DEVELOPMENT OF A DUAL MODALITY SWEPT SOURCE OPTICAL COHERENCE TOMOGRAPHY AND NOVEL FREQUENCY DOMAIN FLUORESCENCE LIFETIME IMAGING SYSTEM USING FIELD PROGRAMMABLE GATE ARRAYS FOR REAL TIME IMAGING OF EX-VIVO HUMAN CORONARY ARTERY ATHEROSCLEROTIC PLAQUES

A Dissertation

by

MICHAEL JOSEPH SERAFINO

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Javier A. Jo
Committee Members,	Brian E. Applegate
	Vladislav V. Yakovlev
	Jim Ji
Head of Department,	Mike McShane

May 2019

Major Subject: Biomedical Engineering

ABSTRACT

Atherosclerosis is the leading cause of morbidity and mortality in the United States. It is a systemic and progressive disease process where the arterial wall thickens through inflammation, oxidative stress, and dyslipidemia. This process can lead to the formation of plaques with fibrous caps and lipid-laden cores. Plaques may rupture and result in myocardial infarction, stroke, or limb injury. Future development of systemic or localized therapies for atherosclerosis will depend on a more detailed understanding of plaque development in-vivo. Optical coherence tomography (OCT) and fluorescence lifetime imaging (FLIM) are optical imaging modalities that have the potential to extract complimentary morphological and biochemical information, respectively, from plaques without exogenous contrast agents. OCT has been used intravascularly in humans for years, and combined intravascular OCT and FLIM imaging was recently demonstrated in a rabbit model.

Due to the relatively high data rates of OCT and FLIM, real time processing is usually not achievable on a typical computer. However, high speed processing is a necessity in facilitating the clinical applications of dual modality OCT and FLIM systems. Field programmable gate arrays (FPGA)s are reconfigurable integrated circuits that are commonly used for data acquisition and processing applications. This work describes a novel implementation of a frequency domain (FD) FLIM system, real time FPGA signal processing algorithms for swept source OCT and FD FLIM, and ex-vivo human coronary artery macrophage classification using FD FLIM.

ACKNOWLEDGMENTS

I would like to thank:

- My parents for paying for everything throughout my undergraduate education. Without them I would likely not have embarked on my graduate studies.
- My graduate advisor Dr. Javier Jo, and my committee member Dr. Brian Applegate for providing funding support and allowing me to partake in the work detailed in this document.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

Significant contributors are listed below in no particular order.

- Chapters 2, 3, and 4
  - Postmortem human coronary arteries: Dr. Jessee Adame MD and Candi (Arlington National Funeral Home, Houston TX)
  - Histological processing of human coronary arteries: Betsy Molinari and Pamela Potts (Texas Heart Institute, Houston TX)
- Chapters 2, 3
  - Interpretation of CD68 results for classification purposes: Dr. Javier Jo
- Section 5.14.2
  - Development of the mathematical model for the 4 DIT FFT which was also the basis for the 2 DIT FFT in section 5.7: Dr. Paritosh Pande
- Providing lab space in the Houston TX area: Dr. Brian Walton and Rosalva Munoz
- Guidance on the subject of TD FLIM: Dr. Javier Jo
- Guidance on the subject of OCT: Dr. Brian Applegate
- Guidance on optical alignment of TD FLIM and SS OCT systems, and handling of postmortem human coronary arteries: Dr. Sebina Shrestha

All other work conducted for this dissertation was completed by the student independently.

Funding Sources

This work was funded by NIH grants 1R01HL111361 and 1R21CA132433, and AHA Beginning Grant-in-Aid 0765102Y.

NOMENCLATURE

9CA	9-Anthracenecarbonitrile
AA7	Abbreviation for arty atrix-7, a model of FPGA made by Digilent Inc
AC	Alternating current, in this work referring to AC coupled electronics that act as high pass filters
ADC	Analog to digital converter
AI	Analog input
ANSI	American National Standards Institute
ANT	Abbreviation for anthracene
APD	Avalanche photodiode
API	Application programming interface
ASIC	Application specific integrated circuit
AUC	Area under the curve, usually in reference to ROC curve
AVI	Audio video interleave
BG	Background, usually referring to background subtraction
BPF	Bandpass filter
BW	Bandwidth
CD68	Cluster of differentiation 68, a protein highly expressed by monocytes; also referring to immunostain used to detect the presence of CD68
CL	Classification line
CP	Classification path

CP-OCT	Classification path for OCT
CPS	Classification path segment
CPU	Central processing unit, in reference to a computer processor
CSV	Comma separated value
CW	Continuous wave, usually referring to laser output, as opposed to pulsed laser output
DAC	Digital to analog converter
DAQ	Data acquisition
DC	Direct current, sometimes referring the 0 Hz content of an electrical signal
DDS	Direct digital synthesis
DEP	Data extent path
DFD	Digital frequency domain, an implementation of FLIM
DFT	Discrete Fourier Transform
DIT	Decimated in time, usually involving computing a DFT by splitting the input data into several sub-streams
DM	Dichroic mirror, an optic that can reflect certain wavelengths of light and transmit others
DMA	Direct memory access
DPA	9,10-Diphenylanthracene
DRAM	Dynamic random access memory
DSP	Digital signal processor, see also DSP48
DSP48	An arithmetic logic unit embedded into Xilinx FPGA fabric, used to perform specific math operations more efficiently than general purpose FPGA fabric
DSP48E1	A specific type of DSP48 slice

DUT	Abbreviation for duty cycle
DV	Data valid
EFL	Effective focal length
FAD	Flavin adenine dinucleotide
FD	Frequency domain, usually in reference to a FLIM system
FD-FLIM	Frequency domain fluorescence lifetime imaging
FDF	Abbreviation for FDFLIM
FDFLIM	Frequency domain fluorescence lifetime imaging
FDOCT	Frequency domain OCT, usually when the detector is a line camera/spectrometer and the light source is continuous wave (i.e. not swept source)
FFT	Fast Fourier Transform, a more computationally efficient implementation of the DFT
FIFO	First in first out
FIR	Finite impulse response, usually referring to a type of digital filter
FLIM	Fluorescence lifetime imaging
FMC	FPGA Mezzanine Card
FPGA	Field programmable gate array
FWHM	Full width at half maximum
GNU	GNU is Not Unix
GPGPUs	General purpose computing on GPUs
GPU	Graphics processing unit
GS/s	Abbreviation for giga-samples per second (i.e. 10e9 samples per second)
GUI	Graphical user interface

HDD	Hard disk drive
HDL	Hardware description language
IDE	Integrated development environment, usually referring to computer software that aids in development of computer software
IP	Intellectual property
ISE	Integrated synthesis environment, referring to an IDE created by Xilinx for FPGA development
JPEG	Joint Photographics Experts Group, although JPEG usually refers to a file format for saving digital images
LDL	Low density lipoprotein
LED	Light emitting diode
LEP	Lumen extent path
LOAO	Leave one artery out, referring to a type of LOO cross validation performed in this work
LOO	Leave one out, referring to a type of cross validation for assessing classifier performance
LOSO	Leave one subject out, referring to a type of LOO cross validation performed in this work
LSB	Least significant bit
LUT	Lookup table, usually referring to logic primitives on FPGAs
LUTRAM	Lookup tables used for implementing RAM on FPGAs
LabVIEW	Laboratory virtual instrument engineering workbench, a graphical programming language developed by NI
MAC	Multiply accumulate
MATLAB	Proprietary programming language and computing environment developed by MathWorks Inc

MB	Abbreviation for mega-byte, i.e. 1e6 bytes
MB/s	Abbreviation for MB per second, referring to a data transfer rate
MHz	Abbreviation for mega-Hertz, i.e. one million Hertz
MMF	Multi-mode optical fiber
MOVATS	Pentachrome stain used to highlight various cellular components during histology analysis
MPE	Maximum permissible exposure, usually referring to human eye or skin exposure to laser radiation
MS/s	Abbreviation for mega-samples per second, usually referring to ADC sampling rate
MSB	Most significant bit
MZI	Mach Zehnder interferrometer
MZI-clock	Referring to an MZI that is sometimes used for calibrating a SS OCT laser
NADH	Nicotinamide adenine dinucleotide
NI	National Instruments Corporation
NIH	National Institutes of Health
OCT	Optical coherence tomography
OD	Outer diameter
ODL	Optical delay line, a component commonly used in OCT systems
OFDI	Optical frequency domain imaging, another way of saying OCT
PBS	Phosphate buffered saline, a solvent used for some fluorescent dyes
PCI	Peripheral component interconnect

PCIe	PCI express
PFI	Programmable function input, referring to a general purpose digital input or output line on some NI hardware
PLL	Phase locked loop
PMT	Photomultiplier tube
PNG	Portable network graphics
POPOP	1,4-Bis(5-phenyl-2-oxazolyl)benzene
PXI	PCI Express Extensions for Instrumentation
PXIe	PXI express
RAID	Redundant array of independent disks
RAM	Random access memory
RM	Removable mirror
ROC	Receiver operating characteristic, a type of performance assessment that can be computed for binary classifiers
RS-232	Recommended standard 232, nowadays synonymous with serial port communication
SD-OCT	Another name for FD-OCT
SDOCT	Another name for FD-OCT
SMA	Subminiature version A, referring to a type of electrical or a type of optical connector (different connectors but same general name)
SNR	Signal to noise ratio
SPIE	Society of photographic instrumentation engineers
SS	Swept source, referring to a type of light source used in OCT systems
SSD	Solid state drive
SSOCT	Term for SS OCT

SVG	Scalable vector graphics
TCSPC	Time correlated single photon counting
TD	Time domain, usually in reference to a FLIM system
TDF	Time domain fluorescence lifetime imaging
TDFLIM	Time domain fluorescence lifetime imaging
TTL	Transistor-transistor logic
USB	Universal serial bus
USB2.0	A specific version, 2.0, of USB
UV-NIR	Referring to light wavelengths between the ultraviolet and infrared
VCSEL	Vertical cavity surface emitting laser
VHDL	VHSIC hardware description language, VHSIC meaning very high speed integrated circuit
VIPS	Referring to the VIPS open source image processing software package

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION

1.1 Dissertation outline

This dissertation is organized into four main chapters.

- Chapter 2 describes a novel frequency domain (FD) fluorescence lifetime imaging (FLIM) system that utilizes field programmable gate arrays for real time processing of lifetime data in the nanosecond range.

- Chapter 3 compares the FD FLIM system against our previously developed time domain (TD) FLIM system. Comparisons were performed by imaging 87 ex-vivo human coronary arteries from 23 human subjects with both FLIM systems. The lifetime maps were compared by computing correlation coefficients between the TD and FD FLIM lifetime maps, and by comparing the CD68 (immunostain for macrophages) classification performance of each FLIM system.

- Chapter 4 describes a dual modality swept source (SS) optical coherence tomography (OCT) and FD FLIM system, and explains additional synchronization concerns relative to our previous generation SS OCT and TD FLIM system. An additional 63 ex-vivo human coronary artery segments from 20 human subjects were imaged with the SS OCT and FD FLIM system, but have yet to be evaluated against histology.

- Finally, chapter 5 details an implementation of SS OCT processing on an FPGA intended for intravascular systems with 2 polarization channels. However, the processing design was only validated on a single channel system, and will likely require improvements in the interpolation processing stage before it is ready for intravascular imaging.

1.2 Background

This section will give a brief overview the theory of TD and FD FLIM, and SS OCT.

Single photon FLIM theory is well documented and understood [1]. In general, FLIM systems can be classified as either TD or FD. Both TD and FD can be implemented in either wide field or point configurations. Wide field systems typically use a camera to acquire multiple spatial points simultaneously, while point configurations acquire data from one spatial location in a serial fashion. This section will focus only on point configurations as they are most relevant to fiber optic based imaging probes used for cardiovascular imaging.

TD implementations strive to acquire the impulse response of a fluorophore as a function of time. Several sub-categories exist: time correlated single photon counting (TCSPC), gated detection, and pulse sampling [2]. This work involves a system that uses the pulse sampling method, which involves exciting the sample with a nanosecond (in our application of atherosclerosis) pulse and digitizing the fluorescence emission pulse as a function of time as depicted in figure 1.1. In the case of human coronary arteries, the main endogenous fluorophores of interest are collagen, elastin, and lipid [3].

FD implementations ultimately acquire the frequency response of a target fluorophore, i.e. its demodulation and phase delay as a function of frequency. The frequency response of a fluorophore can then be used to compute its impulse response if desired [1]. Figure 1.2 shows a simple illustration of an excitation and emission waveform if one excitation frequency was used. However, multiple modulation frequencies are required if resolution of multiple decay components (i.e. multiple lifetimes in the same sample) is needed.

The implementation of swept source (SS) optical coherence tomography (OCT) utilized in this work is summarized in figure 1.3. A more comprehensive overview of OCT can be found in [4].

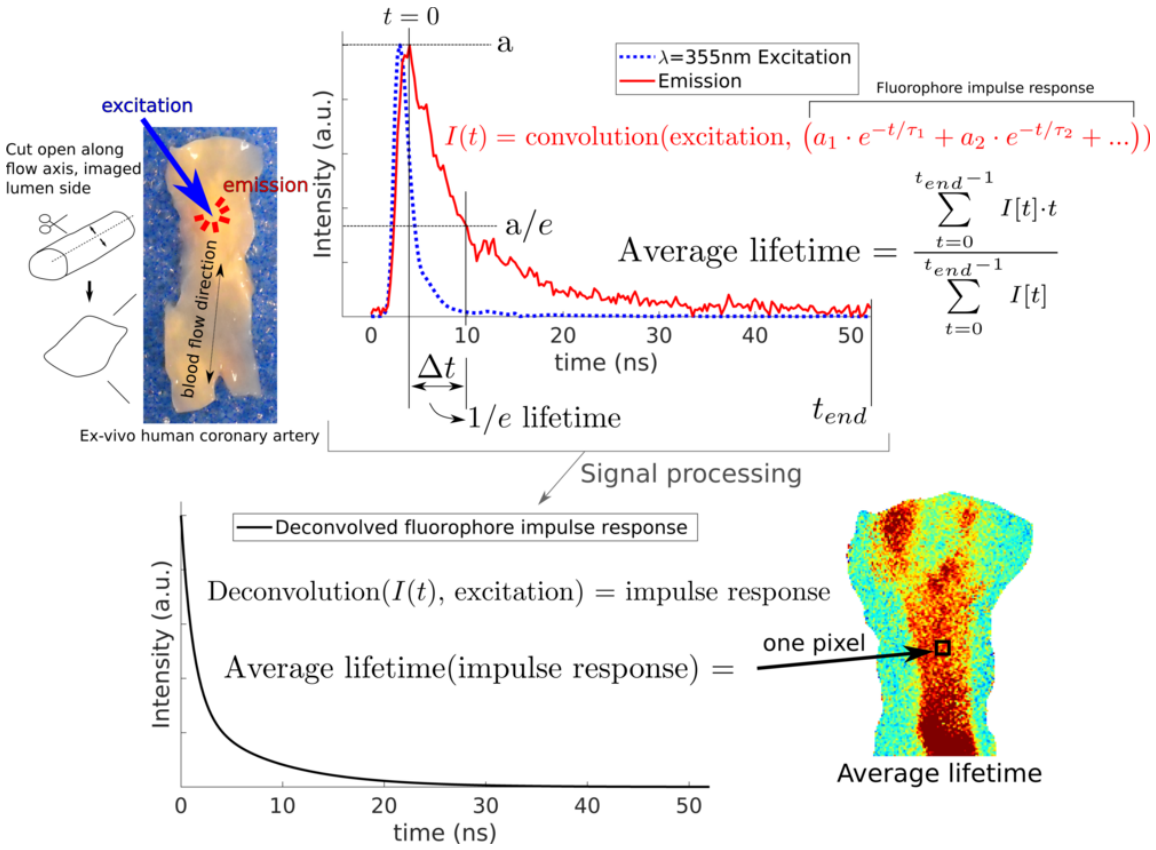

Figure 1.1: Simplified overview of TD FLIM theory for the pulse sampling implementation. A pulse of light is directed to a sample which induces fluorescence emission (ex: Elforlight SPOT-10-100-355 laser with 355nm wavelength, <1.5ns pulse width, and 1μJ per pulse). The emission is collected and digitized as a function of time. The excitation and emission are denoted by the blue and red curves, respectively. At this stage, multiple signal processing approaches can be used depending on the application. Processing can be as simple as integrating the emission to calculate the intensity and calculating the average lifetime using the equation shown (upper right). However if more accurate lifetime measurements are desired, more advanced processing techniques such as deconvolution need to be used. These approaches involve removing the system response (the laser pulse width is not infinitely small, and the detector response is not infinitely fast) from the detected emission.

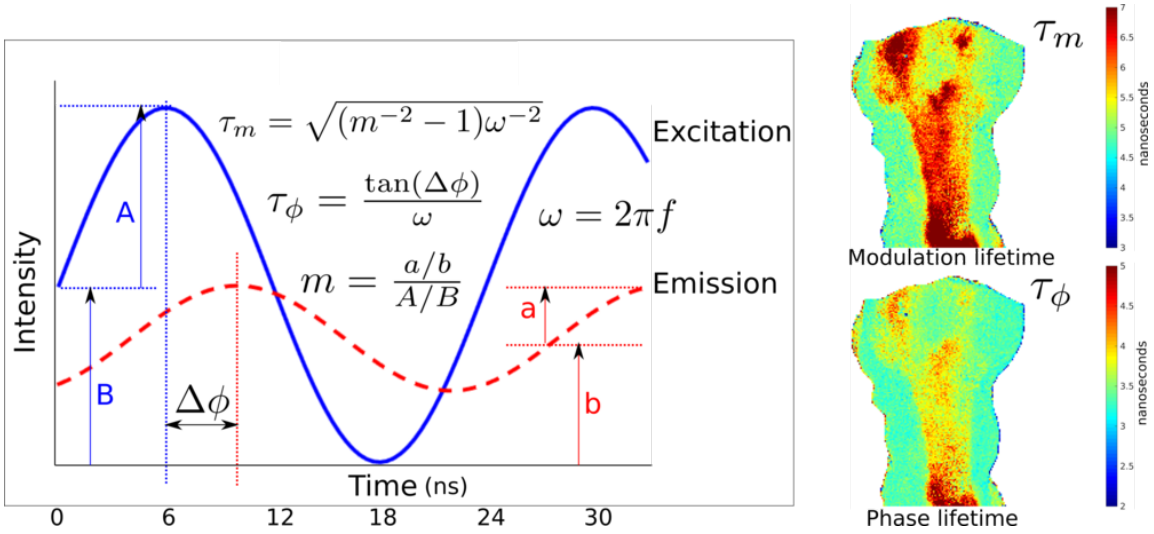

Figure 1.2: Simplified overview of FD FLIM theory. The sample is excited with a modulated light source (blue sine wave), and the resulting emission occurs at the same frequency with some demodulation m (reduction in amplitude) and phase delay ϕ. The apparent modulation $\tau_m(w), w = 2\pi f, f =$ frequency Hz and phase $\tau_\phi(w)$ lifetimes can be calculated at each individual modulation frequency f. The bottom of this figure depicts the approach used in 2 where an FPGA computes the DFT at each modulation frequency. If the exponential components of the sample are desired rather than $\tau_m(w)$ and $\tau_\phi(w)$, more advanced processing methods are needed (similar to deconvolution for TD FLIM) to fit a multi-component model to a given set of $m(w)$ and $\phi(w)$ frequency domain measurements.

The usual first step in SS OCT processing is background subtraction. This involves subtracting any terms that are not sample dependent. The background term can be obtained by blocking the sample arm and collecting raw data, averaging the raw data, and then subtracting the average from future data. In some cases the background term can fluctuate over time, so periodic re-acquisition may be needed during prolonged imaging sessions.

After background subtraction, linearization is performed. This is a process of correcting for the SS OCT laser's non-linear (in k-space) sweep pattern. The Fourier transform makes the assumption that the input data is equally spaced in time (or frequency). Not performing this correction can result in significant artifacts (see section 5.13.1). Usually the raw A-line data is interpolated to extract a version of the spectrum that is equally spaced in k-space. Typically a reference interferogram (referred to as an mzi-clock or k-clock in this document) is acquired alongside the raw A-line data. The mzi-clock is essentially a separate interferometer with a mirror at its sample position. This results in an output interferogram that has a fixed path length (therefore the resulting interferogram should be linear). The mzi-clock interferogram is then used to infer the correct sampling required to linearize the back reflected interferogram from SS OCT system's sample position. Some SS OCT lasers exist that provide a known sweep pattern and known valid points (Insight SS OCT lasers), that reduce the linearization problem to a point-discarding problem, greatly reducing the computational burden of this processing step.

Dispersion correction is performed by first acquiring an interferogram with a mirror at the sample position. Any further non-linearity in the raw A-line data after linearization is performed can be attributed to dispersion. For example, the hypothetical system in figure 1.3 has an objective lens in the sample arm but not in the reference arm. The wavelength dependent index of refraction results in certain wavelengths of the back-reflected sample arm light being delayed relative to the reference arm light when they arrive at the detector. A process similar to that used in linearization is employed to find a complex correction fac-

tor (complex number multiplied by the raw data before the Fourier transform) that makes the mirror's interferogram linear in k-space. Alternatively, dispersion can be corrected for by non-software means by physically inserting glass into the reference arm to manually account for dispersion.

The Fourier transform is then applied to the complex dispersion corrected data. In the SS OCT system described here, only the magnitude of the Fourier transform is used to obtain a 3D morphological image of the sample.

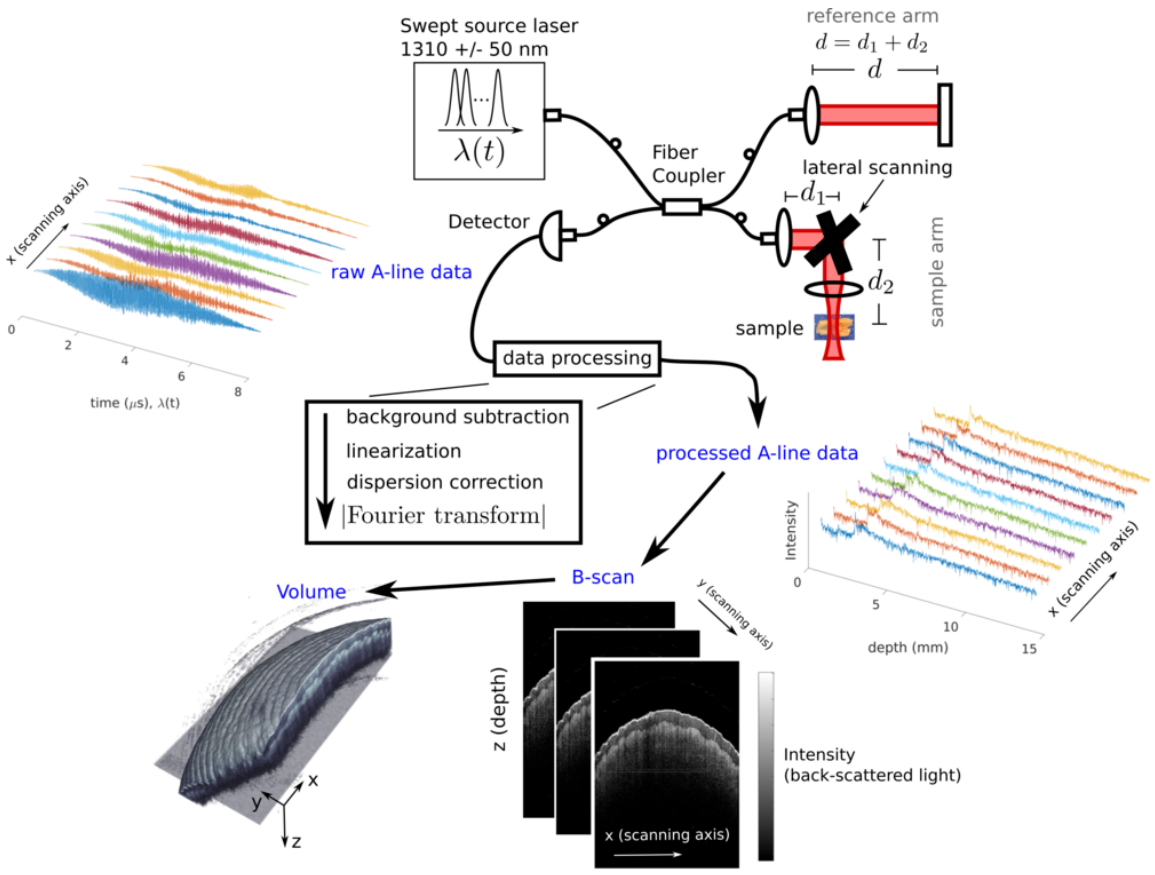

Figure 1.3: Simplified overview of SS OCT theory. A low coherence light source is split into two paths (a.k.a. arms). The light source used for SS OCT 'sweeps', i.e. changes, its output as a function of time in a repeatable fashion. The sample arm of the interferometer (a more general term for the SS OCT system), is directed to the sample position. The sample then reflects light based on differences in its optical properties (refractive index). The back reflected light that is collected by the objective lens is directed back to the detector. At the same time, the reference arm light has traveled to the mirror at the end of the reference arm (top right) and back to the detector. The resulting interference pattern incident on the detector generates an output signal as a function of incident light intensity, and time (therefore wavelength as the laser is sweeping its output). The data processing part of the figure represents the combination of data acquisition card and a host computer. The raw A-line data is processed to give depth profiles that contain the intensity of back reflected light as a function of depth for a given lateral position on the sample. Multiple A-lines are assembled into a B-scan, and multiple B-scans into a volume (sometimes referred to as C-scan). The resulting 3D volume is then used for further other application specific purposes (diagnostics, etc.).

2. NANOSECOND POINT SCANNING FREQUENCY DOMAIN FLUORESCENCE LIFETIME IMAGING USING FIELD PROGRAMMABLE GATE ARRAYS FOR REAL TIME PROCESSING AT MULTIPLE SIMULTANEOUS MODULATION FREQUENCIES AND SPECTRAL BANDS

2.1 Introduction

Typical FDFLIM system implementations employ heterodyne or homodyne detection using a lock-in amplifier (or equivalent electronics) and require that the detector's gain be modulatable and in some way synchronized to the modulated excitation light source [5, 6, 7, 8, 9, 10]. This approach typically provides the fastest pixel rates when used with a camera based imaging system. However, an array detector approach cannot be used in applications where the dimensions of the array detector are prohibitive, such as in intravascular fluorescence lifetime imaging of human coronary arteries which have diameters of just a few millimeters. In such applications, a point scanning method is typically used whereby each spatial pixel is acquired serially rather than in parallel. Point scanning heterodyne or homodyne detection was economical when 100 MHz bandwidth electronics were cost prohibitive as it allows for detection of higher frequency modulation and phase shifts at lower sampling frequencies. Nowadays, however, the cost of higher bandwidth electronics has decreased, making the increased complexity associated with point scanning heterodyne and homodyne methods less justified. As higher bandwidth electronics became more accessible, other excitation and detection schemes have more recently emerged, in which simultaneous measurement of a sample of the excitation light source is required as a reference or trigger [11, 12]. This approach, however, also increases FDFLIM implementation cost, complexity, and physical size by requiring an extra analog input channel and maintenance of a scattering solution, or alignment of a reflector. Most recently proposed

FDFLIM implementations do not use heterodyne or homodyne detection, or require acquiring a reference. For instance, the authors of [13] implemented an FDFLIM system that required two analog inputs, one for a reference signal, and one for the fluorescence emission. Their light source was internally modulated and provided an output signal that was used as a reference. They utilized a sampling rate (2 GS/s) much higher than the applied modulation frequency (10 - 60 MHz) and averaged 128 acquisitions per measurement to improve signal quality. However, this strategy would not scale well to real-time FDFLIM imaging applications due to the lack of real-time processing and need for digitization of a reference signal. The authors of [14] use an FPGA and refer to their system as digital frequency domain FLIM, and their approach resembles a mixture of heterodyning and time correlated single photon counting. A given measurement consisted of building a histogram of discrete photon arrival events using a cross correlation approach implemented on an FPGA using detectors capable of single photon counting. They then applied phasor analysis to the histogram to obtain phase and magnitude data for various frequencies. Although the aforementioned facts do not affect the practical applications of their method, we felt the need to make those distinctions as they relate to our claims of novelty because our work also uses the terms frequency domain FLIM and involves FPGAs. The work described in [15] also implemented a photon counting approach, but acquired sinusoidal waveforms instead of histograms from the fluorescence emission. While our method lacks the sensitivity of the single photon counting approaches [14, 15], it is cheaper and simpler implement, and most importantly is capable of imaging clinically relevant ex-vivo human coronary arteries at multiple modulation frequencies simultaneously.

In this work we demonstrate that our approach is capable of simultaneous detection of multiple spectral emission bands, real time processing, and detection using lower cost fixed gain avalanche photodiodes instead of photomultiplier tubes. The implementation of our FDFLIM system is summarized as follows. The sample was excited with a CW

diode laser that was modulated with a digital pulse train generated with an FPGA. Three emission bands were simultaneously detected by fixed gain avalanche photodiodes, digitized at 250 MS/s, and processed at 251.42 MS/s. The laser digital pulse train, digitizer sample clock, and digitizer trigger were synchronous to the same clock. This synchronization resulted in very low time jitter between successive pixels (< 10 ps), which simplified instrumentation, system calibration, and data processing. Synchronization was achieved using commonly available functionality in FPGAs, namely PLLs and reference clocks. This made the synchronization of the system internal to the FPGAs, and did not require any special detection scheme or extra equipment. External clock synchronization mechanisms are available even on lower end ($100) FPGA evaluation boards, as they are essential components in a wide array of other applications. To the best of our knowledge, such an FPGA based multi-spectral, multi-frequency, and real-time processing FDFLIM system has not been previously reported.

2.2 Background

FDFLIM theory is well documented and understood [16]. A fluorescent sample is first excited by a light source modulated at specific frequency(s). The sample then emits fluorescence at the same frequency(s), but with a phase delay and reduced amplitude. The fluorescence lifetime is then calculated using the observed phase delays and reduced amplitudes (relative to the modulated light excitation) at each excitation frequency. In the FDFLIM system discussed here, the fluorescence was detected and digitized as a function of time, $d_{(t,x,y,\lambda)}$. The discrete Fourier transform was then applied to give $D_{(x,y,\lambda,f)}$. The sample's fluorescence lifetime was obtained by computing the relative phase and modulation at the aforementioned frequency(s). The phase τ_p and modulation τ_m lifetimes were calculated using the equations below. The notation (t, x, y, λ) or (x, y, λ, f) denotes emission band λ, image spatial coordinates x, y (assuming 2D image), modulation frequency f

(Hz), and time t (s). The $\phi_{(x,y,\lambda,f)}$ and $m_{(x,y,\lambda,f)}$ terms denote the relative phase delay and modulation of the observed fluorescence emission, respectively.

$$\text{Fluorescence intensity } = I_{(x,y,\lambda,f)} \tag{2.1}$$

Modulation of fluorescence (2.2)

$$m_{(x,y,\lambda,f)} \propto \frac{I_{(x,y,\lambda,f)}}{I_{(x,y,\lambda,f=0)}}$$

$$\text{Phase delay of fluorescence } = \phi_{(x,y,\lambda,f)} \tag{2.3}$$

Phase lifetime (2.4)

$$\tau_{p,(x,y,\lambda,f)} = \frac{1}{2\pi f}\tan(\phi_{(x,y,\lambda,f)})$$

Modulation lifetime (2.5)

$$\tau_{m,(x,y,\lambda,f)} = \sqrt{((m_{(x,y,\lambda,f)})^{-2} - 1)(2\pi f)^{-2}}$$

Usually the physical implementation of the FDFLIM system prohibits directly measuring the absolute values of $\phi_{(x,y,\lambda,f)}$ and $m_{(x,y,\lambda,f)}$ required to calculate τ_p and τ_m. Typically, a reference fluorophore with a known lifetime is used to calibrate the FDFLIM system (for example, [17]). By imaging a reference fluorophore with a known lifetime and decay kinetics (preferably mono-exponential for simplicity), correction factors can be computed to obtain absolute values of $\phi_{(x,y,\lambda,f)}$ and $m_{(x,y,\lambda,f)}$. These calibration procedures are specific to the particular implementation of the FDFLIM system. We discuss the calibration procedures for our system in more detail later in section 2.3.2.

The lifetimes of the fluorophores of interest dictate the required modulation frequencies for an FDFLIM system. Collagen, elastin, and lipid are the main fluorophores of interest when imaging atherosclerotic plaques in human coronary arteries. These fluorophores have average lifetimes on the order of 2 ns to 10 ns [3]. The optimal modulation

frequency can be calculated as explained in [8] and shown below. Although these optimal frequencies are for monoexponential decays, they still provide a ballpark estimate for fluorophores with more complex fluorescence decays.

Optimal frequency for phase measurement: (2.6)

$$f_{opt,\phi}$$

Optimal frequency for modulation measurement: (2.7)

$$f_{opt,m}$$

Ideal SNR, $\tau \in [2, 10]$ ns (2.8)

$$f_{opt,\phi} = \frac{1}{2\pi\tau} \rightarrow [80,\ 16] \text{ MHz}$$

$$f_{opt,m} = \frac{\sqrt{2}}{2\pi\tau} \rightarrow [113,\ 23] \text{ MHz}$$

Non-ideal SNR, $\tau \in [2, 10]$ ns (2.9)

$$\hat{f}_{opt,\phi} = \frac{0.7}{2\pi\tau} \rightarrow [56,\ 11] \text{ MHz}$$

$$\hat{f}_{opt,m} = \frac{1.4}{2\pi\tau} \rightarrow [111,\ 22] \text{ MHz}$$

2.3 Materials and Methods

The FDFLIM system is explained in terms of the optics and electronics, and data processing as shown in figures 2.2 and 2.4, respectively. Optical components are shown in figure 2.2A, and the FPGA synchronization scheme is shown in figure 2.2B. Both FPGA1 and FPGA2 were used with a PXIe chassis (National Instruments PXIe-1082) for interfacing with the host computer over the PCIe bus using the NI PXIe-PCIe8388 and NI PXIe-PCIe8389 controller. The analog output card (National Instruments PXI-6713) used for controlling the galvonometer scanning mirrors was also connected to the PXIe chassis,

but is not shown in figure 2.2 for brevity. The host computer consisted of an ASRock X79 Extreme6 (motherboard), 56 GB of RAM (7x8 GB Corsair CMZ64 GX3M 8A18 66C9), and an Intel i7-4820k 3.70 GHz processor. The custom top level FPGA designs on both FPGA1 and FPGA2 were implemented in LabVIEW FPGA (LabVIEW 2014 and Xilinx tools version 14.7), and various lower level modules in the design were implemented in VHDL. The custom host computer and user interface was written in LabVIEW 2014. Data processing, as described in section 2.3.2, was performed in two parts. First, the discrete Fourier transform (DFT) was calculated on the FPGA, and then the system corrections and lifetime calculations were applied in Matlab on the host computer.

2.3.1 Instrumentation

The sample was excited by a digitally modulated 375nm CW diode laser (Toptica iBeam Smart 375). The laser was modulated by FPGA1 (National Instruments NI 5772 digitizer and PXIe 7966 FPGA combination). The 375nm excitation light was reflected off of DM1 and scanned across the sample using a pair of galvonometer mirrors (Cambridge Tech. 6220H), and an objective lens L2 (Edmund Optics 64-837, 45mm EFL, MgF_2 coated, UV-NIR corrected triplet). The 1/e beam width at the objective focal plane was measured to be 18.5μm using a Thorlabs beam profiler BP104-VIS. The excitation power at the sample position was 20mW when the 375nm laser was operating in CW mode. During imaging, an average duty cycle between 60-73% and pixel rate of 12.5kHz were used, resulting in 0.96μJ to 1.17μJ of energy per pixel. Figure 2.1 shows an example of the digitized fluorescence emission for one pixel as a function of time and frequency.

The maximum permissible exposure (MPE) for skin was calculated according to the guidelines for the American National Standards Institute (ANSI) for Safe Use of Lasers. Both thermal and photochemical MPEs were considered. Using the 12.5 kHz pixel rate, the thermal MPE was calculated as $0.56 \times 0.272^{0.25} = 405$mJ, where 0.272 was the imag-

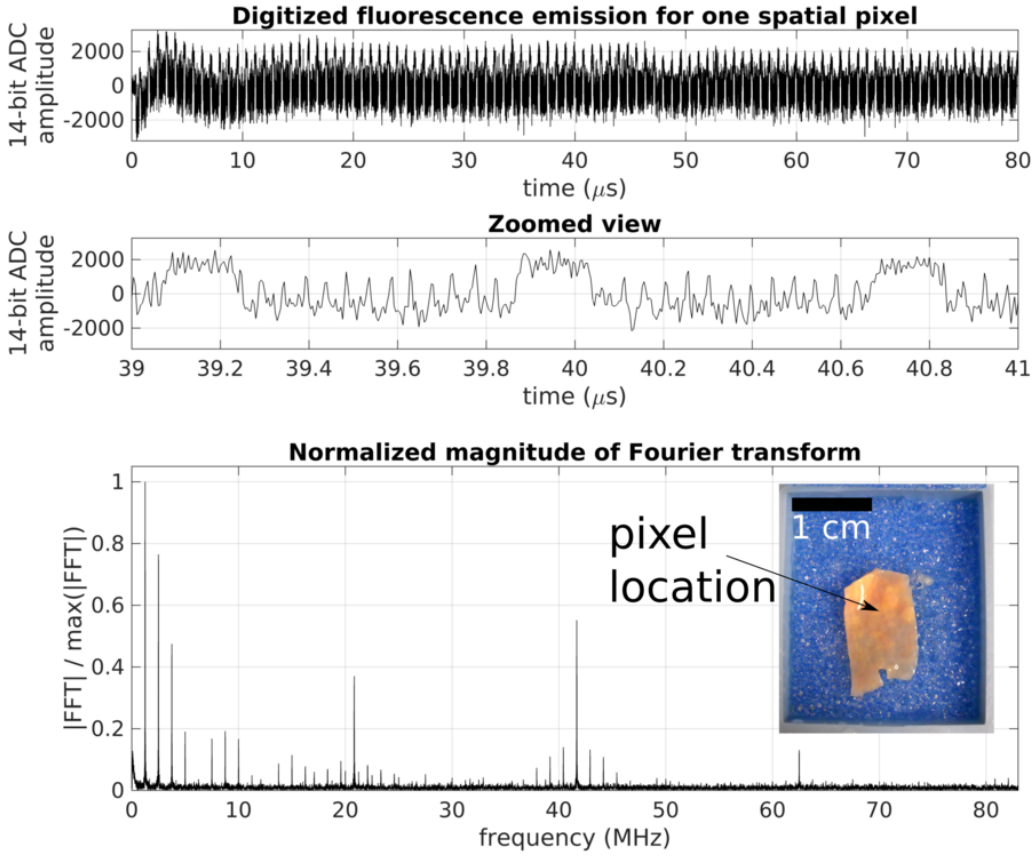

Figure 2.1: Example of the digitized (250 MS/s) fluorescence emission for one spatial pixel (no averaging) as a function of time and frequency. The digital modulation signal used in this example consisted of a pulse train with 3 fundamental frequencies (1.25 MHz, 20.83 MHz, and 41.67 MHz). Raw time domain data was acquired and transferred to the host computer. Then the time domain data (top row) was zero padded to 65536 points and the fast Fourier transform (FFT) was computed. The bottom plot shows the magnitude spectrum of the FFT normalized by its maximum value. The pixel was acquired from an ex-vivo human coronary artery (shown over the right of the FFT plot) as described in section 2.3.4.

ing time in seconds, assuming no flyback dead time. Any flyback dead time would increase the total imaging time without increasing the exposure, making the assumption of no dead time the most conservative case. The photochemical MPE under the system exposure conditions was 1.0J. The exposure over a 3.5mm limiting aperture was calculated by multiplying the energy per pixel by the number of pixels in the 3.5mm aperture assuming 60um pixel spacing. The number of pixels in the 3.5mm aperture was: $\pi[(3.5\text{e}{-3}]/2)^2/[\pi((60\text{e}{-6})/2)^2] = 3403$. The exposures over the limiting 3.5mm aperture for the duty cycle ranges described earlier were 3.2mJ to 4.0mJ, over 2 orders of magnitude below the thermal and photochemical MPEs.

Images were acquired by raster scanning the ganvonometer mirrors, the mirrors were voltage stepped for each pixel. The fluorescence emission was collected by the objective lens, transmitted through DM1, coupled into a 200μm core multi-mode fiber (Thorlabs - M25L02 - 200), collimated, and separated into 3 emission bands by DM2-4 and BPF1-3. The resulting emission bands were 405/20nm, 440/40nm, and 525/50nm (center/FWHM). Each emission band was coupled using a fixed fiber collimator (Thorlabs F220SMA-A) into a 200μm multimode fiber, and directed onto a fixed gain APD (Hamamatsu C12702-11). The gain for each APD was optimized on an ex vivo human coronary artery segment and remained fixed for all subsequent measurements. The output of each multi-mode fiber was placed in close proximity ($<$1mm) to the active area of each APD using a custom adapter plate, thus no coupling lens was needed between each multi-mode fiber and APD. The output of each APD was connected in series to two fixed gain amplifiers (Minicircuits ZFL-500LN+) and one low pass filter (Minicircuits BLP-90+). The output of each low pass filter was connected to an analog input channel on FPGA2 (National Instruments NI 5761 digitizer and PXIe 7962 FPGA combination).

Figure 2.2B describes the FPGA synchronization scheme. FPGA1 outputted a digital pulse train (denoted as the digital modulation signal) to the digital input of the laser.

Modulation frequencies were generated on FPGA1 using counters implemented in general purpose FPGA fabric. The frequency, duty cycle, and delay for each frequency were set by resetting the counter at a specific count, comparing the current count to a fixed value, and starting the counter at a fixed value, respectively. FPGA1 also outputted a trigger and 10 MHz reference signal to FPGA2. All outputs of FPGA1 were generated in the same 250 MHz clock domain. All outputs (10 MHz ref, Trig., and digital modulation signal) were routed out of from FPGA1 using the auxiliary input/output connector of the NI 5772 adapter module through a screw terminal block (National Instruments SCB-19). The names of the output ports (PFI 0, PFI 1, etc.) are included for reference. The ADC on FPGA2 was configured to lock its internal sample clock (operating at 250 MHz) to the external 10 MHz reference from FPGA1. A feedback path exists through the PXIe chassis from FPGA2 to FPGA1 as shown in figure 2.2B. This allowed FPGA2 to disable the digital modulation signal outputted from FPGA1 when data was not being acquired to limit exposure of the sample to the 375nm diode laser.

2.3.2 Data Processing

FDFLIM data must be calibrated for system effects before applying equations (2.4)-(2.5) to obtain the phase and modulation lifetimes. System effects consisted of unwanted background signal, non-flat modulation frequency power spectrum, and phase delays due to optical and electrical path lengths. System effects were removed using data from a reference image that contained a blank (area in the image with no fluorescence) and a fluorophore with a known lifetime. Typically the blank and known fluorophore were located in the same image for simplicity, but they could also be acquired separately if desired. This reference image(s) was then used to calibrate all future images. Spatial averaging was not performed, each pixel was the result of a single measurement.

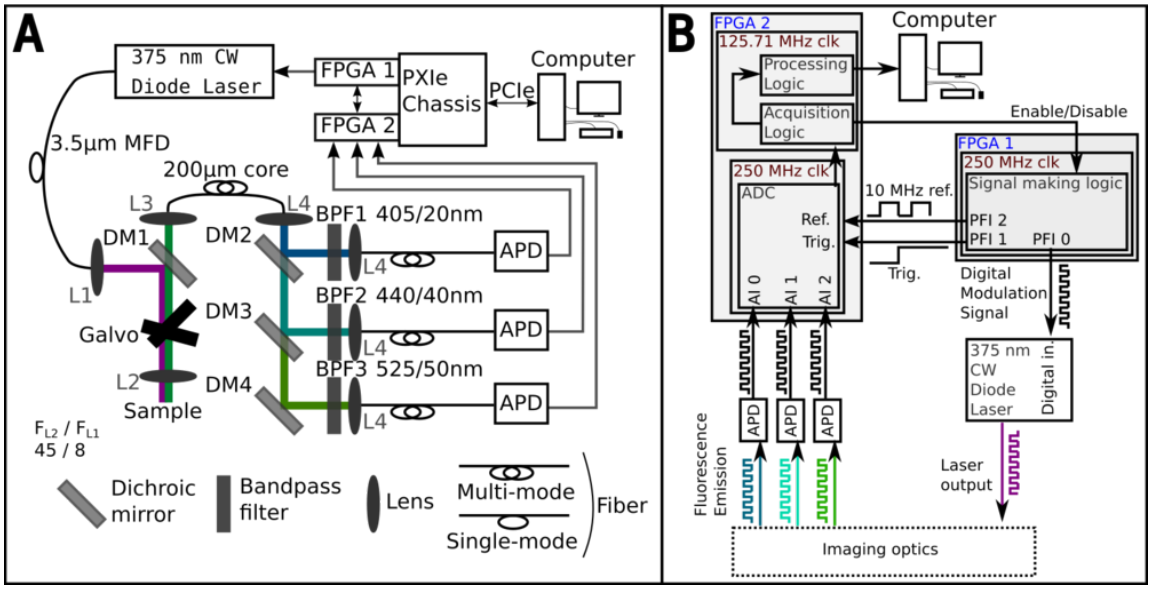

Figure 2.2: General schematic of system (A), and schematic of FPGA synchronization scheme (B). FPGA1 (NI 5772 and PXIe 7966) was configured to generate a 10 MHz clock and a pixel trigger for FPGA2. The modulated laser output was collimated by L1, reflected by DM1, and focused on the sample by L2. The fluorescence emission collected by L2 was focused by L3 into a 200µm core multimode fiber, and separated into 3 spectral bands by DM2-4 and BPF1-3. The center wavelength and FWHM (center/FWHM nm) are shown in panel (A). The output of each DM and BPF pair was focused into a 200µm core multimode fiber by L5-7. The output of each multimode fiber was SMA connectorized and secured in close proximity (<1 mm) to the active area of each APD using custom adapter plates. The output of each APD was connected serially to two fixed gain amplifiers (Minicircuits ZFL-500LN+) and one low pass filter (Minicircuits BLP-90+). The resulting fluorescence signals were digitized at 250 MHz (14-bit) by FPGA2. The raw 14-bit data for each fluorescence emission band was then transferred in pairs of 2 to a 125.71 MHz clock domain for processing before sending the results to the host computer for further analysis and visualization.

2.3.2.1 *Background Subtraction*

A typical FDFLIM measurement consisted of synchronous background signal in addition to the fluorescence emission. The background resulted from fluorescence from optical components in the system and from incomplete shielding of electronics. The background signal was measured by spatially averaging pixels containing no fluorophores from the reference image. The averaged background signal was then subtracted from all pixels of subsequent images as shown below.

$$\underbrace{d_{(t,x,y,\lambda)} = d^{\text{Measured}}_{(t,x,y,\lambda)} - d^{\text{Background}}_{(t,\lambda)}}_{\text{time domain}} \tag{2.10}$$

$$\underbrace{D_{(x,y,\lambda,f)} = D^{\text{Measured}}_{(x,y,\lambda,f)} - D^{\text{Background}}_{(\lambda,f)}}_{\text{frequency domain}}$$

The resulting magnitude and phase are then further processed before calculating the modulation and phase lifetimes.

$$\underbrace{I_{(x,y,\lambda,f)} \equiv |D_{(x,y,\lambda,f)}|}_{\text{magnitude spectra}}, \underbrace{\phi^{\text{measured}}_{(x,y,\lambda,f)} \equiv \angle\, D_{(x,y,\lambda,f)}}_{\text{phase spectra}} \tag{2.11}$$

These subsequent processing steps consist of removing system effects, and are discussed in the following sections.

2.3.2.2 *Modulation Lifetime Corrections*

Two corrections were applied to the magnitude spectrum of the observed fluorescence emission before applying equation (2.5) to obtain the modulation lifetimes. **Normalization** corrected for the non-uniform modulation frequency content in the laser excitation. **DC approximation** was used to approximate the magnitude of the fluorescence emission at 0 Hz with a frequency $>$0 Hz. This was needed because various components in

the system were AC coupled, prohibiting measurements at DC (0 Hz). The normalization correction was computed beforehand and applied to all subsequent images. The DC approximation was applied on a pixel-wise basis, i.e. each pixel required its own DC approximation.

2.3.2.2.1 DC approximation The modulation lifetime required a measurement of the fluorescence emission at $f = 0$, the frequency content at DC. The APDs, amplifiers, and digitizer were all AC coupled, thus it was not possible to acquire frequency content at $f = 0$. The magnitude of the fluorescence emission at DC was approximated using a sufficiently low frequency, $f_{\approx DC}$. The $f_{\approx DC}$ was chosen such that the expected fluorophores exhibited insignificant modulation at that frequency. Figure 2.3 shows the modulation lifetime error when $f_{\approx DC} = 1.25$ MHz. The correction was performed by dividing the magnitude spectrum of the observed fluorescence emission by the magnitude at $f_{\approx DC}$ as shown later in equation (2.14).

