalczyk, and M. Wojtkowski. Scanning protocols dedicated to smart velocity ranging in spectral oct. *Optics Express*, 17(26):23736–23754, 2008.

[69] S. Makita, Y. Hong, M. Yamanari, T. Yatagai, , and Y. Yasunoi. Optical coherence angiography. *Optics Express*, 14(17):7821–7840, 2006.

[70] L. An and R. K. Wang. In vivo volumetric imaging of vascular perfusion within human retina and choroids with optical micro-angiography. *Optics Express*, 16(15):11438–11452, 2008.

[71] R. K. Wang, L. An, P. Francis, , and D. J. Wilson. Depth-resolved imaging of capillary networks in retina and choroid using ultrahigh sensitive optical microangiography. *Optics Letters*, 35(9):1467–1469, 2010.

[72] L. Yu and Z. Chen. Doppler variance imaging for three-dimensional retina and choroid angiography. *Journal of Biomedical Optics*, 15(1):016029, 2010.

[73] G. Liu, W. Qi, L. Yu, , and Z. Chen. Real-time bulk-motion-correction free doppler variance optical coherence tomography for choroidal capillary vasculature imaging. *Optics Express*, 19(4):3657–3666, 2011.

[74] J. Fingler, D. Schwartz, C. Yang, and S. E. Fraser. Mobility and transverse flow visualization using phase variance contrast with spectral domain optical coherence tomography. *Optics Express*, 15(20):12636–12653, 2007.

[75] B. J. Vakoc, R. M. Lanning, J. A. Tyrell, T. P. Padera, L. A. Bartlett, T. Stylianopoulos, L. L. Munn, G. J. Tearney, D. Fukumura, R. K. Jain, and B. E. Bouma. Three-dimensional microscopy of the tumor microenvironment in vivo using optical frequency domain imaging. *Nature Medicine*, 15(10):1219–1223, 2009.

[76] Y. Yasuno, Y. Hong, S. Makita, M. Yamanari, M. Akiba, M. Miura, , and T. Yatagai. In vivo high-contrast imaging of deep posterior eye by 1- um swept source optical coherence tomography and scattering optical coherence angiography. *Optics Express*, 15(10):6121–6139, 2007.

[77] Y. Hong, S. Makita, M. Yamanari, M. Miura, S. Kim, T. Yatagai, , and Y. Yasuno. Three-dimensional visualization of choroidal vessels by using standard and ultra-high resolution scattering optical coherence angiography. *Optics Express*, 15(12):7538–7550, 2007.

[78] A. Mariampillai, B.A. Standish, E.H. Moriyama, M. Khurana, N.R. Munce, M.K.K. Leung, J. Jiang, A. Cable, B.C. Wilson, I.A. Vitkin, and V.X.D. Yang. Speckel variance detection of microvasculature using swept-source optical coherence tomography. *Optics letters*, 33(13):1530–1532, 2008.

[79] H. C. Hendargo, R. Estrada, S. J. Chiu, C. Tomasi, S. Farsiu, and J. A. Izatt. Automated non-rigid registration and mosaicing for robust imaging of distinct retinal capillary beds using speckle variance optical coherence tomography. *Biomedical Optics Express*, 4(6):803–821, 2013.

[80] E. Jonathan, J. Enfield, , and M. J. Leahy. Correlation mapping method for generating microcirculation morphology from optical coherence tomography (oct) intensity images. *Journal of Biophotonics*, 4(9):583–587, 2011.

[81] Y. Jia, O. Tan, J. Tokayer, B. Potsaid, Y. Wang, J. J. Liu, M. F. Kraus, H. Subhash, J. G. Fujimoto, J. Hornegger, and D. Huang. Split-spectrum amplitude-decorrelation angiography with optical coherence tomography. *Optics Express*, 20(4):4710–4725, 2012.

[82] J. Tokayer, Y. Jia, A. Dhalla, and D. Huang. Blood flow velocity quantification using split-spectrum amplitude-decorrelation angiography with optical coherence tomography. *Biomedical Optics Express*, 4(10):1909–1924, 2013.

[83] W. Li, C. T. Lancee, E. I. Cespedes, A. F. W. van der Steen, and N. Bom. Decorrelation properties of intravascular echo signals: Potentials for blood velocity estimation. *Journal of Acoustic Society of America*, 71(14):70D–86D, 1993.

[84] W. Li, A. F. W. van der Steen, C. T. Lancee, E. I. Cespedes, and N. Bom. Blood flow imaging and volume flow quantitation with intravascular ultrasound. *Ultrasound in Medicine and Biology*, 24(2):203–214, 1998.

[85] E. I. Cespedes A. F. W. van der Steen W. Li, C. T. Lancee and N. Bom. Decorrelation of intravascular ultrasound signals: A computer simulation study. In *Proc. IEEE Ultrasonics Symposium*, volume 2, pages 1165–1168, 1997.