2.3.2.2.2 Normalization The digital modulation signal (see figure 2.2B) consisted of several modulation frequencies. The power in the laser output at each modulation frequency was not equal, and thus required normalization before applying equation (2.5). A reference fluorophore with a known lifetime was used to compute normalization factors for each modulation frequency and emission band $I^{Norm}_{(\lambda,f)}$. The normalization factors $I^{Norm}_{(\lambda,f)}$ are computed by solving for the constants that make the ratio at each modulation frequency in the observed fluorescence emission equal to the theoretical modulation of the reference fluorophore, $I^{\text{True ref}}_{(\lambda,f)}$.

The normalization and DC approximation corrections are summarized below. The fully corrected modulation is denoted as $m_{(x,y,\lambda,f)}$.

$$I^{Norm}_{(\lambda,f)} = I^{\text{True ref}}_{(\lambda,f)} / I^{\text{Measured ref}}_{(\lambda,f)} \tag{2.12}$$

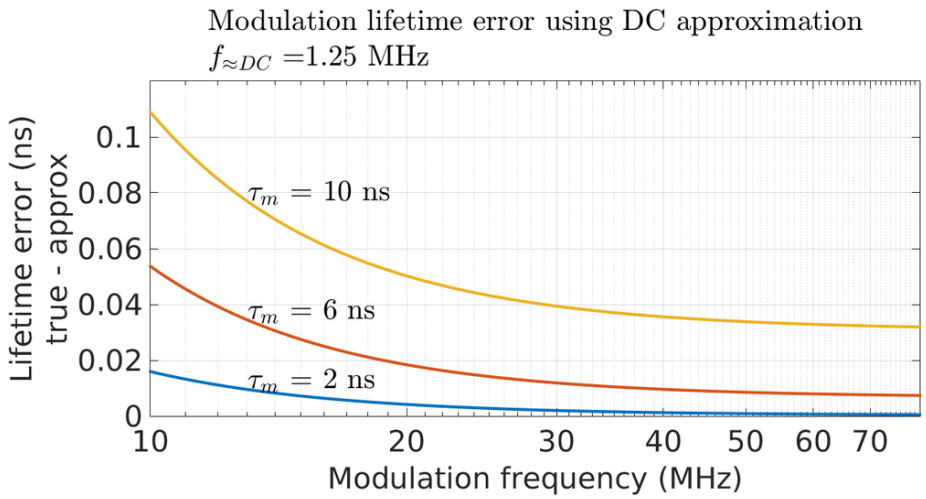

Figure 2.3: Simulated modulation lifetime error when the fluorescence emission intensity at 0 Hz (DC) is approximated by the fluorescence emission intensity at 1.25 MHz for 2-10 ns mono-exponential lifetimes. The DC approximation resulted in an underestimation of lifetimes, with more underestimation for longer lifetimes. However, the magnitude of the error was below 0.1 ns for a majority of the 10 - 100 MHz range. Similarly, the relative error for 2-10 ns mono-exponential lifetimes was below 1% for the majority of the 10 - 100 MHz range. Furthermore, the ordering of lifetimes was not effected by the DC approximation, it only reduced lifetime contrast. In the context of the endogenous fluorophores in the human coronary artery, collagen, elastin, and lipid, this < 0.1 ns error was deemed acceptable.

$$\text{DC Approximation} = I_{(x,y,\lambda,\boldsymbol{f\approx DC})} \tag{2.13}$$

$$m_{(x,y,\lambda,f)} = \frac{I^{Norm}_{(\lambda,f)}}{I_{(x,y,\lambda,\boldsymbol{f\approx DC})}} * \ I_{(x,y,\lambda,f)} \tag{2.14}$$

2.3.2.3 Phase Lifetime Corrections

The optics and electronics in the system introduced a phase delay in the digitized fluorescence that was not due to the imaged fluorophores. This delay needed to be removed before applying equation (2.4) to obtain the phase lifetime. Similar to the modulation corrections, the phase delay due to the system was measured by imaging a reference fluorophore with a known lifetime. The $\phi^{\text{True ref}}_{(\lambda,f)}$ term, the theoretical phase delay of the reference fluorophore at the given frequency and emission band, was calculated from the known lifetime of the reference fluorophore. With the $\phi^{\text{Measured ref}}_{(\lambda,f)}$ and $\phi^{\text{True ref}}_{(\lambda,f)}$ terms known, the $\phi^{\text{system}}_{(\lambda,f)}$ was calculated and saved for correcting future measurements. The corrected phase was denoted as $\phi_{(x,y,\lambda,f)}$, and was obtained by subtracting the system phase from the measured phase. The F operation, shown below, was required due to the arbitrary value of $\phi^{\text{system}}_{(\lambda,f)}$ as calculated from $\phi^{\text{Measured ref}}_{(\lambda,f)}$. For example, if $\phi^{\text{system}}_{(\lambda,f)}$ was close to $-\pi$, the measured phase of fluorophores could be positive values less than π. In such cases, $\phi^{\text{system}}_{(\lambda,f)} - \phi^{\text{measured}}_{(x,y,\lambda,f)} < -\pi$, and 2π was then added to calculate the correct phase before applying equation (2.4). Similarly, if $\phi^{\text{system}}_{(\lambda,f)} > \pi$, shorter lifetime fluorophores can have measured phases close to $-\pi$. In these instances, $\phi^{\text{system}}_{(\lambda,f)} - \phi^{\text{measured}}_{(x,y,\lambda,f)} > \pi$, and -2π was added before applying equation (2.4). The $\phi^{\text{system}}_{(\lambda,f)}$ will never be $< -\pi$, because that would require $\phi^{\text{Measured ref}}_{(\lambda,f)} < -\pi$, which cannot occur since all measured phase values are on the interval $[-\pi, \pi]$.

$$\phi_{(x,y,\lambda,f)} = \text{F}\left[\underbrace{\phi^{\text{Measured ref}}_{(\lambda,f)} + \phi^{\text{True ref}}_{(\lambda,f)}}_{\phi^{\text{system}}_{(\lambda,f)}} - \phi^{\text{measured}}_{(x,y,\lambda,f)}\right] \tag{2.15}$$

$$\mathrm{F}(\phi) = \begin{cases} \phi + 2\pi, & \phi < -\pi \\ \phi \quad\quad , & -\pi \leq \phi \leq \pi \\ \phi - 2\pi, & \phi > \pi \end{cases} \tag{2.16}$$

2.3.2.4 Normalized Intensity

The ratio of the fluorescence emission between spectral bands provided information in addition to the fluorescence lifetime. The normalized intensity, $\hat{I}_{(x,y,\lambda)}$, for a given emission band was defined as the fluorescence intensity in the given emission band divided by the sum of the fluorescence across all bands.

$$\hat{I}_{(x,y,\lambda)} = \frac{I_{(x,y,\lambda,f=f_{\approx DC})}}{\sum_{\lambda=1}^{N_\lambda} I_{(x,y,\lambda,f=f_{\approx DC})}} \tag{2.17}$$

$N_\lambda \equiv$ Number of emission bands in system

2.3.3 FPGA Data Processing

FPGA processing was developed to allow for real time imaging, to eventually lower equipment costs, and to facilitate adapting the FDFLIM system to a clinical setting. This section first explains the data throughput requirement of the FDFLIM system, and how FPGA accelerated processing reduces the data rate. Then the FDFLIM processing from section 2.3.2 that we have implemented on the FPGA is explained.

The main signal processing challenge of this direct implementation of FDFLIM was the relatively high data rate. Continuous acquisition of 14-bit data at 250 MS/s from one emission band in the FDFLIM system resulted in a data rate of 0.407 GiB/s.

$$0.407 \text{ GiB/s} = \underbrace{14}_{\text{bits}} * \underbrace{250}_{\text{MS/s}} / \underbrace{8}_{\text{bits/B}} / \underbrace{2^{30}}_{\text{B/GiB}}$$

Data was transferred in byte multiples to the host computer (making 14 bit equivalent to transferring 16 bit data), which increased the rate to 0.466 GiB/s. Simultaneous acquisition of 3 emission bands tripled the data rate to 1.397 GiB/s from the FPGA to host computer.

Performing the first step in FDFLIM processing, the DFT, on the FPGA reduced throughput by 2-3 orders of magnitude. The actual throughput reduction depended on the bit width of the processed data, pixel rate, and the number of frequencies computed by the DFT. If a 12.5 kHz pixel rate was used with a 250 MS/s sampling rate, one pixel of raw FDFLIM data consisted of 20000 data points per emission band. Once the DFT was performed, the data per pixel per channel was reduced to one complex (real/imaginary) data point per modulation frequency. The data rate for processed data assuming 12.5 kHz pixel rate and 10 DFT frequencies is shown below.

$$1.431\ \text{MiB/s} = \underbrace{2}_{\text{re and im}} * \underbrace{48}_{\text{bits per re and im}} * \underbrace{10}_{\text{frequencies}} * \underbrace{12500}_{\text{Pixels/s}} \div \underbrace{8}_{\text{bits/B}} \div \underbrace{2^{20}}_{\text{B/MiB}}$$

However, due to software limitations, the data was transferred to the host computer as 64 bit numbers, rather than 48 bit numbers, as 48 bits did not fit into any common 8, 16, 32, or 64 bit number types. This increased the data rate to 1.907 MiB/s per emission band, and 5.722 MiB/s for all 3 emission bands.

The DFT was implemented in LabVIEW FPGA using built-in LabVIEW primitives for basic logic elements (registers, etc.) and Xilinx Coregen IP for more complicated operations (multiply accumulate, direct digital synthesizer, etc.). The DFT algorithm on FPGA2 is summarized in Figure 2.4. The resource usage summary of the design on FPGA2 is shown in table 2.1.

The system calibration processing steps outlined in sections 2.3.2.1 to 2.3.2.4 were

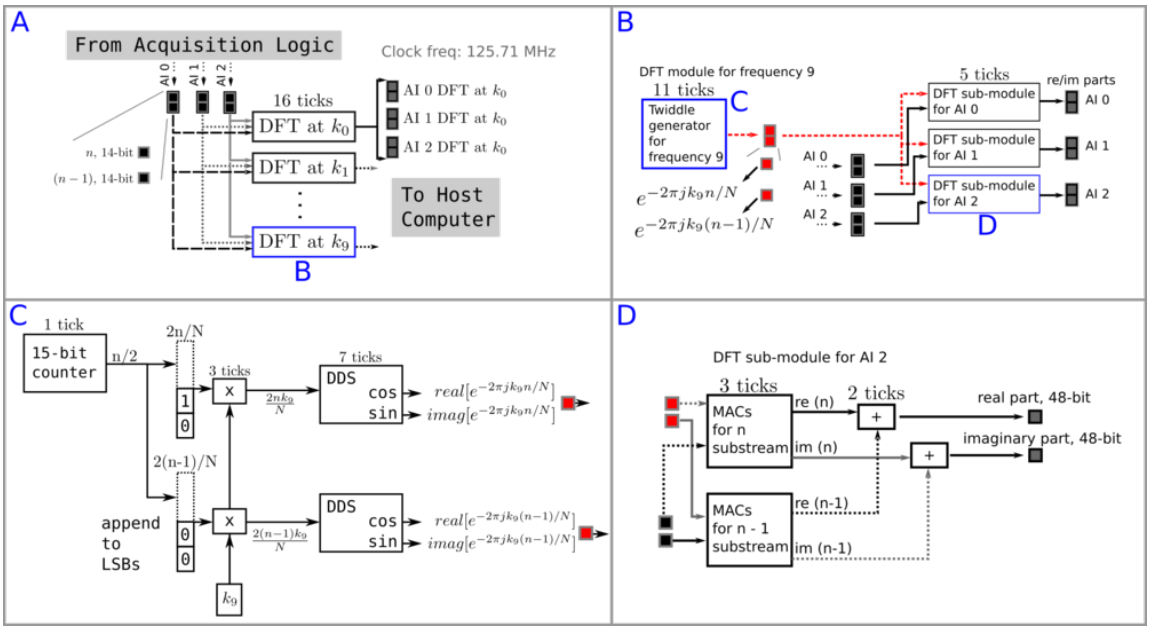

Figure 2.4: Diagram of the 65536-length DFT implemented on FPGA2 corresponding to the processing logic in figure 2.2B. The DFT of each digitized fluorescence emission band was calculated in two decimated sums as $X[k] = \sum_{n=0,2,4,\cdots}^{N-2} \left(x[n] \cdot e^{-2\pi jkn/N}\right) + \sum_{n=1,3,5,\cdots}^{N-1} \left(x[n] \cdot e^{-2\pi jkn/N}\right)$. The $x[n]$ term represents the digitized fluorescence emission signal at time point n, N is the total number of time points in the DFT (65536), k is the index of the current frequency being computed, $j = \sqrt{-1}$, and $X[k]$ is the DFT at frequency index k. The latency of each processing step is shown above in terms of ticks, i.e. clock cycles at 125.71 MHz. The DFT summation was computed across ten k values in two decimated substreams so that the processing logic could run at approximately half of the ADC sampling frequency. We were not able to compile the logic at the ADC sampling frequency (250 MHz), and thus resorted to the two decimated approach. The twiddle factors were generated using a 15 bit counter, one constant k index (one per frequency), one Xilinx LogiCORE Multiplier v11.2 per substream, and one Xilinx LogiCORE IP DDS Compiler v4.0 (direct digital synthesizer) per substream (panel C). A total of 4 multiply accumulate (MAC) modules (Xilinx multiply accumulator 2.0) were used per each frequency and channel. One MAC was used for each real part and imaginary part of each substream. The post-MAC adders (2 per frequency per channel) combined the n and n-1 substreams and were generated using the Xilinx adder subtracter 11.0 core generator.

Table 2.1: FPGA resource usage summary of the 3 channel, 14-bit, 250 MS/s, and 10-frequency 65536-length DFT processing logic on FPGA2 (National Instruments NI 5761 and PXIe7962, Xilinx part: Virtex-5 SX50T). Processing was performed at streaming speeds (processing rate $\geq$ acquisition rate). Each channel of data was acquired at 1 point per clock cycle at 250 MHz, and processed at 2 points per clock cycle at 125.71 MHz. The FPGA design was also compiled without the DFT processing logic for comparison purposes. Each resource (slices, registers, etc.) is listed on a row with the total number present on the FPGA in parenthesis, and the percentage used for each processing and no processing designs.

	Processing (%)	No processing (%)
Slices (8160)	98.3	83.8
Registers (32640)	70.2	51.2
LUTs (32640)	83.2	46.4
Block RAMs (132)	30.3	23.5
DSP48s (288)	93.8	0.0

performed on the host computer in Matlab. The inbuilt tic() and toc() functions were used to measure the processing time of the system calibration steps. The goal of the timing tests was to show that the system calibration steps could be performed in real time (processing time $\leq$ pixel rate), thereby making all processing real time.

2.3.4 Sample preparation and imaging

2.3.4.1 Capillary tubes filled with fluorophores

The following fluorophores were imaged to evaluate the lifetime accuracy of the FD-FLIM system: NADH (nicotinamide adenine dinucleotide) 1mM in PBS (0.44 ns, [18]), POPOP ([1,4-Bis(5-phenyl-2-oxazolyl)benzene]) <0.1mM in ethanol (1.2 - 1.5 ns [19]), FAD (flavin adenine dinucleotide) 1mM in PBS (2.3 - 2.85 ns [19]), anthracene <0.1mM in ethanol (5.1 ns [20]), DPA (9,10 - Diphenylanthracene) <0.1mM in ethanol (8.19 ns [21]), and 9CA (9 - Anthracenecarbonitrile) <0.1mM in ethanol (11.9 ns [22]). Concentrations for POPOP, anthracene, DPA, and 9CA were diluted from 0.1mM to better fit the

fluorescence emission of all fluorophores in the dynamic range of the detectors. POPOP was used for system calibration, a lifetime of 1.35 ns was assumed for all emission bands and frequencies for both the phase and modulation lifetimes. Fluorophores were placed in fused quartz capillary tubes for imaging (VitroCom CV3040, 300μm ID, 400μm OD). The empty regions in the image (area not occupied by the capillary tubes) were used for background subtraction.

2.3.4.2 Ex vivo human coronary artery

Ex vivo human coronary artery segments were imaged to demonstrate that the system was sensitive enough to image atherosclerotic plaques. All images used a pixel rate of 12.5kHz, no spatial averaging, and a pixel spacing of 60μm in both dimensions. These parameters were chosen to match our existing time domain FLIM system to facilitate future comparison studies [23].

Human coronary artery segments were obtained from autopsy cases within 48 hours of death, according to a protocol approved by the Texas A&M University Institutional Review Board. Excess fat and connective tissue was removed from each artery segment. Then the segments were longitudinally cut open and imaged from the lumen side. Each segment was fixed in 10% formalin within 48 hours after imaging, and subsequently ink-marked for correlation with histopathology. Segments were analyzed using Movat's pentachrome and CD68 stains. Segments were cut perpendicular to the direction of blood flow during histopathological processing.

2.4 Results and Discussion

2.4.1 Capillary tubes filled with fluorophores

The FDFLIM system was validated by imaging capillary tubes containing NADH, POPOP, FAD, anthracene, DPA, and 9CA. The pixels corresponding to POPOP were selected by first excluding pixels with low fluorescence intensity, and then manually drawing

a mask around the capillary tube to separate POPOP from the other fluorophores. Similarly, masks were manually drawn around the other capillary tubes for pixel selection. Data for all POPOP pixels were averaged before calculating $I_{(\lambda,f)}^{\text{Measured ref}}$ and $\phi_{(\lambda,f)}^{\text{Measured ref}}$. The area of the image outside of the capillary tubes was used for background subtraction. The lifetime distributions for each fluorophore are shown in table 2.2. All phase and amplitude lifetimes are shown, even for cases of low signal to noise ratio. Lifetimes were only omitted for spectral channels where fluorophores had negligable fluorescence emission (ex: NADH in the 405/20nm channel).

The NADH phase lifetimes are longer than the reported average lifetime of 0.44 ns [18]. The measured lifetimes of FAD are slightly longer than the 2.3-2.85ns range in [19]. This could be due to different excitation wavelengths, 355nm in the source and 375nm in this study. The measured lifetimes for anthracene and DPA are shorter than reported 4.42-4.59 ns versus 5.1 ns, and 6.66-7.61 ns versus 8.19 ns. This could be due to oxygen quenching in the sample as the source [20] reports that others typically utilize N_2 or argon bubbling to remove oxygen. However, measured lifetimes for 9CA were slightly longer than literature values, 12.63-12.84 ns versus 11.9 ns.

The modulation lifetimes of shorter lifetime fluorophores were overestimated at lower frequencies. The relatively small demodulation with the addition of system noise lead to values of $m_{(x,y,\lambda,f)} > 1$. This caused the modulation lifetime, equation (2.5), to be an imaginary number. Pixels were set to NaN (not a number) in these cases, and were excluded from the lifetime mean and standard deviation calculations in table 2.2. These omissions biased the reported lifetimes towards longer lifetimes. This is most apparent in the measured modulation lifetimes of NADH. However, NADH is out of the 2-10 ns intended lifetime range of the system. The optimal modulation frequencies for a 0.5 ns lifetime fluorophore are 223 MHz and 446 MHz for phase and modulation lifetimes, respectively. The phase lifetimes for NADH are closer to the reported average lifetime of

0.44 ns. The modulation lifetime bias could be mitigated by observing the amount of modulation values greater than 1. A larger portion of greater than 1 modulation values would suggest that the modulation lifetimes may be inaccurate due to low signal to noise ratio or lower than optimal modulation frequency. In such cases the phase lifetime would be more reliable.

2.4.2 Ex vivo human coronary artery

Ex-vivo human coronary artery segments were imaged to demonstrate the system's sensitivity to endogenous fluorophores. FDFLIM maps and histopathology sections of two ex-vivo human coronary artery segments are shown in figures 2.5 and 2.6. The artery segment in figure 2.5 contained several long lifetime regions. The segment in figure 2.6 does not contain long lifetime regions. The FDFLIM maps are shown above the histology images. The normalized intensity maps are shown in the first row. Subsequent pairs of rows show the modulation and phase lifetimes for different modulation frequencies denoted on the right of each color bar. Each column contains data from a spectral channel that is indicated at the top (for example, the first column is 405/20 nm). The dashed red line in the lower left FDFLIM map shows the position of the histological slices. The histological section stained with Movats pentachrome is shown immediately beneath the FDFLIM maps. The CD68 stained section is shown on the bottom.

2.4.3 Instrumentation

The novelty of this FDFLIM approach arises from the acquisition and real time processing of three fluorescence emission bands without an optical reference, gain modulated detection, or lock-in detection hardware. However, the system could be further improved by using lower cost components and a more optimized laser digital modulation signal.

The setup described here used two Vertex-5 FPGAs connected to a PXIe chassis, which was in turn connected via PCI express to a host computer as shown in figure 2.2. The max-

Table 2.2: Measured lifetimes of NADH, POPOP, FAD, anthracene, DPA, and 9CA at 20.83 MHz and 3 higher harmonics (mean ± standard deviation). Phase lifetime (τ_ϕ) and modulation lifetime (τ_m) are shown for each modulation frequency and relevant spectral band (λ_{em}). Spectral bands are labeled in the second column by wavelength as center/FWHM (nm).

	λ_{em}	20.83 MHz τ_ϕ,τ_m (ns)	41.67 MHz τ_ϕ,τ_m (ns)
NADH	440/40	0.74 ± 0.21, 2.84 ± 1.08	0.69 ± 0.12, 1.64 ± 0.62
	525/50	0.61 ± 0.23, 3.00 ± 1.19	0.58 ± 0.14, 1.60 ± 0.63
POPOP	405/20	1.35 ± 0.08, 1.77 ± 0.69	1.35 ± 0.06, 1.34 ± 0.37
	440/40	1.35 ± 0.06, 1.54 ± 0.57	1.35 ± 0.04, 1.34 ± 0.24
	525/50	1.35 ± 0.22, 2.70 ± 1.15	1.35 ± 0.13, 1.65 ± 0.63
FAD	525/50	2.77 ± 0.10, 3.59 ± 0.53	2.67 ± 0.08, 3.10 ± 0.22
ANT	405/20	4.43 ± 0.21, 4.63 ± 0.86	4.54 ± 0.26, 4.44 ± 0.41
	440/40	4.42 ± 0.22, 4.56 ± 0.96	4.59 ± 0.30, 4.44 ± 0.44
DPA	405/20	6.69 ± 0.25, 6.73 ± 0.64	6.94 ± 0.54, 6.72 ± 0.45
	440/40	6.67 ± 0.18, 6.85 ± 0.41	6.99 ± 0.35, 6.76 ± 0.29
	525/50	6.66 ± 0.95, 6.78 ± 2.41	7.29 ± 3.08, 6.76 ± 1.83
9CA	405/20	12.79 ± 1.60, 12.76 ± 1.75	16.40 ± 72.44, 12.96 ± 2.14
	440/40	12.63 ± 0.59, 12.95 ± 0.52	13.75 ± 2.23, 12.84 ± 0.65
	525/50	12.76 ± 1.35, 12.81 ± 1.39	15.12 ± 8.12, 12.62 ± 1.58

	λ_{em}	62.5 MHz τ_ϕ,τ_m (ns)	83.33 MHz τ_ϕ,τ_m (ns)
NADH	440/40	0.66 ± 0.12, 1.29 ± 0.46	0.61 ± 0.19, 1.23 ± 0.44
	525/50	0.55 ± 0.13, 1.16 ± 0.46	0.51 ± 0.22, 1.06 ± 0.45
POPOP	405/20	1.35 ± 0.06, 1.34 ± 0.21	1.35 ± 0.11, 1.35 ± 0.20
	440/40	1.35 ± 0.04, 1.35 ± 0.13	1.35 ± 0.07, 1.35 ± 0.13
	525/50	1.35 ± 0.15, 1.41 ± 0.49	1.37 ± 0.31, 1.37 ± 0.50
FAD	525/50	2.58 ± 0.12, 2.91 ± 0.17	2.57 ± 0.29, 2.89 ± 0.25
ANT	405/20	4.87 ± 0.61, 4.42 ± 0.38	6.92 ± 22.65, 4.49 ± 0.76
	440/40	4.96 ± 0.72, 4.43 ± 0.44	11.26 ± 130.56, 4.68 ± 0.83
DPA	405/20	7.65 ± 1.64, 6.74 ± 0.53	5.19 ± 94.70, 6.87 ± 1.16
	440/40	7.61 ± 0.97, 6.77 ± 0.35	9.91 ± 22.33, 6.91 ± 0.77
	525/50	12.21 ± 95.71, 6.88 ± 2.69	-2.96 ± 53.93, 6.55 ± 5.99
9CA	405/20	-10.81 ± 494.93, 13.15 ± 3.61	-0.92 ± 89.38, 15.61 ± 37.40
	440/40	10.15 ± 348.62, 12.91 ± 1.10	8.96 ± 462.88, 13.15 ± 2.96
	525/50	6.38 ± 79.21, 12.73 ± 2.52	1.10 ± 59.61, 14.78 ± 11.80

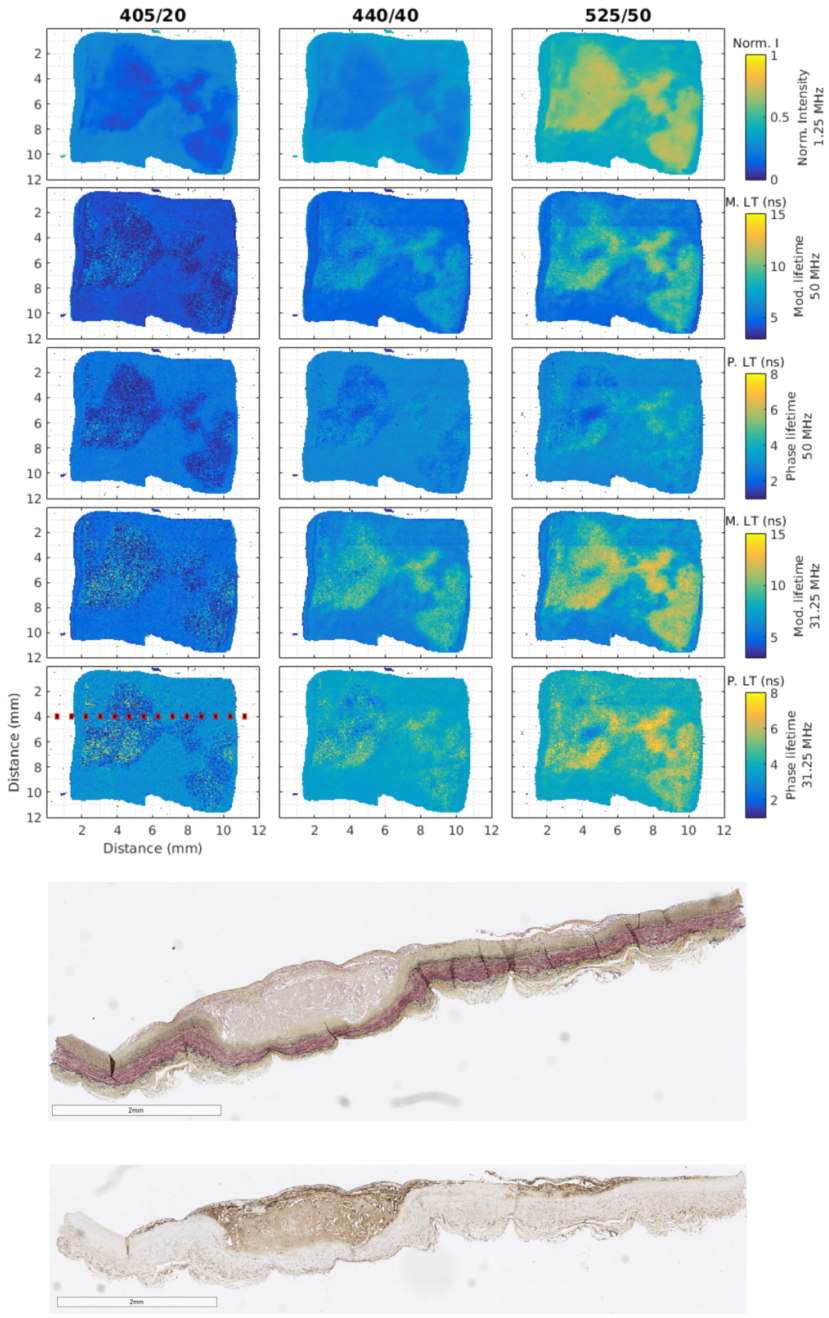

Figure 2.5: FDFLIM maps and histology images of one ex-vivo human coronary artery segment containing longer lifetime regions (blood flow direction was vertical). Each column of FDFLIM maps corresponds to an emission band. The leftmost column is the 405/20 nm (center/FWHM) emission band, the middle column is the 440/40 nm emission band, etc. The normalized intensity maps are located in the first row. The second and third rows contain the modulation and phase lifetime maps, respectively, for 50 MHz. Similarly, the bottom two rows contain the lifetime maps for 31.25 MHz. The color scales for the modulation and phase lifetime maps span 3-15ns and 1-8ns, respectively.

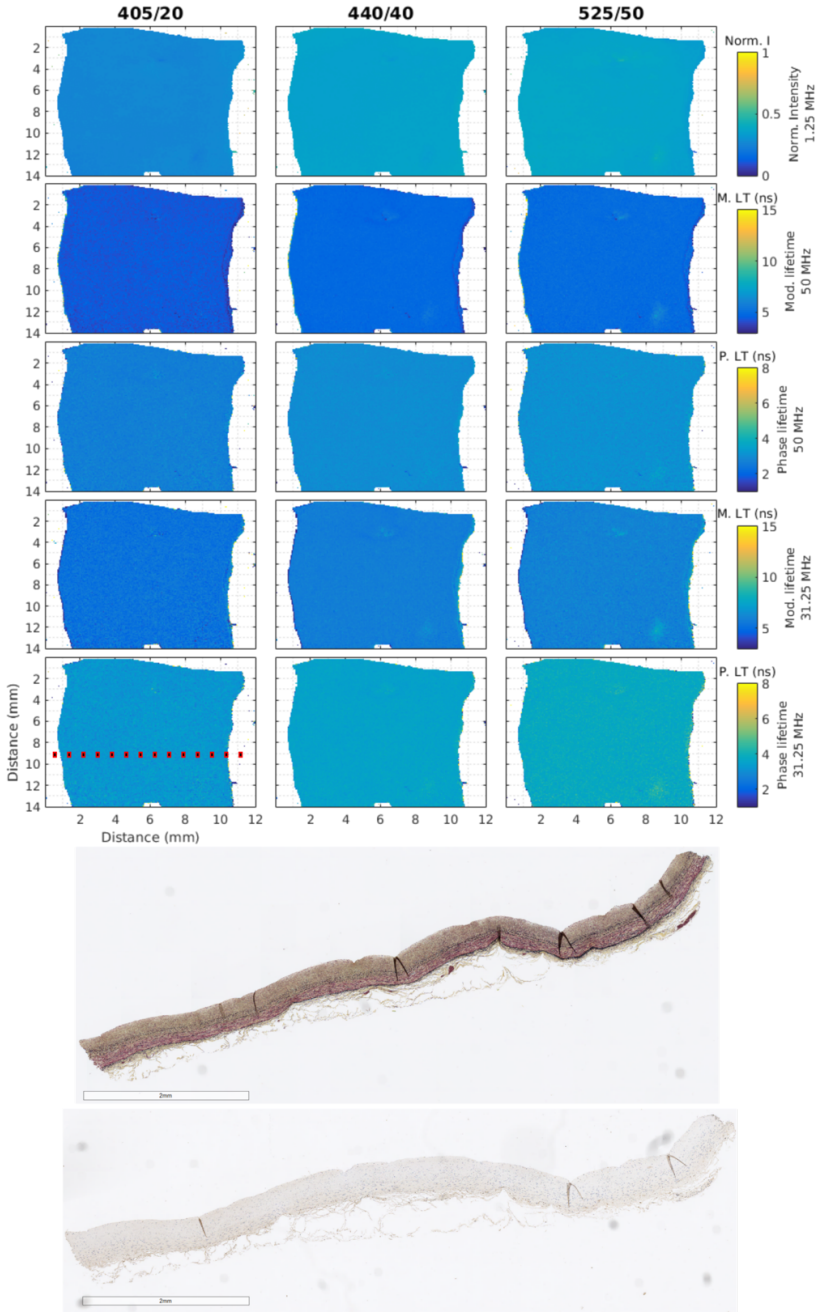

Figure 2.6: FDFLIM maps and histology images of one ex-vivo human coronary artery segment containing no longer lifetime regions (blood flow direction was vertical). Each column of FDFLIM maps corresponds to an emission band. The leftmost column is the 405/20 nm (center/FWHM) emission band, the middle column is the 440/40 nm emission band, etc. The normalized intensity maps are located in the first row. The second and third rows contain the modulation and phase lifetime maps, respectively, for 50 MHz. Similarly, the bottom two rows contain the lifetime maps for 31.25 MHz. The color scales for the modulation and phase lifetime maps span 3-15ns and 1-8ns, respectively.

imum data rate from FPGA2 to the host computer was 800 MiB/s, much greater than the required $<$ 6 MiB/s data rate described in section 2.3.3. System cost and complexity could be reduced by implementing the acquisition and processing logic on a lower cost FPGA, for example the Nexys Video Artix-7 from Digilent ($490) or Kintex-7 KC705 ($1695) with an FMC digitizer card (typically $2.5-5k for an FMC digitizer card with 4 channels and 250 MS/s sampling rate). The $<$6 MiB/s data stream could easily be transfered via USB2.0 to the host computer. The analog output functionality for galvonometer mirror control could be accomplished using two Pmod DA3 modules ($30 each Digilent, in addition to op-amps to bias and amplify the output as needed).

The laser digital modulation, clock, and trigger signals could be generated on the same FPGA responsible for analog to digital conversion of the fluorescence emission. We have in fact done this in previous prototyping work with FPGA1. However, we had to utilize FPGA2 for acquisition and processing due to the insufficient number of analog inputs on FPGA1. The single FPGA approach could also be achieved with the Nexys Video Artix-7 mentioned earlier. In the current design, these signals were generated on FPGA1 while digitization and processing occurred on FPGA2. Moving all logic to one FPGA would reduce system complexity by removing the external 10 MHz clock, trigger, and other auxilary signals. The trigger would be generated internally to the FPGA in the same clock domain as the analog to digital converters (which removes the need for the 10 MHz clock). This was not performed in the current setup because the digital output bandwidth of FPGA2 was 500 kHz (versus the 100 MHz desired bandwidth). FPGA1 was used for convenience as it was already in our laboratory, however its $>$ $10000 price makes it otherwise cost prohibitive. Alternatively, the clock, trigger, and laser digital modulation signal could be generated on a cheaper FPGA while still using FPGA2 for digitization and processing. The Arty Atrix-7 from Digilent ($99), for example, is capable of outputting digital signals on the order of 100 MHz.

The laser digital modulation signal could be optimized to give more control over the power at each modulation frequency. In the current system, the laser digital modulation signal was generated by counters on FPGA1. Multiple counters were set to specific frequencies and then logically AND-ed to mix the frequencies into one digital signal. This approach was simple to implement, but offered poor control over the power in each modulation frequency. A better alternative would be to store an arbitrary digital waveform in the FPGA and output it for each spatial pixel. The arbitrary waveform could be analytically or empirically optimized beforehand to give the desired power over the desired modulation frequencies. In the current design, 65536 is the max number of clock cycles at 250 MHz possible in one spatial pixel, meaning that the longest arbitrary waveform would consist of 65536 bits. This would occupy less than 4% of the block RAM on the $99 Arty Atrix-7 FPGA mentioned earlier.

2.4.4 Data Processing

Data processing was performed in two locations, on FPGA2, and on the host computer. The DFT was performed in real time on FPGA2 to reduce data throughput by a factor of 250. System calibration steps were performed on the host computer because the current FPGA design does not yet include those operations. The system calibration steps were timed on various computers to show that they can be performed in real time, thus making all FDFLIM processing real time. Table 2.3 summarizes the system calibration timing tests. The processing time required on all tested computers for the system calibration was less than the 12.5kHz pixel rate used for imaging. Two times for each computer and frequency combination are listed because the F operation (see equation (2.16)) can be avoided by designing the digital pulse train on FPGA1 so that $\phi^{\text{system}}_{(\lambda,f)} \geq -\pi/2$. This will ensure that all valid phase delays will not wrap from $-\pi$ to π. These computers were selected based on convenience and hardware specifications. At the time of this writing,

the Toshiba Satellite and Macbook Pro 4.1 (early 2008) represented relatively low end computers, the Intel i5-4670K computer represented a mid-level desktop, and the Intel i7-4820K computer represented a higher-end desktop. The Toshiba Satellite, even with a single core processor and 1GB of RAM, was able to perform system calibrations in real time.

The 10 DFT FPGA processing scheme was limited by development time and skill. However, significant optimization is possible. Implementing the DFTs at 125.71 MHz was a design choice that compromised increased resource usage for a slower clock frequency. The DSP48 slices on FPGA2 are able to operate at clock frequencies greater than 125.71 MHz. The DSP48 logic could be moved to a $>$ 250 MHz clock domain while keeping the other logic in the 125.71 MHz clock domain. The post-MAC adders (60 in the current design) could be replaced with a single adder, since the throughput after the MACs is on the order of 1 out of 1000 valid point per clock cycle. All MACs could be configured to share the adder by multiplexing the MAC outputs between the adder. All DFT modules could share the same 15-bit counter (10 in the current design) for twiddle factor generation (see figure 2.4). Operating all processing logic at 250 MHz instead of 125.71 MHz would remove the 2 points per clock processing requirement, reducing the resource requirements by half. However, this would require more design optimization to meet timing constraints.

The FPGA to host computer transfer rate could be further reduced by converting the real and imaginary parts to floating point immediately before transfer. For example, each 48-bit real and imaginary number for each frequency could be represented with some precision loss as a 32 bit floating point number. The 32 bit floating point number would consist of 27 integer bits plus 5 exponential bits to account for the possible 21 power of 2 shift values. The data transfer rate would be halved as each number would be transferred as 32 bit instead of 64 bit. Since the raw ADC data was 14-bit, we do not expect all 48 bits of output data to be needed for accurate and precise lifetime values. Existing data could

Table 2.3: System calibration processing time on various computers for 2, 4, and 10 frequencies. System calibration includes all processing steps listed in section 2.3.2 after the DFT is computed on FPGA2: background subtraction, modulation lifetime corrections, phase lifetime corrections, and normalized intensity. CPU and RAM were operated at default clock frequencies for all computers. The time per pixel for all 3 emission bands is listed in the table below (mean $\pm$ standard deviation). Two times are listed for each frequency and computer combination. The top time includes all processing steps except the F operation in equation (2.16), and the bottom time include all processing steps. A pixel rate of 12.5 kHz (80 µs) was used for all images. A total of 72, 66, and 12 images were used for the 2, 4, and 10 frequency results, respectively. However, on the Toshiba Satellite computer, only 4 images were used for the 10 frequency timing tests due to RAM limitations.

Computer	Frequencies		
	2	4	10
Toshiba Satellite L305-S5917 Intel 585 2.16 GHz, 1 GB RAM, Debian GNU/Linux 8.8 (Jessie) 3.16.43-2 x86_64, Matlab 2017a	~~F~~: 3.984 ± 0.191 µs F: 4.502 ± 0.870 µs	7.957 ± 0.485 µs 9.166 ± 1.916 µs	21.027 ± 0.284 µs 22.375 ± 0.131 µs
Macbook Pro 4.1 (early 2008) Intel Core 2 Duo 2.4 GHz, 6 GB RAM, OS X 10.10.5, Matlab 2012b	~~F~~: 2.281 ± 0.366 µs F: 5.664 ± 0.737 µs	4.478 ± 0.433 µs 10.966 ± 1.781 µs	12.219 ± 1.833 µs 29.577 ± 4.474 µs
Custom computer Intel i5-4670K, 32 GB RAM, Debian GNU/Linux 9 (stretch) 4.9.0-1 x86_64, Matlab 2015a	~~F~~: 0.742 ± 0.103 µs F: 0.785 ± 0.092 µs	1.295 ± 0.143 µs 1.509 ± 0.436 µs	2.957 ± 0.407 µs 3.371 ± 0.341 µs
Custom computer Intel i7-4820K, 56 GB RAM, Windows 7 Enterprise, Matlab 2015a	~~F~~: 0.723 ± 0.064 µs F: 0.851 ± 0.069 µs	1.515 ± 0.161 µs 1.779 ± 0.119 µs	3.930 ± 0.087 µs 4.515 ± 0.208 µs

be analyzed to verify the minimum amount of integer bits required to maintain acceptable lifetime accuracy and signal to noise ratio.

We have presented an FPGA based point scanning FDFLIM system capable of resolving nanosecond lifetimes using multiple simultaneous excitation frequencies, simultaneous detection of multiple emission bands with fixed gain APDs, and real time processing. This novel approach reduces system cost and complexity by not requiring an optical reference, gain modulated detection, or lock-in detection. We have verified the system accuracy by imaging fluorophores NADH, POPOP, FAD, anthracene, DPA, and 9CA. The sensitivity to endogenous fluorophores was demonstrated by imaging ex-vivo human coronary arteries. This implementation of FDFLIM could easily be extended to include multiple excitation sources (for example 375nm for endogenous and 488nm for exogenous fluorophores) by frequency multiplexing the modulation frequencies from each light source. More emission channels could be added by synchronizing additional FPGAs with more ADCs synchronized to the same reference clock.

2.5 Supplemental

2.5.1 FPGA: Processing speed and resource usage

The processing speed and image size limitations will be discussed similarly to the preceding section, but in more detail. The valid pixel rates are defined by the DFT length and the sampling frequency. The pixel rate must be a multiple of $1/fs^{raw}$. The maximum image size in the current FPGA firmware is 4096×4096 and is limited by a 24-bit counter on the FPGA.

$$fs^{raw} = 250 \text{ MHz} = \text{Sampling frequency} \tag{2.18}$$

$$fs^{proc} = 125.71 \text{ MHz} = \text{Processing frequency} \tag{2.19}$$

$$THRPUT^{proc}_{clk} = 2 = \text{Number of time points processed per clock cycle at } fs^{proc} \tag{2.20}$$

$$\text{PostTrig}_{\text{ticks}} = 1 = \text{Clock ticks required after trig. and} \tag{2.21}$$

$$\text{before acquisition at } fs^{proc}$$

$$\text{ReARM}_{\text{ticks}} = 1 = \text{Clock ticks required after acquisition} \tag{2.22}$$

$$\text{before next trig.}$$

$$DFT_{len}^{sub} \leq 2^{15} = 32768 \tag{2.23}$$

$$DFT_{len}^{full} \leq DFT_{len}^{sub} \times THRPUT_{clk}^{proc} = 65536 \tag{2.24}$$

$$DUT^{max} \approx 99.994\% \tag{2.25}$$

$$\approx \frac{DFT_{len}^{full} - THRPUT_{clk}^{proc} \times (\text{PostTrig}_{\text{ticks}} + \text{ReARM}_{\text{ticks}})}{DFT_{len}^{full}}$$

$$DFT_{len}^{full} \times \frac{1}{fs^{raw}} \leq \text{Pixel}_{rate}^{DFT} \tag{2.26}$$

$$\leq (\text{PostTrig}_{\text{ticks}} + \text{ReARM}_{\text{ticks}} + 1) \times \frac{1}{fs^{raw}}$$

$$3.815 \text{ kHz} \leq \text{Pixel}_{rate}^{DFT} \leq 83.333 \text{ MHz} \tag{2.27}$$

$$\min(\text{Pixel}_{rate}^{DFT} \text{ increment}) = \frac{1}{fs^{raw}} = 4\text{ns} \tag{2.28}$$

The resource usage summary for the current implementation of the design is shown in table 2.4. Two versions of the design exist with and without the use of the DRAM on board the NI PXIe 7962 FPGA. The DRAM is utilized when acquiring the raw time domain signal, $d_{i(x,y)}[\hat{t}]$, to provide a larger buffer for transferring the raw data stream from the FPGA to the host computer. The DRAM is not used in the FPGA processing scheme. We still acquire the raw time domain data for each sample, $d_{i(x,y)}[\hat{t}]$, so that we can build a database to verify the FPGA processed data. Rather than use two separate bitfiles, one for FPGA processing and one for raw data acquisition, we combined both into the same bitfile. Due to the current implementation of the design, including the DRAM interface causes timing violations that limit the number of DFT frequencies. If no DRAM interface is used, more DSP48 slices can be utilized and 10 DFT frequencies can be implemented. If the DRAM interface is used, a maximum of 9 DFT frequencies can fit on the FPGA.

In the future it may be possible to optimize parts of the design to fit 10 DFT frequencies alongside the DRAM interface.

2.5.2 FPGA: DFT - data throughput

The DFT is the only processing stage that is currently implemented in the FPGA. However, the remaining processing steps can be performed on the host computer in real time, so the all data processing can be considered real time. The DFT reduces the majority of data throughput by converting the digitized fluorescence emission as a function of time (typically 20000 time points) to one pair of real and imaginary numbers for each modulation frequency (typically ≤ 10 modulation frequencies). The remaining operations consist of magnitude, phase, and lifetime calculations. These computations occur after the data rate has been reduced by the DFT by several orders of magnitude. We will introduce several terms before showing the data throughput calculations.

1. f_s, sampling frequency.
2. $T^{raw}_{TrigFreq}$, trigger frequency (pixel rate).
3. T^{raw}_{bytes}, bytes per point of raw data when transferring to host computer.
4. T_{CH}, analog input channels.
5. DUT, duty cycle - fraction of time data is acquired per pixel.
6. T^{proc}_{bytes}, bytes per frequency per point of processed data when transferring to host computer.
7. T^{proc}_{Nfreq}, number of frequencies computed by the DFT on the FPGA.

Table 2.4: Resource usage summary for the FD FLIM processing algorithm compiled on the NI 5761 and PXIe7962. Three variants of the design are shown, with the DRAM interface and 9 DFT frequencies, with no DRAM and 10 DFT frequencies, and with no DRAM or DFT frequencies. The DRAM interface is used when raw unprocessed data is acquired to provide a buffer when transferring data from the FPGA to the host computer. Raw unprocessed data is used for comparison purposes to validate the quality of FPGA processed images. Once the FPGA processing was fully validated the DRAM is not needed. Omitting the DRAM interface makes it possible to implement 10 DFT frequencies in the design due to resource and timing constraints.

DRAM, 9 DFT frequencies	
Total Slices:	99.9% (8150 out of 8160)
Slice Registers:	75.8% (24749 out of 32640)
Slice LUTs:	89.5% (29201 out of 32640)
Block RAMs:	28.8% (38 out of 132)
DSP48s:	84.7% (244 out of 288)
No DRAM, 10 DFT frequencies	
Total Slices:	98.3% (8023 out of 8160)
Slice Registers:	70.2% (22925 out of 32640)
Slice LUTs:	83.2% (27147 out of 32640)
Block RAMs:	30.3% (40 out of 132)
DSP48s:	93.8% (270 out of 288)
No DRAM, 0 DFT frequencies	
Total Slices:	83.8% (6842 out of 8160)
Slice Registers:	51.2% (16719 out of 32640)
Slice LUTs:	46.4% (15133 out of 32640)
Block RAMs:	23.5% (31 out of 132)
DSP48s:	0.0% (0 out of 288)

When the DFT is not computed on the FPGA, the raw data must be transferred to the host computer for further processing. The required throughput is shown below in a general form, and then the throughput using current FD FLIM system parameters is calculated.

$$THR^{raw}_{bytes/s} = f_s \times T^{raw}_{bytes} \times T_{CH} \times DUT \tag{2.29}$$

$$THR^{raw}_{bytes/s} = 250\ \mathrm{MS/s} \times 2 \times 3 \times 0.99 \approx 1.383\ \mathrm{GiB/s} \tag{2.30}$$

If the DFT is computed on the FPGA, only the real and imaginary parts per each frequency per pixel must be transferred to the host computer. This results in an almost 3 orders of magnitude reduction in required throughput to the host computer. The T^{proc}_{bytes} term includes the real and imaginary parts of the DFT for a given frequency. In the current DFT implementation on the FPGA, the real and imaginary parts are 48-bit numbers, but are transferred as 64-bit numbers to the host computer. A more advanced bit-packing scheme could be implemented if further reduction in throughput is desired (theoretically could be reduced by (64-48)/64 = 25% without affecting precision). However, a data compression scheme would require more FPGA resources and host computer processing time as the data would need to be decompressed after transfer.

$$THR^{proc}_{bytes/s} = T^{proc}_{bytes} \times T_{CH} \times T^{proc}_{Nfreq} \times T^{raw}_{TrigFreq} \tag{2.31}$$

DFT at 10 frequencies

$$THR^{proc}_{bytes/s} = 16 \times 3 \times 10 \times 12.5\ \mathrm{kHz} \approx 0.005588\ \mathrm{GiB/s} \approx 5.722\ \mathrm{MiB/s} \tag{2.32}$$

DFT at 2 frequencies (minimum required for lifetime calculations)

$$THR^{proc}_{bytes/s} = 16 \times 3 \times 2 \times 12.5\ \mathrm{kHz} \approx 1.145\ \mathrm{MiB/s} \tag{2.33}$$

DFT at 10 frequencies at (100kHz)

$$THR^{proc}_{bytes/s} = 16 \times 3 \times 10 \times 100\ \text{kHz} \approx 45.777\ \text{MiB/s} \tag{2.34}$$

Processed data throughput is mainly a function of pixel rate and the number of DFT frequencies on the FPGA. The fastest pixel rate that we expect to use is 100kHz, and the maximum number of DFT frequencies on the current FPGA is 10. Therefore the highest throughput to the host computer will be $45.777\ \text{MiB/s}$. However, in our preliminary experiments we used a pixel rate on the order of $12.5\ \text{kHz}$, and used less than 10 DFT frequencies.

2.5.3 FPGA: DFT - miscellaneous

The DFT at one frequency is defined as shown below.

$$D_{(x,y,\lambda)}[\hat{f}[k]] \equiv \text{DFT}\{d_{(x,y,\lambda)}[\hat{t}]\} = \sum_{n=0}^{N-1} d_{(x,y,\lambda)}[n] \cdot e^{(-2\pi jkn)/N} \tag{2.35}$$

Here the k and n terms represent the indices of the $\hat{f}$ and $\hat{t}$ vectors, respectively. In other words, $n = 0$ corresponds to $\hat{t}[0]$, the first time point. To compute the full DFT across all frequencies, the above equation must be repeated for $k = [0, 1, \ldots, N-1]$. However, in FD FLIM we are typically interested in only a select few frequencies (typically less than 10). Thus it is more efficient to directly implement the DFT as shown above versus using another algorithm such as the FFT. The radix-2 FFT has computational complexity on the order of $log_2(N) \cdot N$, and the DFT has complexity N^2 if all frequencies are computed. However, if only N_{freq} frequencies are computed, the DFT has complexity $N_{freq} \cdot N$. If $N = 2^{16}, N_{freq} = 10$, the complexity of the radix-2 FFT is $16 \cdot 2^{16}$, and the complexity of the DFT is $10 \cdot 2^{16}$. Furthermore, using the DFT allows for lengths that are not powers of 2, which gives the system more flexible pixel rates. A higher radix FFT, such as 4, would be more computationally efficient than the DFT $log_4(N) \cdot N = 8 \cdot 2^{16}$. However, the pixel rate

restrictions make this approach unattractive at this time. For example, we currently use pixel rates on the order of 12.5 kHz. The pixel rates would be limited to integer powers of 4 as shown below assuming $f_s = 250$ MHz.

$$\text{Example pixel rates limited by powers of } 4$$

$$\text{Pixel rate } = \frac{1}{\frac{1}{f_s} \cdot 4^x} \tag{2.36}$$

$$\frac{1}{\frac{1}{250 \text{ MHz}} \cdot 4^5} = 244.14 \text{ kHz} \tag{2.37}$$

$$\frac{1}{\frac{1}{250 \text{ MHz}} \cdot 4^6} = 61.04 \text{ kHz} \tag{2.38}$$

$$\frac{1}{\frac{1}{250 \text{ MHz}} \cdot 4^7} = 15.26 \text{ kHz} \tag{2.39}$$

$$\frac{1}{\frac{1}{250 \text{ MHz}} \cdot 4^8} = 3.81 \text{ kHz} \tag{2.40}$$

The closest option to our current pixel rate is 15.26 kHz, but the next largest and smallest options are relatively far away from this value. This approach would not allow fine tuning of the pixel rate in increments of $\frac{1}{250 \text{ MHz}}$ like the pure DFT does. Fine tuning is typically needed to increase signal quality, or to optimize pixel size when the scanning speed is constrained as in intravascular imaging. The current

The basic structure of the DFT implementation for a single frequency on the FPGA is shown in part B of figure 2.4. The implementation of the DFT consists of two parts. First, the twiddle factors $e^{(-2\pi jkn)/N}$ must be generated (part C). Second, the twiddle factors are multiplied by the raw data and accumulated until the DFT is finished (part D, MAC $\equiv$ multiply accumulate). Since multiple analog input (AI) channels of data were acquired (one channel is one detector in FD FLIM), the twiddle factors were shared to reduce resource usage (part B). The outputs of the DFT for each AI channel and each frequency were one real and one imaginary number.

The twiddle factors for each frequency were generated using two direct digital synthesizer (DDS) modules (Xilinx LogiCORE IP DDS Compiler v4.0), one for each n and $n-1$ data substream. The DDS modules were instantiated as Xilinx IP cores in LabVIEW FPGA. DDS modules were the optimal choice for implementing the twiddle factors given the IP available in LabVIEW FPGA, and the resources of the FPGA. A pure lookup table approach would require excessive amounts of block RAM. For example, the current implementation uses a max DFT length of 2^{16}, and a twiddle factor bit width of 14. A lookup table of size $2^{16} \times 14$ bits consumes $0.875\ \mathrm{MiB}$ out of the available $4.64\ \mathrm{MiB}$ block RAM on the National Instruments PXIe7962. This would allow for 5 frequencies assuming no block RAM is used by anything else in the design, which is not the case. The DDS provides a method of computing the twiddle factors by using a lookup table and DSP48 slices, which is more optimal for our overall design. Another alternative is to implement our own custom lookup table and DSP48 slice method. We have not attempted such a method at this time. Due to our limited skill set, we expect our custom approach would not be as effective as the DDS.

We have not performed a full analysis of the error versus performing all processing on the CPU at this time. However, the DFT is unscaled and the final bit width of the real and imaginary parts is 48 (full width of one DSP48). Preliminary data suggests that under experimental conditions there is no observable difference between lifetimes computed from the DFT on the FPGA and the DFT on the host computer. At this time, all post-DFT calculations are performed on the host computer. Figure 2.7 shows the result of imaging the same sample twice, once with FPGA processing, and again with processing on the host computer. Currently it is not possible to simultaneously transfer both the raw time domain signal $d_{i(x,y)}[\hat{t}]$ and processed frequency domain signal $D_{i(x,y)}[\hat{f}]$ from FPGA from the same image. Therefore, variations due to measurement noise will be present between the two images.

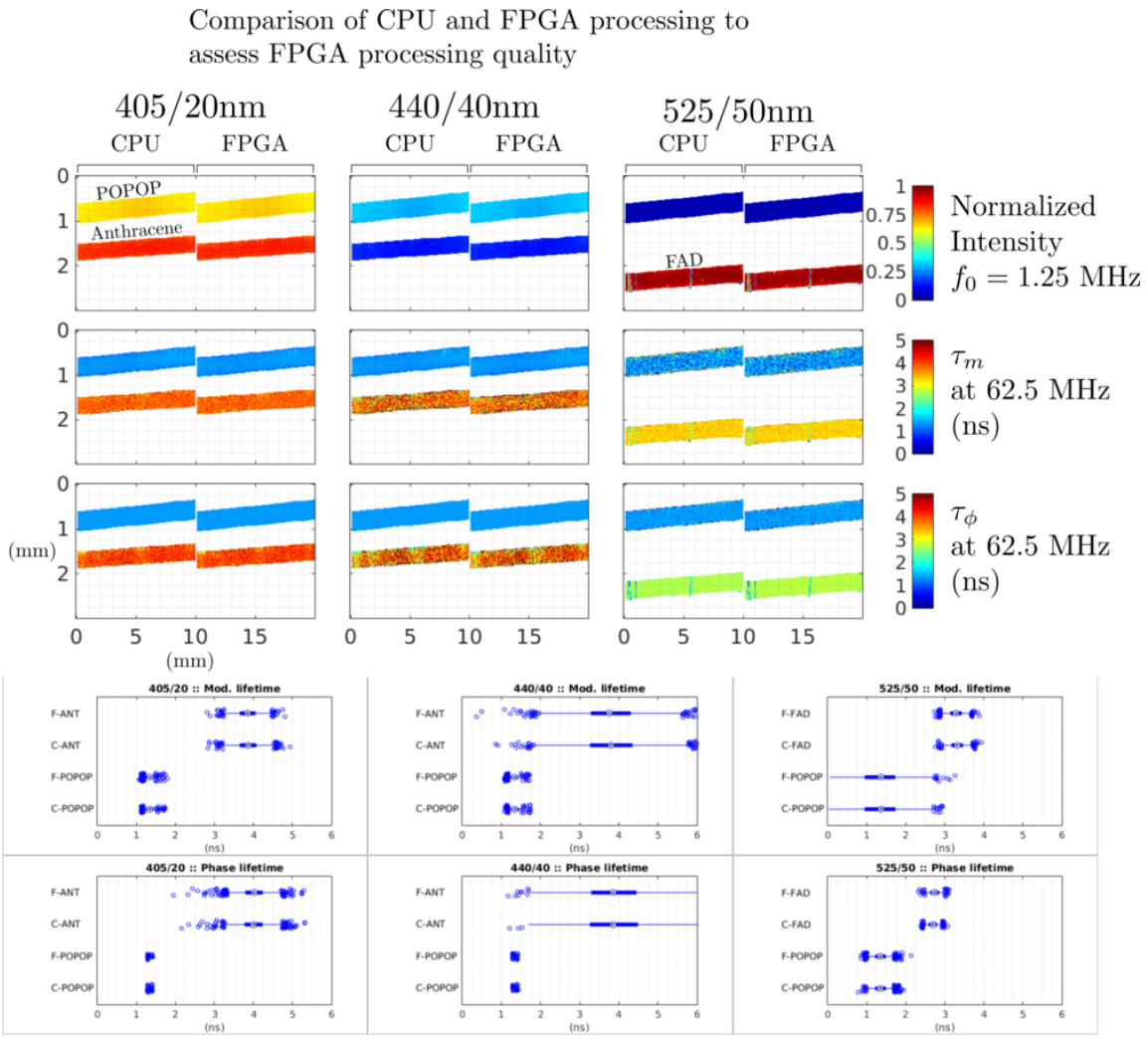

Figure 2.7: Comparison of normalized intensity, modulation lifetime (τ_m), and phase lifetime (τ_ϕ) when performing the Fourier transform on the CPU (Matlab FFT) versus the DFT of the FPGA. The same sample (POPOP, Anthracene, and FAD) was imaged twice. In the first image, the raw digitized fluorescence emission was transferred from the FPGA to host computer where all processing was performed on the CPU. In the second image, the DFT was computed on the FPGA, then the real and imaginary parts of each frequency were transferred back to the host computer for system corrections and lifetime calculations. CPU processed data is shown on the left in each plot, and FPGA processed data is shown on the right. Boxplots (using the Matlab 2017a boxplot function) of the τ_m and τ_ϕ lifetimes are shown in the bottom half of the figure. Each boxplot indicates the following: circles - outliers, lines - 0 to 25th percentile (left) or 75th to 100th (right), box - 25th to 75th percentiles.

3. DETECTION OF LIPID-RICH CORONARY ATHEROSCLEROTIC PLAQUES USING FLUORESCENCE LIFETIME IMAGING (FLIM): A COMPARISON BETWEEN TIME-DOMAIN (TD) AND FREQUENCY-DOMAIN (FD) FLIM IMPLEMENTATIONS

3.1 Introduction

Lower cost equipment and faster data processing speeds are needed for clinical imaging. The implementation of FD FLIM described here reduced cost by approximately $25000 relative to the TD FLIM system, while giving similar CD68 positive classification performance. The cost reduction came from a cheaper light source and cheaper photodetectors. Furthermore, the FD FLIM system was capable of generating modulation lifetime images used for CD68 positive classification in real time.

The primary goal of this study was to evaluate the FD FLIM system's ability to identify CD68 positive regions in ex-vivo human coronary arteries. A study evaluating the TD FLIM system's ability to classify CD68 (an immunostain for macrophages) positive regions has already been completed but is not yet published. The aforementioned TD FLIM study utilized only the 494/41nm emission channel for CD68 classification.

We hypothesized that the 525/50nm emission channel of our FD FLIM system could also be used for CD68 classification. This work investigated the correlation between the 494/41nm TD FLIM emission channel and the FD FLIM 525/50nm emission channel. Then we compared the CD68 classification performance between the 494/41nm TD FLIM and 525/50nm FD FLIM data using histology as the gold standard. Additional lifetime features were also evaluated for CD68 classification: the TD FLIM lifetime without deconvolution, and the FD FLIM phase lifetime at 1.25 MHz. The TD FLIM lifetime without deconvolution gave similar classification performance to the deconvolved lifetime. The FD

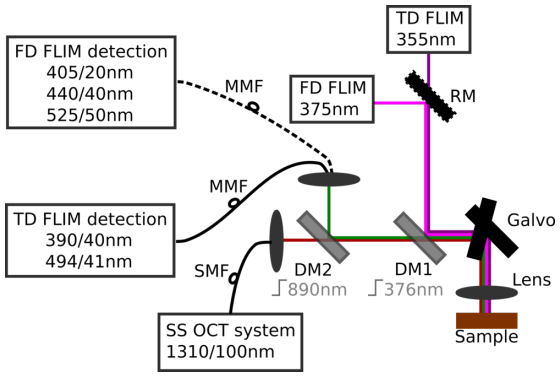

Figure 3.1: Combined TD FLIM and SS OCT, and FD FLIM system. We combined our existing TD FLIM and SS OCT system [24] with our FD FLIM system (see chapter 2). TD and FD images were acquired separately, but with the same galvo scanning mirrors and objective lens. The removable mirror (RM) was removed when using the TD FLIM system, and inserted when using the FD FLIM system. The multimode fiber (MMF) was manually switched between the TD or FD detection optics when using the respective FLIM system. The SS OCT system was not used when acquiring FD FLIM data. Wavelengths are labeled as center/FWHM.

FLIM phase lifetime at 1.25 MHz gave slightly worse performance, but the lower modulation frequency offers the potential to significantly reduce cost by using a lower speed light source and digitizer in the future.

3.2 Materials and Methods

3.2.1 Instrumentation

We combined our existing swept source (SS) optical coherence tomography (OCT) and TD FLIM system [24] (without the exogenous 532nm laser) with our existing FD FLIM (see chapter 2) system as shown in figure 3.1. The majority of components from each system were not altered, and are not shown in the schematic for brevity. The 355nm TD FLIM laser (SPOT-10-50-355, Elforlight Ltd., England) and 375nm FD FLIM laser (Toptica iBeam Smart 375) were directed towards a removable mirror (RM) (Thorlabs NX1N). The RM was added allow for manual switching between using the 375nm excitation for

FD FLIM, and the 355nm excitation for TD FLIM. The RM was removed when using the 355nm laser and added when using the 375nm laser. DM1 (Semrock FF376-Di01) was used to separate the FLIM excitation and emission. DM1 was angled tuned to reflect both 355nm and 375nm. A pair of galvonometer mirrors (Cambridge Tech. 6220H) and an objective lens (Edmund Optics #64-837) were used to raster scan the sample. DM2 (Semrock FF01-890) was used to separate the FLIM emission from the back scattered SS OCT. The multimode fiber (MMF) was manually switched between the TD and FD FLIM detection optics when using the respective FLIM system. More details on the detection systems are located in [24] and chapter 2 for TD FLIM and FD FLIM, respectively.

The lateral resolutions were 120μm and 18.5μm for the TD and FD FLIM systems, respectively. Both FLIM systems used a pixel rate of 12.5kHz, and an excitation energy per pixel of approximately 1μJ. The average duty cycle of the FD FLIM excitation waveform was approximately 73% as described in chapter 2. The relatively small spot size of the FD FLIM system meant that the thermal MPE (for skin) was exceeded by a factor of 8.24 as described in 2.3.1. The TD FLIM system did not exceed the thermal MPE for skin.

3.2.2 Data Processing

The digitized TD FLIM fluorescence emission data was first spatially averaged by four pixels in the fast scanning axis. Then an iterative nonlinear least squares deconvolution was applied using the Matlab lsqcurvefit() function, a bi-exponential model, and a measured instrument response for each emission channel. Finally, the average lifetime was computed from the estimated fluorophore impulse response using the center of mass method listed in equation (3.1). The terms in equation (3.1) are as follows: $\bar{\tau}_{i(x,y)}$ the average lifetime in spectral channel i, spatial pixel (x, y) where x is the pixel index of the fast scanning axis and y is the pixel index of the slow scanning axis, $t(p)$ is the time vector evaluated at index p, the total number of time points N, and the deconvolved fluorophore

impulse response vector h.

$$\text{TD FLIM Average lifetime} \ : \ \bar{\tau}_{i(x,y)} = \frac{\sum_{p=0}^{N-1} h_{i(x,y)}(p) \cdot t(p)}{\sum_{p=0}^{N-1} h_{i(x,y)}(p)} \tag{3.1}$$

The discrete Fourier transform of the FD FLIM fluorescence emission data was computed on the FPGA immediately after digitization. The real and imaginary parts of each frequency component were then transferred to the host computer. System calibrations were applied using data from a previously measured reference fluorophore (<0.1mM POPOP, [1,4-Bis(5-phenyl-2-oxazolyl)benzene] in ethanol, assumed mono-exponential lifetime of 1.35 ns across all spectral channels). The modulation lifetime was then computed using equation (3.2). Up to 10 frequencies were simultaneously acquired in each FD FLIM image, but only 1.25 MHz and 41.67 MHz were used in this study. The modulation frequencies 1.25 MHz and 41.67 MHz will be referred to as 1 MHz and 40 MHz, respectively, for the remainder of this text for brevity. The terms in equation (3.2) are analogous to equation (3.1), except τ_m denotes the modulation lifetime, m denotes the calibrated modulation and f denotes the modulation frequency in Hz.

$$\text{FD FLIM Modulation lifetime} \ : \ \tau_{m,i(x,y)}(f) = \sqrt{((m_{i(x,y)}(f))^{-2} - 1) \cdot (2\pi f)^{-2}} \tag{3.2}$$

$$\text{FD FLIM Phase lifetime} \ : \ \tau_{\theta,i(x,y)}(f) = \tan(\theta) \cdot (2\pi f)^{-1} \tag{3.3}$$

3.2.3 Sample Preparation and Imaging

Human coronary artery segments were obtained from autopsy cases within 48 hours of death, according to a protocol approved by the Texas A&M University Institutional Review Board. Excess fat and connective tissue were removed from each artery segment. Then the segments were longitudinally cut open, placed on a quartz microscope slide, and

imaged from the lumen side. Artery segments were first imaged with the combined SS OCT and TD FLIM system. The artery segments were then stored in plastic cassettes with foam inserts in a 4°C refrigerator. Then, within 24 hours, the artery segments were imaged with the FD FLIM system. When imaging with the FD FLIM system, multiple images were collected containing various modulation frequency combinations to allow for more flexibility in data analysis. Each segment was fixed in 10% formalin within 48 hours of imaging and subsequently ink-marked for correlation with histopathology. Segments were cut perpendicular to the direction of blood flow during histopathological processing. Histology segments were analyzed using Movat's pentachrome and CD68.

3.2.4 Co-registration

Multiple co-registration tasks were performed before conducting the correlation and classification analyses. Alignment tasks are summarized in figure 3.2. In total, five types of data were co-registered: FD FLIM, TD FLIM, OCT, pictures of the inked artery segments, and digital images of histology slices. Alignment of FD FLIM, TD FLIM, and inked artery pictures was required for the correlation analysis, and co-registration of all other data types was needed for the classification analysis.

The TD FLIM and OCT data were simultaneously acquired and thus intrinsically co-registered. FD FLIM data was acquired within 24 hours of the TD FLIM and OCT data, and required manual alignment to be spatially co-registered with the TD FLIM data. This manual alignment was performed using an open source vector graphics editing program called Inkscape (version 0.92.3 [1]). The 525/50nm FD FLIM modulation lifetime map and 494/41nm TD FLIM average lifetime map were first resized using the MATLAB function imresize() (MATLAB® (version 2017a [2])) to have equal pixel spacing of 60μm in both lateral dimensions, and then saved as PNG images. The PNG images were loaded into

[1] https://inkscape.org
[2] https://www.mathworks.com/

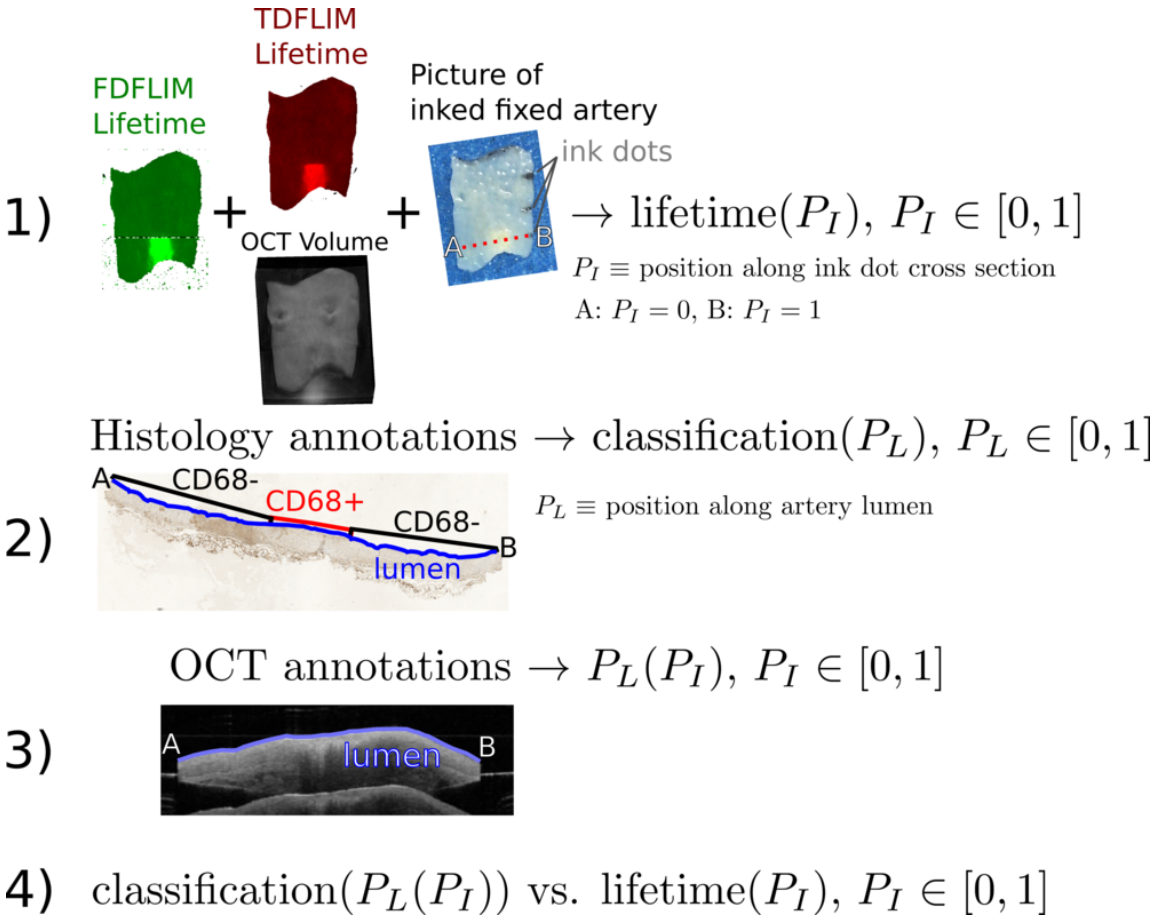

Figure 3.2: Summary of co-registration tasks that were performed for the correlation and classification analyses. First, the FD FLIM lifetime, TD FLIM lifetime, and artery picture were co-registered in Inkscape (row 1). Then a line was drawn for each ink dot (one example is denoted as A $\rightarrow$ B in this figure) corresponding to a histological section. Each histological section was classified and the lumen was traced (row 2, lumen trace is referred to as the lumen extent path, LEP, in the text) for later co-registration with OCT. An OCT cross section was extracted for each ink dot using the line drawn in row 1, as OCT and TD FLIM were acquired simultaneously and were intrinsically co-registered. Then the lumen was traced across the OCT cross section (row 3, referred to as the data extent path, DEP, in the text). The OCT annotations in row 3 allowed for mapping the ink dot position P_I to the lumen position P_L, as the lumen was typically not perfectly flat and perpendicular to the optical axis when imaging. The classification analyses were performed by comparing the lifetime as a function of lumen position (combination of rows 1 and 3) against the classification as a function of lumen position (row 2) as the gold standard.

Inkscape and the result was saved as an SVG file. The FD FLIM map was aligned with the TD FLIM map by manually rotating and translating the FD FLIM image in Inkscape. The rotation and translation amounts were extracted from the Inkscape SVG file using a Python script that employed the xml.etree.ElementTree API. The alignment amounts were then saved to a CSV file. The CSV file was then loaded into MATLAB and used to programmatically align the TD and FD FLIM data from each artery segment. Then the correlation analysis was performed as described later in section 3.2.5.1.

Digitized histological sections were first annotated by an expert reader using Aperio ImageScope [3] to classify CD68 positive regions. Then a series of tasks were performed to combine these classification annotations with co-registration annotations. All histological sections for an artery segment were located on the same microscope slide, and thus in the same microscope slide image as shown in steps 2-3 of figure 3.3. The expert reader annotations were automatically saved as an xml file by Aperio Imagescope. A Python script, using the openslide module, was used to save a thumbnail (lower resolution image) of each annotated slide image. Thumbnails were used so that the Inkscape program would remain responsive while drawing the additional annotations. The aforementioned expert reader annotations were parsed from the xml file and overlaid on the thumbnail before saving the PNG image. The PNG thumbnail image was loaded into Inkscape and saved as an SVG file. Then, also in Inkscape, a rectangle was drawn around each histology section (step 4 in figure 3.3). The thumbnail and rectangle coordinates were then used by another python script to save higher resolution thumbnails of each individual histology section defined by each rectangle. Then additional annotations were drawn on the higher resolution thumbnails in Inkscape to help automate later co-registration tasks (step 5 of figure 3.3). First, a path was drawn along the lumen and denoted as the lumen extent path (LEP). Then, smaller paths were drawn adjacent to the LEP corresponding to the expert reader classi-

[3](version 12.2.2.5015) https://www.leicabiosystems.com/digital-pathology/manage/aperio-imagescope/

fications. For example, in each area marked as CD68 positive, a path was drawn along the LEP and labeled as CPS_cd68p_#, where CPS denotes a classification path segment, and # denotes a number (object identifiers in Inkscape needed to be unique). Similarly, cd68n (negative CD68) paths were drawn where the expert reader did not indicate positive CD68 regions. Then a python script, using the svg.path module, was used to compute the classification path for each artery section. Each CPS was mapped onto the LEP to apply the classification information. The LEP, stored in the SVG file as a path consisting of different move instructions, was first approximated as a series of 1000 line segments. Then, for each CPS, all LEP points between the CPS's end points were set to the classification type of the CPS. This was repeated for each CPS (figure 3.3 step 6). Any leftover LEP regions were classified as NaN.

All five data types were co-registered before performing the classification analysis. In most cases, the physical size, shape, and orientation of each histology slice did not sufficiently match with the corresponding FLIM image. The OCT data was utilized to provide more accurate co-registration between the histology and FLIM data. First, the inked artery segment picture was co-registered with the TD FLIM data in Inkscape (similar to the FD and TD FLIM co-registration described earlier). Then one line was drawn for each ink dot on each TD FLIM image. One example line is shown in part 1 (A to B) of figure 3.2. The coordinates of each line, relative to the TD FLIM image (and therefore the OCT volume), were used to extract an OCT cross section (part 3 in figure 3.2). This cross section represented the OCT alines corresponding to the given histological slice. Then a path, denoted as data extent path (DEP), was drawn on the artery lumen of the OCT image. Care was taken to draw the DEP as similarly as possible to the LEP. The physical extent of the DEP was assumed to be equal to the LEP. In other words, the start of the DEP corresponded to the same physical position on the artery lumen as the start of the LEP. Similarly, the ends of the DEP and LEP were assumed to be at the same physical

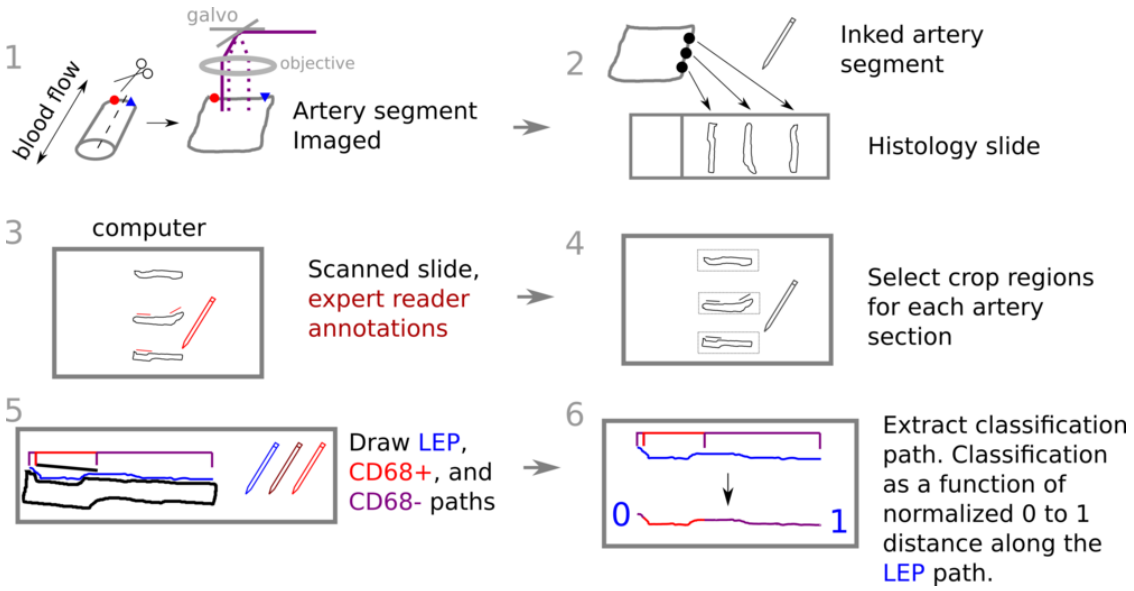

Figure 3.3: Summary of histology annotations referenced in step 2 of figure 3.2. The artery was first cut parallel to the blood flow direction and imaged with the TD FLIM and FD FLIM systems (step 1). Then the artery was fixed in formalin and inked to mark locations for histology (step 2). The digitized histology images were annotated by an expert reader using the Aperio ImageScope program (step 3). Then each artery section was cropped from the main scanned slide file and saved as a separate image to facilitate easier post processing (step 4). Each artery section image was then annotated with one lumen extent path (LEP) and at least one classification path segment (CPS) (step 5). The lumen extent path was drawn on the lumen of artery section, and a CD68+ CPS was drawn where the expert reader indicated positive CD68. The remaining regions were assumed CD68 negative and were denoted with a CD68- CPS. A classification path was generated by applying all CPSes to the LEP (step 6). This classification path was later used to generate a classification vector for use in the classification analysis.

position on the artery lumen. A classification path (CP) was generated for each LEP by applying the classification path segments to the LEP as indicated in figure 3.3. The CP contained the artery classification as a function of normalized $0 \rightarrow 1$ distance d, i.e. CP(d). The CP was then mapped to the DEP to generate a classification path for the OCT cross section CP-OCT. Finally, the FLIM classification vector was taken as the projection of the CP-OCT along the lateral dimension perpendicular to the optical axis. This allowed for comparing the histology classification with the TD FLIM or FD FLIM lifetime (part 4 of figure 3.2).

3.2.5 Statistical Analysis

3.2.5.1 Correlation

87 arteries from 23 human subjects were imaged for this study. Histological analysis involved 264 artery sections taken from the imaged arteries. The TD FLIM average lifetime from the 494/41 nm emission channel was compared with the FD FLIM 40 MHz modulation lifetime from the 525/50 nm emission channel. An outline of the correlation analysis is shown in the top half of figure 3.4. TD FLIM and FD FLIM data were compared at three spatial scopes: *between arteries*, *all arteries*, and *all artery sections*.

In the *between arteries* case, one correlation coefficient was computed for each artery for a pixel level comparison between the TD FLIM and FD FLIM lifetime map of the given artery. This was repeated with no spatial median filtering, 3x3 median filtering, and 9x9 median filtering (filter sizes indicate fast axis pixels x slow axis pixels). For the *all arteries* scope, pixels from all TD FLIM lifetime maps and all FD FLIM lifetime maps were compared and one correlation coefficient and linear fit were generated for no spatial median filtering, 3x3 median filtering, and 9x9 median filtering. The *all artery sections* scope consisted of first computing the median lifetime of each region corresponding to location with histology. Similarly to the all arteries scope, a correlation coefficient and linear

fit were generated. The linear correlation coefficients and the linear fits were calculated in Matlab using the corrcoef() and polyfit() Matlab functions, respectively.

The three spatial scopes were used in an attempt to avoid bias due to a majority of the arteries not having long lifetime regions (*between arteries*), possibly inadequate co-registration (spatial filtering), and for a more specific comparison with histology (*all artery sections*). The *all arteries* case provided the most comprehensive comparison, but was biased by the fact that a majority of arteries did not have long lifetime regions. The *between arteries* case helped compare the TD FLIM and FD FLIM lifetime maps in a more individualized manner that allowed for a comparison of correlations between arteries with and without long lifetime regions. The *all artery sections* scope provided a correlation analysis that encompassed the same pixels used for the classification analysis described later in this paper.

3.2.5.2 Classification

The bottom half of figure 3.4 shows an outline of the classification tasks. Leave one out cross validation was used to quantify classification performance. Two types of leave one out cross validation were performed, leave one subject out (LOSO, abbreviated as subject in later sections) and leave one artery out (LOAO, abbreviated as artery in later section). In LOSO, all arteries from a given subject were taken as the test set, while the remaining arteries from all other subjects were used as the training set. Similarly for LOAO, one artery was used as the test set and all other arteries were used as the training set.

Classification was performed on two scopes: pixel, and section. Individual pixel classification was performed by classifying each pixel in each artery section using the FLIM classification vector (generated earlier during co-registration with histology) as the gold standard. In the section approach, an artery section was considered CD68+ if any point in the FLIM classification vector was marked as CD68+. Artery section classification

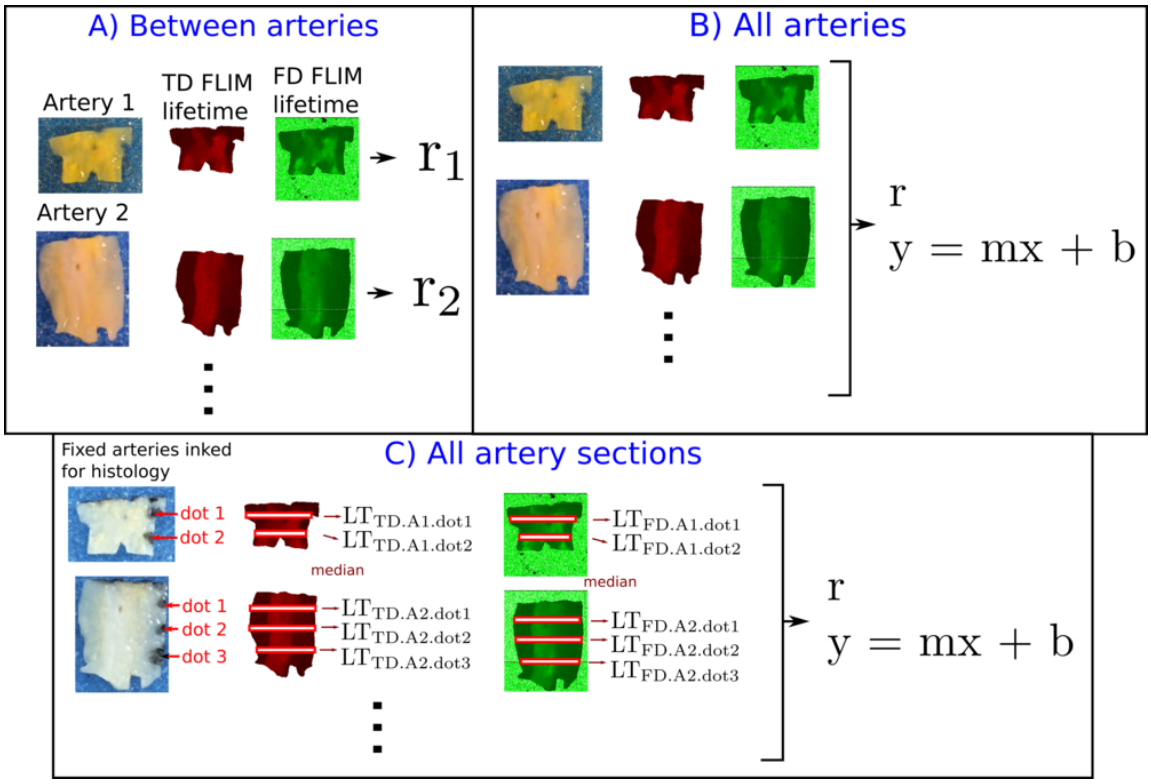

Figure 3.4: Outline of correlation scopes. Correlation analysis was performed across three spatial scopes: *between arteries*, *all arteries*, and *all artery sections*. The correlation coefficient (denoted as r) and linear fit (denoted as y = mx + b) were used as performance metrics. In the *between arteries* case (A), one correlation coefficient was computed for each artery and no linear fit was performed. In the *all arteries* case (B), one correlation coefficient and linear fit was calculated after comparing all arteries. Both the *between arteries* (A) and *all arteries* (B) analyses were performed for various amounts of spatial filtering. (A) and (B) were repeated for no spatial filtering, 3 by 3 median filtering, and 9 by 9 median filtering. The *all artery sections* scope (C) involved comparing the medians of the TD FLIM and FD FLIM lifetime maps only where histology analysis was performed (denoted by the dot numbers and red/white rectangles). One correlation coefficient and linear fit was computed for the *all artery sections* case.

was performed to assess classification performance in a manner less dependent on co-registration accuracy than the pixel approach. Different pixel number thresholds were used to classify a whole artery section. For example, a pixel number threshold of 3 meant that if 3 or more pixels from a given artery section were classified as CD68+, then the entire artery section was classified as CD68+.

The optimal lifetime threshold, the area under the ROC curve, accuracy, sensitivity, and specificity were computed to quantify the results of each cross validation run. The optimal lifetime threshold was determined by ROC analysis. The lifetime threshold was varied from 0ns to 15ns in 0.1ns increments, and the optimal threshold was taken as the point on the ROC curve closest to 0 false positive rate and 1 true positive rate (the upper left corner of the ROC plot). In the section classification, one ROC curve was generated for each pixel number threshold. The pixel number threshold was varied as the following Matlab style vector notation describes: [1:20,25:5:50,75,100]. The optimal pixel number threshold was determined by selecting the ROC with the maximum AUC. Then the optimal lifetime threshold was selected as in the pixel classification case. The AUC was calculated by integrating the ROC curve true positive rate with respect to the false positive rate. The trapz() Matlab function was used to compute the AUC. The accuracy, sensitivity, and specificity were computed using the typical confusion matrix approach shown in table 3.1. The accuracy was the number of correctly classified units over the total number of units (units being pixels in the pixel classification case, or artery sections in the section case). Sensitivity was the number of correctly classified CD68+ units divided by the number of true CD68+ units. Similarly, specificity was the number of correctly classified CD68- units divided by the number of true CD68- units.

Table 3.1: Generic confusion matrix, each entry's subscripts i and j ($x_{i,j}$) denote its predicted and true class, respectively. Accuracy was calculated as $(x_{1,1} + x_{0,0})/(x_{1,1} + x_{0,1} + x_{1,0} + x_{0,0})$. Sensitivity was calculated as $x_{1,1}/(x_{1,1} + x_{0,1})$, and specificity was calculated as $x_{0,0}/(x_{0,0} + x_{1,0})$.

	True 1	True 0
Predicted 1	$x_{1,1}$	$x_{1,0}$
Predicted 0	$x_{0,1}$	$x_{0,0}$

3.3 Results and Discussion

3.3.1 Co-registration

Criticism could be made as to the complexity of the various co-registration tasks. Aperio Imagescope was utilized for it's ability to access and annotate the full resolution histology slide images, and Inkscape was used for its superior arbitrary drawing abilities and SVG file format. The digitized histology slide images were large (on the order of $>$300MiB), and required special purpose programs to view them in a responsive manner. Aperio Imagescope was the only program that we could find that allowed for accessing and annotating the digitized slide files in a user friendly and responsive manner. Other free viewer programs existed, but none were as fast or contained the annotation features of Aperio Imagescope. The Python Openslide server did not perform as well with the slide files and did not offer annotations abilities. VIPS [25], using NIP2 as a graphical front end, was also attempted. Rendering speed was decent, but it lacked out of the box annotation functionality. Thus we chose to continue using Aperio Imagescope. Furthermore, we did not wish to impose additional annotation restrictions on the expert reader. For example, asking the reader to label segments as CD68+ or CD68-, or to draw the LEP for easier post processing. We allowed the expert reader to simply draw lines indicating CD68+ without any additional labeling via Aperio Imagescope. Then we traced over the anno-

tations, outlined the lumen, and inserted classification labels in Inkscape. Our approach served to make the processes as easy as possible on the expert reader. Aperio Imagescope stored its annotations (lines and labels) in an xml file which could be parsed and used for post-processing. However, the program did not possess Inkscape's ability to overlay and arrange multiple images and shapes.

The co-registration methods outlined in figure 3.2 provide a precise way of mapping the histology slide to the TD FLIM lifetime map by looking at OCT data that was acquired with each TD FLIM image. However, there was no way to account for the 3D orientation of the artery when the FD FLIM image was acquired (although we tried to keep the orientation the same). This was the main source of the difficulty in the correlation analysis described in the next section. Although the histology annotation protocol allowed for precise annotations, certain arteries were difficult to align with histology due to their orientation when imaging. Arteries with significant plaque content or calcification typically remained in a U-shape during imaging, making the lumen start and end points difficult to ascertain in the OCT and histology images.

3.3.2 Statistical Analysis

3.3.2.1 Correlation

The correlation coefficient provided limited insight into the TD FLIM 494/41nm average lifetime and FD FLIM 525/50nm 40 MHz modulation lifetime correspondence due to imperfect co-registration and homogeneous artery segments. However, the lifetime similarity was obvious when comparing the lifetime maps of each artery. First, examples of arteries and the result of pixel correlation analysis are shown. This illustrated that imperfect co-registration or homogeneous artery segments yielded low correlation coefficients for otherwise very similar TD and FD lifetime maps. Then the correlation analysis described previously in figure 3.4 is presented.

We attempted to analyze the effect of homogeneous lifetimes and imperfect co - registration by splitting the arteries into two groups and applying various amounts of spatial averaging, respectively, and then comparing the TD FLIM versus FD FLIM correlation coefficients. A TD FLIM lifetime threshold of 5.5ns, gleaned from the results of the classification analysis explained later, was used to split the artery segments into two groups. Arteries with at least 25% of their TD FLIM lifetime pixels above 5.5ns were used as the high group (n = 38), and the others were placed into the low group (n = 49). The threshold of 25% was chosen heuristically. Then the lifetime maps were spatially filtered with no filtering, 3 by 3 median filtering, and 9 by 9 median filtering. Figure 3.5 shows the distribution of correlation coefficients between the high and low lifetime groups for the various amounts of median filtering. Figure 3.6 and 3.7 show one artery from the $<$ 5.5 and $>=$ 5.5 groups, respectively. The TD FLIM and FD FLIM lifetime maps are shown in top half of the figures, a difference map is shown in the bottom left, and a two dimensional histogram comparing the TD and FD lifetime maps is shown in the bottom right. Non-overlapping pixels were ignored when generating the difference map and histogram. Figure 3.6 represents a typical homogeneous artery segment with no long lifetime regions. It is visually obvious that there is a high degree of similarity between the TD and FD lifetime maps. However, the homogeneity resulted in a low correlation coefficient even after applying 9x9 median filtering. In contrast, the $>=$ 5.5 artery in figure 3.7 contained one region of elevated lifetimes in the upper middle of the artery. This provided enough lifetime variation to give a good correlation coefficient (> 0.8) without any spatial median filtering, and > 0.9 with spatial median filtering.

Subtle non-trivial displacements made an accurate pixel level TD FLIM versus FD FLIM comparison difficult across our database of arteries. The co-registration methods in this study could only account for two dimensional displacements perpendicular to the optical axis. Recall that the arteries were first measured with the TD FLIM system, and then

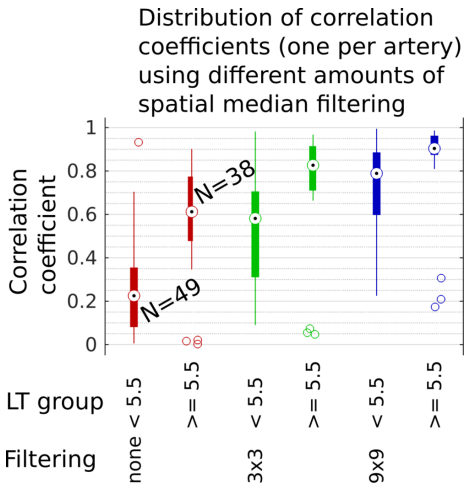

Figure 3.5: Correlation coefficients for pixel comparison of TD FLIM 494/41nm average lifetime and FD FLIM 525/50nm 40 MHz modulation lifetime maps across all arteries. Arteries were divided into two groups according to their TD FLIM 494/41nm average lifetimes. The 38 arteries consisting of lifetimes $>= 5.5$ ns across $>= 25\%$ of their pixels were placed in the $>= 5.5$ group. The remaining 49 arteries were placed in the < 5.5 group.

imaged with the FD FLIM system within 24 hours. This was due to FD FLIM stability concerns as our version of the technology was in its infancy. We often observed changes in the FD FLIM system that required re-calibration if the system was re-initialized. Initialization would happen when switching between acquiring TD FLIM and SS OCT, and acquiring FD FLIM. This is why we did not simply image an artery with the TD FLIM system, and then image with the FD FLIM system without moving the artery. The $>= 5.5$ artery shown in figure 3.8 contains areas that are not spatially aligned despite most of the artery being spatially aligned. A strong qualitative correlation is obvious between the TD FLIM and FD FLIM images. The red arrows in figure 3.8 denote similar areas of relatively long lifetimes. The black arrows and circles denote areas of obvious misalignment in the difference plot.

The correlation plot comparing the TD FLIM 494/41nm average lifetime with the FD

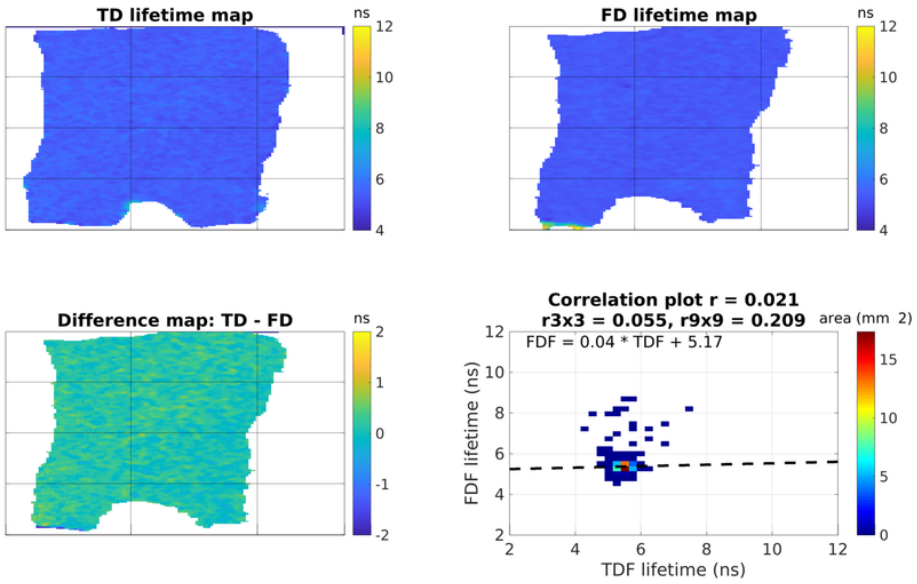

Figure 3.6: Correlation plot for one artery from the < 5.5 group after 3x3 median filtering. Grid lines were plotted every 3mm. The top left plot shows the TD FLIM lifetime map. The top right plot shows the FD FLIM lifetime map. The difference map (TD FLIM lifetime - FD FLIM lifetime) is shown in the bottom left. Finally the two dimensional histogram comparing each TD FLIM pixel with each FD FLIM pixel is shown in the bottom right. The colorscale is labeled in terms of area mm^2. Points in the histogram with 0 were plotted as transparent (white). The minimum square area was one pixel, $60\mu m$ by $60\mu m$. The correlation coefficients for all median filtering runs (none, 3x3, and 9x9) are shown in the title. Despite the low correlation coefficient, it is visibly obvious that the lifetime maps were similar.

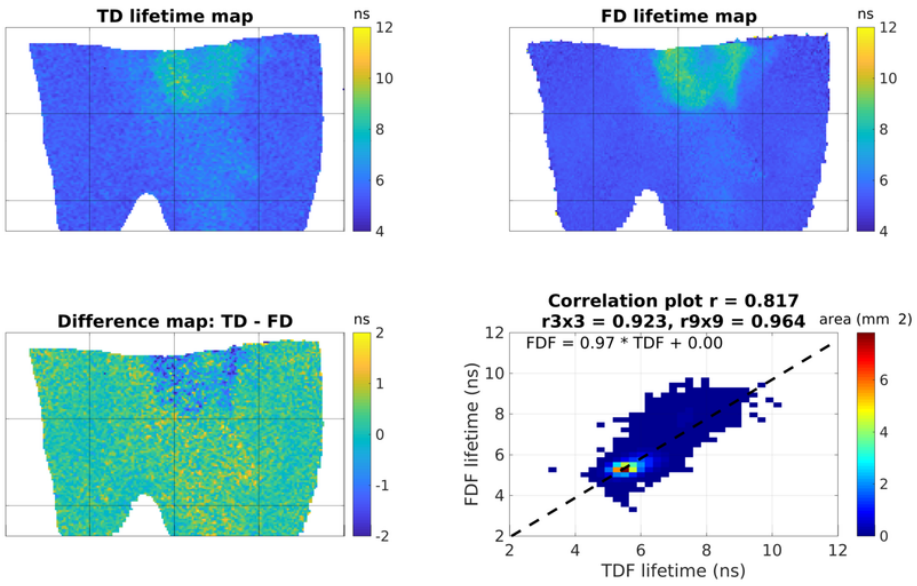

Figure 3.7: Correlation plot for one artery from the >= 5.5 group after 3x3 median filtering. Grid lines were plotted every 3mm. This artery resulted in a relatively high correlation coefficient >0.8 without median filtering, and >0.9 with median filtering. In this case, due to the presence of the long lifetime region, the correlation coefficient provides a good quantification of the strong visible correlation between the TD FLIM and FD FLIM lifetime maps.