[86] O. Arend, A. Remky, D. Evans, R, Stber, and A. Harris. Contrast sensitivity loss is coupled with capillary dropout in patients with diabetes. *JAMA*, 38(9):1819–1824, 1997.

[87] J. Zhao, D. A. Frambach, P. P. Lee, M. Lee, and P. F. Lopez. Delayed macular choriocapillary circulation in age-related macular degeneration. *Int. Ophthalmol.*, 19(1):1–12, 1995.

[88] N. M. Bressler. Age-related macular degeneration is the leading cause of blindness. *JAMA*, 291(15):1900–1901, 2004.

[89] R. K. Wang and L. An. Multifunctional imaging of human retina and choroid with 1050-nm spectral domain optical coherence tomography at 92-khz line scan rate. *J. Biomed. Opt*, 16(5):050503, 2011.

[90] D. Y. Kim, J. Fingler, J. S. Werner, D. M. Schwartz, S. E. Fraser, and R. J. Zawadzki. In vivo volumetric imaging of human retinal circulation with phase-variance optical coherence tomography. *Biomed. Opt. Express*, 2(6):1504–1513, 2011.

[91] L. Laatikainen and J. Larinkari. Capillary-free area of the fovea with advancing age. *Invest. Ophthalmol. Vis. Sci.*, 16(12):1154–1157, 1977.

[92] Rohand Weiter. *Retinal and choroidal circulation.* Mosby Elsevier, St. Louis, MO, 2008.

[93] R. H. W. Funk. Blood supply of the retina. *Ophthalmic Res.*, 29(5):320–325, 1997.

[94] G. Liu, L. Chou, W. Jia, W. Qi, B. Choi, and Z. Chen. Intensity-based modified doppler variance algorithm: application to phase instable and phase stable optical coherence tomography systems. *Optics Express*, 19(12):11429–11140, 2011.

[95] G. Liu, W. Jia, V. Sun, B. Choi, and Z. Chen. High-resolution imaging of microvasculature in human skin in vivo with optical coherence tomography. *Optics Express*, 20(7):7694–7705, 2012.

[96] G. Liu, A. J. Lin, B. J. Tromberg, and Z. Chen. A comparison of doppler optical coherence tomography methods. *Optics Express*, 3(10):2669–2680, 2012.

[97] S. Yazdanfar, C. Yang, M. V. Sarunic, and J. A. Izatt. Frequency estimation precision in doppler optical coherence tomography using the cramer-rao lower bound. *Optics Express*, 13(2):410–416, 2005.

[98] M. L. Turgeon. *Clinical Hematology: Theory and Procedures.* Lippincott, Williams and Wilkins, Baltimore, MD, 2005.

[99] C. S. Kim, W. Qi, J. Zhang, Y. J. Kwon, and Z. Chen. Imaging and quantifying brownian motion of micro- and nanoparticles using phase-resolved doppler variance optical coherence tomography. *Journal of Biomedical Optics*, 18(3):030504, 2013.

[100] Y. Jia, J. C. Morrison, J. Tokayer, O. Tan, L. Lombardi, B. Baumann, C. D. Lu, W. Choi, J. G. Fujimoto, and D. Huang. Quantitative oct angiography of optic nerve head blood flow. *Biomedical Optics Express*, 3(12):3127–3137, 2012.

[101] E. Wei, Y. Jia, O. Tan, B. Potsaid, J. J. Liu, W. Choi, J. G. Fujimoto, and D. Huang. Parafoveal retinal vascular response to pattern visual stimulation assessed with oct angiography. *PLOS ONE (submitted).*

[102] C.E. Riva and B. Petrig. Blue field entopic phenomenon and blood velocity in the retinal capillaries. *Journal of Optical Society of America*, 70(10):1234–1238, 1980.

[103] R. Flower, E. Peiretti, M. Magnani, L. Rossi, S. Serafini, Z. Gryczynski, , and I. Gryczynski. Observation of erythrocyte dynamics in the retinal capillaries and choriocapillaris using icg-loaded erythrocyte ghost cells. *Investigatigve Ophthalmology and Visual Science*, 49(12):5510–5516, 2008.

[104] J. Tam, P. Tiruveedhula, , and A. Roorda. Characterization of single-file flow through human retinal parafoveal capillaries using an adaptive optics scanning laser ophthalmoscope. *Biomedical Optics Express*, 2(4):781–793, 2011.

[105] T. Klein, W. Wieser, C. M. Eigenwillig, B. R. Biedermann, and R. Huber. Megahertz oct for ultrawide-field retinal imaging with a 1050nm fourier domain mode-locker laser. *Optics Express*, 19(4):3044–3062, 2011.

[106] B. Potsaid, V. Jayaraman, J. G. Fujimoto, J. Jiang, P. J. S. Heim, and A. E. Cable. Mems tunable vcsel light source for ultrahigh speed 60khz 1mhz axial scan rate and long range centimeter class oct imaging. In *Proc. SPIE 8213, Optical Coherence Tomography and Coherence Domain Optical Methods in Biomedicine XVI, 82130M*, 2012.