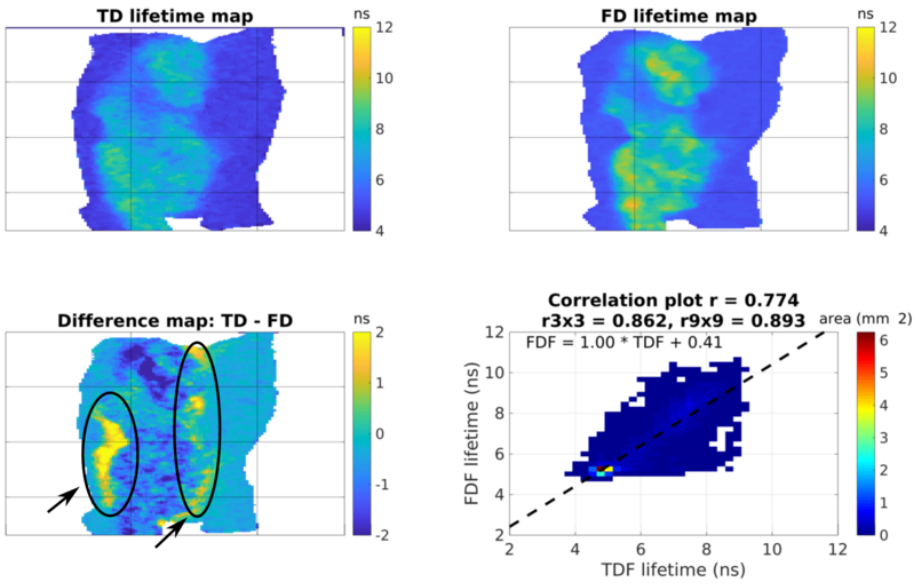

Figure 3.8: Correlation plot for one artery from the >= 5.5 group showing co-registration misalignment (also after 3x3 median filtering). Grid lines were plotted every 3mm. This artery contains areas of co-registration misalignment. The misaligned areas are indicated with a black arrows and circles in the difference map. Qualitatively, it is obvious that there is a strong correlation between the two lifetime maps of this artery. However, subtle non-trivial displacements made an accurate pixel level comparison difficult.

FLIM 525/50nm 40 MHz average lifetime is shown in figure 3.10. This corresponds to the *all artery sections* scope. As expected, there is significantly less variation in correlation coefficients and linear fit coefficients across the no median filtering, 3x3 median filtering, and 9x9 median filtering runs. Although this quantification of correlation is less biased by spatial co-registration misalignment between TD and FD FLIM, it is still biased by the large number of homogeneous arteries with minimal long lifetime regions. Most of the correlation plot straddles the trend line with a slope of 1, but the few artery sections with long median lifetimes have longer FD FLIM lifetimes.

3.3.2.2 Classification

Several lifetime and classification maps are shown in figures 3.11-3.12. These are included to show examples of classification results, and also to support our claims of strong visual correlation between the TD FLIM and FD FLIM lifetime maps, despite minor displacements that made exact spatial co-registration difficult. The ROC curves from the classification analysis are shown in figure 3.17.

The optimal lifetime, AUC, accuracy, sensitivity, and specificity for all classification runs are summarized in table 3.2. Each entry is the median $\pm$ standard deviation. Ideally the pixel and section analysis should be identical. However, imperfect co-registration and possible classification errors contributed to the poorer performance of the pixel level analysis. The leave one subject out and leave one artery out runs performed similarly. This suggests that our database had sufficient information to classify each subject independently. This could also suggest that variations between subjects in the lifetime of CD68+ regions was minimal. Alternatively, improving data-histology co-registration may improve the leave one artery out analysis more as the current results may be limited by co-registration accuracy. Additional spatial averaging was performed to examine the effect of signal quality on classification performance in our database. The cross validation

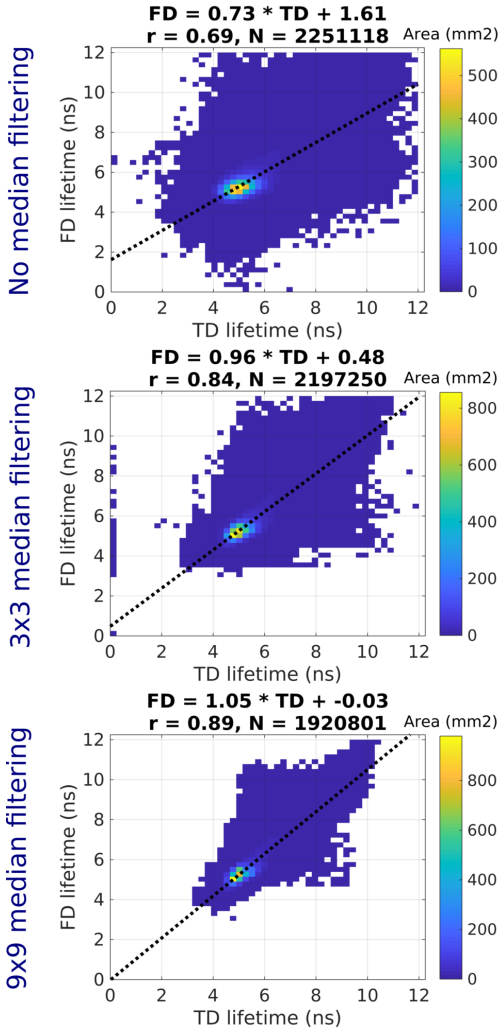

Figure 3.9: 2D histgram correlation plot of the FD FLIM 525/50nm 40 MHz modulation lifetime versus the TD FLIM 494/41nm average lifetime for the *all arteries* approach for various levels of median filtering. The top plot was generated using no median filtering, the middle utilized 3x3 median filtering, and the bottom used 9x9 median filtering.

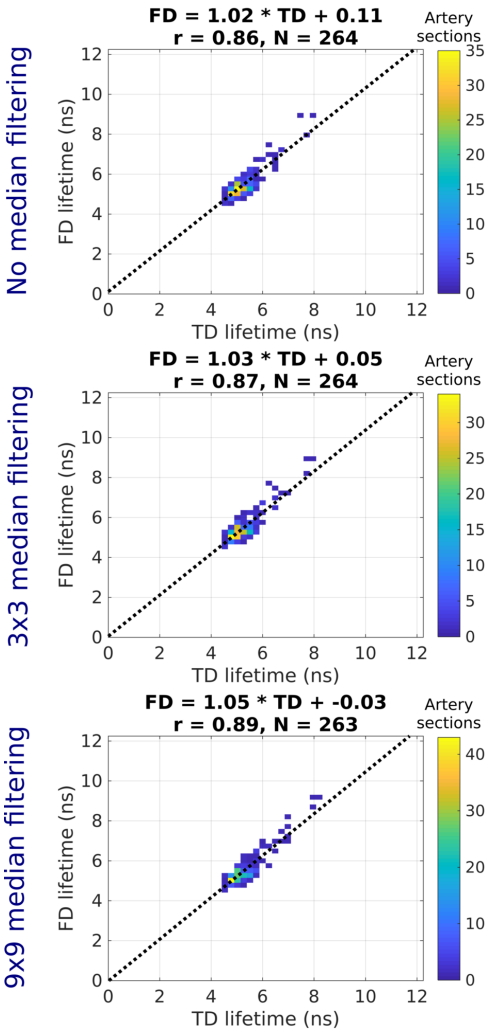

Figure 3.10: 2D histgram correlation plot of the FD FLIM 525/50nm 40 MHz modulation lifetime versus the TD FLIM 494/41nm average lifetime for the median of each artery section for various levels of median filtering. The top plot was generated using no median filtering, the middle utilized 3x3 median filtering, and the bottom used 9x9 median filtering.

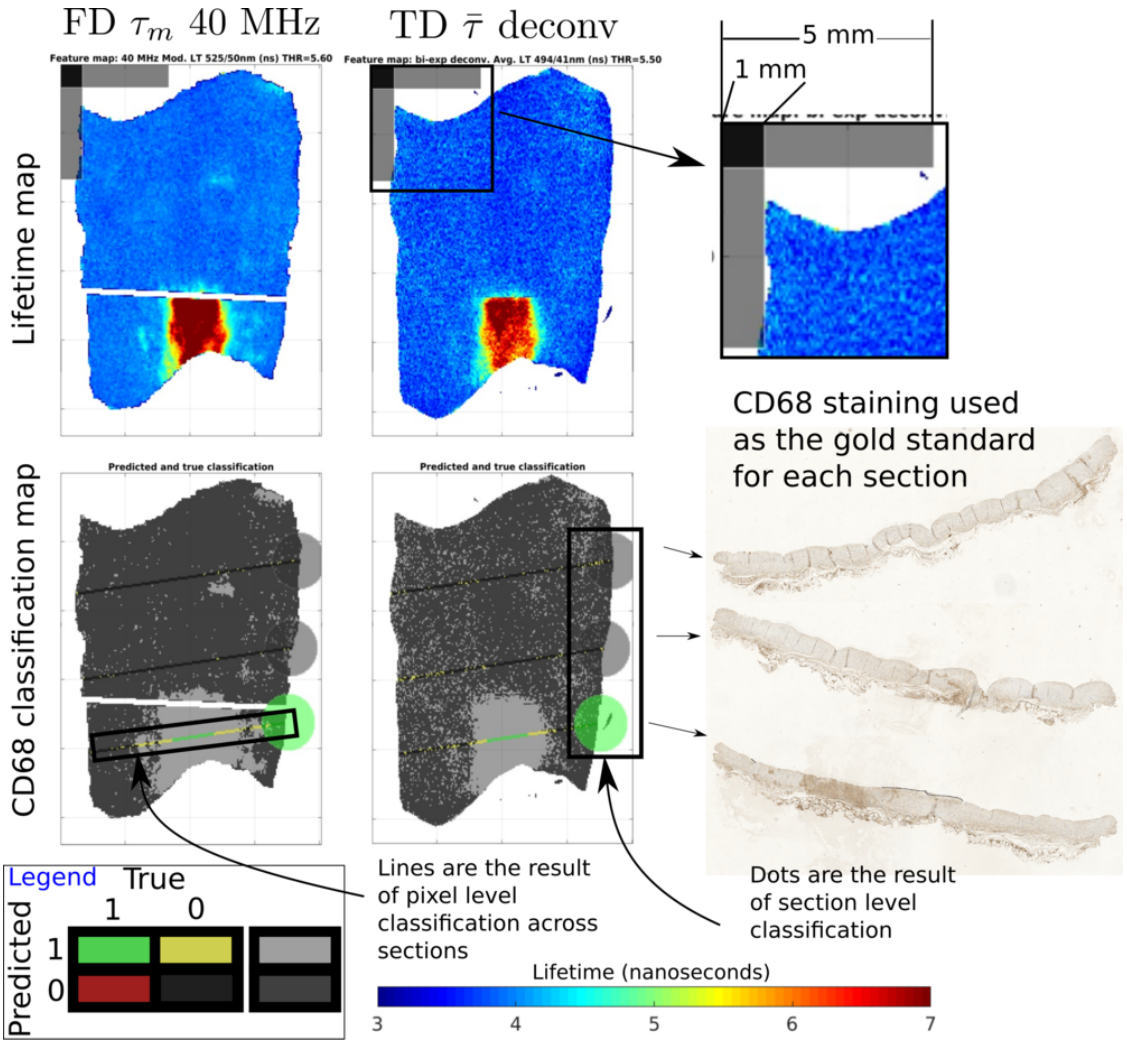

Figure 3.11: Pixel and section CD68 classification results for one artery segment. The FD FLIM τ_m 40 MHz lifetime and classification map are shown on the left side of the figure. The TD FLIM average lifetime ($\bar{\tau}$) and classification map are shown in the center. The legend for the classification map is shown in the bottom left. Section (i.e. histology slice) level classification is indicated by the circles on the right of each classification map, and corresponds to the green (true positive), yellow (false positive), red (false negative), and black (true negative) legend entries. Recall that for section level classification, a cross section was considered true (gold standard) CD68 if any regions were marked CD68+ by the expert reader. Then the section was considered predicted true if a given number of pixels were classified as true (referred to as the pixel level threshold **pix** in table 3.2). Pixel level classification is indicated with the same color scheme to the left of each circle across the approximate location of each histology slice. Pixel level classification across the remaining area of the artery is light gray for predicted true, and dark gray for predicted false.

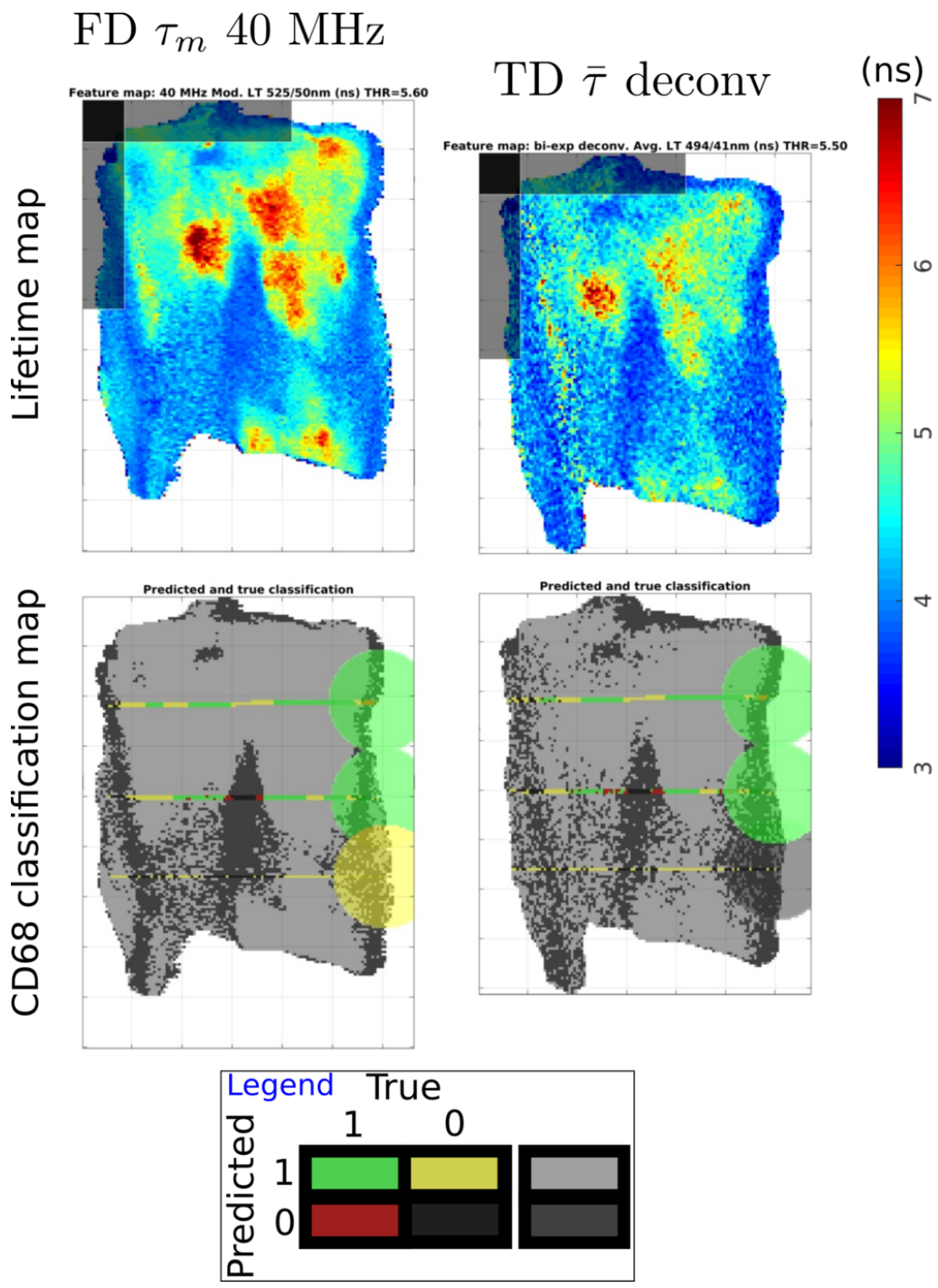

Figure 3.12: Lifetime and classification maps for one artery with significant CD68+ classification. The format is similar to figure 3.11.

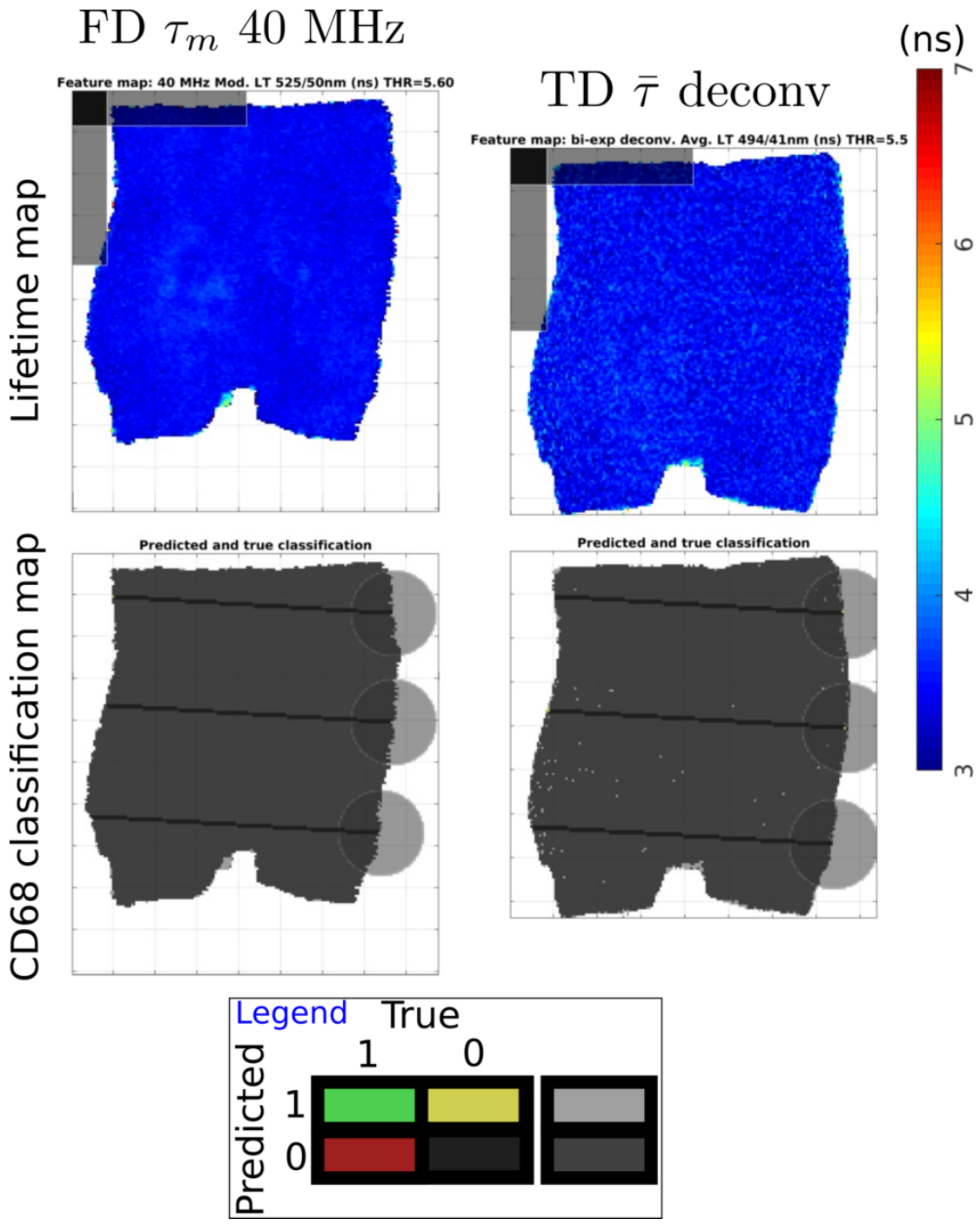

Figure 3.13: Lifetime and classification maps for one artery with no CD68+ classification. The format is similar to figure 3.11.

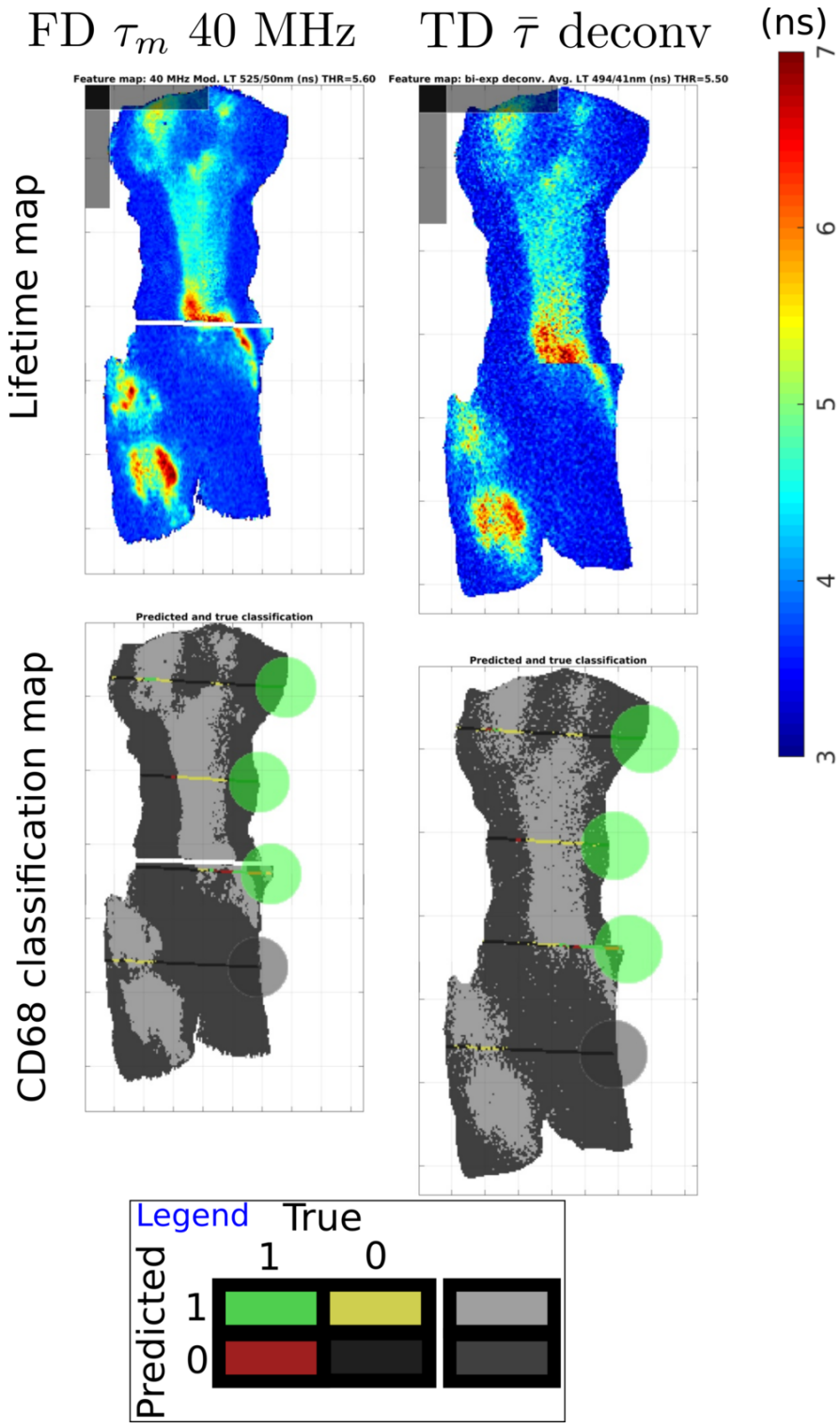

Figure 3.14: Lifetime and classification maps for one artery with no CD68+ classification. The format is similar to figure 3.11.

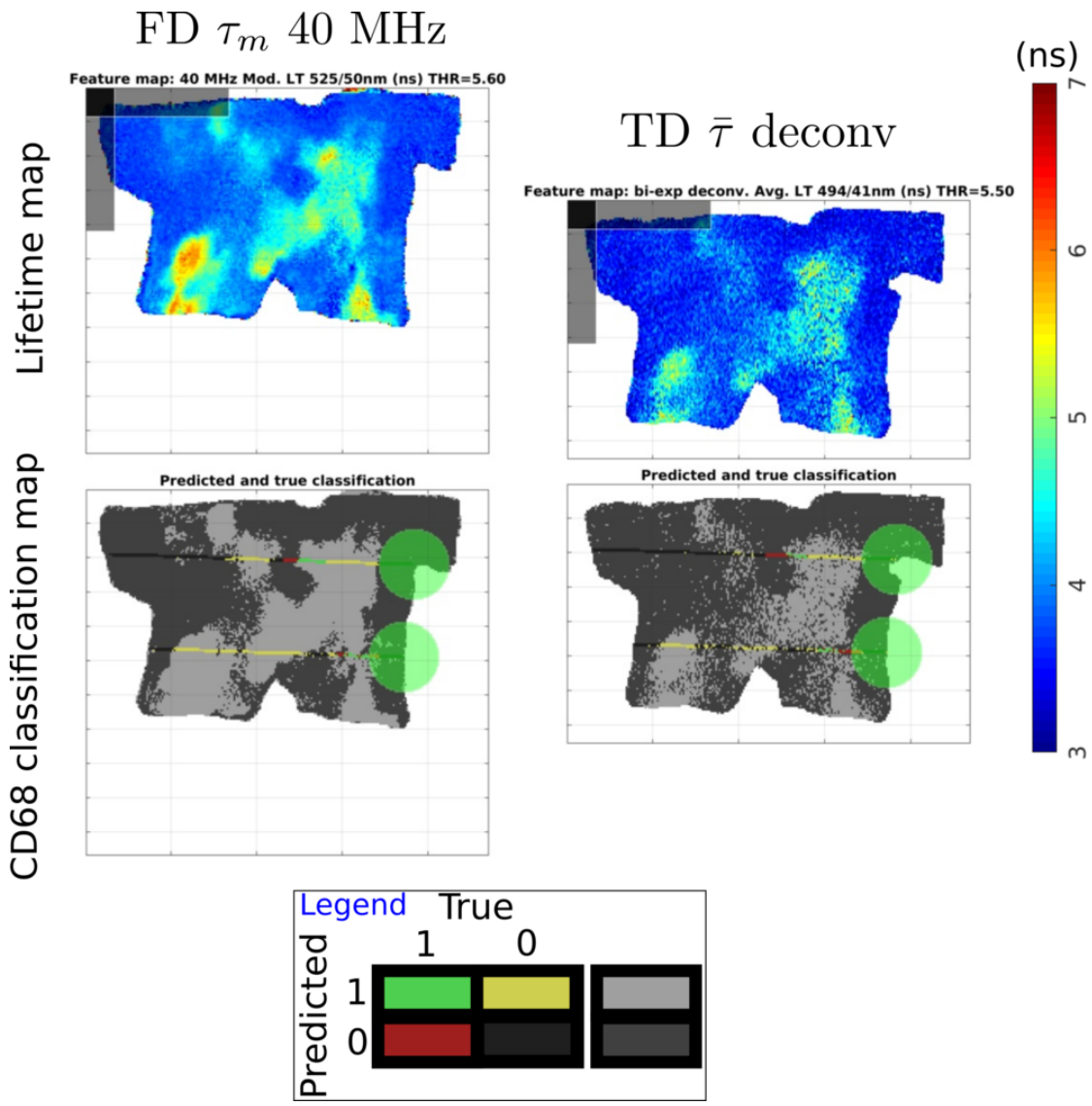

Figure 3.15: Lifetime and classification maps for one artery with significant CD68+ classification. The format is similar to figure 3.11.

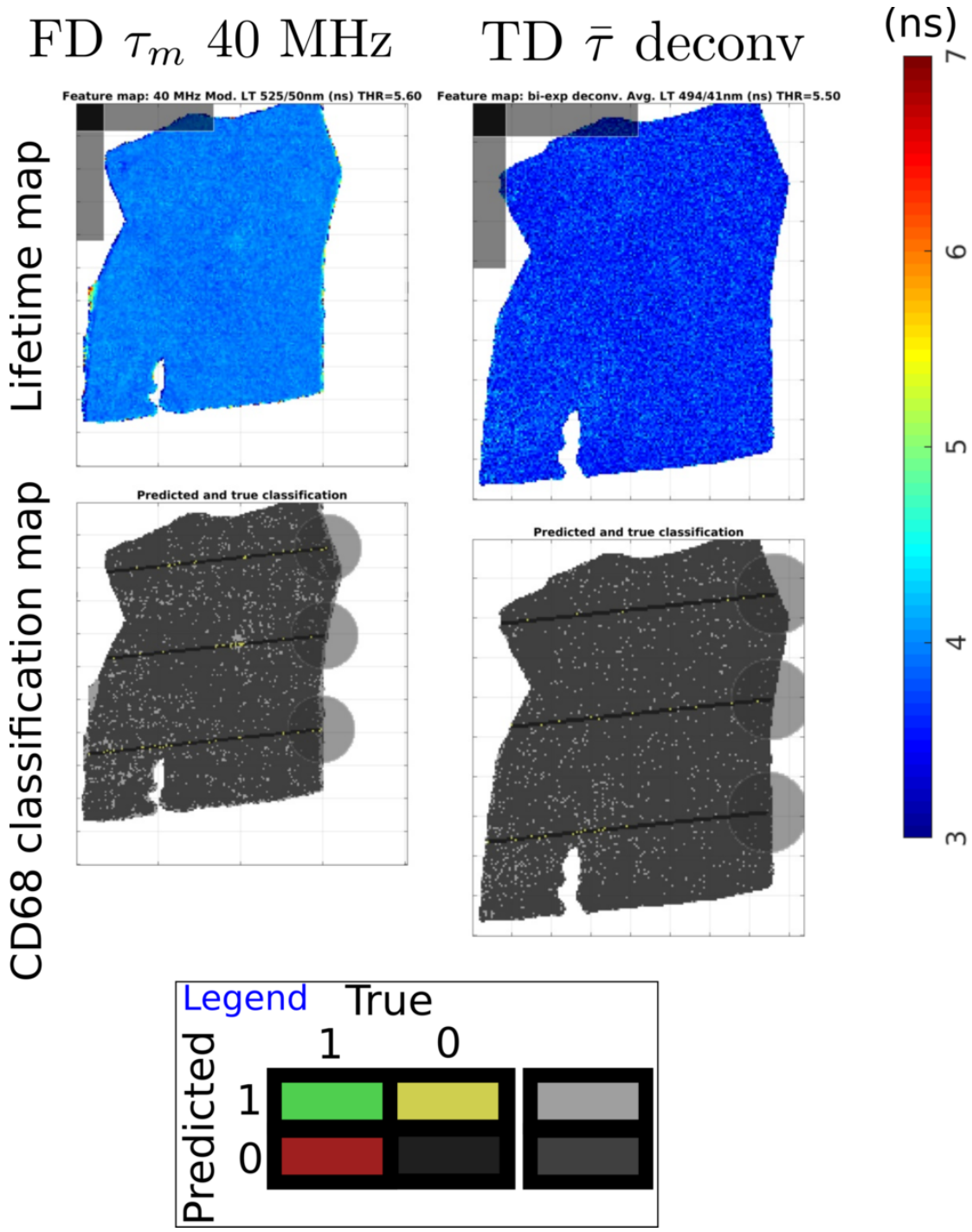

Figure 3.16: Lifetime and classification maps for one artery with significant CD68+ classification. The format is similar to figure 3.11.

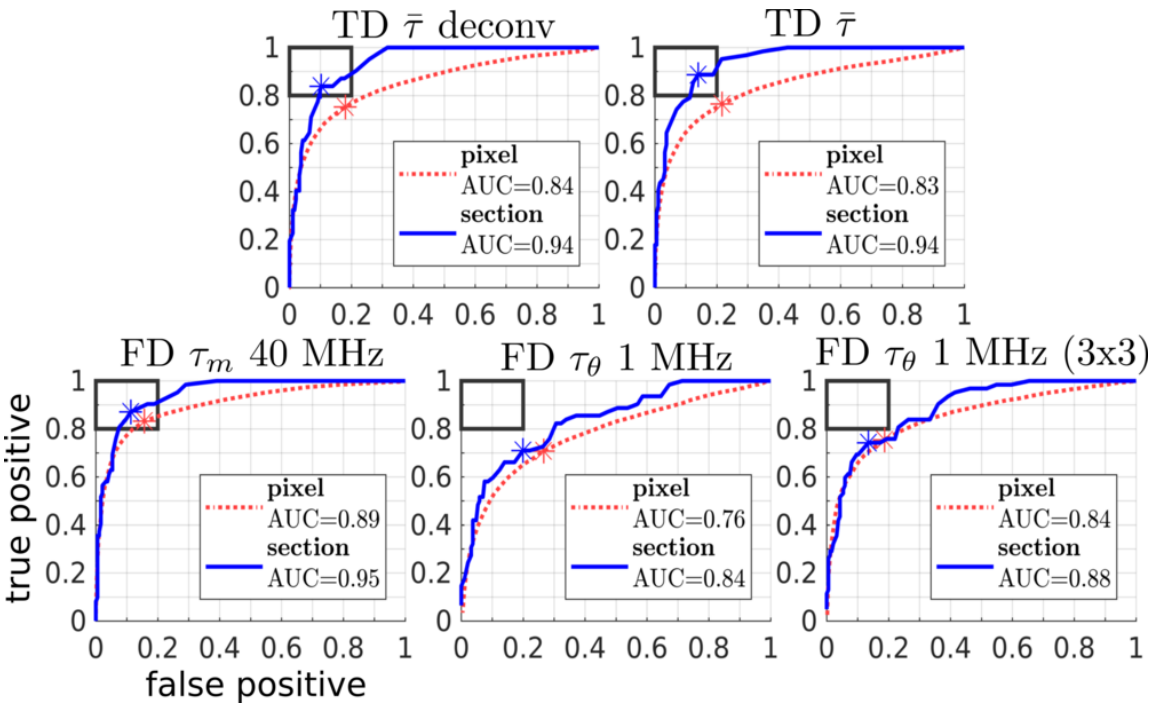

Figure 3.17: ROC curves for classification analysis (pixel and section levels) without leave one out cross validation (i.e. all samples were included, corresponding to the blank LOO rows in table 3.2). A rectangle was plotted in the upper left to highlight the upper 80% sensitivity (true positive) and specificity (1 - false positive) area of the ROC plot to make visually comparing ROC curves easier.

analysis was repeated after spatially median filtering the TD and FD data with various window sizes (none, 3x3, and 9x9). However, this also had the effect of relaxing the data and histology co-registration.

In this study, we only collected at least two modulation frequencies for each artery. Ideally, a more modulation frequencies would have been collected for each FD FLIM image to maximize flexibility in post processing data analysis. Raw time domain data and FPGA processed data were collected in separate images for each image due to the relative newness and uncertainty of the FPGA processing implementation. This necessitated longer data acquisition and save times, as well as larger storage capacity. Thus we chose to only acquire a small subset of possible modulation frequencies. It is likely that more optimal frequency combinations exist, as our choice of modulation frequencies was based on a single exponential model and apriori knowledge that the high lipid regions would have lifetimes >6 ns [3].

An extensive correlation and binary CD68 classification analysis was performed between a TD FLIM system that excited at 355nm and detected at 494/41nm, and an FD FLIM system that excited at 375nm and detected at 525/50nm. For the TD FLIM system, the average lifetimes with and without bi-exponential deconvolution were used, and the 41.67 MHz modulation lifetime and 1.25 MHz phase lifetime were used for FD FLIM. Despite spatial co-registration difficulties, we found a strong correlation between the 41.67 MHz modulation lifetime in FD FLIM (FD τ_m 40 MHz) and the average lifetime after bi-exponential deconvolution in TD FLIM (TD $\bar{\tau}$ deconv). The longer lifetime regions showed longer FD τ_m 40 MHz lifetimes than TD $\bar{\tau}$ deconv lifetimes. This could be due to the longer excitation wavelength (375nm versuss 355nm) used in the FD FLIM system, and the longer emission channel (525/50nm versus 494/41nm) used in the FD FLIM system. Also, the measurement noise tended to be larger in the TD images (most obvious when viewing the classification maps), even though 4 spatial points were averaged prior

to bi-exponential fitting. The decent classification performance of the TD $\bar{\tau}$ and FD τ_θ 1 MHz (3x3) lifetimes suggests that system instrumentation cost could be lowered if only CD68 classification is desired. Further classification analysis is possible with this ex-vivo human coronary artery database. Lipid content is only one aspect of plaque biochemical composition. The FD FLIM system's ability to classify collagen and elastin should also be evaluated against the TD FLIM system and histology to further prove its utility as a lower cost FLIM implementation.

Table 3.2: Classification performance metrics for the leave one out cross validation analyses. The FLIM lifetime feature is listed in the leftmost column. TD $\bar{\tau}$ deconv indicates the TD FLIM system average lifetime with deconvolution, TD $\bar{\tau}$ indicates the TD FLIM system average lifetime without deconvolution, FD τ_m 40 MHz indicates the FD FLIM 40 MHz modulation lifetime, FD τ_θ 1 MHz indicates the FD FLIM 1.25 MHz phase lifetime, and FD τ_θ 1 MHz (3x3) indicates the same 1.25 MHz phase lifetime data with 3 by 3 median filtering. The leave one out type is listed in the next column: empty for nothing left out, art for one artery left out, and sub for one human subject left out. The third column is the scope: pixel or section classification. The remaining columns (left to right) list the number of pixels threshold for artery section classification, the optimal lifetime, accuracy, sensitivity, and specificity for each cross validation type. The values of the last three columns (usually on the interval 0 to 1) were multiplied by 100 to make the table more legible.

	LOO	scope	pix	LT (ns)	acc	sens	spec
TD $\bar{\tau}$ deconv		pixel		5.5	81.5	75.2	82.0
TD $\bar{\tau}$ deconv		section	8.0	6.5	88.3	83.9	89.8
TD $\bar{\tau}$ deconv	art	pixel		5.5 ± 0.0	81.5 ± 0.5	75.2 ± 0.3	82.0 ± 0.5
TD $\bar{\tau}$ deconv	art	section	8.0 ± 0.5	6.5 ± 0.0	88.2 ± 0.3	83.9 ± 0.5	89.7 ± 0.3
TD $\bar{\tau}$ deconv	sub	pixel		5.5 ± 0.0	81.3 ± 1.3	75.4 ± 0.8	81.8 ± 1.4
TD $\bar{\tau}$ deconv	sub	section	8.0 ± 2.7	6.5 ± 0.1	88.2 ± 0.6	83.9 ± 1.1	89.5 ± 0.9
TD $\bar{\tau}$		pixel		7.3	78.2	76.5	78.3
TD $\bar{\tau}$		section	10.0	8.2	86.7	88.7	86.0
TD $\bar{\tau}$	art	pixel		7.3 ± 0.0	78.2 ± 0.8	76.5 ± 0.7	78.3 ± 0.9
TD $\bar{\tau}$	art	section	10.0 ± 0.0	8.2 ± 0.0	86.6 ± 0.3	88.7 ± 0.5	85.9 ± 0.3
TD $\bar{\tau}$	sub	pixel		7.3 ± 0.0	78.8 ± 1.6	75.6 ± 1.6	79.0 ± 1.8
TD $\bar{\tau}$	sub	section	10.0 ± 1.2	8.2 ± 0.1	86.6 ± 0.6	88.5 ± 1.6	86.0 ± 1.0

FD τ_m 40 MHz		pixel		5.6	84.3	83.2	84.4
FD τ_m 40 MHz		section	6.0	6.3	88.3	87.1	88.7
FD τ_m 40 MHz	art	pixel		5.6 ± 0.0	84.2 ± 0.2	83.2 ± 0.5	84.3 ± 0.2
FD τ_m 40 MHz	art	section	6.0 ± 0.0	6.3 ± 0.0	88.2 ± 0.2	87.1 ± 0.5	88.6 ± 0.3
FD τ_m 40 MHz	sub	pixel		5.6 ± 0.0	84.3 ± 1.0	83.3 ± 1.4	84.4 ± 1.1
FD τ_m 40 MHz	sub	section	6.0 ± 5.0	6.3 ± 0.1	88.2 ± 0.5	86.9 ± 1.2	88.6 ± 0.7
FD τ_θ 1 MHz		pixel		8.5	73.2	70.8	73.4
FD τ_θ 1 MHz		section	15.0	11.0	77.8	71.0	80.1
FD τ_θ 1 MHz	art	pixel		8.6 ± 0.1	74.4 ± 0.7	69.9 ± 0.8	74.9 ± 0.9
FD τ_θ 1 MHz	art	section	15.0 ± 0.0	11.0 ± 0.2	77.6 ± 2.5	71.1 ± 5.2	79.8 ± 5.0
FD τ_θ 1 MHz	sub	pixel		8.6 ± 0.0	74.3 ± 0.8	69.6 ± 1.3	74.7 ± 0.8
FD τ_θ 1 MHz	sub	section	15.0 ± 0.0	11.0 ± 0.3	77.1 ± 3.0	73.2 ± 5.3	78.9 ± 5.6
FD τ_θ 1 MHz (3x3)		pixel		8.4	80.9	75.5	81.3
FD τ_θ 1 MHz (3x3)		section	10.0	10.1	83.5	74.2	86.6
FD τ_θ 1 MHz (3x3)	art	pixel		8.4 ± 0.0	80.9 ± 0.3	75.5 ± 0.5	81.3 ± 0.3
FD τ_θ 1 MHz (3x3)	art	section	10.0 ± 1.1	10.1 ± 0.2	83.3 ± 1.2	74.2 ± 1.7	86.4 ± 2.0
FD τ_θ 1 MHz (3x3)	sub	pixel		8.4 ± 0.1	80.7 ± 1.2	75.7 ± 0.9	81.1 ± 1.3
FD τ_θ 1 MHz (3x3)	sub	section	10.0 ± 1.9	10.1 ± 0.3	83.1 ± 2.2	74.2 ± 3.2	86.3 ± 3.8

4. DUAL MODALITY SWEPT SOURCE OPTICAL COHERENCE TOMOGRAPHY AND FREQUENCY DOMAIN FLUORESCENCE LIFETIME IMAGING SYSTEM

4.1 Introduction

A combination of the FD FLIM system mentioned in chapter 2, and the SS OCT system mentioned in [24] (with a different laser) is described here. In short, the TD FLIM subsystem from [24] was replaced with the FD FLIM subsystem. This was fairly straightforward, but introduced additional synchronization issues. This chapter describes the instrumentation of the FD FLIM and SS OCT system, additional synchronization concerns, and results from imaging reference fluorophores and ex-vivo human coronary arteries to demonstrate the system's co-registration and sensitivity.

4.2 Materials and methods

4.2.1 Instrumentation

The SS OCT and FD FLIM system will be described in 3 parts: the SS OCT subsystem, the FD FLIM subsystem, and the common components.

4.2.1.1 SS OCT

A schematic of the FD FLIM and SS OCT system is shown in figure 4.1. A 1300 nm swept source laser was used as the light source for the SS OCT system: Thorlabs SL1310V1-10048, 100 kHz unidirectional sweep rate (decimated to 50 kHz, as explained later in this section), 100 nm tuning range, $>$ 100 mm coherence length in air. The remainder of the SS OCT system is identical to [24]. The fiber coupled laser output connected to a Mach-Zehnder interferometer (MZI) (Thorlabs, INT-MZI-1300) to generate a wavelength reference for k-space linearization. The electrical output of the MZI was connected to a

low pass filter (Minicircuits BLP-150+, DC-140 MHz), an attenuator (Minicircuits HAT-8+) to avoid saturating the digitizer, and then to AI1 of the SS OCT digitizer and FPGA (National Instruments NI 5772 and PXIe7966, respectively). ***Reference arm*:** The optical output of the MZI was connected to a 2x2 90:10 coupler (Thorlabs 10202A-90-APC) to split the swept source light into the sample and reference arms of the OCT system. The 10% output of the coupler was connected to an attenuator (Thorlabs VOA50-APC) to allow for optimizing the power in the reference arm. The attenuator was connected to an optical delay line (ODL) (General Photonics VDL-001-35-60-FC/APC/APC-SS) that allowed for fine tuning of the reference arm length to match the sample arm length. The output of the ODL was routed through a polarization controller (model similar to the Thorlabs FPC030) and connected to port 1 of a circulator (Thorlabs CIR1310PM-APC). The output of port 2 of the circulator was collimated with L1 and focused onto a mirror M1 with another lens (not shown). The distance between L1 and M1 was set such that the reference and sample arm path lengths in air were approximately matched. The light reflected from M1 was coupled back into port 2 of the circular, emerged from port 3, and was connected to a 50:50 2x2 coupler (General Photonics, NPMC-22-F-13-50/50-FC/APC). ***Sample arm*:** The 90% output of the first couplier (after the MZI) was connected to port 1 of a circulator. The infrared light output from port 2 was collimated with L3 and directed to DM2 in the common portion the system. The light returning from the common subsystem reentered port 2, exited port 3, was routed through a polarization controller, and was connected to the same 50:50 2x2 coupler as the light exiting out of port 3 of the reference arm circulator. The outputs of the 50:50 coupler (recombined the sample and reference arms) were connected to a balanced photodetector (Weiserlabs WL-BPD600MA). The electrical output of the balanced detector was connected to a low pass filter (same model as used for the MZI) and then to AI0 of the NI 5772 and PXIe 7966 digitizer and FPGA. The analog signals from the MZI and balanced detector were digitized at 12-bit 800 MS/s after detect-

ing a rising edge trigger from the swept source laser's sweep trigger output. The sweep trigger was also digitally sampled at 800 MS/s in the same clock domain as the ADCs. The MZI and balanced detector signals were then decimated (1 out of every 2 data points was omitted) to 400 MS/s in the PXIe 7966 FPGA before they were transferred to the host computer.

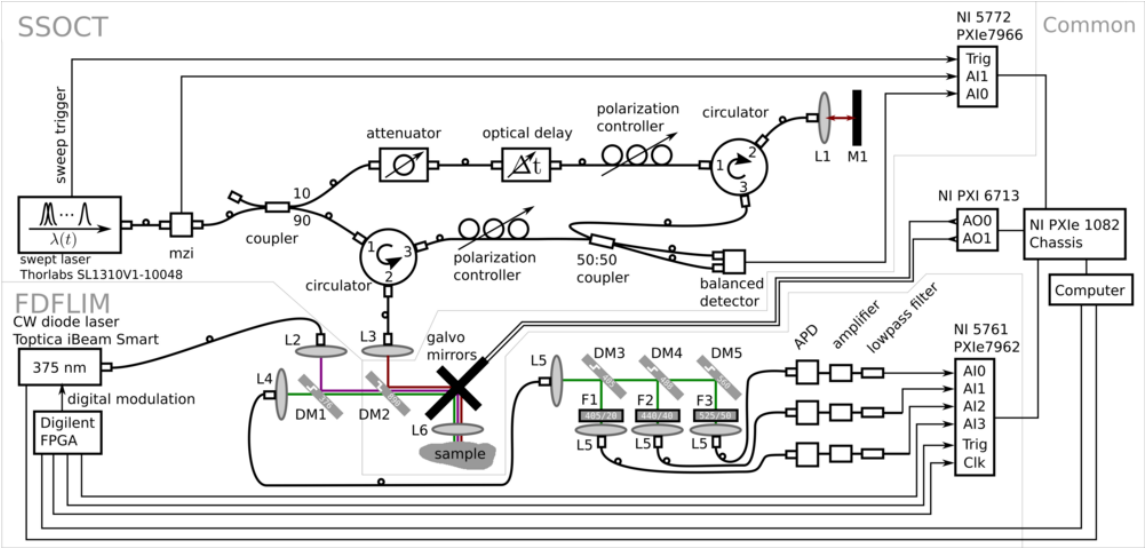

Figure 4.1: Schematic of dual modality SS OCT and FD FLIM system.

An effective sweep rate of 50 kHz was used during imaging in order to avoid overflow when transferring data between the PXIe 7966 FPGA to the host computer. The SS OCT FPGA, analog output card, and FD FLIM FPGA were all housed in a PXIe chassis (National Instruments NI PXIe 1082). Each slot in the chassis had a maximum chassis/computer transfer speed of 2 GiB/s. However, the PXIe 7966 had a maximum FPGA/computer transfer speed of 800 MiB/s (the slowest and therefore rate limiting component in this case). At 50 kHz (assuming a 100% duty cycle for the SS OCT laser), the throughput was almost equal to 800 MiB/s (shown below, 12-bit analog input data was

transferred as 16-bit).

$$\underbrace{16}_{\text{bits/point}} \cdot \underbrace{4000}_{\text{points per sweep}} \cdot \underbrace{2}_{\text{AI channels}} \cdot \underbrace{50e3}_{\text{sweeps per second}} \div \underbrace{8}_{\text{bits/byte}} \div \underbrace{2^{20}}_{\text{bytes/MiB}} \approx 763 \text{ MiB/s}$$

4.2.1.2 FD FLIM

A 375nm CW diode laser (Toptica iBeam Smart 375) was digitally modulated at frequencies between 1 MHz and 100 MHz by an atrix-7 FPGA (Digilent Arty Atrix-7). The diode laser was coupled into a single mode fiber (2.5μm mode field diameter) and collimated by L2 (Thorlabs A240TM-A) as shown in figure 4.1. The 375nm excitation light reflected off of DM1 (Semrock FF376-Di01), passed through DM2, and was scanned across the sample by the galvanometer mirrors and objective lens L6. The fluorescence emission was collected by L6 and coupled into a multimode fiber (200μm core diameter) by L4. The emission was collimated by L5 (Thorlabs F220-SMA-A) and split into three spectral channels by a series of dichroic mirrors and bandpass filters: DM3 (Chroma ZT405rdc) and F1 (Chroma ZET405/20x), DM4 (Chroma ZT488rdc) and F2 (Semrock FF01-440/40), and DM5 (Chroma T560LPXR) and F3 (Semrock FF03-525/50). The bandpass filters resulted in the following spectral channels: 405/20nm, 440/40nm, and 525/50nm (center/FWHM). The emission for each channel was coupled into a multi mode fiber (Thorlabs M25L02, 200μm core) by L5 and secured in close proximity to a fixed gain avalanche photodiode (Hamamatsu C12701-11). The output of each APD was connected to a fixed gain amplifier (Minicircuits ZFL-500LN+), then to a lowpass filter (Minicircuits BLP-90+), and finally to an analog input channel of the NI 5761 (250 MS/s) and PXIe 7962 digitizer and FPGA. The digitizer was configured to lock its sample clock to a 10 MHz reference clock provided by the atrix-7 FPGA.

4.2.1.3 Common

The SS OCT light and FD FLIM 375nm excitation light were combined onto the same optical path by DM2 (Semrock FF01-890). The combined light was then directed towards a pair of galvanometer scanning mirrors (Cambridge Technology 6220H), and scanned across an objective lens (Edmund Optics 64-837) to image the sample.

With respect to the first generation of TD FLIM and SS OCT system, additional synchronization steps were implemented due to the free running nature of both the SS OCT and FD FLIM systems. In the previous generation of the TD FLIM and SS OCT system, the SS OCT laser's sweep trigger was responsible for timing the analog outputs for galvanometer scanning and triggering of each TD FLIM laser pulse. This was accomplished by having the NI 5772 and PXIe 7966 (SS OCT digitizer and FPGA) control the start trigger and sample clock of the analog output card (PXI 6713) via two PXI trigger lines (in the backplane of the PXIe 1082 chassis).

Custom data acquisition and control was implemented in LabVIEW FPGA 2014 with Xilinx tools version 14.7 (on the PXIe 7966 and PXIe 7962 FPGAs) and LabVIEW 2014 (on the host computer). The host computer consisted of an ASRock X79 Extreme6 (motherboard), 56 GB of RAM (7x8 GB Corsair CMZ64 GX3M 8A18 66C9), and an Intel i7-4820k 3.70 GHz processor. For the remainder of this discussion the PXIe 7966 and PXIe 7962 will be referred to as the SS OCT FPGA and the FD FLIM FPGA, respectively. An outline of the additional synchronization steps required for the combined SS OCT and FD FLIM system operation is listed below.

1. The user armed the imaging system on the host computer.

2. The host computer then configured (set various parameters such as image size, etc.) and armed the FD FLIM FPGA.

3. The FD FLIM FPGA was configured to wait for a secondary trigger from the SS OCT FPGA to ensure that both FPGAs began data acquisition simultaneously.

4. Then the host computer configured and armed the SS OCT FPGA.

5. Once a swept source trigger was detected by the SS OCT FPGA, a trigger was sent to the FD FLIM FPGA via a PXI trigger line (in the backplane of the PXIe 1082 chassis), indicating that FD FLIM should begin acquiring data.

6. The SS OCT FPGA began collecting data after triggering the FD FLIM FPGA.

7. The FD FLIM FPGA then began acquiring data once a trigger was detected from the Arty Atrix-7 FPGA.

The SS OCT and FD FLIM systems operated independently once the initial triggering sequence was completed. The SS OCT and FD FLIM image pixel sizes and FD FLIM trigger rate were set so that there were four SS OCT pixels for a given FD FLIM pixel in the fast axis of the raster scan (similar to the previous generation TD FLIM and SS OCT system). The pixel spacing in the slow raster scan axis was equal. This gave a FD FLIM trigger rate of 12.5 kHz.

4.2.2 Data processing

4.2.2.1 SS OCT

Each backscattered spectral interferrogram was processed on the host computer in a manner similar to [24] and [26]. The various processing steps are outlined below.

1. Background subtraction using the average of all interferograms for a given volume.

2. Cubic spline interpolation, each interferogram was linearized in k-space using its co-acquired mzi interferogram signal as a phase reference.

3. Dispersion compensation using a pre-computed phase correction calculated using data from a mirror at the sample position.

4. Discrete Fourier transform, only the single sided depths were used (indices 0 to N/2 - 1, where N was the number of time points in the linearized interferogram).

4.2.2.2 FD FLIM

The digitized fluorescence emission for each spatial pixel was processed exactly as described in section 2.3.2.

4.2.3 Sample preparation and imaging

The data presented in this study utilized the same sample preparation protocol as section 2.3.4 for fluorescent dyes and ex-vivo human coronary arteries. The same fluorescent dyes (concentrations and solvents), but different ex-vivo human coronary arteries were used in this study.

4.3 Results and discussion

4.3.1 Capillary tubes filled with fluorophores

NADH, POPOP, FAD, anthracene, DPA, and 9CA were loaded into quartz capillary tubes and imaged to demonstrate the accuracy of the FD FLIM system and the co-registration between the SS OCT and FD FLIM systems. The fluorophore image is shown in figure 4.2, and the lifetime values are summarized in table 4.1. The SS OCT volume of the same capillary tubes is shown in figure 4.3, along with a bscan of a finger. The FD FLIM phase lifetime, τ_ϕ, at 41.67 MHz in the 525/50nm emission channel is shown on top of the volume and in the upper part of the bscan.

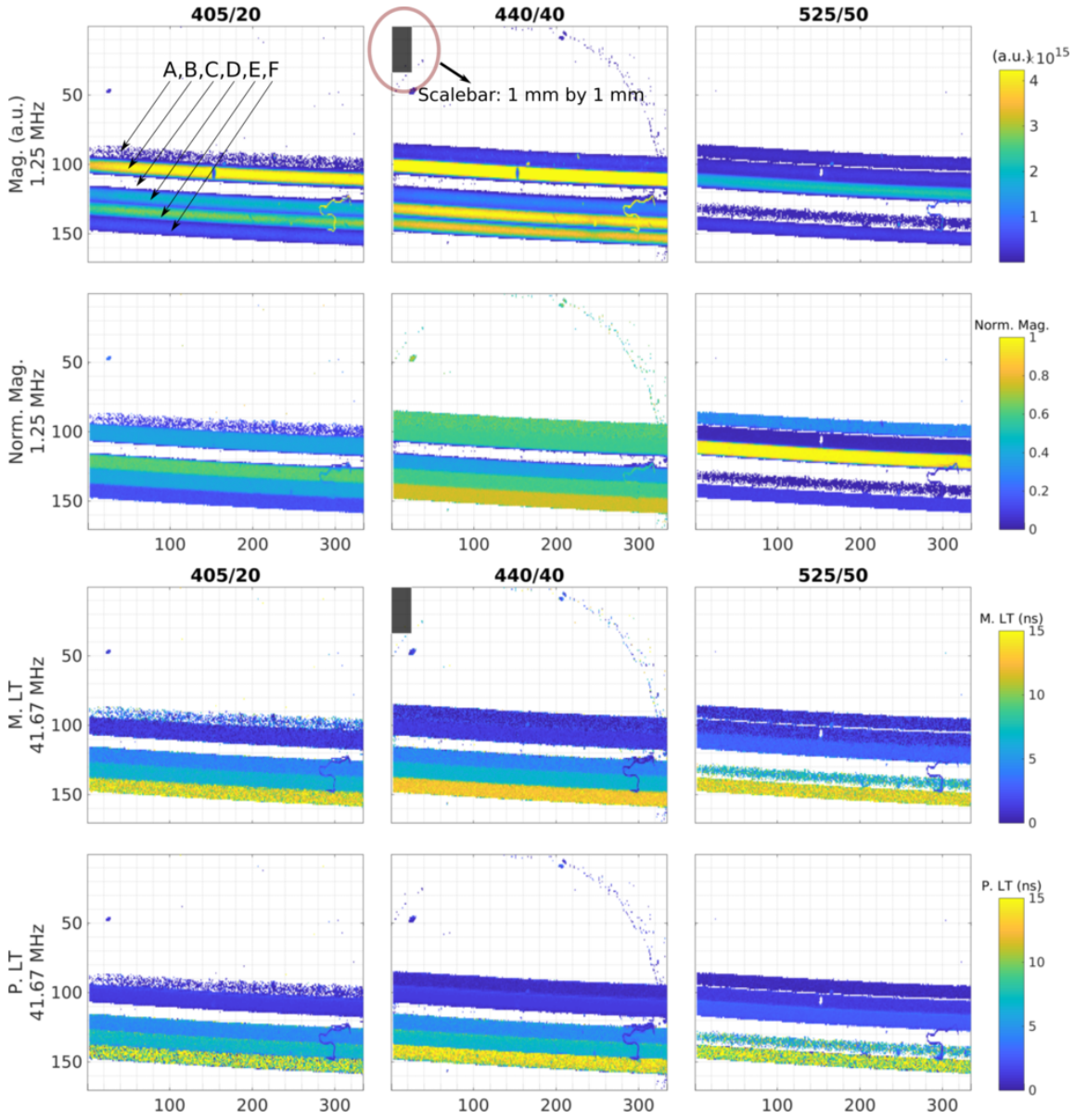

Figure 4.2: Image of fluorophores in quartz capilary tubes. A - NADH, B - POPOP, C - FAD, D - anthracene, E - DPA, F - 9CA. See section 2.3.4 for concentration and solvent information. The top row of maps contain the absolute intensities (amplitude at 1.25 MHz). The second row of maps contain the normalized intensities, but no spectral calibration was performed (i.e. to make the intensity ratios true between channels). The last two rows contain the magnitude and phase lifetimes (M. LT $\equiv \tau_m$ and P. LT $\equiv \tau_\phi$) at 41.67 MHz. Each column of maps is a spectral channel, denoted at the top as center wavelength/FWHM in nanometers. Although only lifetimes at 41.67 MHz are shown, 20.83 MHz , 62.50 MHz, and 83.33 MHz were also collected simultaneously in each pixel of the image.

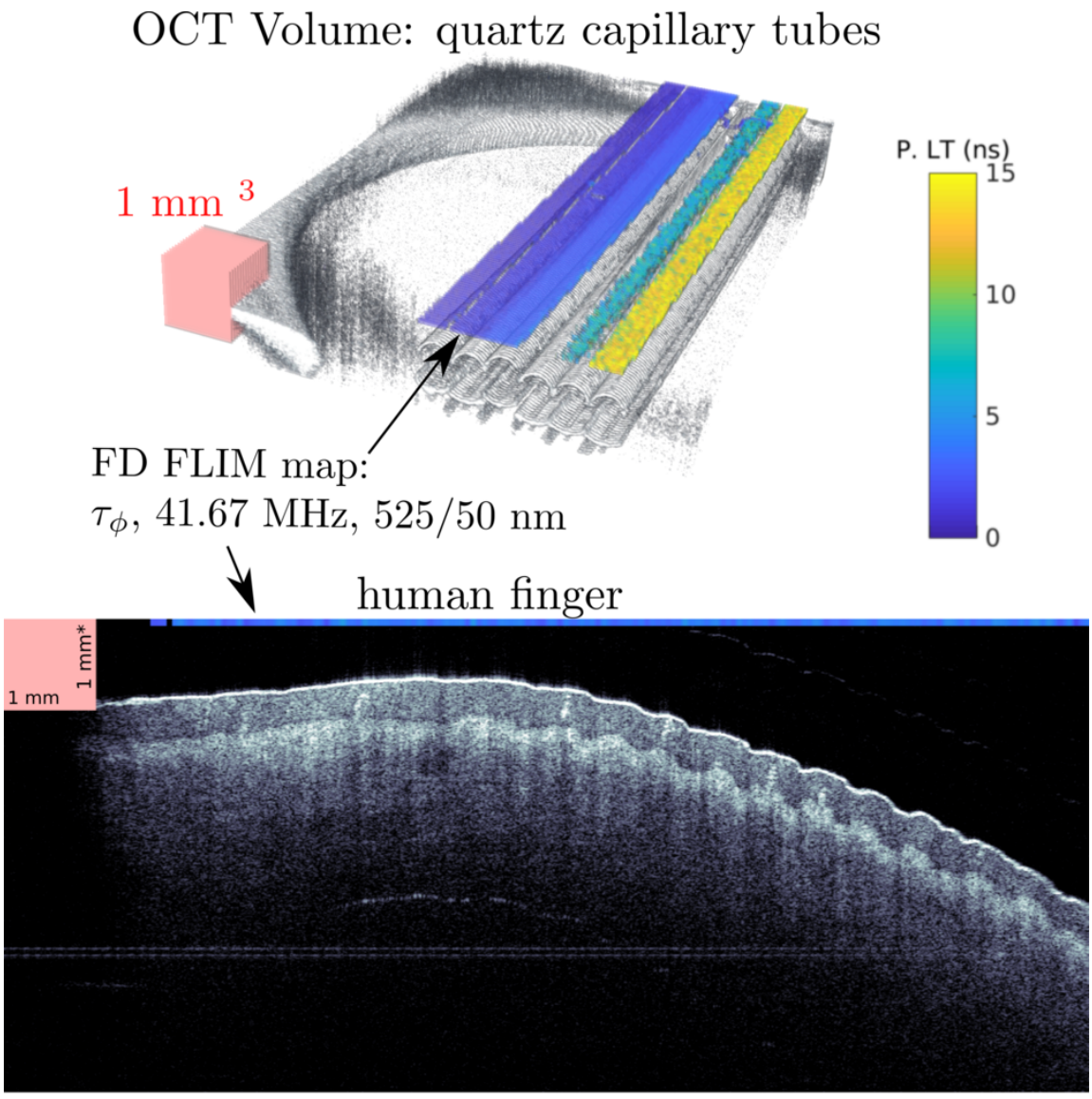

Figure 4.3: SS OCT volume of fluorophores in quartz capilary tubes with FD FLIM lifetime map overlaid on top (upper half) and SS OCT bscan with FD FLIM lifetime trace overlaid on top (bottom half).

Table 4.1: Fluorophore lifetime values corresponding to figure 4.2. A mask was manually drawn around each capillary tube during post processing to select only pixels for a given fluorophore. Lifetime values are listed for each fluorophore for one frequency across only spectral bands with sufficient signal intensity (ex: FAD only has significant emission in the 525/50nm channel). The modulation frequency shown for each fluorophore was selected based on the lowest lifetime standard deviation (ex: 20.8 MHz was optimal for 9CA because more lifetime variation was present at 41.7 MHz, 62.5 MHz, and 83.3 MHz frequencies, as expected based on the ≈12 ns lifetime). The literature lifetimes were take from the following sources: NADH [18], POPOP and FAD [19], DPA [21], and 9CA and anthracene [20].

Fluorophore	Solvent	Concentration	Phase lifetime			Modulation lifetime			Frequency (MHz)	Literature lifetime
			405/20	440/40	525/50	405/20	440/40	525/50		
A. NADH	PBS	1mM			0.593 (0.230)				83.3	0.44
B. POPOP	Ethanol	< 0.01mM	1.35	1.35	1.35	1.35	1.35	1.35	ALL	1.2-1.5
C. FAD	PBS	1mM			2.407 (0.188)			2.796 (0.195)	83.3	2.3-2.85
D. Anthracene	Ethanol	< 0.01mM	4.838 (0.726)			4.406 (0.521)			62.5	5.1 (0.3) Methanol
E. DPA	Ethanol	< 0.01mM		6.637 (0.280)	6.715 (1.594)		6.791 (0.445)	6.933 (2.590)	20.8	8.14
F. 9CA	Ethanol	< 0.01mM		12.421 (2.255)	12.543 (2.748)		12.918 (0.742)	12.805 (4.431)	20.8	12.7 (0.7)

mean (standard deviation) Lifetimes in nanoseconds

4.3.2 Ex vivo human coronary artery

63 ex-vivo human coronary artery segments from 20 human subjects were imaged with the dual modality SS OCT and FD FLIM imaging system. Images of two human coronary arteries, one with no positive CD68+ regions and one with positive CD68+ regions, are shown in figures 4.4-4.5 and 4.6-4.7, respectively.

4.3.3 Conclusion and future work

The synchronization scheme described here ensured $\leq$1 FD FLIM pixel of jitter between the start of the SS OCT acquisition and FD FLIM acquisition. This start jitter manifests itself as apparent physical misalignment between the SS OCT and FD FLIM images in the fast scanning dimension (with respect to the galvonometer mirrors). Significantly less jitter ($\ll$1 FD FLIM pixel) could be achieved with minor modifications to the FD FLIM driving firmware on the Arty Atrix-7 FPGA. Instead of operating at a fixed pixel rate (12.5 kHz in this study), FD FLIM could be configured to acquire a pixel for a fixed time interval and then wait for the next SS OCT sweep trigger before beginning

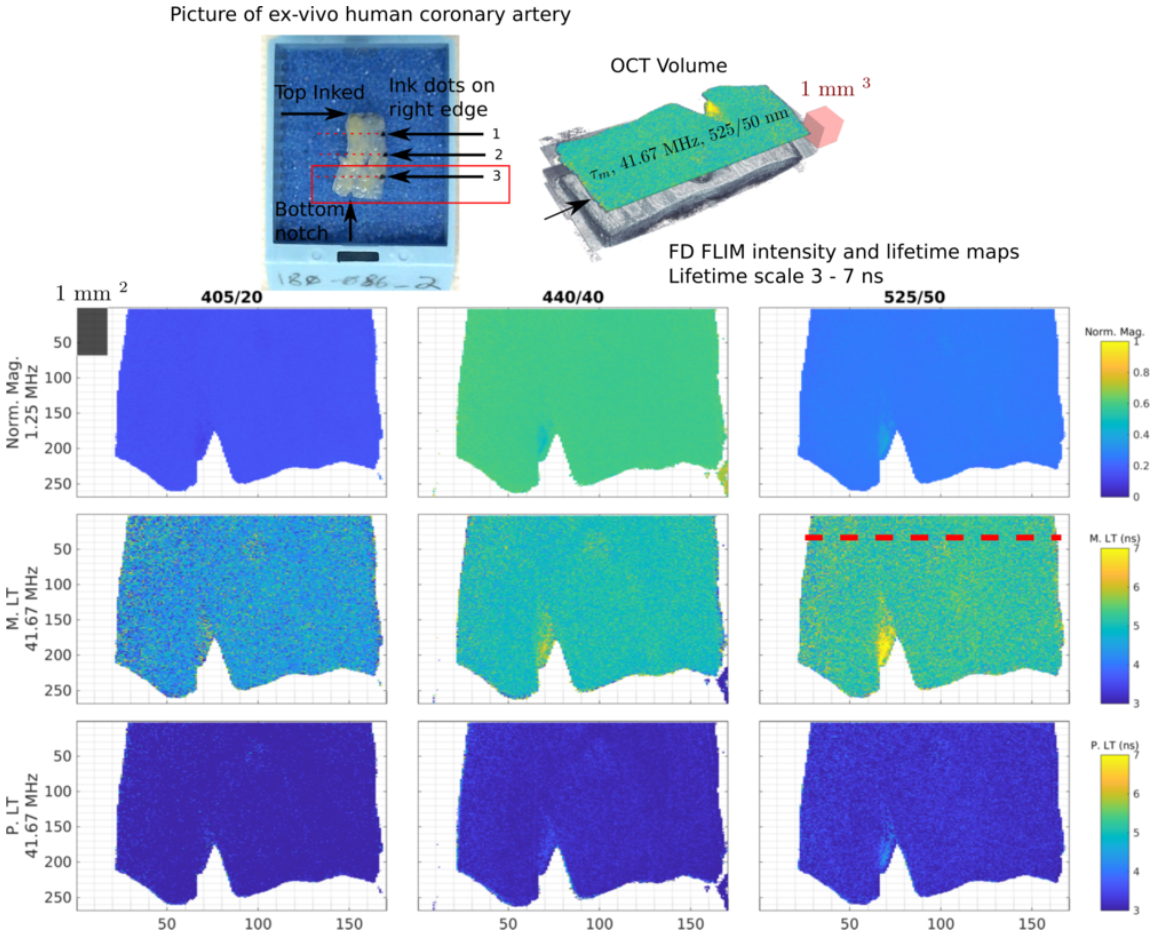

Figure 4.4: One SS OCT and FD FLIM acquisition of an ex-vivo human coronary artery showing no CD68+ staining. A picture of the artery is shown in the top left, along with annotations used for histology co-registration (one arrow per histology slice). A three dimensional rendering of the SS OCT volume with the FD FLIM τ_m 41.67 MHz lifetime is shown to the right of the artery picture. The black arrow indicates the approximate location of the histology slice and SS OCT bscan cross sections shown in figure 4.5.

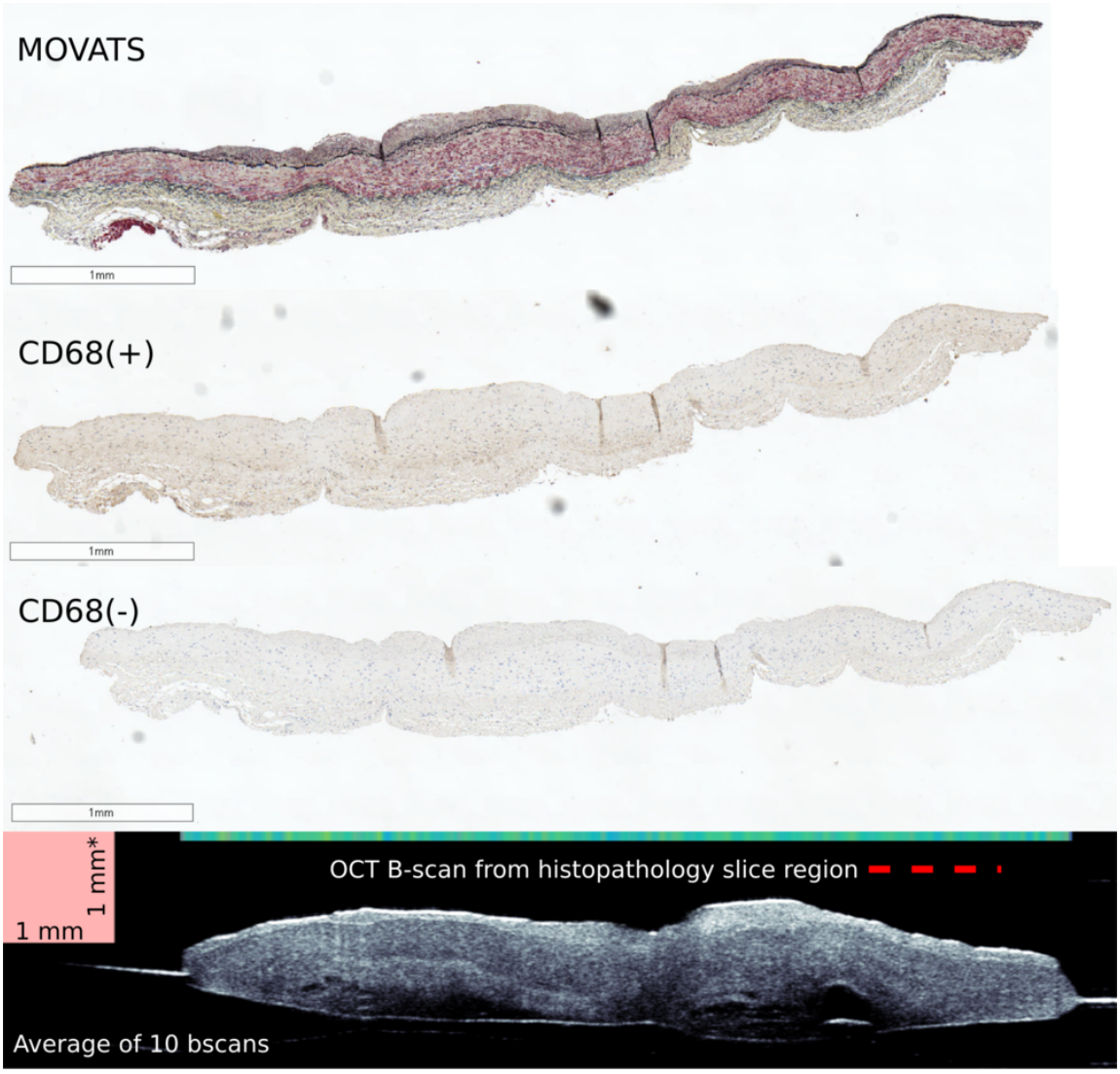

Figure 4.5: Histology and SS OCT bscan images from the red dotted line area of the ex-vivo human coronary artery shown in figure 4.4. The scalebars in the histology images are 1mm in the horizontal dimension. The FD FLIM τ_m 525/50nm 41.67 MHz lifetime is plotted as a color ribbon at the top of the SS OCT bscan, with the same colorscale as the lifetime maps in figure 4.4. The lack of excessive brown areas in the CD68(+) image suggest that this region of this artery did not contain any macrophages. The lifetime values of approximately 5 ns support this conclusion. We would expect a lifetime closer to 6 ns to be correlated to negative CD68+ staining (see table 3.2).

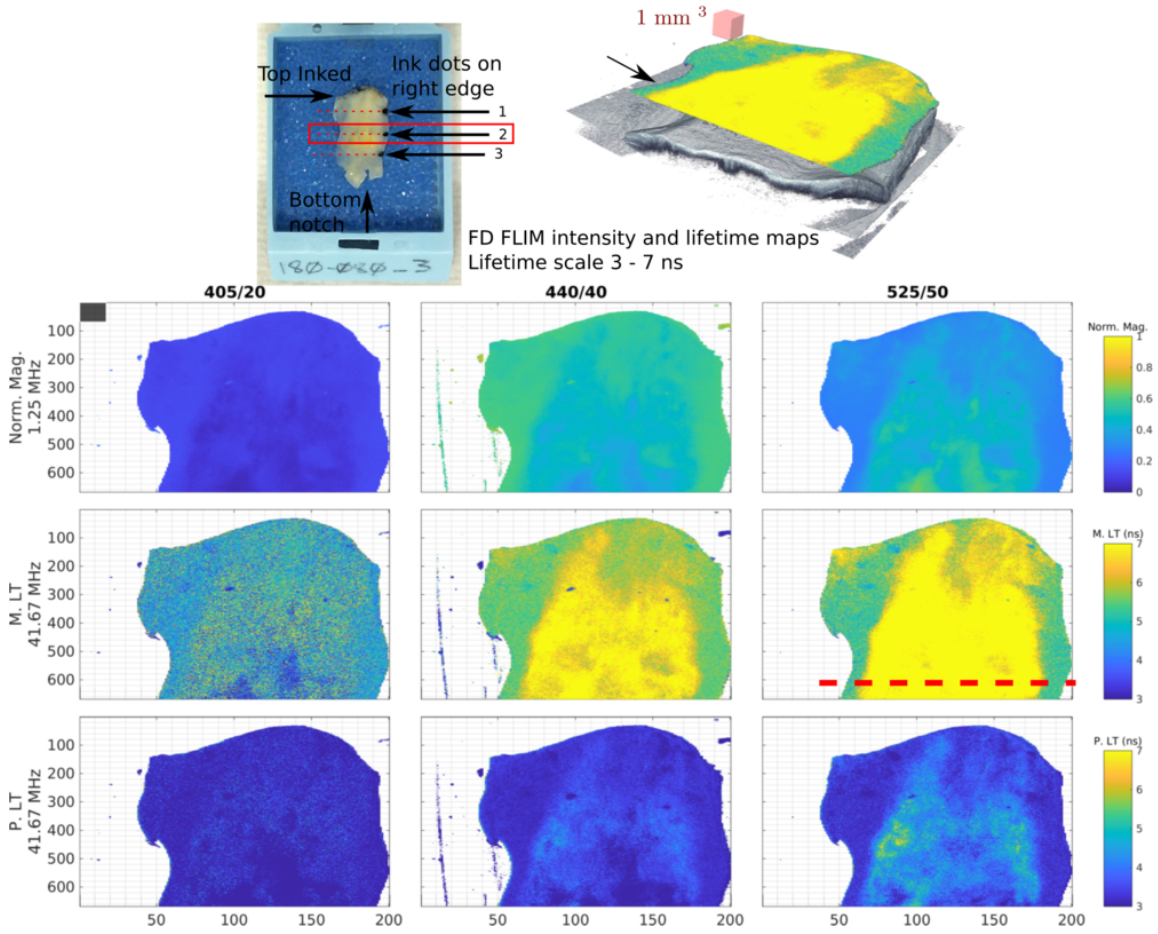

Figure 4.6: One SS OCT and FD FLIM acquisition of an ex-vivo human coronary artery showing positive CD68+ staining (darker brown areas indicated under the red bracket labeled lipid rich in figure 4.7). A picture of the artery is shown in the top left, along with annotations used for histology co-registration (one arrow per histology slice). A three dimensional rendering of the SS OCT volume with the FD FLIM τ_m 41.67 MHz lifetime is shown to the right of the artery picture. The black arrow indicates the approximate location of the histology slice and SS OCT bscan cross sections shown in figure 4.7.

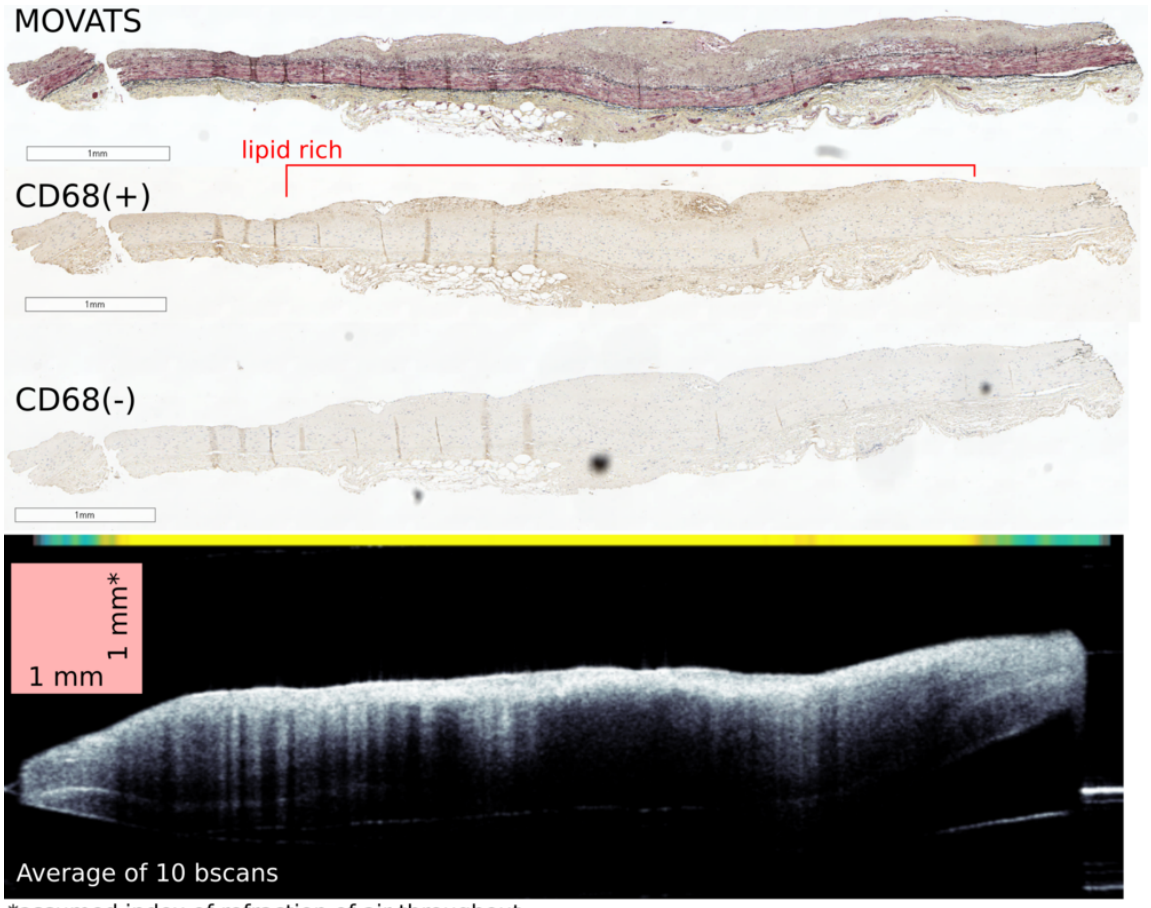

Figure 4.7: Histology and SS OCT bscan images from the red dotted line area of the ex-vivo human coronary artery shown in figure 4.6. The scalebars in the histology images are 1mm in the horizontal dimension. The FD FLIM τ_m 525/50nm 41.67 MHz lifetime is plotted as a color ribbon at the top of the SS OCT bscan, with the same colorscale as the lifetime maps in figure 4.7. The presence of brown areas in the CD68+ image suggest that this artery section contains macrophages. The lifetime values $\geq$7 ns support this conclusion. We would expect lifetimes greater than approximately 6ns to be correlated to positive CD68+ staining (see table 3.2).

another pixel acquisition. We will refer to this functionality as an adaptive pixel rate. For example, FD FLIM could acquire for 79μs (instead of 80μs) and then wait for the next SS OCT sweep trigger (every 10μs) before beginning another 79μs acquisition. This behavior would also solve another issue that we encountered, sweep frequency error in the SS OCT laser.

The Thorlabs SL1310V1-10048 SS OCT laser has a nominal sweep frequency of 100 kHz. However, we observed that it was 99.998 kHz using a Tektronix TDS 2022C oscilloscope. We also configured the arty atrix-7 FPGA to output a 100 kHz square wave and measured the frequency to be 100.000 kHz. This 2 Hz error would result in approximately 90μs of error in a typical 1 cm^2 image (668 by 688 pixels, assuming no flyback). 90μs is equivalent to 9 pixels of error for SS OCT, and 1.1 pixels of error for FD FLIM (operating at 12.5 kHz). However, we decimated the 100 kHz SS OCT trigger to 50 kHz to avoid overflow when transferring data from the FPGA to the host computer. This resulted in the same 1 cm^2 acquisition taking twice as long, and made the spatial misalignment twice as severe. Another consequence of this error and our particular control scheme was that this spatial misalignment appeared in FD FLIM rather than SS OCT. This was because the SS OCT FPGA was responsible for controlling the scanning mirrors. In other words, FD FLIM was operating at a more accurate 12.5 kHz while SS OCT was operating at 99.998 kHz and driving the scanning mirrors at that frequency. The aforementioned adaptive pixel rate approach for FD FLIM would solve this issue and result in FD FLIM operating at 99.998 / 8 kHz.

Additional considerations would arise if another scanning approach was used. In intravascular imaging (for example: [27],[28],[29],[30]), scanning is typically performed in a helical fashion where spinning is set at a specific rotational speed by a motor for fast axis scanning and a linear translation stage is moved for slow axis scanning. Assuming SS OCT would not control the scanning mechanism, physical misalignment would occur

in both SS OCT and FD FLIM (assuming FD FLIM was utilizing the adaptive pixel rate approach). Such cases would require post-processing corrections to correct the physical pixel coordinates.

At the time of this writing, 63 ex-vivo human coronary artery segments from 20 human subjects were imaged with the SS OCT and FD FLIM system. This added to the existing database of 87 arteries from 23 human subjects from our earlier study in chapter 3. This will allow for more future studies of the classification performance of the FD FLIM system in the context of atherosclerosis.

5. REAL TIME SIGNAL PROCESSING OF SWEPT SOURCE OPTICAL COHERENCE TOMOGRAPHY DATA USING A FIELD PROGRAMMABLE GATE ARRAY

5.1 Overview

This chapter describes the various signal processing steps implemented on the National Instruments NI 5772 and PXIe 7966 digitizer and FPGA combination. First, a short explanation of axial resolution and single sided imaging depth are given as they relate to signal processing. Second, the processing speed and FPGA resource usage are listed. Then, each processing step is explained: background subtraction, interpolation, dispersion correction, fast Fourier transform, magnitude estimation (and phase estimation), magnitude combination, and logarithmic compression. Then, two higher speed algorithms are described as they may be of use to others developing SS OCT FPGA processing designs. Finally, future work and possible ways to extend various components of the single processing steps are discussed. The notable features and limitations of the SS OCT FPGA signal processing algorithm are as follows:

1. **Sweep pattern is assumed to be fixed.** In other words, the same time versus k-space correspondence is used to interpolate all sweeps. The sweep pattern can be updated via the host computer, but not in real time during the acquisition of an image. Whether or not this is acceptable depends on the inter-sweep variability of the given SS OCT laser. For SS OCT lasers with inter-sweep variations, this processing approach is likely not suitable for phase-sensitive measurements.

2. **Two polarization channels.** The processing steps described here were intended for a SS OCT system that employed two polarization channels. The background sub-

traction, interpolation, dispersion correction, and fast Fourier transform, and magnitude estimation processing steps were implemented for each polarization channel (analog input data stream).

3. **2048-length FFT, 100 kHz sweep rate, 200 MHz sampling frequency, upwards of 6.8mm imaging depth assuming 1310±100nm and 80% sweep duty cycle**. See sections 5.2.2 and 5.3 for more information.

5.2 Imaging depth and axial resolution

5.2.1 Axial resolution l_c

Axial resolution can be calculated as described in [4, chapter 2 pp. 55]. We will use the same notation as the aforementioned book for convenience. Here we assume 1310nm center wavelength and 50nm FWHM bandwidth. We assume that the SS OCT laser has a 100nm sweep range with a hanning window applied during data processing, resulting in a FWHM bandwidth of 50nm.

$$l_c = coherence\ length = axial\ resolution = \frac{2ln(2)}{\pi}\frac{\lambda_0^2}{\Delta\lambda_{windowed}} \tag{5.1}$$

$$\boldsymbol{15.15\mu m} = \frac{2ln(2)}{\pi}\frac{(1310e-9)^2}{50e-9} \tag{5.2}$$

5.2.2 Single sided imaging depth z_{max}

In SS OCT the k spacing $\delta_s k$ is used to compute the single sided imaging depth z_{max}. Typically, the average k spacing is used. We will make a distinction between the average k spacing, $\bar{\delta_s k}$, and the largest k spacing in the sweep, $\hat{\delta_s k}$. These terms emphasize the fact that some SS OCT lasers do not sweep linearly. In other words, the single sided imaging depth - the depth at which aliasing starts to occur, varies along the sweep. First we will calculate z_{max} using $\bar{\delta_s k}$ to demonstrate the overall calculation. Then we will give an

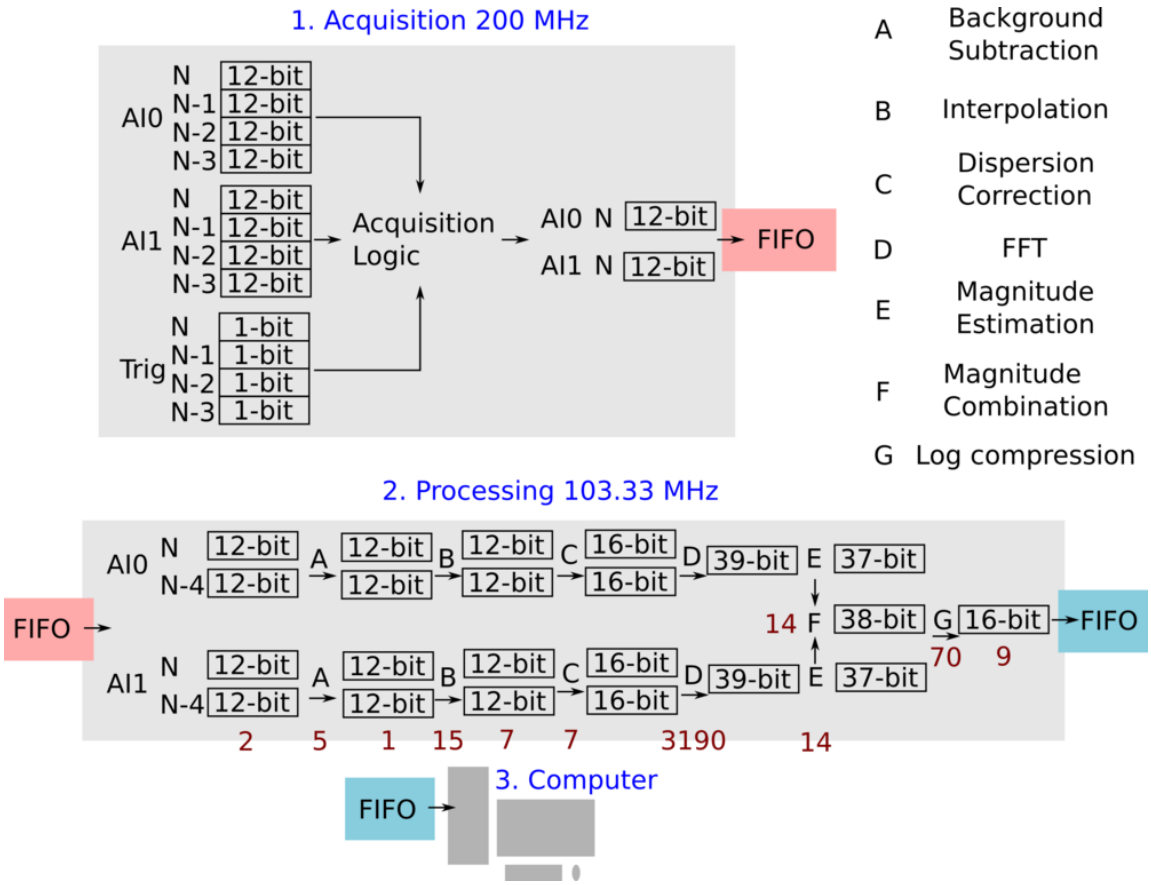

Figure 5.1: Overview of all FPGA processing steps. Two channels of 12-bit analog input data are acquired at 800 MS/s (4 points per clock cycle at 200 MHz). The data streams are decimated to one 12-bit data point per 200 MHz clock cycle before being transferred to the processing logic running at 103.33 MHz. Data is then extracted at 2 points per 103.33 MHz clock cycle for processing. Each processing stage is indicated by a letter A-G. The data precision is indicated between each processing step, and the latency is located below or adjacent to each step as a red number. Latency is in terms of clock ticks at 103.33 MHz (3334, equivalent to 32.3μs of latency). The precision of the log transformed data (after step G) was 16-bit, but 4 data points (64-bit) were accumulated before transferring the data to the host computer (FPGA to computer FIFOs are only 64-bit). Furthermore, the latency of 9 after log compression includes the depth cropping processing step (section 5.12). The latency of 7 between B and C is due to accumulators, the same as those shown in figure 5.35.

example method for calculating $\hat{\delta_s}k$ using one of our in-house SS OCT lasers. The reader can then substitute $\hat{\delta_s}k$ for $\bar{\delta_s}k$ in the example below if desired.

To compute $\bar{\delta_s}k$ the spectrum is transformed from λ, wavelength, to k space. Then the k space bandwidth is divided by the number of points used to sample the spectrum, M, to obtain $\bar{\delta_s}k$. M can be computed given the sampling frequency of the digitizer f_s, sweep frequency f_{sweep}, and sweep duty cycle $f^{\%}_{sweep}$. The duty cycle $f^{\%}_{sweep}$ is mentioned here because some SS OCT lasers do not sweep for the entire duration of their sweep period $1/f_{sweep}$.

$$k = \frac{2\pi}{\lambda},\ (definition) \tag{5.3}$$

$$k_{start} = (2\pi)/(\lambda_0 - \frac{\Delta\lambda}{2}) \tag{5.4}$$

$$k_{end} = (2\pi)/(\lambda_0 + \frac{\Delta\lambda}{2}) \tag{5.5}$$

$$\bar{\delta_s}k = \frac{|k_{start} - k_{end}|}{M} \tag{5.6}$$

$$M = \frac{1}{f_{sweep}} \times f^{\%}_{sweep} \times f_s \tag{5.7}$$

$$\bar{z_{max}} = \pi/(2\bar{\delta_s}k),\ [4, \text{chapter 2 pp. 62}] \tag{5.8}$$

In equations (5.4) and (5.5) the full, not windowed, $\Delta\lambda$ is used. This is important as windowing before the FFT does not change the single sided imaging depth. Note that $\bar{z_{max}}$ denotes the single sided imaging depth calculated with $\bar{\delta_s}k$. Now let's expand upon our previous example where $\lambda_0 = 1310nm$ and $\Delta\lambda = 100nm$. Assume we have a $200MS/s$ digitizer $f_s = 200e6$, an 80 percent duty cycle $f^{\%}_{sweep} = 0.8$, and a sweep frequency $f_{sweep} = 100kHz$.

$$M = 1600 = \frac{1}{100e3} \times 0.8 \times 200e6 \tag{5.9}$$

$$k_{start} = 4986655 \tag{5.10}$$

$$k_{end} = 4619989 \tag{5.11}$$

$$\bar{\delta_s k} = 229 \tag{5.12}$$

$$\bar{\boldsymbol{z_{max}}} = \mathbf{6.85}\boldsymbol{mm} \rightarrow \text{ Assuming a linear sweep profile}^1 \tag{5.13}$$

The maximum single sided imaging depth based is $6.85mm$. Here the 80% duty cycle imposes more difficulty on the system because it means the entire sweep occurs in a shorter amount of time relative to a 99% duty cycle. In other words, all things equal, a shorter duty cycle decreases $\bar{z_{max}}$.

Our approach to calculate $\hat{\delta_s k}$ for an actual SS OCT laser is as follows. First acquire a reference interferogram as mentioned in section 5.5.1. Then assume that we know the precise time when the sweep starts and stops. This is determined by manually observing the interferogram. Next, generate a sweep profile $k[t]$ that specifies k space as a function of time during the sweep. The sweep profile will be obtained by using the unwrapped phase of the hilbert transform of the reference interferogram. Since we know k_{start} and k_{end} from $\lambda_0, \Delta\lambda$ we can set these to correspond to the min and max of the unwrapped phase to obtain $k[t]$. The maximum of $d(k[t])/dt$, the time derivative of $k[t]$, is $\hat{\delta_s k}$. Figure 5.2 shows the results of applying this approach to a reference interferogram.

$$max(\hat{\delta_s k}) = 270.8 \text{ (From data in figure 5.2)} \tag{5.14}$$

$$\hat{\boldsymbol{z_{max}}} = \mathbf{5.8}\boldsymbol{mm} = \pi/(2 \times max(\hat{\delta_s k})) \tag{5.15}$$

The actual single sided imaging depth based on the sweep profile for this particular SS OCT laser is 15% shorter than the depth calculated assuming a linear sweep profile. $\hat{z_{max}}$

[1] See figure 5.2 and equation (5.15) for an example of computing the maximum single sided imaging depth using a non-linear sweep profile.

is the minimum depth at which parts of the interferogram begin to alias, and z_{max}^{-} is the depth at which the interferogram is aliased assuming a linear sweep profile.

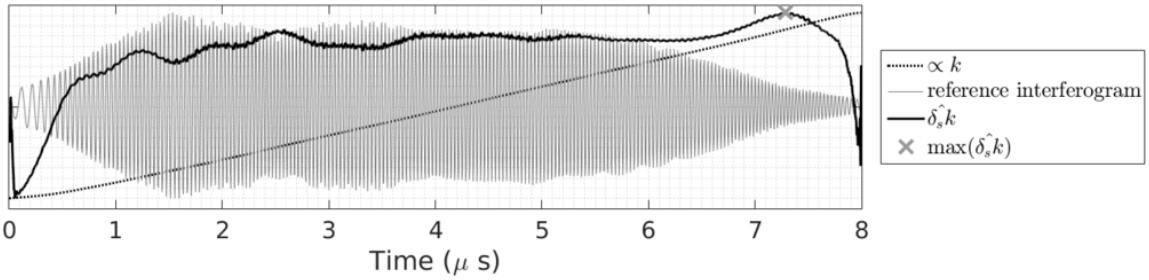

Figure 5.2: Illustration of finding $\hat{\delta_s k}$ using a reference interferogram from our in house SS OCT laser. The vertical axis in this figure is not labeled because all curves were scaled individually so that they could be plotted together. The $\propto k$ curve is the result of a hilbert transform based method for extracting the instantaneous phase from the reference interferogram. This gives a curve that is proportional to $k[t]$ as a function of time. The $\propto k$ curve was rescaled to approximate the real $k[t]$ curve by assuming k_{start}, k_{end} according to the known laser $\lambda_0, \Delta\lambda$. The $\hat{\delta_s k}$ curve is the time derivative of the $\propto k$ curve. Higher values of $\hat{\delta_s k}$ correspond to points in the sweep when the laser is sweeping faster. A faster sweep rate requires a faster sampling frequency to maintain the same single sided imaging depth. The abrupt transitions in the $\hat{\delta_s k}$ curve at the beginning and end are due to bandpass filtering that was performed on the reference interferogram to reduce artifacts when generating the $\propto k$ curve from the hilbert transform method.

5.3 Processing speed and resource usage

To fully characterize the SS OCT processing algorithm, processing speed, resource usage, and quality need to be quantified. This section lists the resource usage and explains the limits of the processing speed of the current implementation of the SS OCT processing algorithm as a whole. Processing quality is discussed in the later sections where each processing step is individually explained. An analysis of cumulative processing quality in terms of SS OCT relevant parameters (ex: SNR) is not finished at this time.

The current processing scheme uses LabVIEW high throughput IP for several adders.

Table 5.1: Resource usage summary for the SS OCT processing algorithm compiled on the NI 5772 / PXIe7966 (Virtex-5 SX95T speed grade -2). Usage with and without the processing parts is shown. The no processing column includes only the acquisition logic, i.e. the 103.33 MHz clock domain and all processing logic are not included in the design. The processing column contains the acquisition and processing logic. The percentage of resources used is shown as mean $\pm$ standard deviation. The total number of each resource is shown in parenthesis in the leftmost column. 5 compilations were used for the no processing case, and 4 compilations were used for the processing case.

	No processing 5 compiles mean $\pm$ sd (%)	Processing 4 compiles
Slices (14720)	47.1 $\pm$ 0.2	98.0 $\pm$ 1.2
Registers (58880)	22.9 $\pm$ 0.0	74.0 $\pm$ 0.0
LUTs (58880)	22.6 $\pm$ 0.0	73.7 $\pm$ 0.1
Block RAM (244)	17.6 $\pm$ 0.0	34.3 $\pm$ 0.2
DSP48 (640)	00.0 $\pm$ 0.0	32.2 $\pm$ 0.0

This IP does not support using DSP48 slices. However, the LabVIEW adder IP can be replaced with Xilinx adder IP to use DSP48 slices in favor of general purpose logic. This would reduce the slice registers and slice LUT usage and increase DSP48 usage. Using less general purpose logic would allow the design to compile at higher frequencies and may allow for increasing the processing speed in the future.

Terms

1. $T^{raw}_{bits} = 12 =$ bits per point of raw interoferogram
2. $T^{raw}_{sweepFreq} = 100kHz =$ sweep frequency of swept source laser
3. $T^{raw}_{DUT} =$ duty cycle of swept source laser
4. $T^{raw}_{len} =$ points per raw interoferogram
5. $T^{raw}_{channels} = 2 =$ Simultaneously acquired interoferograms

6. $T_{bits}^{proc} = 16 =$ bits per point of processed interferogram (magnitude from combined polarizations)

7. $T_{bits}^{proc} = 16 =$ bits per point of processed interferogram (phase from one polarization)

8. $T_{len}^{proc} \leq FFT_{len}^{fixed}/2 =$ points per processed interferogram, set by user

9. $fs^{raw} = 200MS/s =$ sampling frequency when acquiring raw interferogram

10. $fs^{proc} = 103.33MHz =$ Clock frequency of processing logic

11. $THRPUT_{clk}^{proc} = 2 =$ Number of points processed per clock cycle by processing logic

12. $PostTrig_{ticks} = 1 =$ Number of clock ticks required after trig before start of acquisition

13. $ReARM_{ticks} = 1 =$ Number of clock ticks required for rearming FPGA between interoferograms

14. $FFT_{len}^{fixed} = 2048 =$ Fixed FFT length in terms of interferogram points

15. $DUT^{max} \approx 99.9\%$
$\approx \frac{FFT_{len}^{fixed} - (ReARM_{ticks} + PostTrig_{ticks})}{FFT_{len}^{fixed}} =$
=Maximum acquisition duty cycle

16. $TFR_{T->H}^{max\ MiB/s} = 800MiB/s$
= Maximum data transfer rate from FPGA to host computer

All processing is performed at $fs^{proc} \times THRPUT_{clk}^{proc} = 206.66MS/s$ for each individual polarization channel. This means that data is processed as fast as it is acquired. Between sweeps, the FPGA requires $ReARM_{ticks}$ clock cycles at fs^{raw} to re-arm and prepare for the next sweep. At the beginning of each sweep, the FPGA requires $PostTrig_{ticks}$ clock cycles at fs^{raw} before data can be acquired. This limits the duty cycle to the number of desired points of data plus $PostTrig_{ticks} + ReARM_{ticks}$ 200MHz clock cycles. The maximum number of points per interoferogram is fixed at 2048 due to the use of a fixed length version of the Xilinx FFT IP core.

Compatible sweep frequencies at DUT^{max}: 97.752 kHz → 100.908 kHz

The sweep frequency range is constrained by FFT_{len}^{fixed}, DUT^{max}, fs^{raw}, fs^{proc}, and $THRPUT_{clk}^{proc}$. The maximum sweep frequency is limited by how quickly the processing logic can compute the FFT_{len}^{fixed} point FFT. The minimum sweep frequency at the max duty cycle occurs when the maximum amount of points per sweep, $DUT^{max} \times FFT_{len}^{fixed}$, are acquired.

$$\max(T_{sweepFreq}^{raw}) = \frac{1}{FFT_{len}^{fixed}/(fs^{proc}) \times THRPUT_{clk}^{proc}} \tag{5.16}$$

$$\max(T_{sweepFreq}^{raw}) = \frac{1}{2048/(103.33\text{ MHz}) \times 2} = 100.908\text{ kHz} \tag{5.17}$$

$$\min(T_{sweepFreq}^{raw}) = \frac{1}{FFT_{len}^{fixed} \times DUT^{max}/(fs^{raw})} \tag{5.18}$$

$$\min(T_{sweepFreq}^{raw}) = \frac{1}{2046/(200\text{ MHz})} = 97.752\text{ kHz} \tag{5.19}$$

Acquiring data at other sweep frequencies is also possible, but the maximum number of points per sweep will always be limited to 2048. For example, an SS OCT laser with a 200kHz sweep frequency could be used. The acquisition logic can acquire at 400 MS/s, but sweeps would need to be decimated by 1 (the acquisition logic can do this), resulting

in an effective sweep rate of 100kHz. Slower sweep rates down to 25 kHz can also be acquired with a 200 MS/s sampling rate. However, the maximum number of points per sweep after interpolation would be limited to 2048 due to the fixed length FFT.

Maximum number of pixels per image - $2^{24} = 16777216 = 4096 \times 4096$. The FPGA is capable of processing data until certain limits are reached depending on the host computer memory size, and the bit widths of various FPGA counters. During an acquisition, the FPGA counts each sweep in order to determine when to stop the acquisition and perform other actions. The current width of this counter is 24-bits, which means up to 2^{24} sweeps can be acquired in one acquisition. This equates to a square image size of 4096x4096 pixels, and a total imaging time of $2^{24}/(100\text{ kHz}) = 167.77$ seconds assuming a 100 kHz sweep rate. In intravascular imaging, the maximum duration of a single image is limited by the flush time, usually $\approx$ 4 seconds. The 24-bit counter could be increased if needed. For example, a 32-bit counter would allow for a 65536x65536 image, or $\approx$ 11.93 hours.

The host computer data storage can also limit the number of pixels per image. Assuming phase data is not returned to the host computer, each spatial pixel consists of T_{len}^{proc} points at 2 bytes per point. Assuming the host data storage is limited to RAM and that the host computer has 64 GiB of RAM, the image size limits are shown below. If phase is returned, the equivalent result is half of the limit shown below. However, the pixel size can be reduced by depth cropping. We will assume no depth is cropped for these calculations.

$$\begin{aligned} \text{Pixel limit by current FPGA counter} &= 2^{24} = 16777216 \\ &= 4096 \times 4096 \text{ (image pixels)} \\ &\approx 167.77 \text{ seconds} \end{aligned} \tag{5.20}$$

$$\begin{aligned} \text{Pixel limit by 32 bit FPGA counter} &= 65536 \times 65536 \\ &\approx 11.93 \text{ hours} \end{aligned} \tag{5.21}$$

$$H^{mem} = 64 \text{ GiB} = \text{host memory} \tag{5.22}$$

$$P_{bytes} = T_{len}^{proc} \times T_{bits}^{proc}/8 \tag{5.23}$$

$$= 2048 = \text{bytes per pixel}$$

$$\text{Pixel limit by host memory} = H^{mem}/P_{bytes} = 64 \times 2^{30}/2048 \tag{5.24}$$

$$= 33554432$$

$$\text{Pixel limit by host memory} \approx 5792 \times 5792 \text{ (image pixels)} \tag{5.25}$$

$$\approx 335.54 \text{ seconds}$$

The required throughput to the host computer is also of interest. If it is sufficiently low, traditional hard drives may be used to store data instead of RAM. The throughput is calculated below assuming $T_{sweepFreq}^{raw} = 100$ kHz.

$$\text{Full depth throughput mag. only} = T_{len}^{proc} \times T_{bits}^{proc}/8 \times T_{sweepFreq}^{raw} \tag{5.26}$$

$$\text{Full depth throughput mag. only} = 195.3125 \text{ MiB/s} \tag{5.27}$$

$$\text{Full depth throughput mag. and phase} = 390.625 \text{ MiB/s} \tag{5.28}$$

This throughput may be possible with traditional hard drives in a RAID setup, or with more modern SSDs. The maximum transfer speed from the PXIe7966 to the host computer is $TFR_{T->H}^{max\ MiB/s} = 800MB/s$. If the transfer rate is greater than $TFR_{T->H}^{max\ MiB/s}$, the acquisition size may be limited by onboard FPGA memory (512 MiB on the PXIe7966). Since we are not limited by this rate we are not discussing this limitation.

5.4 FPGA: Background subtraction

Before background subtraction is performed, a background interferogram must be collected by blocking the sample arm of the SS OCT system and saving the observed inter-

ferogram. The saved background interferogram is then uploaded from the host computer to the FPGA to be used for background subtraction. Background subtraction is performed by subtracting a vector stored in the FPGA from each interferogram as the interferograms are acquired. In other words, the same background vector is used to process all acquired data.

5.4.1 Precision

The background subtraction data is stored on the FPGA as 12-bit signed integer, the same as the raw data from the ADCs. This means that there is quantization error that occurs from when the host computer computes the background vector in double precision floating point, and when background subtraction occurs on the FPGA. This is because several interferograms are typically averaged to generate the background vector (upwards of 1000). Thus, the background vector will contain non-integer values that are truncated before uploading the data to the FPGA. We have not further analyzed the error at this time.

5.5 FPGA: Interpolation

After background subtraction is performed, the interferogram is interpolated so that it is equally sampled in k-space. This is required before taking the FFT later in the processing pipeline. Linear interpolation, involves knowing the values of a function $y = f(x)$ at specific points, x, and deriving unknown values of y denoted as y_i at specific query points, x_i. A summary and example is shown below.

$$\begin{aligned} &Known\ function \\ &\qquad y = f(x),\ x = [x_0, x_1, \ldots, x_{N-1}] \\ &\qquad N = number\ of\ points\ in\ y \\ &Known\ function\ for\ example \end{aligned} \tag{5.29}$$

$$y = [0, 2, 4, 6, 8, 13] \tag{5.30}$$

$$x = [0, 1, 2, 3, 4, 5] \tag{5.31}$$

$Desired\ function$

$$y_i = f(x_i),\ x_i = [x_{i0}, x_{i1}, \ldots, x_{iM-1}] \tag{5.32}$$

$$M = number\ of\ query\ points$$

$Desired\ function\ for\ example$

$$y_i = unknown \tag{5.33}$$

$$x_i = [0.5, 1.1, 2.7] = query\ points \tag{5.34}$$

To calculate the y_{i1}, the value of the function at $x_{i1} = 1.1$, we need to first find which points in $y = f(x)$ surround $x_{i1} = 1.1$. From our example, we can see that $x_1 \leq x_{i1} < x_2$, which means x_1 and x_2 surround x_{i1}. In our example we will set the lower bound condition to include $equal\ to$, but this is arbitrary as we could set the upper bound $equal\ to$ instead. The only concern is that we are consistent when interpolating all points, especially when we move to the example in the FPGA. Now we can interpolate to get y_{i1} as shown below.

$$y_{i1} = \frac{y_2 - y_1}{x_2 - x_1} \times (x_{i1} - x_1) + y_1 = \frac{2}{1} \times 0.1 + 2 = 2.2 \tag{5.35}$$

Now if we extend this example to SS OCT data and we assume the same k-space versus time correspondence for each interferogram, we can make the following observations about each term during linear interpolation.

$$Unknown\ until\ runtime \quad y_2 - y_1,\ y_1 \tag{5.36}$$

$$Constant\ between\ all\ interferograms \quad x_2 - x_1 \tag{5.37}$$

$$Constant\ between\ all\ interferograms \quad x_{i1} - x_1 \tag{5.38}$$

This means that the $x_2 - x_1$ and $x_{i1} - x_1$ terms can be pre-computed on the host computer and uploaded to memory in the FPGA. This saves FPGA resources and at the cost of assuming a fixed k-space versus time correspondence. This approach may not work on all SS OCT lasers. Two out of the three in-house lasers that we have tested are capable of using this interpolation method. Furthermore, we have only tested the performance related to the magnitude spectra of the processed interferograms, not the phase spectra. In other words, it is possible that this approach will not work for phase sensitive measurements with any of our in-house lasers. Now that we have introduced linear interpolation we will describe our FPGA implementation used for linearizing the SS OCT interferograms in k-space.

5.5.1 Reference interferogram

A reference interferogram must be acquired to perform interpolation during future imaging sessions. Typically we use a commercial fixed path length interferometer such as the Thorlabs MZI-clock, INT-MZI-1300 ($1810 as of 2016/03/03). The MZI-clock is inserted into the system directly after the SS OCT laser so that the reference interferogram contains no dispersion related k-space nonlinearities. Once acquired, a fixed interpolation vector is extracted in Matlab [2]. Then interpolation instructions are generated from the fixed interpolation vector in Matlab. These instructions are then saved to disk and uploaded later to the FPGA to perform interpolation.

5.5.2 Interpolation instructions

The format of the interpolation instructions is detailed in this section. It may be easier for the reader to skip this section and continue to the interpolation architecture section. Once an understanding of the architecture is achieved the format of the interpolation instructions is more intuitive.

[2] Documentation for the Matlab algorithms is not complete at this time.

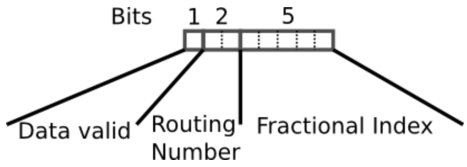

Figure 5.3: Format of interpolation instructions. One instruction consists of 8 bits. One bit contains the data valid status of the resulting output point. Two bits contain the routing number, which specifies the two input points to interpolate from the input data stream. The remaining 5 bits are the fractional index, a number 0 to 1.

Each interpolation cell receives one interpolation instruction per clock cycle. The interpolation instruction contains the information used for interpolating one output point from two input points. The current format consists of 8 bits per instruction and is illustrated in figure 5.3. It uses 5 bits for specifying the fractional index, 2 for the routing number, and 1 bit for the data valid status of the output point. The fractional index is relative weight between the two input points to use for interpolating the output point. For example, a fractional index of 0 means that the most-previous point is passed directly to the output. A fractional index of 0.5 means that the output point is the average of both input points.

The fractional index is interpreted as an unsigned fixed point number spanning approximately 0 to 1. This means that the precision of the interpolation, in terms of the query points, is limited to increments of 2^{-5}. In other words, the fractional index is limited to this set of numbers $(0, 0.03125, 0.0625, 0.09375, \ldots, 0.96875)$.

The choice of 5 bits for the fractional index is based on preliminary simulations using data acquired from our SS OCT system. The SS OCT laser used for these simulations was the ESS320024-00 [3]. Figures 5.4-5.6 show the same interferograms processed with different fractional index bit widths. A behavioral model of the interpolation algorithm described in figure 5.8 was implemented in Matlab. Reference interferograms were acquired

[3]Swept source laser used for simulations: ESS320024-00: ESS-1310nm/ A-scan: 100kHz/ Sweep range: 110nm/ Coherence length: 5-6mm/ Pavg: 15-20mW/ k-clock.

using our SS OCT system. A mirror was placed at the sample position and the reference arm length was varied to move the peak in depth. The interferograms were then processed using different fractional index bit depths. Figure 5.4 shows interpolation using 0 bits for the fractional index, equivalent to nearest neighbor interpolation. Figure 5.5 shows the result using 5 bits, and figure 5.6 shows the result using double floating point precision. Figure 5.7 shows the SNR calculated from various bit depths. Each point represents the mean SNR of 100 alines, and the error bars denote the standard deviation. The SNR was computed by dividing the peak amplitude by the standard deviation of the deepest 20% of depth pixels and then taking $20\log()$ of the quotent. The simulations show that a precision of 5 bits is enough to mitigate higher frequency artifacts observed when insufficient precision is used. However, since data was not available across the entire depth range, depths below approximately half of Nyquist were not simulated.

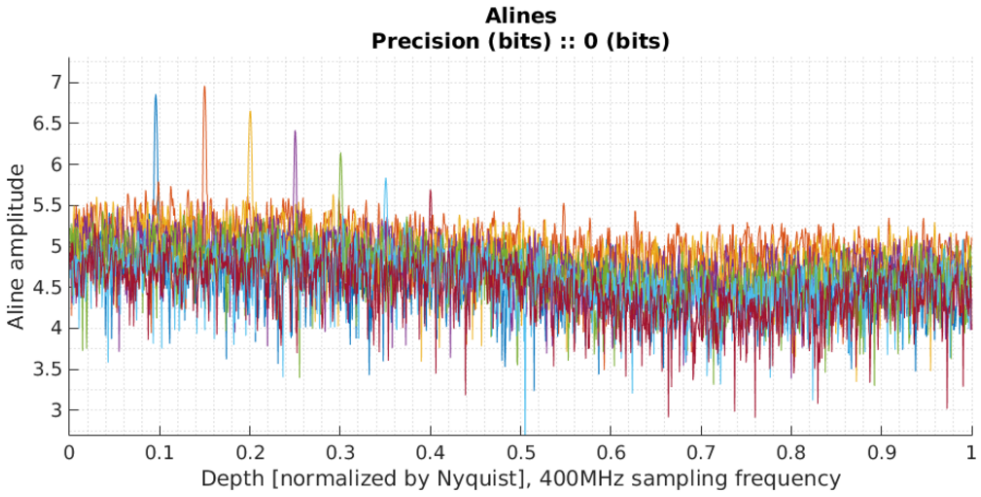

Figure 5.4: Reference interferograms processed using a fractional index of 0 bits, equivalent to nearest neighbor interpolation. Aline amplitude (arbitrary units) plotted versus imaging depth normalized by Nyquist.

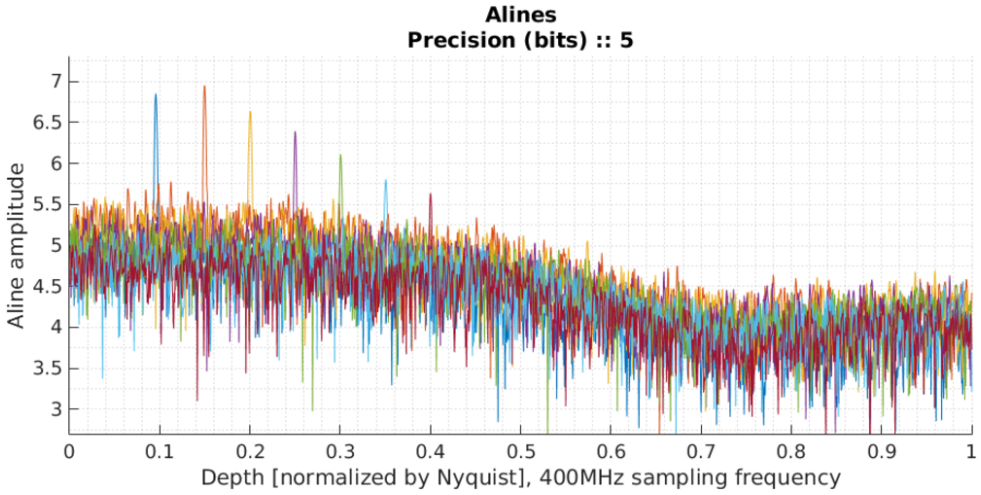

Figure 5.5: Reference interferograms processed using a fractional index of 5 bits. Aline amplitude (arbitrary units) plotted versus imaging depth normalized by Nyquist.

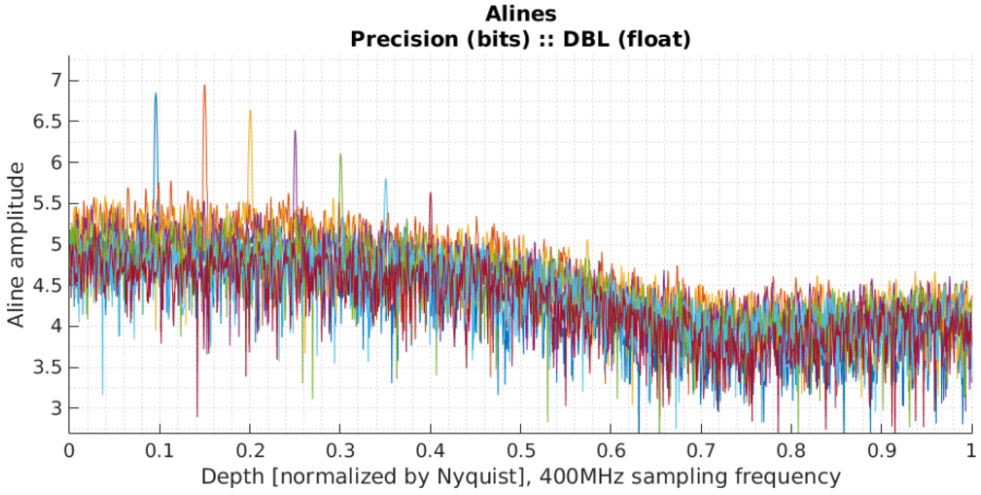

Figure 5.6: Reference interferograms processed using floating point precision for the fractional index. This is used as the gold standard for comparison. Aline amplitude (arbitrary units) plotted versus imaging depth normalized by Nyquist.

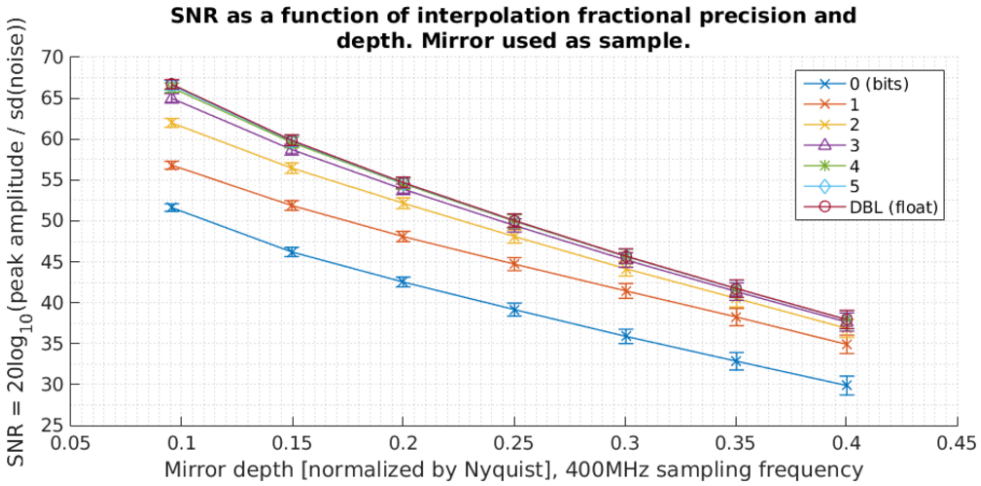

Figure 5.7: SNR calculated from the interferograms at each depth location for all fractional index bit widths.

5.5.3 Interpolation architecture

This section describes the organization of the interpolation algorithm on the FPGA. The input data rate and lack of storage space on our particular FPGA means that a streaming architecture is best suited to implement interpolation. Streaming means that data can be processed as fast as it is received with no overflow. For each of the 2 polarization channels, 2 time points are processed per clock cycle in the interpolation algorithm. The algorithm also stores the previous 2 time points, which are needed if the points to interpolate fall between the current and previous chunk of 2 time points. See figure 5.8 for a schematic of the interpolation architecture.

5.5.4 Limitations

In our current implementation a maximum of 4 output points are generated for a given set of 2 input points. This means that the algorithm may not be applicable to extreme nonlinear sweep patterns that require more than 4 points per any given pair of input points.

The current algorithm assumes that all sweeps of the SS OCT laser can be adequately

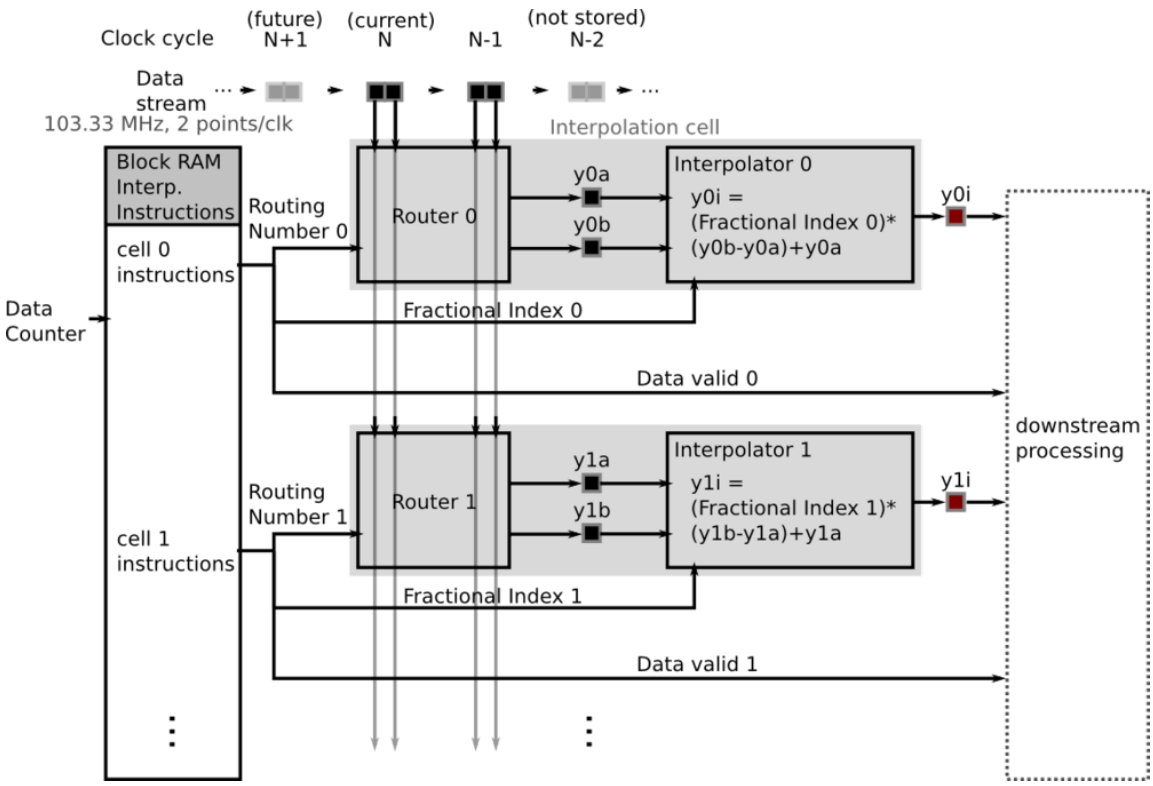

Figure 5.8: Schematic of FPGA interpolation architecture. Only two interpolations cells are shown in this illustration. The current implementation uses 4 interpolation cells per polarization channel. Input data is processed at two points per clock cycle and is denoted at the top of the diagram by the black squares (data stream). N denotes the current clock cycle, N-1 denotes the previous clock cycle, etc. A data counter is maintained in the processing pipeline to keep track of the current time point in the current interferogram. The counter is used to index block RAM to extract the interpolation instructions for each cell for the current data chunk, N. The lines drawn down from the data stream illustrate that all 4 data points are inputted to each router. Once the interpolation instructions are extracted they are sent to each interpolation cell, each cell receives unique instructions. Each interpolation cell consists of a routing block and interpolation block. The routing block selects 2 points from the 4 total input points, according to the routing number, from the current and previous data chunks, N and N-1, respectively. These are denoted as $y\#a$ and $y\#b$, where $\#$ denotes the number of the interpolation cell (0 to 3, 4 total interpolation cells in this implementation). The $y\#a$ term is the earliest time point selected interpolation, and $y\#b$ is the latest time point. The interpolator block performs 3 operations as shown in the schematic. It subtracts the most earliest time point from the latest time point, multiplies the result by the fractional index, and adds the product to the most previous time point. The result $y\#i$ and the corresponding data valid signal are then outputted to the downstream processing modules.

interpolated using the same interpolation instructions. SS OCT lasers that require each sweep to be linearized in an individualized manner would require interpolation instructions to be generated on the fly (unique to each sweep). Also, this increased interpolation accuracy will likely be required if the given SS OCT system is used for phase measurements. Generating the instructions immediately before interpolation could be accomplished by using an approach similar to [31].

5.5.5 Future work

This architecture can be extended to interpolate more non-linear sweep patterns by increasing the number of interpolation cells or by adding flow control to pause the input data stream to allow for extracting more output data from the current set of input data.

5.5.5.1 Increasing output points per input points: more interpolation cells

For example, instead of having 4 cells per 2 input points, 8 cells could be used. This would require modifying the Matlab code for generating the interpolation instructions. If more than 8 interpolation cells are used then it will require modifying the post-interpolation sorting code on the FPGA, but nothing else in the processing pipeline would be effected (dispersion correction, FFT, etc.). The post-interpolation sorting code acts to accumulate points before the next stages of processing. For example, some clock cycles result in $0, 1, 2, 3,$ *or* 4 output points. However, the next stages of processing require a fixed number of points per clock cycle, usually 2 in this implementation. The sorter acts as a throughput adjuster, assuring that when data is valid it always consists of 8 valid points (8 is more optimal for some of the memory interfaces used in the design). Later in processing this throughput is reduced to 2 points per clock cycle. The sorting code is not described in more detail here.

5.5.5.2 Increasing output points per input points: flow control

The current architecture accepts fresh input data on every clock cycle and only keeps the N and N-1 pairs of data points for use in the interpolation operations. Flow control could be added to the design to allow for the input data stream to be halted while output data points are generated with no modifications to other parts of the design (assuming the number of output points is still $\leq$ the FFT length of 2048). This could be accomplished by:

1. Adding a FIFO at the beginning of the interpolation logic to absorb fluctuations in the input data stream.

2. Including an increment bit in the interpolation instruction (figure 5.3) to signal when the data stream should be incremented to take in a fresh pair of data points.

5.6 FPGA: Dispersion correction

Dispersion correction is implemented similarly to background subtraction. Pre-calculated correction factors are generated on the host computer and uploaded to the FPGA prior to an imaging session. The FPGA then performs the complex multiplication and sends the real and imaginary numbers downstream to the FFT.

Prior to the imaging session, the correction factors are computed by first acquiring several interferograms with a mirror placed at the sample position. The interferograms are then saved to the host computer and used in a dispersion correction algorithm to compute a custom window and phase correction. The custom window and phase correction are then separated into real and imaginary parts and uploaded to the FPGA. The same correction factors are used to correct all interferograms in subsequent images. Each polarization channel is capable of having a unique set of correction factors as separate block RAM modules are used for each channel.

5.7 FPGA: FFT

The stock Xilinx core IP FFT 7.1 module is used along with custom logic to implement the FFT in the processing algorithm. Each FFT core was configured with the following options:

1. Channels: 1
2. Transform length: 1024
3. Target clock frequency: 100 MHz
4. Implementation: Pipelined, Streaming I/O
5. Run time configurable transform length: no
6. Data format: fixed point
7. Input data width: 16
8. Phase factor width: 16
9. Scaling options: unscaled
10. Rounding modes: truncation
11. Output ordering: Natural order (no cyclic prefix insertion)
12. Input Data Timing: no offset
13. Number of stages using block RAM: 5 (maximum)
14. Reorder buffer: block RAM
15. Hybrid memories: no

16. Complex multipliers: use 4-multiplier structure

17. Butterfly arithmetic: use XtremeDSP slices

Due to the required throughput of the design and timing constraints, 2 points per clock cycle need to be processed per polarization channel. The Xilinx core IP FFT is not capable of operating in a *decimated in time* (DIT) FFT approach. Thus we have implemented our own logic to create a decimated in time FFT module from 2 separate Xilinx core IP FFT modules. The input data stream of 2 points per clock cycle is split by sending each $2n$ and $2n-1$ data streams to different Xilinx core IP FFT modules. Then the output data streams from each Xilinx core IP FFT module are combined using custom logic to compute the full FFT. Figure 5.9 illustrates and further explains the FFT module. The rejoiner logic is explained in section 5.14.2.

5.8 FPGA: Magnitude estimation

The magnitude estimation method used in the processing pipeline is taken from here, [32, p. 679, High-speed vector magnitude approximation]. In short, the magnitude is approximated as shown below.

$$mag = \sqrt{a^2 + b^2} \tag{5.39}$$

$$mag \approx \alpha max(|a|, |b|) + \beta min(|a|, |b|) \tag{5.40}$$

$$a, b \in \mathbb{R}$$

This approximation can be implemented in several ways that trade off increased complexity and resource usage for improved accuracy. We have identified the following considerations.

1. Number of regions.

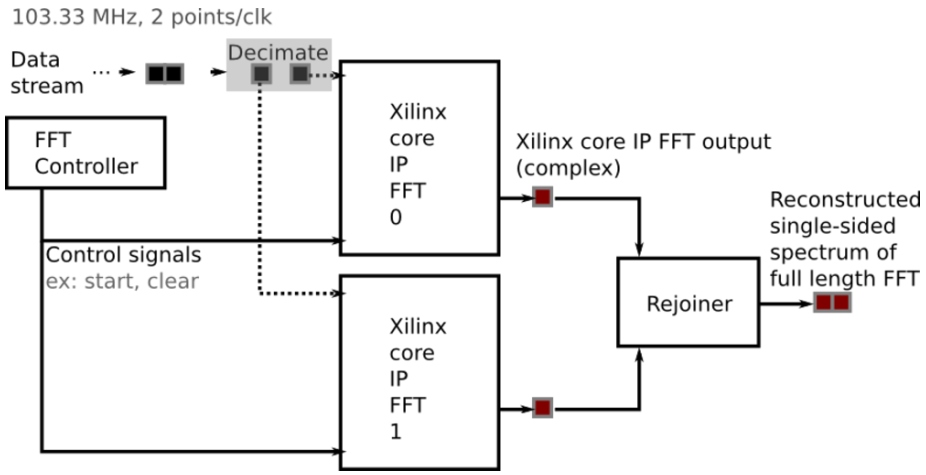

Figure 5.9: Schematic of FPGA FFT implementation for one polarization channel. Two of these modules exist in the design. Two stock Xilinx core IP FFT modules are configured as 1024 length, streaming, unscaled fixed point, radix-4, and set to use block RAM and DSP48 slices to conserve general logic resources (block RAM and DSP48 slices are used for all butterfly multiplier stages). Each pair of input data points is decimated between the two Xilinx FFT cores so that the total throughput of the overall processing pipeline can be maintained. The FFT controller module maintains an internal counter and logic for generating the start and clear signals for the Xilinx FFT cores. The same control signals are sent to both cores at the same time. The outputs of the two half length FFTs are combined in the rejoiner module by multiplying by the appropriate twiddle factors. The resulting full length, 2048, FFT is then outputted to the next processing module downstream. **The rejoiner module only computes the single sided spectrum of the full length FFT**. In other words, only 1024 points out of the full 2048 are outputted downstream. See section 5.14.2 for more information on the rejoiner.

2. Bit width of α, β.

Before continuing, it is worth mentioning that we only need to consider cases where $a = 1$ and $b = 0 \ldots 1$ (for simulation purposes). All subsequent plots are plotted versus b. This is because the magnitude depends only on $|a|, |b|$. If we divide both $|a|, |b|$ by $max(|a|, |b|)$ then one of $|a|, |b|$ will be 1, and the other will be between 0 and 1. So it is appropriate to only consider $a = 1$ and $b = 0 \ldots 1$ (implying that we've set a to the maximum of the pair and b to the minimum of the pair).

5.8.1 Number of regions

In the simplest implementation of the approximation, the terms α, β are constants. This means that the same value of α and β is used for all a, b. The approximation can be improved if different values of α, β are used according to the values of a, b. In a 2 region approach we would use α_0, β_0 when $0 \leq b \leq a/2$, and then use α_1, β_1 when $a/2 < b \leq a$. This is relatively simple on the FPGA if the number of sub-regions is a multiple of 2. However, for our initial implementation described here we use the same α, β for all a, b. Figures 5.10 and 5.11 show the error of the approximation with various numbers of sub-regions. The relative error is below $-20dB$ for all a, b even for the single region approach.

5.8.2 Bit width of α, β

Until now, we have not mentioned the number of bits used to represent α, β. Figures 5.10 and 5.11 assume double precision floating point. However, floating point operations are likely not feasible with our current FPGA due to resource and timing constraints. **We have determined that 4 bits of precision are acceptable for α, β *in the single region approach*.** This is because the optimal α, β for the single region approach fall close to numbers that can be obtained with a 4 bit fixed point number. Table 5.2 lists the optimal values for α, β for different bit widths.

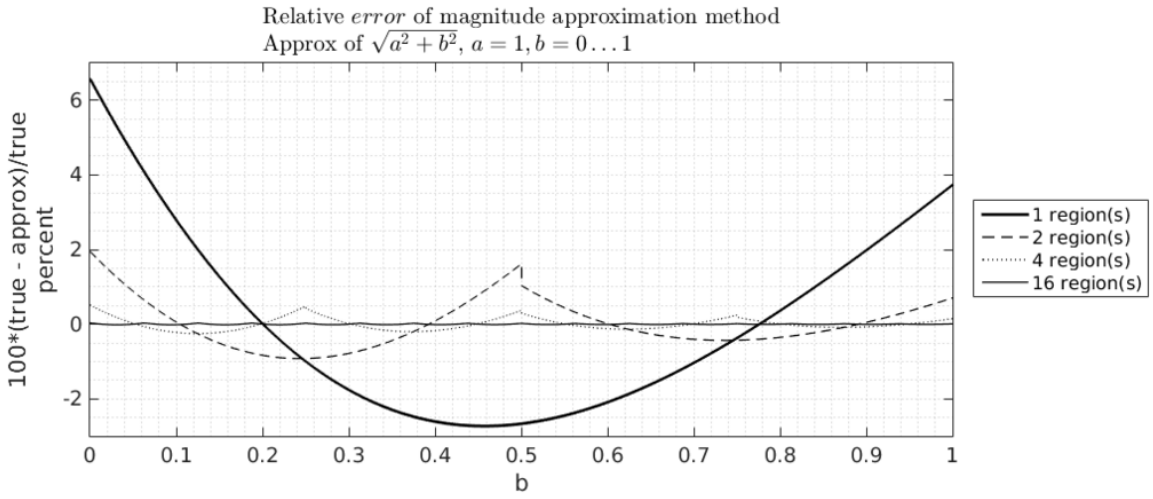

Figure 5.10: Comparison of magnitude approximation when using different numbers of sub-regions, under the condition that sub-regions are divided by halving existing regions. Increasing the number of sub-regions allows for optimal α, β values to be used for each individual region, increasing accuracy while also increasing resource usage on the FPGA.

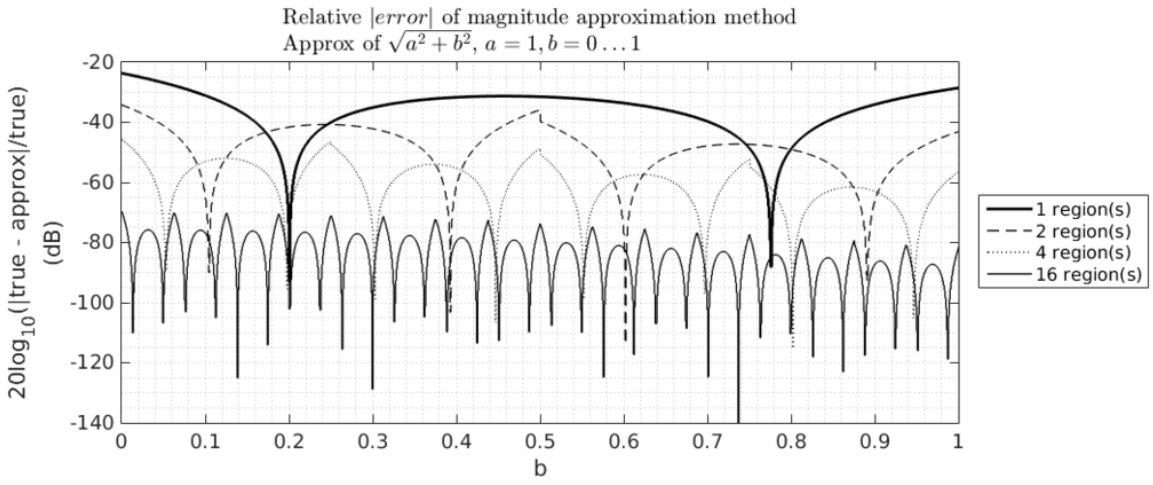

Figure 5.11: This figure shows the same data as figure 5.10, except here the absolute value or the error in dB is shown. The 1 region approach guarantees less than $-20dB$ of error across all values of a, b. The current processing approach uses the 1 region approach because of its simplicity. We do not plan to modify this portion of the algorithm until a comprehensive error analysis of all processing is performed.

Table 5.2: Optimal values of α, β for the single region magnitude estimation approach. Double precision floating point values are listed at the bottom of the table. These were obtained by optimizing against the mean square error in Matlab using the `lsqcurvefit()` function with the default `optimoptions`. The fixed point optimal values were optimized by an exhaustive search of all α, β pairs against the root mean square error. An alternative optimization strategy for the fixed point cases would be to round the optimum DBL FLOAT values to the closest fixed point value. We would use this strategy if larger bit widths > 8 were of interest.

Bits	Optimal α	Optimal β
1	0.5	0.5
2	0.75	0.75
3	0.875	0.5
4	0.9375	0.4375
5	0.9375	0.4375
6	0.9375	0.421875
7	0.9375	0.421875
8	0.933594	0.429688
DBL FLOAT	0.934398526544529	0.426908840694482

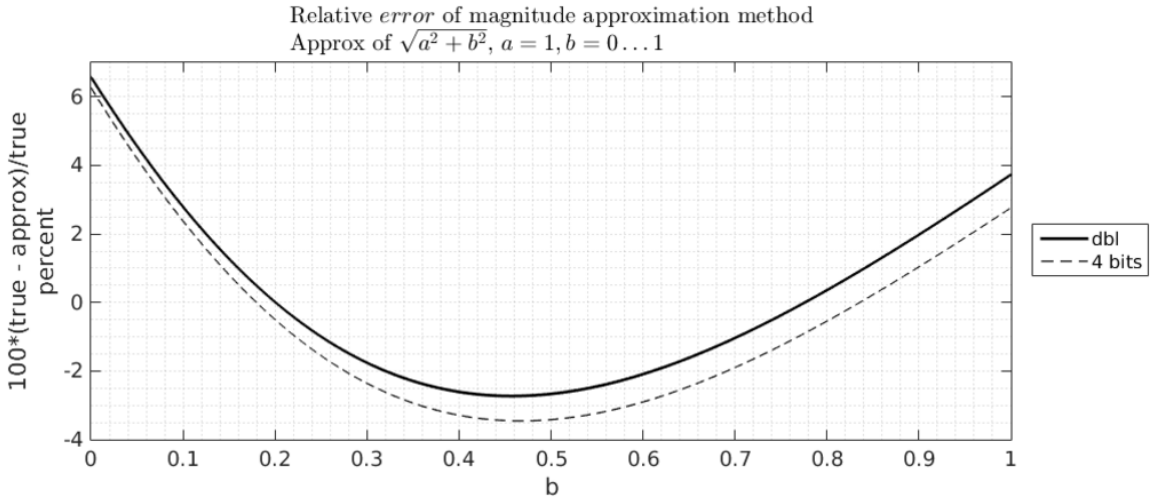

Figure 5.12: Comparison of the error in the single region magnitude approximation when α, β are represented as floating point versus fixed point with 4 bits. See figure 5.13 for the absolute value of the error. Due to the optimal values of α, β in the single region approach, a 4 bit fixed point number provides a decent approximation. In other words, the optimal α, β fall close to the numbers represented by $[0, 2^{-4}, 2 * 2^{-4}, 3 * 2^{-4}, \ldots, 1 - 2^{-1}]$. If a higher order region approach is used then they may require higher bit widths for α, β to maintain accuracy.

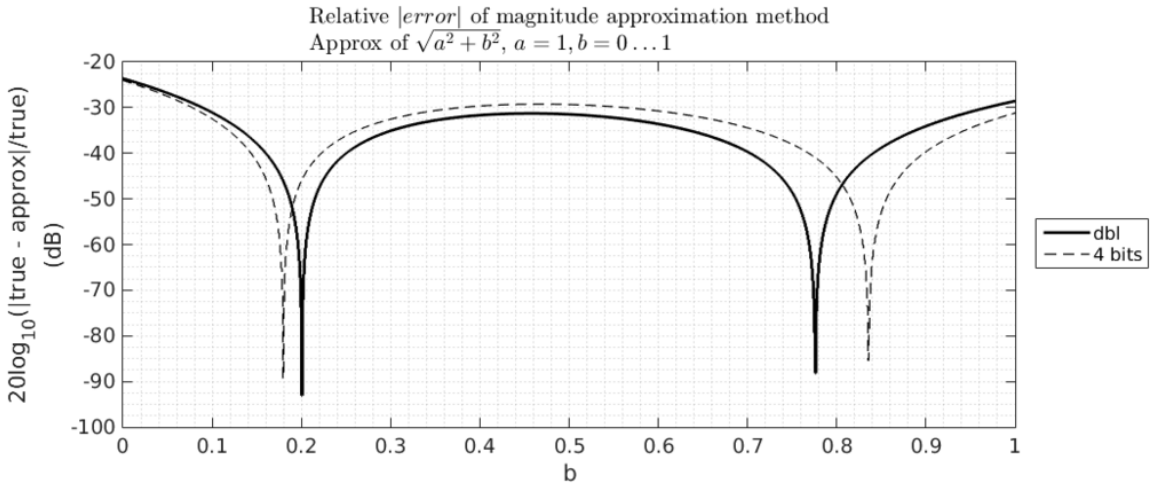

Figure 5.13: Comparison of the absolute value of the error in the single region magnitude approximation when α, β are represented as floating point versus fixed point with 4 bits. The error is plotted as dB. See figure 5.12 for the linear scale error. The error when using 4 bits to represent α, β is below $-20dB$ throughout the full range of possible inputs $a = 1, b = 0 \ldots 1$.

5.9 FPGA: Phase estimation

Phase estimation is currently implemented for one polarization channel only. Phase estimation is not required in the current processing scheme, but may be useful in the future with other OCT systems. The LabVIEW FPGA arctangent IP is used to compute the phase. The phase module is shown in figure 5.14.

Definition of π_{rad}

$$\pi_{rad} = \frac{\text{rad}}{\pi} \tag{5.41}$$

Example, x $= 48$ degrees

$$\text{x} = \underbrace{48}_{\text{degrees}} \equiv \underbrace{0.838}_{\text{rad}} \equiv \underbrace{0.267}_{\pi_{rad}} \tag{5.42}$$

The raw LabVIEW block diagram is shown because of the simplicity of the module.

The real and imaginary parts of the FFT of one of the first polarization channel are inputted to the arctangent module (27 bit numbers from the `phase_approx_rejoiner_output` cluster). The arctangent module is configured for streaming operation, and 16-bits of precision for each output point. The inputs and outputs are registered, this results in a latency of 18 clock cycles (denoted by the `18L` comment). The output was configured at 16-bit so it could easily be used and tested with existing bit-packing code for transfer to the host computer. However, a larger output bit width may be desirable for future applications.

The multiplier, `High Throughput Multiply`, is used to convert from π-radians, π_{rad}, to radians before transferring to the host computer. This is done so that the host computer does not need to perform this conversion. This conversion is necessary because the LabVIEW arctangent module outputs are in units of π_{rad}. Computations using fixed point are typically done in π_{rad} as it can be made more resource efficient. The current disadvantage of performing this computation on the FPGA is that a partial bit of precision is lost. The current 16 bit number returned to the host in radians is $[-4, 0.00012207, 3.99988]$ where the notation is [minimum, increment, maximum]. The MSB of the 16 bit number is only partially filled. The minimum and maximum of the phase is $\pm\pi$, which never reaches the full $[-4, 3.99988]$ range. However, if the π_{rad} to radian conversion is performed on the host computer, the full range of the 16 bit number is used $[-1, 3.05176e-05, 0.999969]$. Thus, we will consider performing the conversion on the host computer in the future if more precision is needed.

5.10 FPGA: Magnitude combination

Magnitude combination is implemented using the same approach as the magnitude estimation processing stage in section 5.8. The only difference is that the precision of the input and output data is increased here. In the original magnitude estimation stage, the magnitude was computed from the real and imaginary parts of the FFT on each of the two

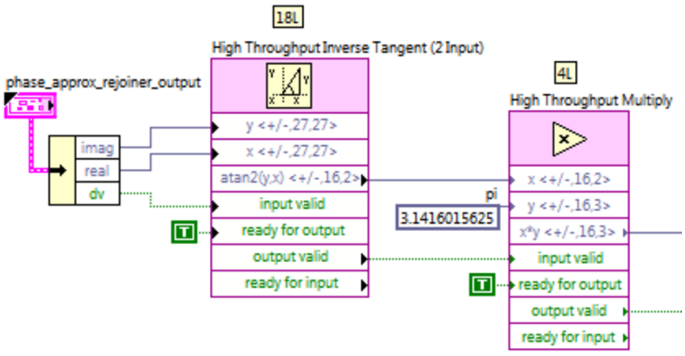

Figure 5.14: Phase estimation LabVIEW FPGA block diagram. The LabVIEW FPGA block diagram is shown here because of the simplicity of the phase estimation processing stage. The real and imaginary parts of the first polarization channel's FFT are inputted to the phase estimation module. The LabVIEW `High Throughput Inverse Tangent (2 Input)` module is used to compute the arctangent in π_{rad}. The following multiplier is used to convert from π_{rad} to rad so that the host computer does not need to perform the conversion.

polarization channels (two instances in the overall design). In magnitude *combination*, the magnitude is computed from the magnitudes of the two polarization channels (one instance in the overall design).

5.11 FPGA: Logarithmic compression

Logarithmic compression is performed on the combined polarization magnitude spectrum in OCT to reduce the dynamic range of the processed interferograms before visualization. The FPGA implementation of logarithmic compression in our algorithm was dictated by the availability of pre-existing IP. We used the LabVIEW FPGA IP for computing the natural log [4]. The LabVIEW IP computes the fractional part of the natural log, but not the integer portion. We have implemented our own algorithm for computing the

[4] http://zone.ni.com/reference/en-XX/help/371599G-01/lvfpga/ht_natural_log/

integer part of the natural log. The natural log is computed as shown below.

$$ln(x) = ln(M) + E \times ln(2) \tag{5.43}$$

$$E = \text{Integer part computed by custom logic (signed)} \tag{5.44}$$

$$M = \text{Fractional part computed by custom logic (unsigned)} \tag{5.45}$$

$$ln(M) = \text{Computed by LabVIEW FPGA IP} \tag{5.46}$$

$$ln(2) = \text{Constant} \tag{5.47}$$

A diagram of the current implementation in the FPGA is shown in figure 5.15. The input data stream is one point per clock and 38 bits per point. The normalizer computes the bit position of the most significant 1 in the input data point. The number is then shifted so that the fractional part of the log resides in the lowest 19 bits of the data point. The lowest 19 bits are used because it is half of the input 38 bit number. Setting the effective decimal point to the middle of the input 38 bit number biases the log calculation evenly between the integer and fractional portions. Biasing the calculation more towards the fractional part (decimal point towards MSB) requires more resources in the LabVIEW FPGA IP. Biasing the calculation more towards the integer part requires less resources but introduces more artifacts in the output number. **We have not** performed any analysis as to the optimal biasing of the decimal point prior to computing the log. Such an analysis would need to be performed on actual OCT data.

The absolute error of the log approximation is shown in figure 5.16 across the entire range of x in $\ln(x)$. For simulation purposes, the error was only computed for decimated values of x. This is because computing all values for the 38 bit number would be memory intensive, $2^{38} \times 8$ bytes assuming double precision floating point for each error value. The

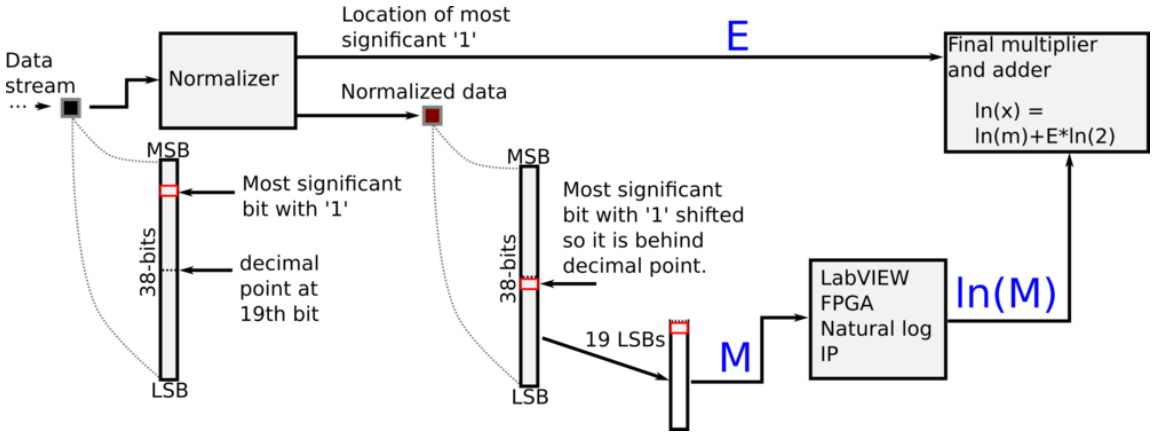

Figure 5.15: Diagram of logarithmic compression algorithm. The terms from equation (5.43) are shown in blue in larger font. This diagram emphasizes the parts of the computation performed by custom logic and LabVIEW FPGA IP, and the effect of the decimal point position on the final result. Placing the decimal point closer to the MSB means that more resources will be used by the LabVIEW FPGA IP to compute $ln(M)$. This will also change how the input data is compressed, and may or may not be optimal depending on the characteristics of the input data (which are not yet defined). The current implementation shown above sets the decimal point to the middle of the input 38-bit data point. Custom logic (normalizer) finds the most significant 1 in the input 38-bit number, and logically shifts the 38-bit number so that the most significant 1 is behind the decimal point. The shift amount is the E term. The 19 LSBs of the normalized data are then inputted to the LabVIEW FPGA IP to compute the natural log. Finally, multipliers and adders are used to perform the final computations to generate $ln(x)$.

peak absolute error is $5.481e-4$ over all $x = [0, 2^{-19}, 2 \times 2^{-19}, 3 \times 2^{-19}, \ldots, 2^{19} - 2^{-19}]$ in $\ln(x)$.

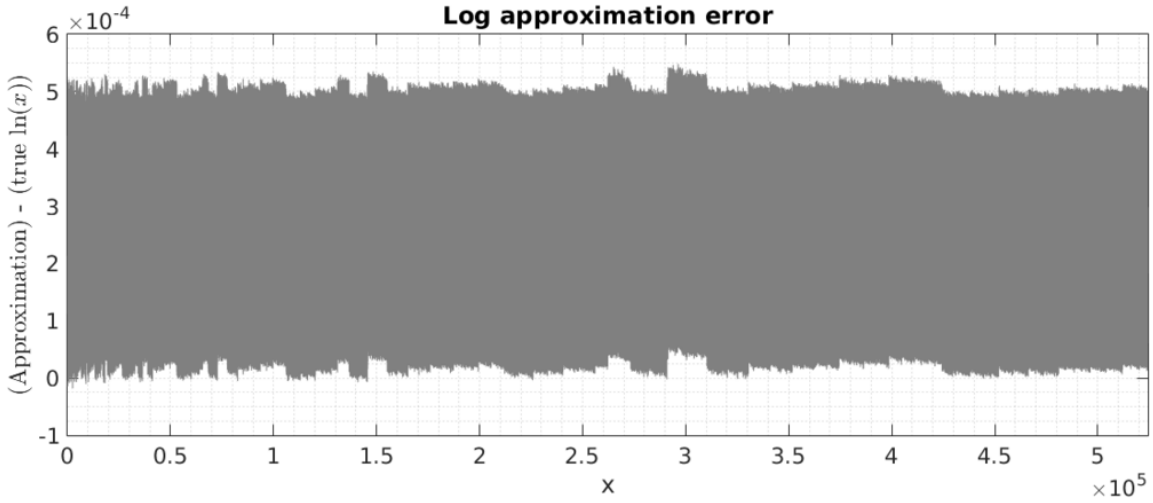

Figure 5.16: Error of log approximation method across the full range of input values. The peak error is $5.481e-4$. The argument x is 2^{38} with a decimal point at the 19^{th} bit. In other words, $x = [0, 2^{-19}, 2 \times 2^{-19}, 3 \times 2^{-19}, \ldots, 2^{19} - 2^{-19}]$. All of the 2^{38} numbers were not evaluated as this would be computationally intense. Evaluating all 2^{38} numbers (using double floating point precision) would require $8 \times 2^{38} = 2048$ GiB of memory.

5.12 FPGA: Depth cropping

Depth cropping is the last processing stage. It allows for cropping the data before it is transferred from the FPGA to the host computer. This is advantageous if initial or final depth points do not contain desired signal. The restriction on depth cropping is that the number of passed points must be a multiple of 4. This is due to how the magnitude data is 16-bits, and the optimal FPGA to host transfer format is 64-bits. Currently the bit packing logic is limited to transferring full 64-bit numbers only.

5.13 Validation tests

5.13.1 1310±50nm Thorlabs VCSEL laser (1 polarization channel)

This section describes a limited set of experiments that qualitatively test the performance of the FPGA processing design against performing all processing on the CPU. The SS OCT system from chapter 4 was used for this section with minor modifications:

- The full 100 kHz sweep rate was used (instead of decimating to 50 kHz).
- The sampling frequency was further reduced from 400 MS/s to 200 MS/s by decimating 1 out of every 2 time points.
- The anti-aliasing filters (after the mzi-clock and after the balanced detector) were replaced with minicircuits BLP-90+ lowpass filters (81 MHz 3dB cutoff) to avoid aliasing when sampling at 200 MS/s.

Before presenting the FPGA data, various CPU data comparisons were made to help compare FPGA image quality to CPU processing quality when the same processing compromises (one mzi-clock for all interferograms, linear interpolation) were made.

The following terms are used in this analysis:

- Individual mzi: refers to using the co-acquired mzi clock signal to interpolate each interferogram.
- Global mzi: refers to using one mzi signal to interpolate all interferograms in a given volume.

The CPU processing figures are grouped by the placement of the optical delay line (ODL, refer to chapter 4 for more information). The ODL was changed to 3 positions: 1/4, 1/2, and 3/4 of the total single sided imaging depth. This helped demonstrate the

performance of the various processing combinations as a function of imaging depth. Then volumes of finger were acquired with FPGA processing at approximately the same ODL locations as the CPU processing volumes. The FPGA processing algorithm corresponds to the CPU processing case of global mzi linear interpolation. Figures were then plotted that compare the FPGA processed bscans against the CPU processed bscans using global mzi linear interpolation and individual mzi linear interpolation (all cases used dispersion correction). An outline of the all processing figures is shown in table 5.3.

Table 5.3: Outline of figures for SS OCT processing validation with VCSEL laser.

Figure	CPU or FPGA	ODL	interp.	disp.
5.17, 5.18	CPU	1/4	linear	on/off
5.19, 5.20	CPU	2/4	linear	on/off
5.21, 5.22	CPU	3/4	linear	on/off
5.23, 5.24	CPU	3/4	spline	on
5.25, 5.26	FPGA	1/4	linear	on
5.27, 5.28	FPGA	2/4	linear	on
5.29, 5.30	FPGA	3/4	linear	on

CPU processing comparisons are shown in figures 5.17-5.24. As expected, no interpolation is the worst case. Enabling dispersion correction made the bscan sharper, but introduced echo artifacts (panels B to C). Using global mzi rather than individual mzi interpolation resulted in what appears as smearing in the axial dimension. This is more evident at ODL positions 2/4 and 3/4 than 1/4. Figure 5.31 show two bscans of a finger with and without dispersion correction (not the exact same bscan, but acquired in quick sucession $<$ 1 second). Similar to the CPU cases, increased echo artifacts are visible when applying dispersion correction on the FPGA. However, the echos on the FPGA appear to have higher intensity and appear more frequently in the bscan.

Overall, FPGA processed data at ODL locations $>$2/4 performed slightly worse than

CPU processing with global mzi and linear interpolation due to increased echo artifacts. The echo artifacts are likely due to insufficient precision in either the interpolation stage or the dispersion correction processing stage. More time would be needed to fully optimize the precision of the various FPGA processing stages. But additional precision is feasible given the amount of free resources mentioned in 5.1. The results are promising considering this is the first time that this SS OCT FPGA processing design has been evaluated on a functional SS OCT system. However, this set of tests did not demonstrate the 2 channel combining stage of the processing pipeline as the SS OCT system had only 1 channel. It is realistic to expect that the 2 channel combining stage would perform adequately since it is the same code that computes the magnitude of the Fourier transform data after the FFT processing stage.

5.14 Supplemental

This section describes other higher throughput processing methods that were implemented but were not included in the previous SS OCT discussion. This information is intended to facilitate further development of higher speed (greater than the 200 MHz used in this work) SS OCT processing systems.

5.14.1 1.6 GS/s (two channels at 800 MS/s) data valid processing for Insight SS OCT lasers

5.14.1.1 Introduction

At the time of this writing, the vendor Insight was capable of producing swept source lasers that provided a stable sampling clock for use with most analog to digital converter boards. In this discussion stable means a clock with a consistent frequency, typically 400 MHz at the time of this writing, with newer models having an adjustable frequency (ex: 10 MHz). This is in contrast to typical swept source lasers that offer a clock output in the form of a k-clock, which outputs clock signals according to the output wavelength of

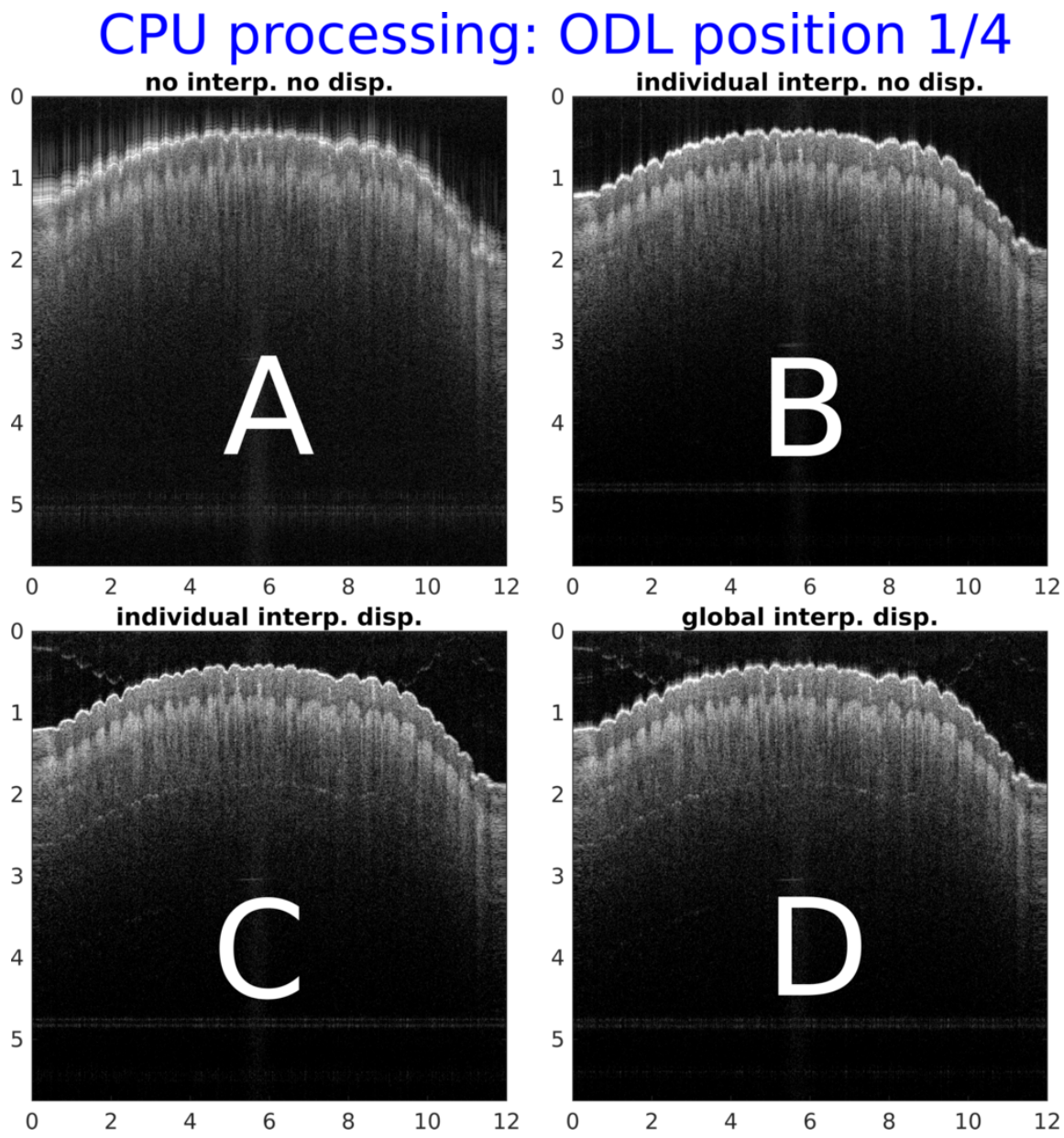

Figure 5.17: SS OCT bscans of finger with various processing stages disabled to illustrate the impact on image quality. All interpolation performed here was linear. The ODL was adjusted so that the focal plane of the objective lens was approximately 1/4 of the nyquist sampling frequency (this is the meaning of ODL position 1/4). The horizontal and vertical axes are labeled in mm (index of refraction of air assumed throughout). Total imaging depth was measured to be approximately 5.75 mm, meaning 1/4 ODL is at a depth of 1.44 mm in the bscans.

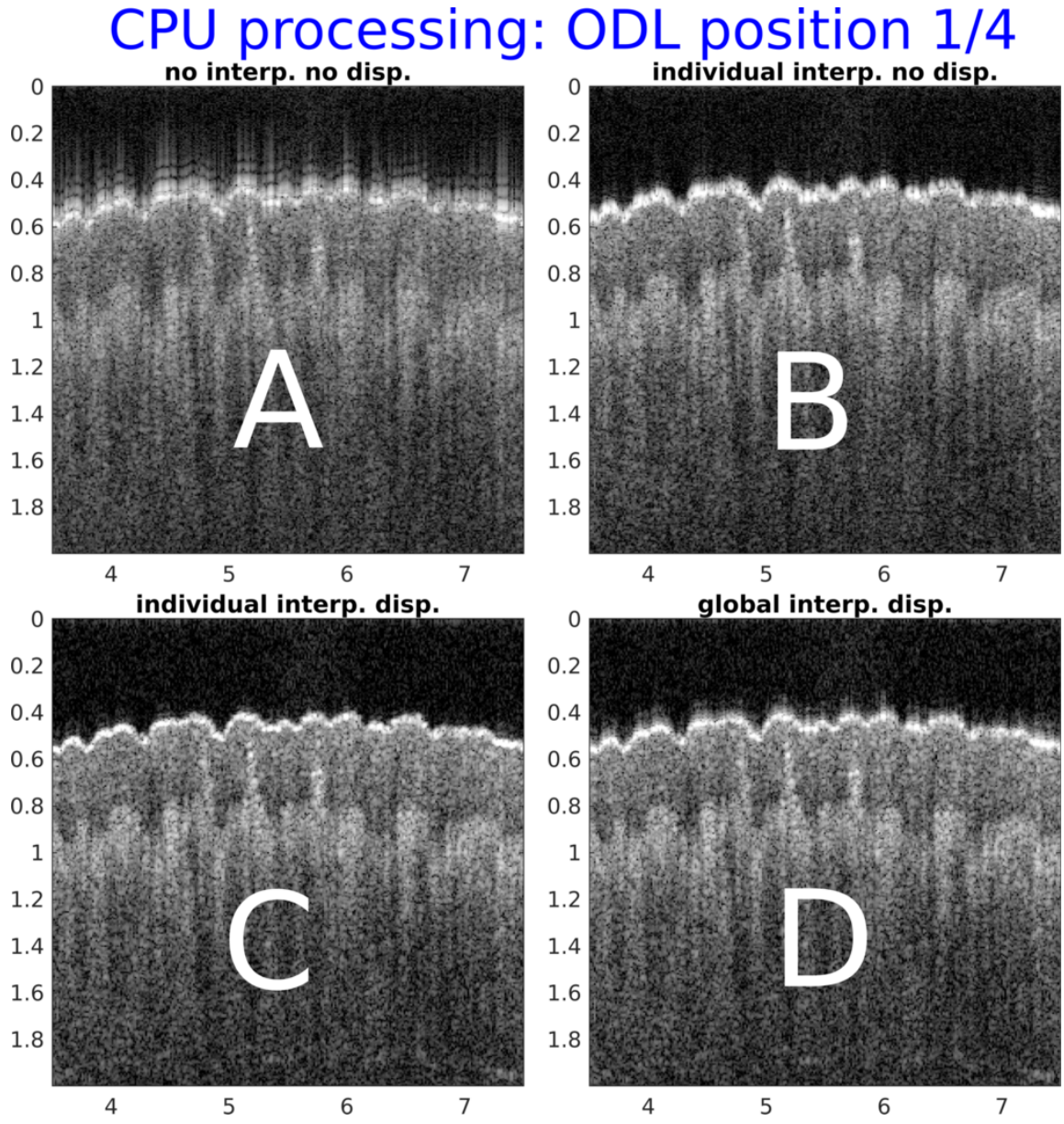

Figure 5.18: Zoomed view of figure 5.17.

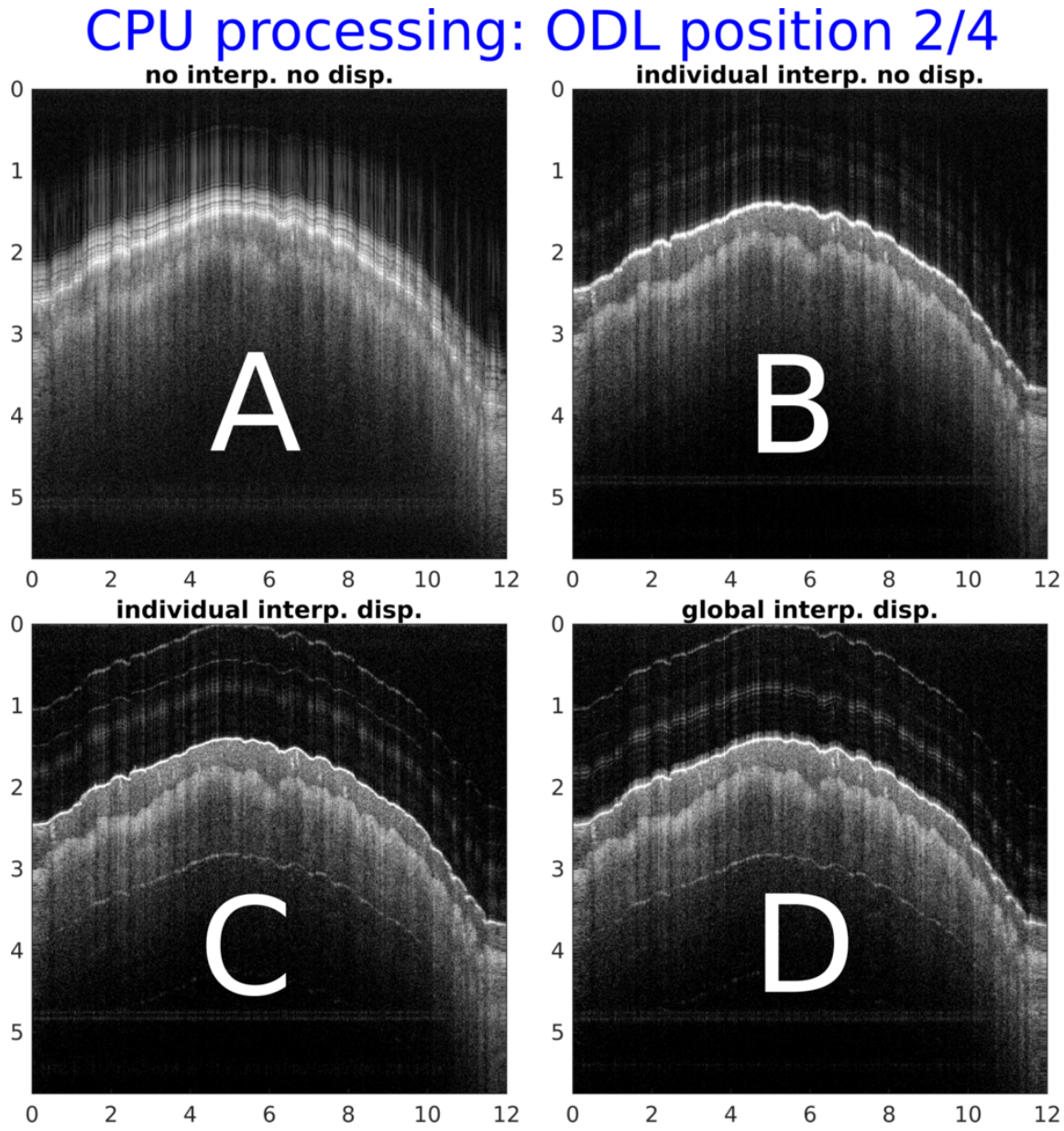

Figure 5.19: SS OCT bscans of finger with various processing stages disabled to illustrate the impact on image quality. All interpolation performed here was linear. The ODL was adjusted so that the focal plane of the objective lens was approximately 2/4 of the nyquist sampling frequency (this is the meaning of ODL position 2/4). The horizontal and vertical axes are labeled in mm (index of refraction of air assumed throughout). Total imaging depth was measured to be approximately 5.75 mm, meaning 2/4 ODL is at a depth of 2.88 mm in the bscans. The full single sided imaging depth is shown, and each bscan consisted of 800 alines of length 1024.

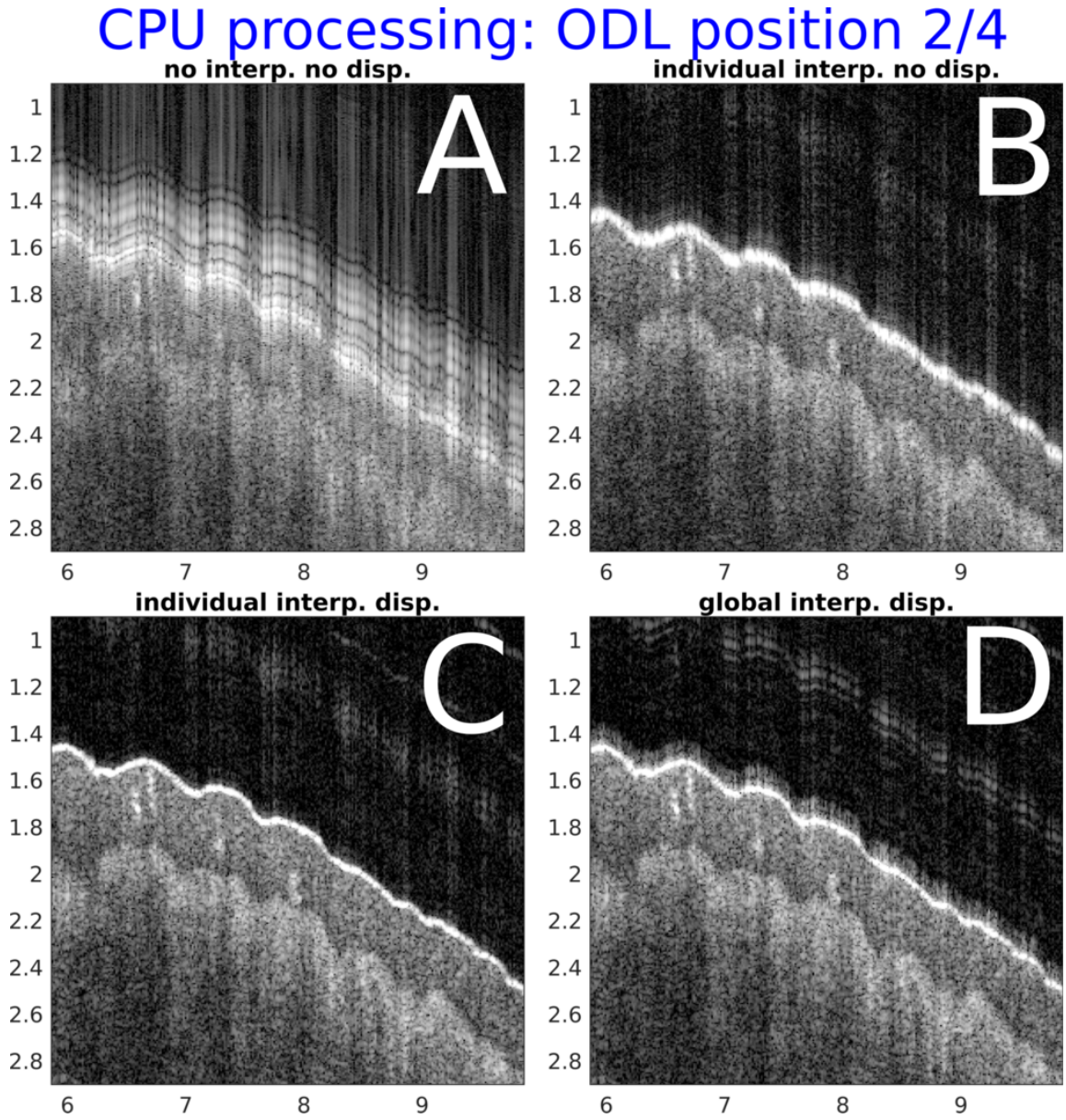

Figure 5.20: Zoomed view of figure 5.19.

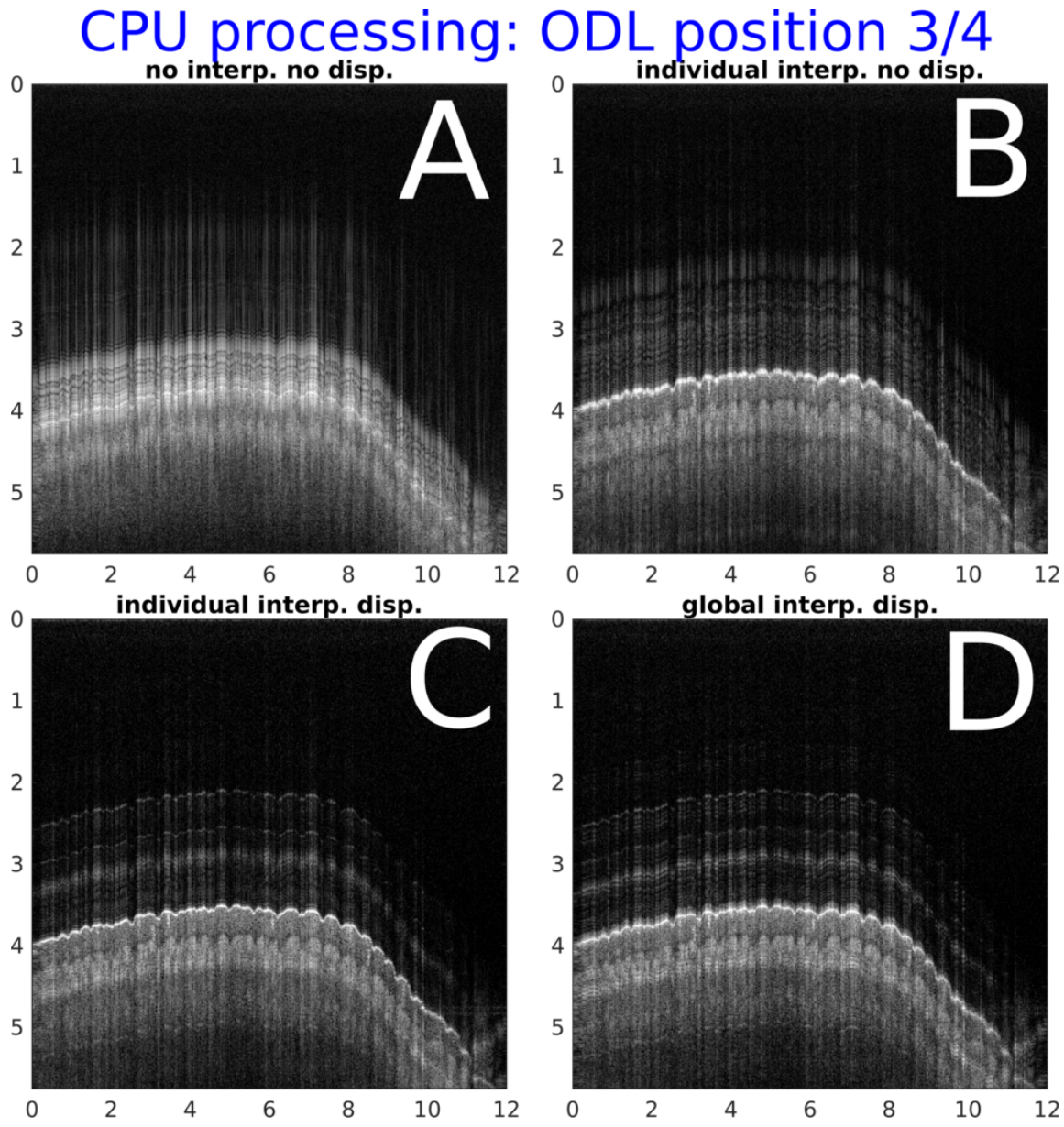

Figure 5.21: SS OCT bscans of finger with various processing stages disabled to illustrate the impact on image quality. All interpolation performed here was linear. The ODL was adjusted so that the focal plane of the objective lens was approximately 3/4 of the nyquist sampling frequency (this is the meaning of ODL position 3/4). The horizontal and vertical axes are labeled in mm (index of refraction of air assumed throughout). Total imaging depth was measured to be approximately 5.75 mm, meaning 3/4 ODL is at a depth of 4.31 mm in the bscans. The full single sided imaging depth is shown, and each bscan consisted of 800 alines of length 1024.

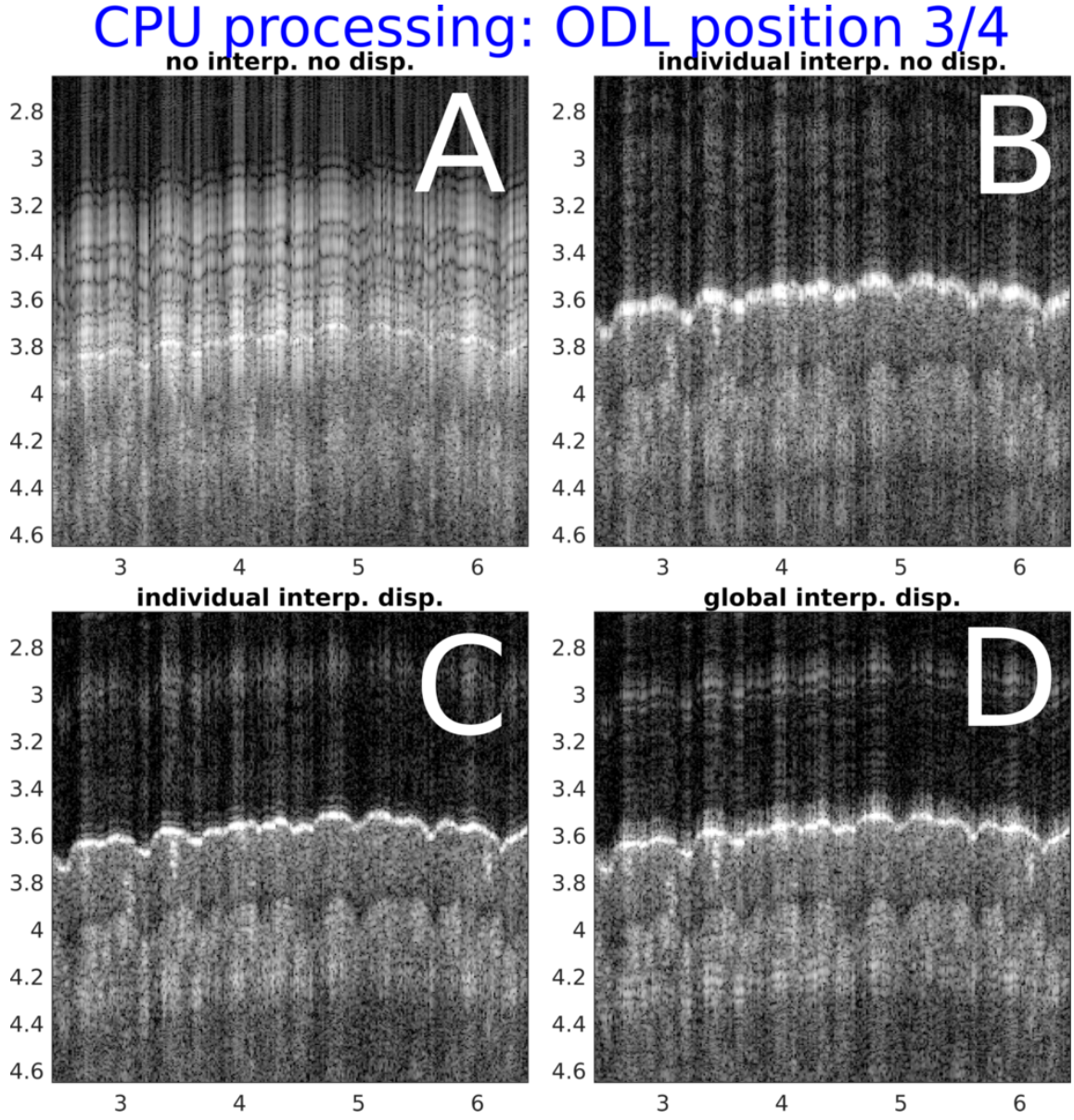

Figure 5.22: Zoomed view of figure 5.17.

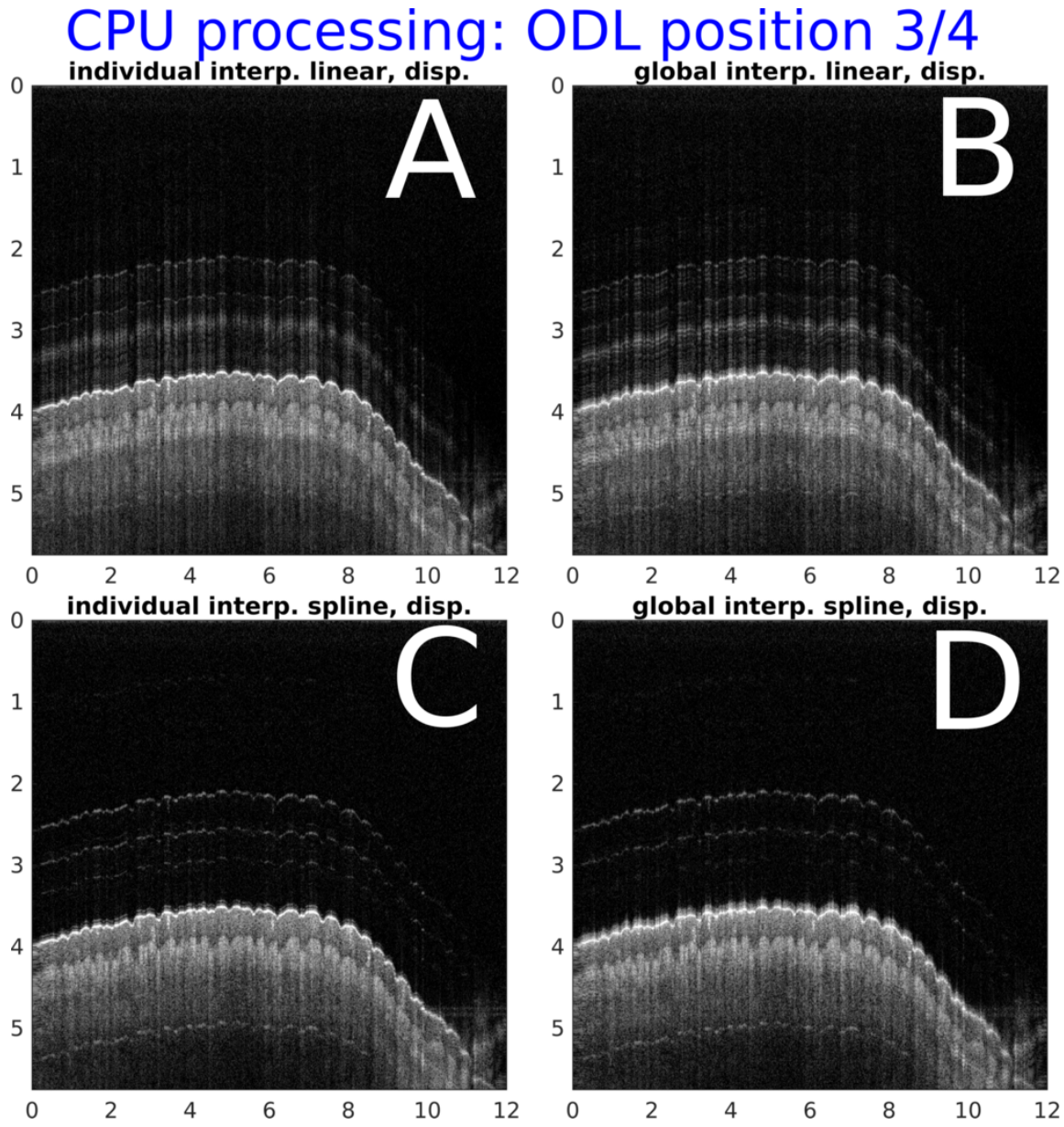

Figure 5.23: SS OCT bscans of finger comparing linear interpolation and spline interpolation as well as individual versus global mzi interpolation. The ODL was adjusted so that the focal plane of the objective lens was approximately 3/4 of the nyquist sampling frequency (this is the meaning of ODL position 3/4). The horizontal and vertical axes are labeled in mm (index of refraction of air assumed throughout). Total imaging depth was measured to be approximately 5.75 mm, meaning 3/4 ODL is at a depth of 4.31 mm in the bscans. The full single sided imaging depth is shown, and each bscan consisted of 800 alines of length 1024.

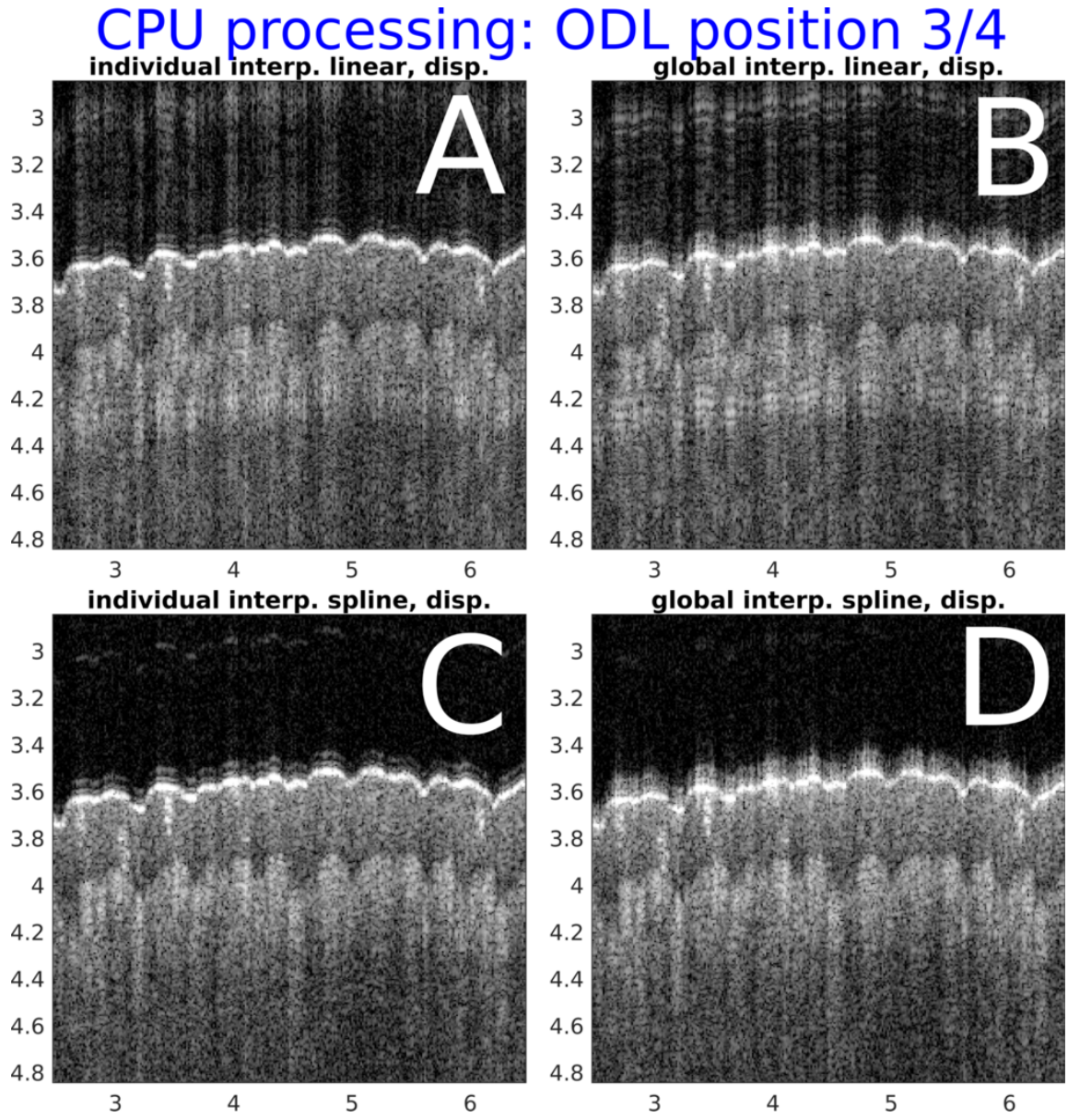

Figure 5.24: Zoomed view of figure 5.23.

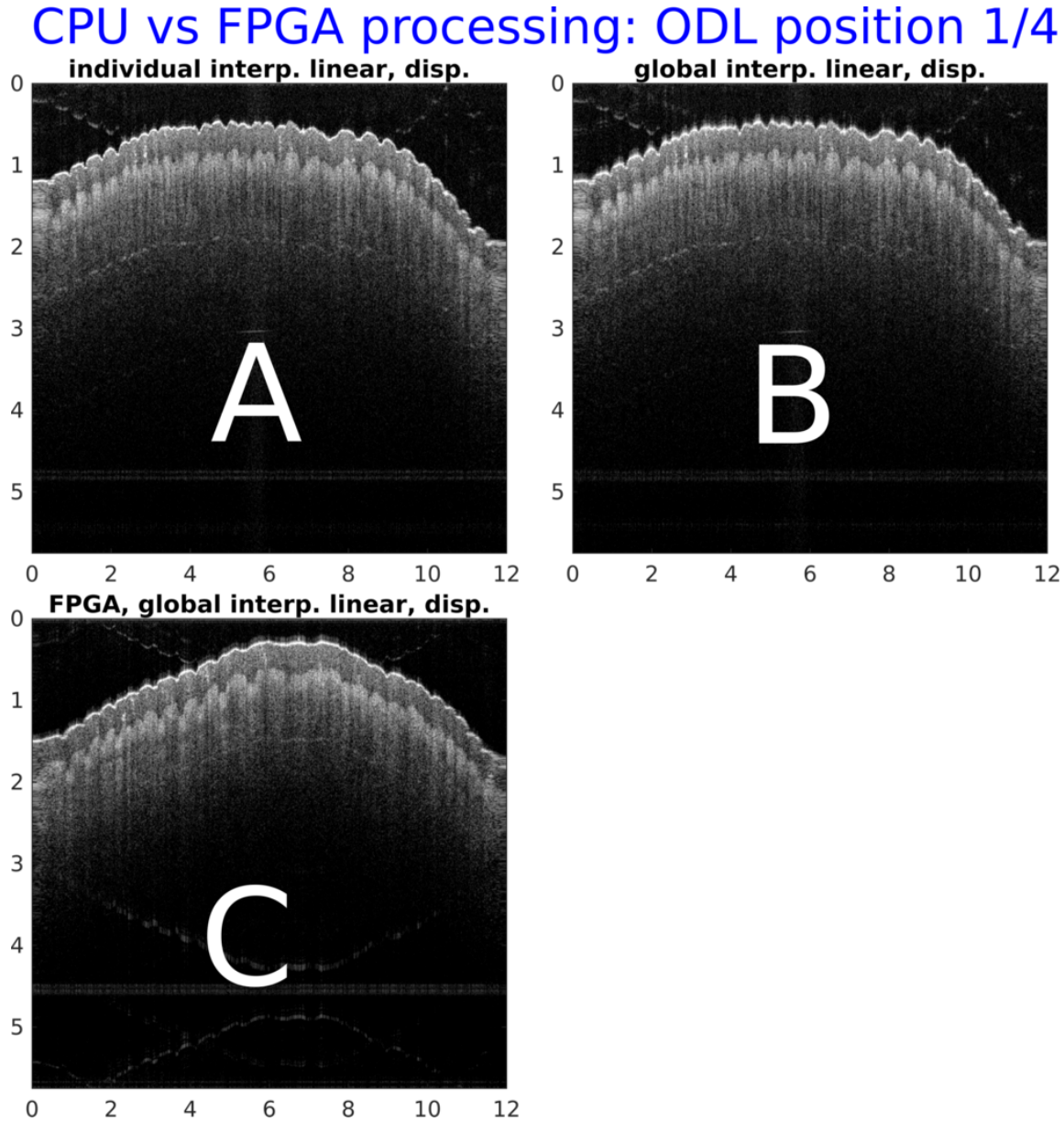

Figure 5.25: SS OCT bscans of finger comparing CPU processing with individual mzi linear interpolation (panel A), CPU processing with global mzi linear interpolation (panel B), and FPGA processing with global mzi and linear interpolation (panel C). The same bscan was used for A and B, but a separate bscan was acquired for the FPGA processing case in C. The ODL was adjusted so that the focal plane of the objective lens was approximately 1/4 of the nyquist sampling frequency (this is the meaning of ODL position 1/4). The horizontal and vertical axes are labeled in mm (index of refraction of air assumed throughout). Total imaging depth was measured to be approximately 5.75 mm, meaning 1/4 ODL is at a depth of 1.44 mm in the bscans. The full single sided imaging depth is shown, and each bscan consisted of 800 alines of length 1024.

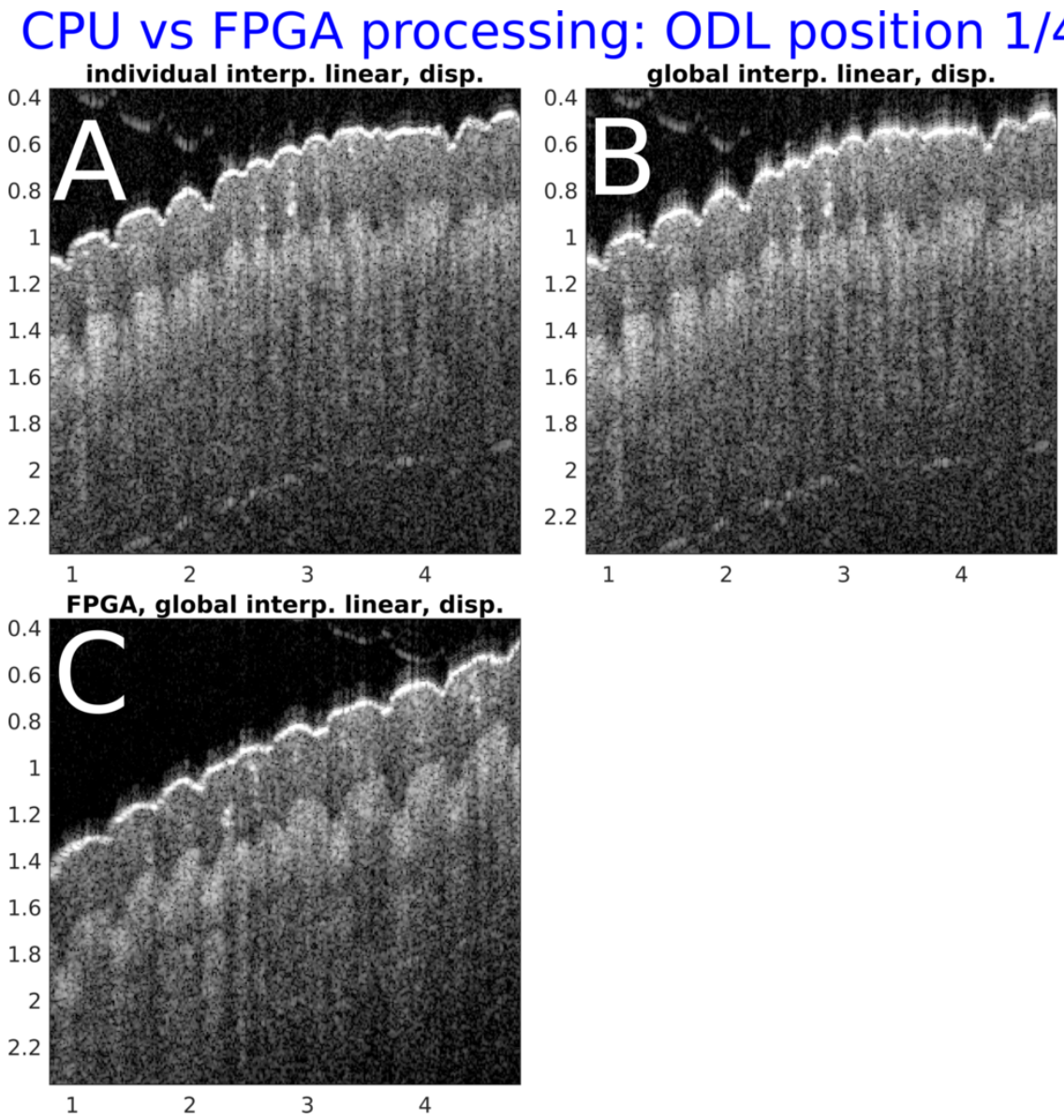

Figure 5.26: Zoomed view of figure 5.25.

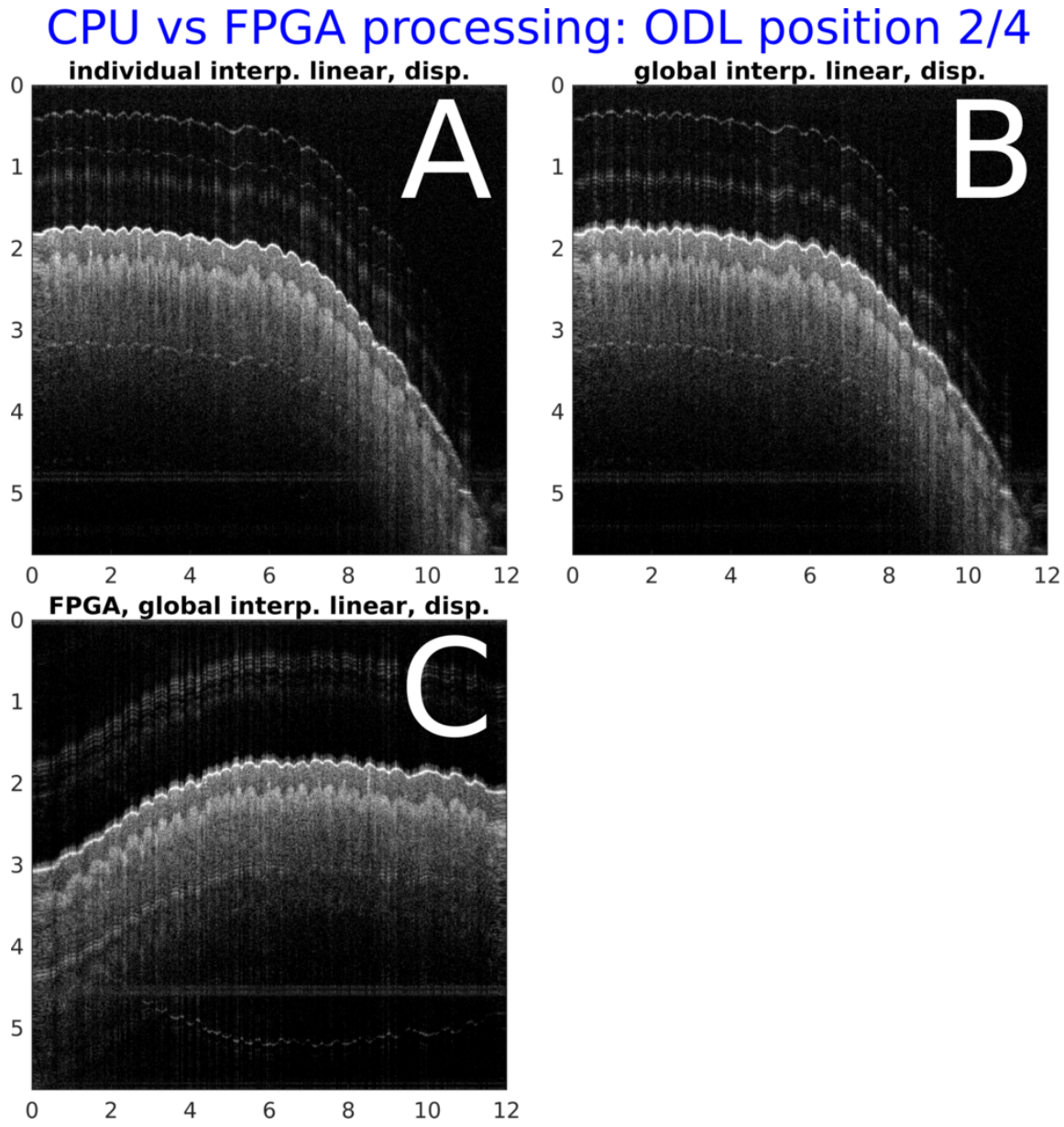

Figure 5.27: SS OCT bscans of finger comparing CPU processing with individual mzi linear interpolation (panel A), CPU processing with global mzi linear interpolation (panel B), and FPGA processing with global mzi and linear interpolation (panel C). The same bscan was used for A and B, but a separate bscan was acquired for the FPGA processing case in C. The ODL was adjusted so that the focal plane of the objective lens was approximately 2/4 of the nyquist sampling frequency (this is the meaning of ODL position 2/4). The horizontal and vertical axes are labeled in mm (index of refraction of air assumed throughout). Total imaging depth was measured to be approximately 5.75 mm, meaning 2/4 ODL is at a depth of 2.88 mm in the bscans. The full single sided imaging depth is shown, and each bscan consisted of 800 alines of length 1024.

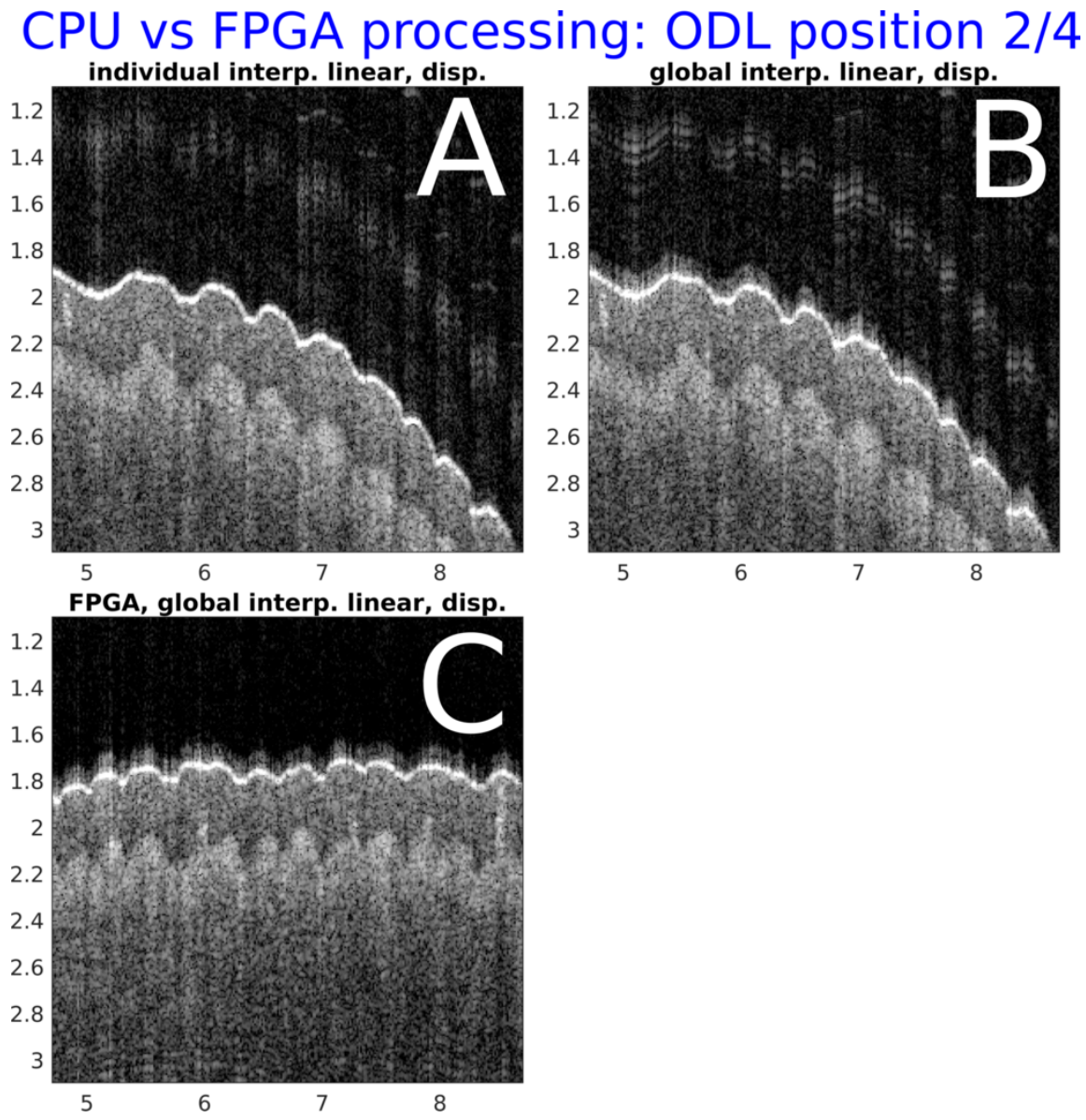

Figure 5.28: Zoomed view of figure 5.27.

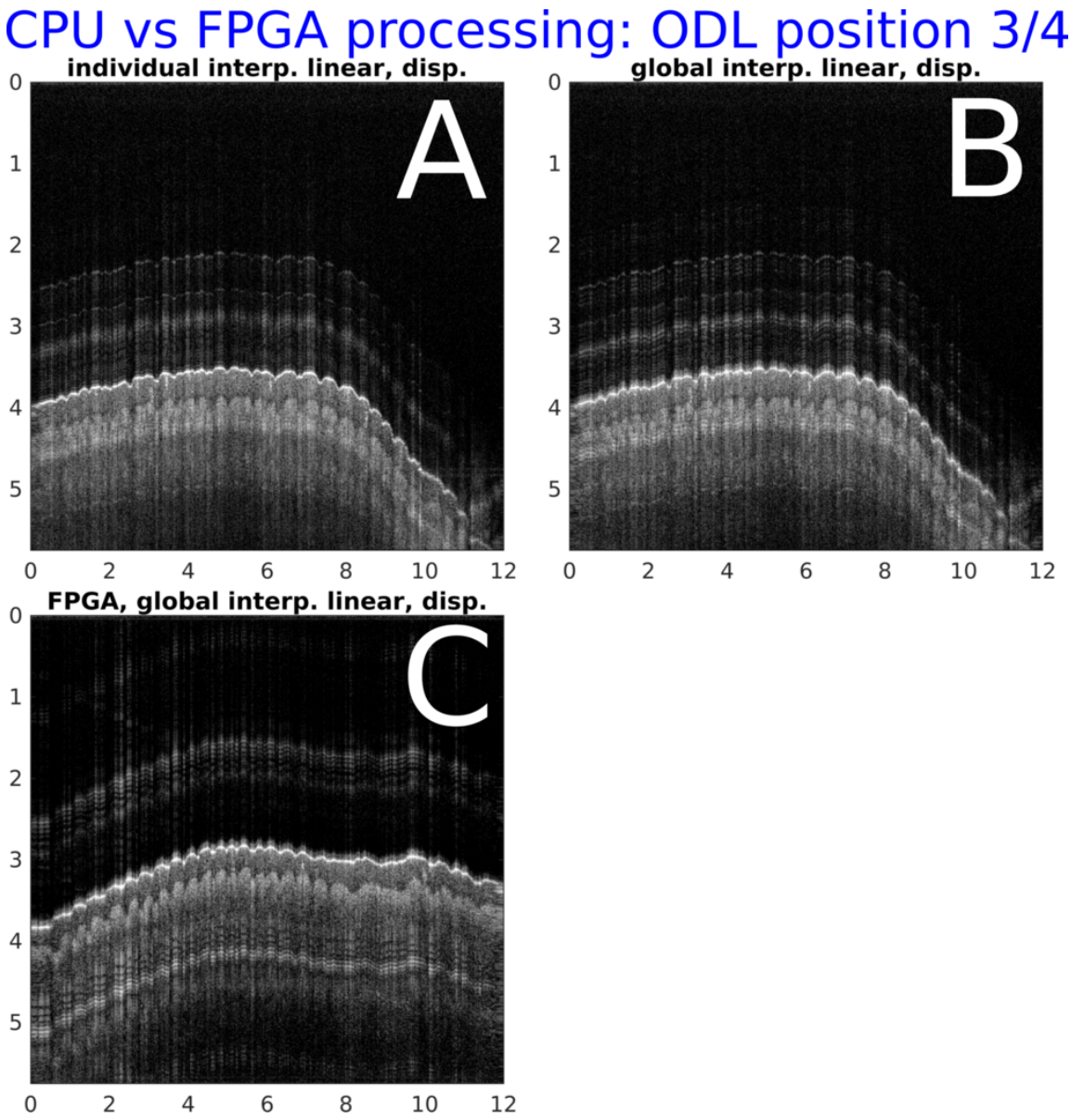

Figure 5.29: SS OCT bscans of finger comparing CPU processing with individual mzi linear interpolation (panel A), CPU processing with global mzi linear interpolation (panel B), and FPGA processing with global mzi and linear interpolation (panel C). The same bscan was used for A and B, but a separate bscan was acquired for the FPGA processing case in C. The ODL was adjusted so that the focal plane of the objective lens was approximately 3/4 of the nyquist sampling frequency (this is the meaning of ODL position 3/4). The horizontal and vertical axes are labeled in mm (index of refraction of air assumed throughout). Total imaging depth was measured to be approximately 5.75 mm, meaning 3/4 ODL is at a depth of 4.31 mm in the bscans. The full single sided imaging depth is shown, and each bscan consisted of 800 alines of length 1024.

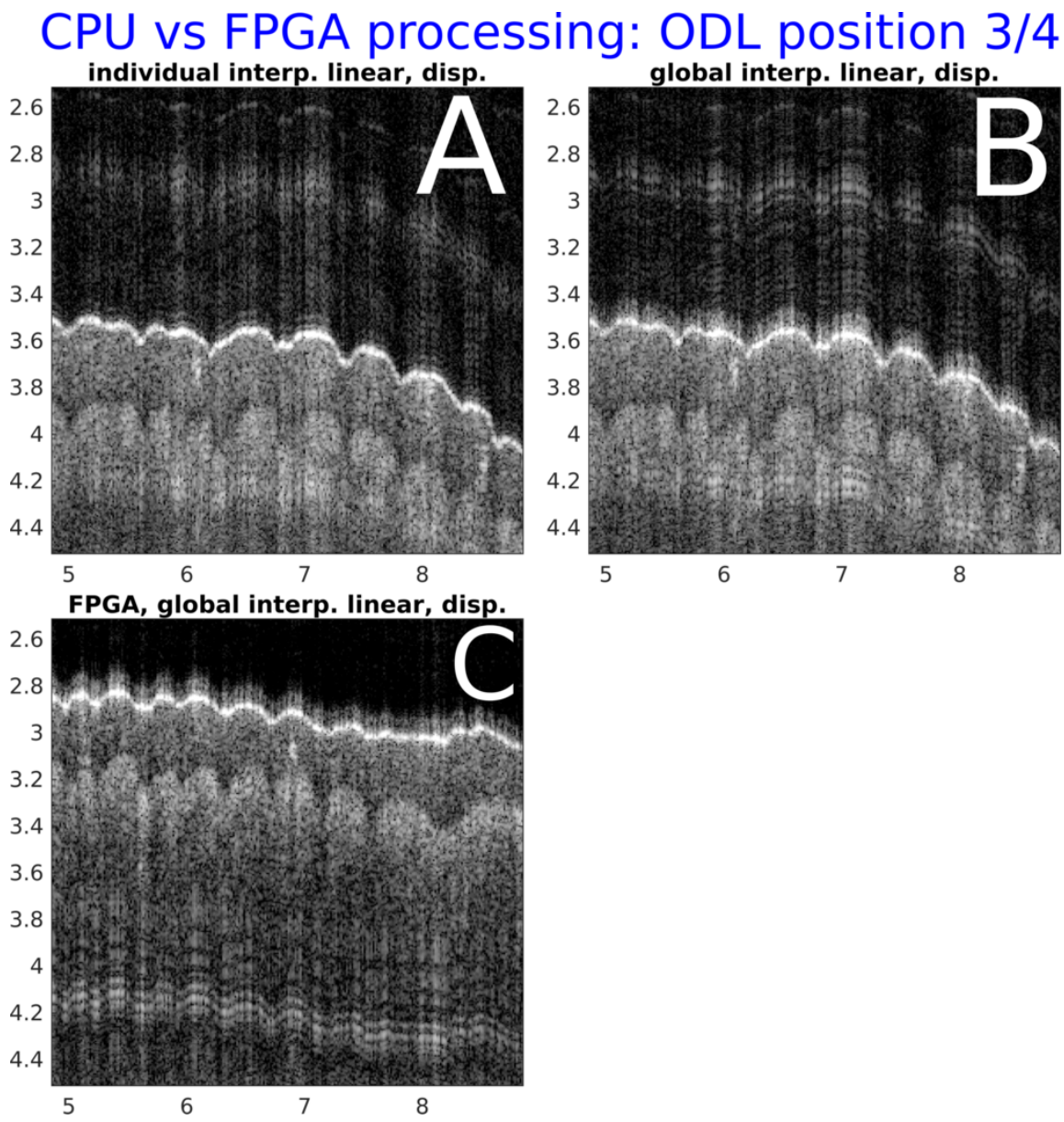

Figure 5.30: Zoomed view of figure 5.29.

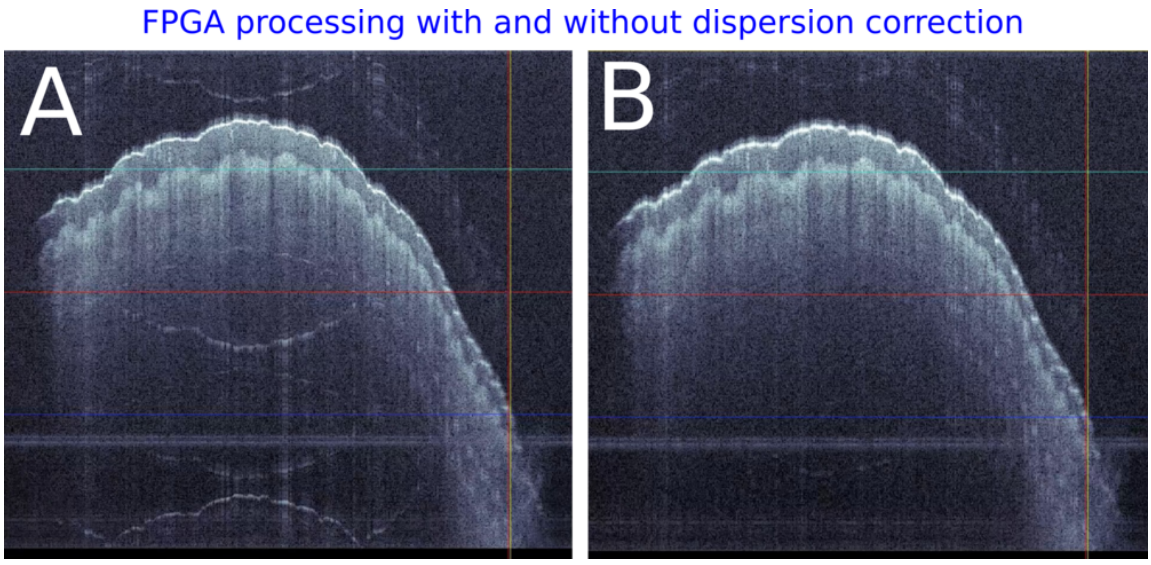

Figure 5.31: SS OCT bscans of finger comparing FPGA processing with and without dispersion correction. Bscan A (left) was acquired with dispersion correction enabled. Bscan B (right) was acquired in close temporal succession (not the exact same bscan) and had dispersion correction disabled. More echo artifacts are noticeable in bscan A. These bscans are screenshots from our LabVIEW acquisition interface, and thus have a slightly different color scheme relative to the bscans shown earlier in this section. Axes are not labeled, but correspond to the same range in the lateral and depth dimensions as the previous non-zoomed bscan figures (800 lateral and 1024 depth pixels).

the laser. In these lasers, the output may change between sweeps, and the k-clock output is always low between sweeps. Some analog to digital converter boards are able to work with this signal, but most others require a consistent clock for use with an internal PLL and will not function with such a k-clock.

If k-clock sampling is not possible, the digitized data must be interpolated to be linear in k-space before performing the Fourier Transform step in OCT processing. Interpolation is a resource intensive process, and can introduce artifacts into the k-linear output data (lower axial resolution, echos at higher frequencies, etc.).

The advantage of the Insight lasers is that they allow for discarding sweep points to obtain a k-linear output sweep without having to perform interpolation. Several discarding strategies exist:

1. Using DV signal from Insight laser on acquisition card.
2. Using archived DV vector.
    (a) Discard on FPGA ← **this is the method implemented here**.
    (b) Discard after acquiring on host computer.

5.14.1.1.1 Method 1 is conceptually the easiest, but requires an extra digital input that is synchronous to the analog to digital conversion. In this case the laser provides a TTL output signal that is high when points in the sweep are valid and low when they should be discarded. The idea is that this DV signal is acquired along with the OCT signal and used to discard points in real time (assuming the user is able to accomplish this with their hardware). However, if the user's hardware does not allow for measuring the DV signal along with the data then this discarding approach cannot be performed. One possible alternative would be to logically AND the sweep trigger and DV signal outputs from the laser before they are input to the digitizer. Then the trigger input could be used to both start

the acquisition of a sweep and discard invalid time points. Any significant skew (electrical and/or optical) between the DV signal and the OCT signal would need to be corrected by adding a delay in the user's hardware (ex: FPGA or an external delay line).

5.14.1.1.2 Method 2a involves first acquiring a known DV vector from the laser itself, and then using that vector to discard points in future data. The stability of the laser makes this feasible at each sweep is identical in its temporal k-space value relative to the sweep trigger synchronous to the laser's clock (which the ADC should be synchronized to). The DV vector is typically obtained from the laser's computer interface after it performs an internal calibration. The resulting text file contains index numbers that indicate which points in a given sweep are valid. The DV vector can then be downloaded to the FPGA and used to ignore points in real time. The stability of the laser allows for using the same DV vector for extended periods of time (several days at least). Any significant skew between the DV vector and the OCT signal would need to be corrected for by adding a delay to the digitized OCT signal in the FPGA before discarding points.

5.14.1.1.3 Method 2b is equivalent to performing method 2a after data is acquired and transferred to the host computer. This will likely take the most amount of time as it requires transferring more data to the host computer, and then discarding invalid points on the host computer. However, this is likely choice when first prototyping or when troubleshooting. So it is useful to have a rudimentary implementation of this method.

5.14.1.2 Implementation

Method 2a was implemented on the NI 5772 and PXIe 7966 digitizer and FPGA as shown in figure 5.32.

In step 1 (figure 5.32) the acquisition logic handled trigger detection and acquisition of input data from analog input (AI) 0 and AI1. Data was packaged from a chunk of 4 into a chunk of 8 12-bit data points before transferring to the processing logic in the 100

MHz clock domain via FIFO.1. This allowed the 100 MHz loop to maintain the same data throughput as the 200 MHz loop.

Step 2, data valid filtering consisted of marking various points as valid or invalid and sorting them so that they occupied continuous sub-chunks of the 8 chunk data (see figures 5.33 and 5.34). The data valid information was uploaded to the FPGA from the host computer prior to acquiring data (the same data valid vector was used to filter all sweeps of the laser).

The chunk of 8 data points was then sent to an accumulator that functioned to gather 8 data points from the possible 0 to 8 valid input points (see figure 5.35). Once 8 valid points were accumulated it was passed downstream to FIFO.2 (or alternatively to more processing logic), which sent the data back to the 200 MHz clock domain to be transferred to the host computer (as shown in figure 5.32. LabVIEW FPGA (2014 version) only provides one type of FPGA to host computer FIFO. Data is always transferred in 64-bit numbers. Thus the only way to maintain the full data throughput was to implement the FPGA to host transfer in the 200 MHz clock domain.

5.14.2 400 MS/s FFT (4 points per clock cycle at 100 MHz)

5.14.2.1 Theory

The DFT of a sequence of N points (with equal time spacing) is defined as:

$$X[k] = \sum_{n=0}^{n=N-1} x[n]e^{-j2\pi kn/N},\ k = \{0, 1, ..., N-1\},\ j = \sqrt{-1} \tag{5.48}$$

where $x[n]$ denotes the signal in the time domain, n denotes the time index, k denotes the frequency index, and $X[k]$ denotes the DFT of the x signal at frequency index k. The sequence of N points can be decimated in time and split into 4 streams such that the first stream consists of $4n$, the second $4n + 1$, the third $4n + 2$, and the fourth $4n + 3$. In this

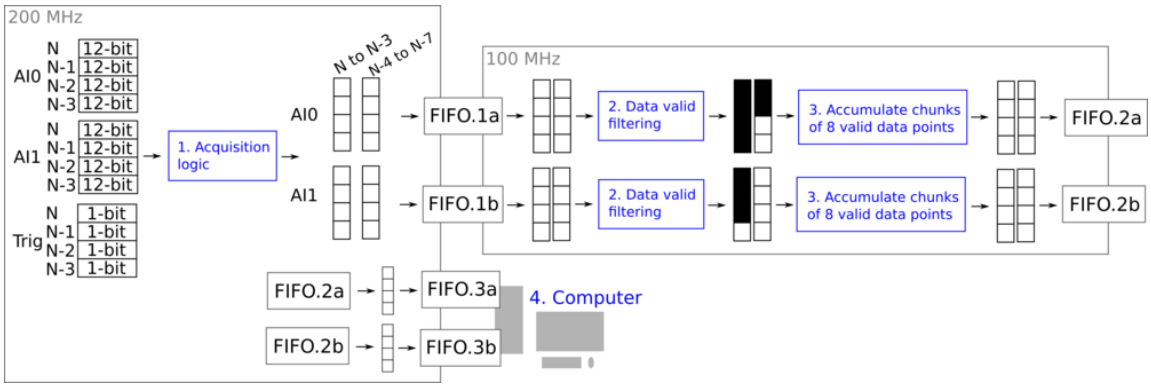

Figure 5.32: Overview of the FPGA data valid processing scheme for use with Insight style SS OCT lasers and the NI 5772 and PXIe 7966 digitizer and FPGA. First, data was acquired from both analog inputs at 800 MS/s (4 data points per clock cycle at 200 MHz). Then two chunks of 4 data points per analog input channel were accumulated before being transferred to the 100 MHz loop via FIFO.1. Chunks of 8 data points per analog input channel were read in the 100 MHz loop and filtered according to a pre-uploaded (computer to FPGA) list of points to ignore. The resulting output data was accumulated so that only valid points were present in chunks of 8 (per analog input channel) before being transferred back to the 200 MHz clock domain via FIFO.2. Finally, data was read from FIFO.2 and transferred to the host computer via FIFO.3 in chunks of 4 data points as a 64-bit number for both data channels.

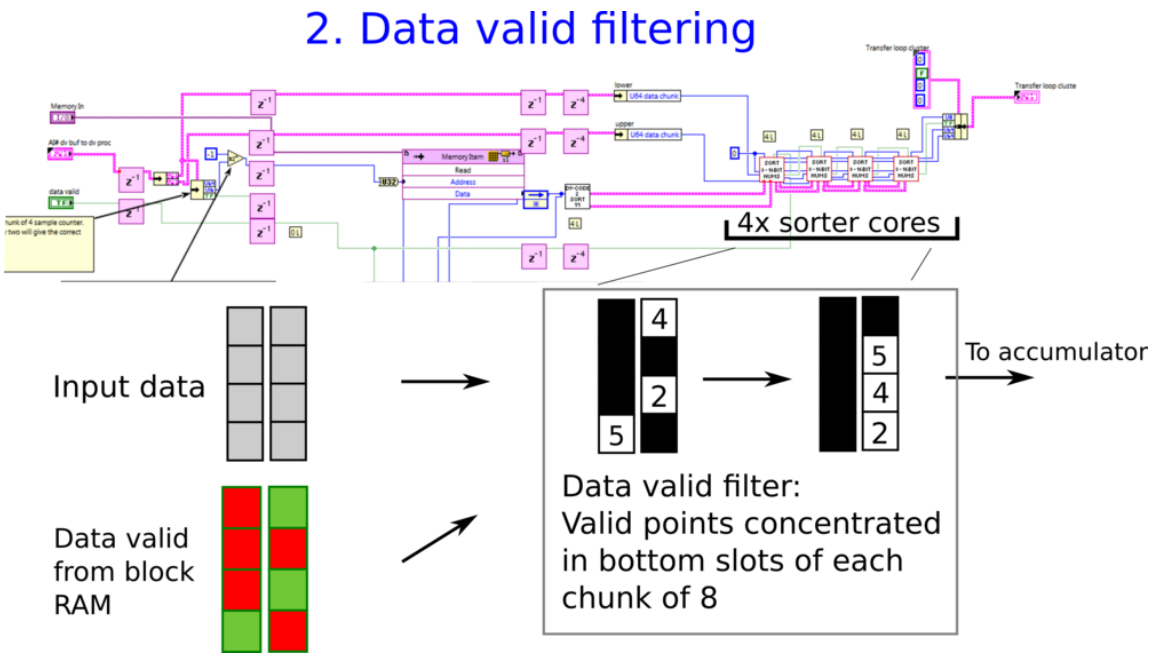

Figure 5.33: Overview of the data valid filtering step. A snippet of the LabVIEW block diagram is shown in the top portion of the figure, and a summary is shown in the bottom. The input data, along with a data counter (used to index block RAM for the data valid information), is fed into the first sorter core. The first sorter core is then connect serially to 3 more cores. The output chunk of 8 12-bit data points consists of valid points (if any) contiguously arranged in the lower slots of the chunk of 8 points (recall each slot is 12-bits because the original ADC data was 12-bits). The output chunk of 8 is then connected to an accumulator that packs data into chunks of 8 valid 12-bit points.

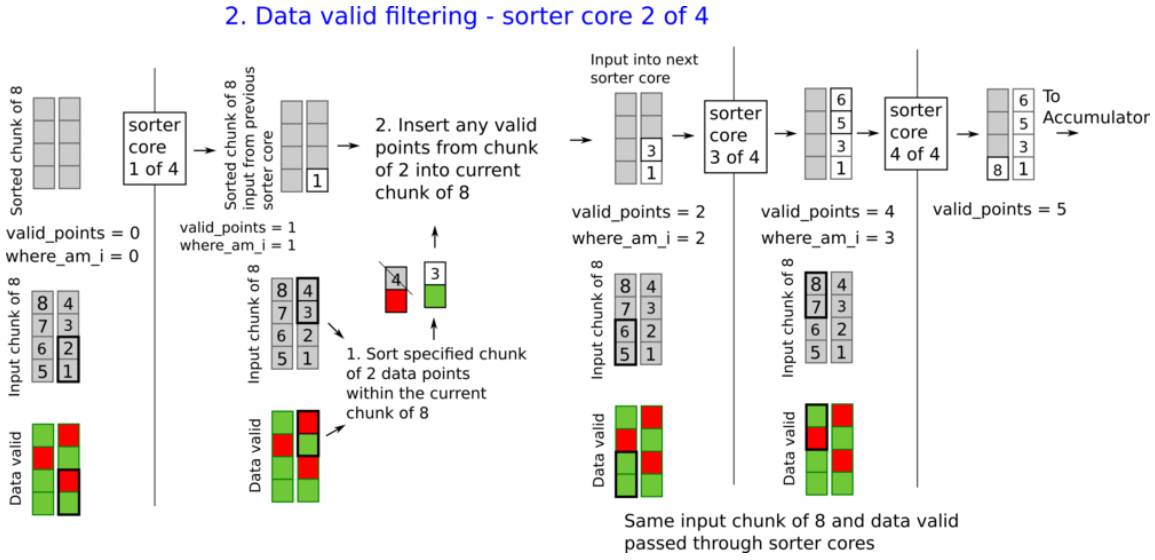

Figure 5.34: Diagram showing an example chunk of 8 data points being processed by the data valid filtering sorter core 2 (the other 3 cores are shown to the left and right). A total of 4 sorter cores are used to filter a chunk of 8 data points. Each sorter core sorts a pair of two data points for each chunk of 8, and inserts them into an output chunk of 8. The first core sorts the first 2 of 8, the second core operates on the second 2 of 8, etc. All 8 data points are sorted after passing through all 4 cores. The last sorter core sends its data (the chunk of 8 and the number of valid points between 0 and 8) to the accumulator.

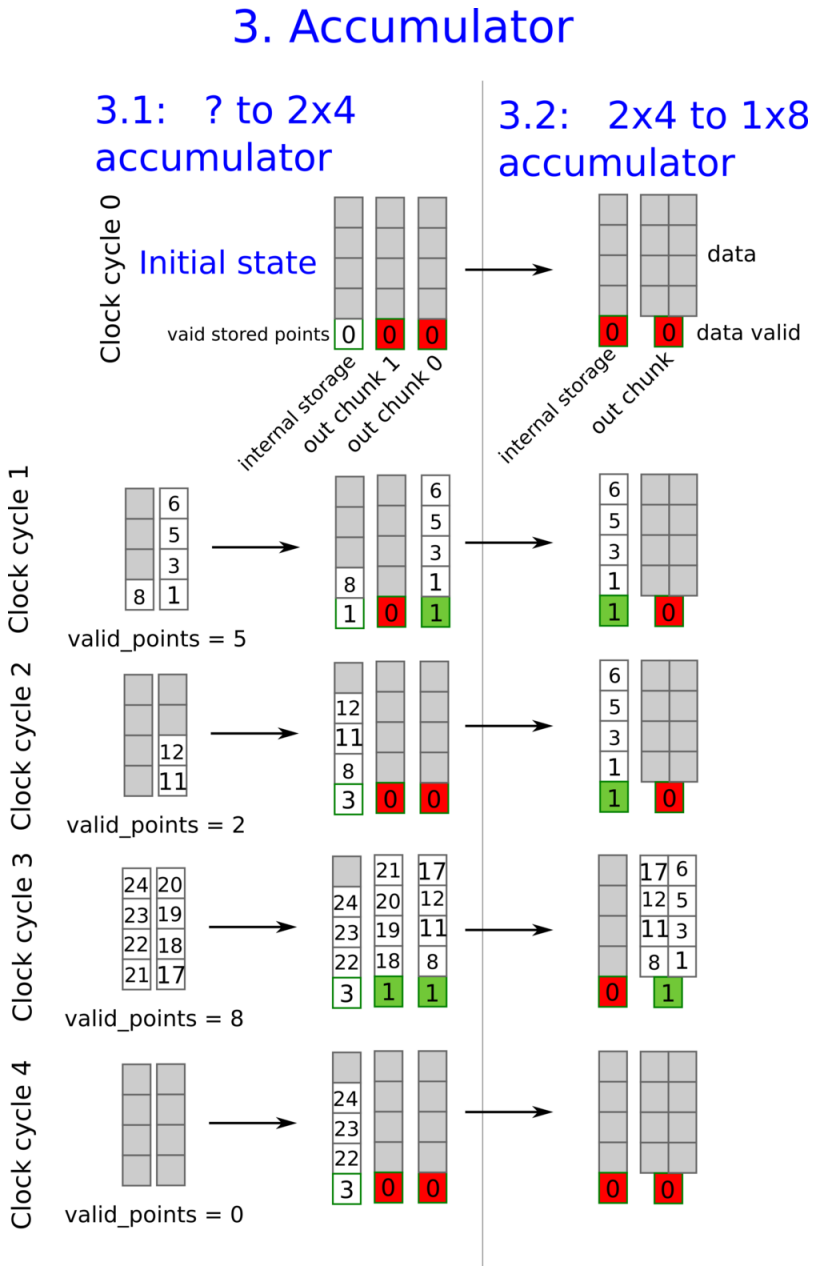

Figure 5.35: Diagram showing example behavior of the accumulator logic for one channel of data (figure 5.32 part 3). The accumulator logic is implemented in two parts, first a ? to 2x4 accumulator converts the arbitrary number of valid data points into 1 to 2 chunks of 4 valid data points. Then a 2x4 to 1x8 accumulator takes the 1 or 2 chunks of 4 valid data points and converts them into 1 chunk of 8 valid data points.

case the $X[k]$ can be rewritten as shown below, assuming N is divisible by 4.

$$\begin{aligned} X[k] = & \sum_{n=0}^{n=N/4-1} x[4n]e^{-j2\pi k(4n)/N} + \\ & \sum_{n=0}^{n=N/4-1} x[4n+1]e^{-j2\pi k(4n+1)/N} + \\ & \sum_{n=0}^{n=N/4-1} x[4n+2]e^{-j2\pi k(4n+2)/N} + \\ & \sum_{n=0}^{n=N/4-1} x[4n+3]e^{-j2\pi k(4n+3)/N}, \; k = \{0, 1, ..., N-1\} \end{aligned} \tag{5.49}$$

If the 4 streams are denoted as $x_1[n]$, $x_2[n]$, $x_3[n]$, and $x_4[n]$ for $n = \{0, 1, ..., N/4 - 1\}$ (where $x_1[n]$ denotes $x[4n]$, etc.) then the previous equation can be rewritten as follows.

$$\begin{aligned} X[k] = & \left(\sum_{n=0}^{n=N/4-1} x_1[n]e^{-j2\pi kn/(N/4)} \right) + \\ & \left(\sum_{n=0}^{n=N/4-1} x_2[n]e^{-j2\pi kn/(N/4)} \right) e^{-j2\pi k/N} + \\ & \left(\sum_{n=0}^{n=N/4-1} x_3[n]e^{-j2\pi kn/(N/4)} \right) e^{-j2\pi 2k/N} + \\ & \left(\sum_{n=0}^{n=N/4-1} x_4[n]e^{-j2\pi kn/(N/4)} \right) e^{-j2\pi 3k/N}, \; k = \{0, 1, ..., N-1\} \end{aligned} \tag{5.50}$$

The terms inside the parenthesis are the DFTs of the decimated data streams. We will denote the DFT of each stream of data as $X_i[k_p]$, where $i = 0, 1, 2, 3$ and $k_p = \{0, 1, ..., N/4 - 1\}$. The 4 separate DFT streams must be recombined in order to obtain $X[k]$. One way to accomplish this is by using the periodicity property of the DFT and to construct $X[k]$ in 4 segments simultaneously ($k = \{0, 1, ..., N/4 - 1\}$, $k = \{N/4, N/4 + 1, ...N/2 - 1\}$, etc.). However, for our specific application of SS OCT we only care about the first half of the

spectrum $k = \{0, 1, ..., N/2 - 1\}$, so we will not assemble the second half. The first and second quarters of the output spectrum can be computed as shown below.

$$\begin{aligned} k \in & \{0, 1, ...N/4 - 1\} \\ X[k] = & X_1[k_p] + X_2[k_p]e^{-j2\pi k_p/N} + \\ & X_3[k_p]e^{-j2\pi 2k_p/N} + X_4[k_p]e^{-j2\pi 3k_p/N} \end{aligned} \tag{5.51}$$

$$\begin{aligned} k \in & \{N/4, N/4 + 1, ...N/2 - 1\} \\ X[k] = & X_1[k_p] + X_2[k_p]e^{-j2\pi (k_p+N/4)/N} + \\ & X_3[k_p]e^{-j2\pi 2(k_p+N/4)/N} + X_4[k_p]e^{-j2\pi 3(k_p+N/4)/N} \\ = & X_1[k_p] + X_2[k_p]e^{-j2\pi k_p/N}(-j) + \\ & X_3[k_p]e^{-j2\pi 2k_p/N}(-1) + X_4[k_p]e^{-j2\pi 3k_p/N}(j) \end{aligned} \tag{5.52}$$

Assembling the first half of $X[k]$ is accomplished by multiplying the 4 stream DFTs by complex exponentials. Since both quarters are computed simultaneously, reordering logic is needed to concatenate the parallel 1/4 and 2/4 parts of the spectrum when this algorithm is implemented on an FPGA. This approach is detailed later in this section.

5.14.2.2 Implementation

Section 5.7 discussed an implementation of a 200 MS/s FFT at two points per clock cycle at a frequency of 100 MHz. The 2048 length FFT was achieved by calculating two 1024 length FFTs, and joining them to get the 1024 length single sided spectrum. The 400 MS/s FFT described in this section is an extension of the 200 MS/s FFT (actually, the 400 MS/s FFT was implemented first). The reordering FIFOs described here were not needed for the 200 MS/s FFT.

First, the SS OCT signal was digitized at a rate of 800 MS/s (4 points per clock cycle at 200 MHz), with 12-bit amplitude resolution and a fixed voltage range of ± 2 V by the

National Instruments NI 5772 digitizer and PXIe 7966 FPGA. The 800 MS/s data stream was reduced to 400 MS/s by discarding every 1 out of 2 data points. The data stream was then transferred to a 103.33 MHz clock domain for further processing.

The data stream was 4 points per 103.33 MHz clock after being transferred from the 200 MHz clock domain into the 103.33 MHz clock domain. Each chunk of four data points contained the digitized SS OCT signal at times N-3, N-2, N-1, and N. Pre-computed complex exponential terms ($e^{-j2\pi\alpha k_p/N},\ \alpha = \{1, 2, 3\}$) were uploaded to the FPGA from the host computer before the imaging session. Four instances of the Xilinx FFT 7.1 module were used to compute the FFT. The FFT modules were configured as described in section 5.7. Figure 5.36 provides an overview of the 400 MS/s FFT.

5.15 Conclusion and future work

The algorithms described in this work are pipelined in a low latency and high throughput fashion. However, relatively low latency (on the order of 1 to 2 sweeps of the OCT laser) is not necessarily a requirement. Others typically utilize higher latency approaches with equivalent throughput [33]. In such a scheme, several 1 point per clock cycle slower processing cores could be run in parallel instead of implementing a single multi-point per clock cycle approach as described in this chapter. This strategy is typically easier to scale as the core algorithm can be implemented in a simple one point per clock cycle fashion, and then placed multiple times on the FPGA to increase throughput (with extra logic to multiplex between the inputs and outputs of the cores) as shown in figure 5.37. However, this scale-by-replication approach may result in copying the use of some resources like block RAM. For example, the background subtraction processing step reads pre-stored background from block RAM. Block RAM typically cannot have an arbitrary number of read interfaces. Thus, additional independently running background subtraction cores would require their own block RAMs even though the contents would be identical. Ulti-

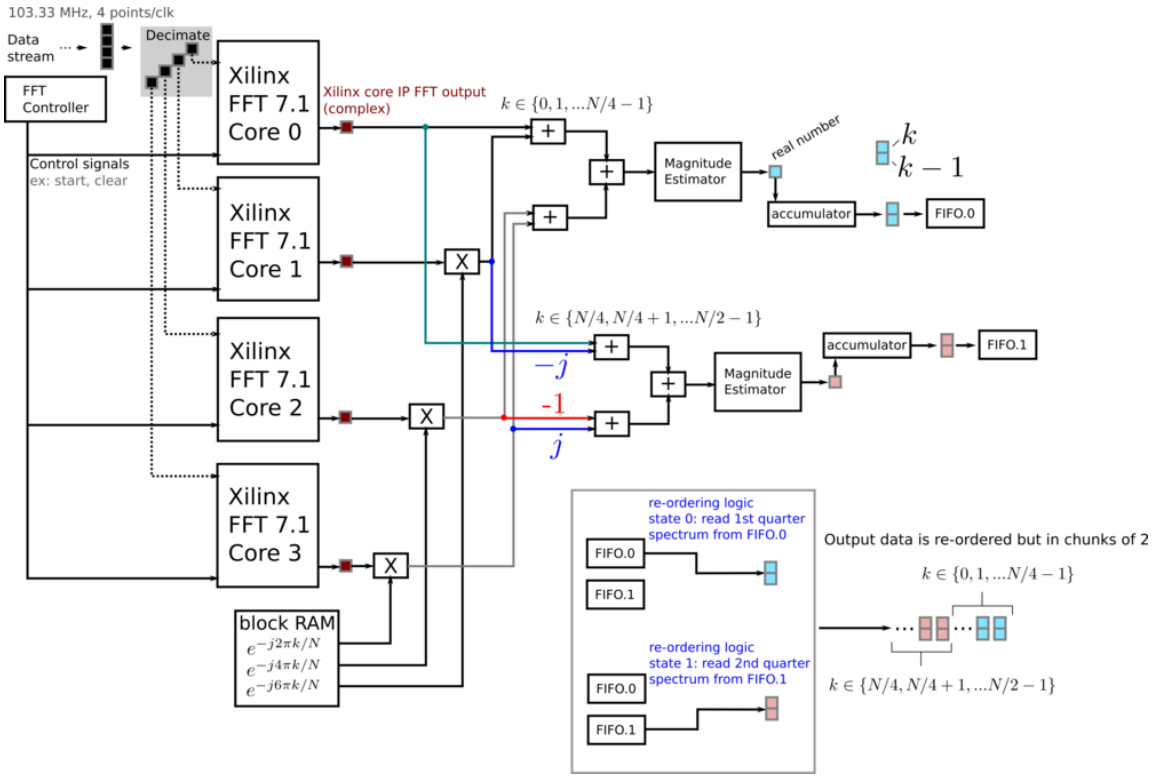

Figure 5.36: Diagram of the 4 decimated in time (DIT) FFT. The input data stream of 4 data points per clock cycle was input into 4 separate Xilinx FFT 7.1 modules (configured as described in section 5.7). The first half of the total spectrum was assembled from the 4 separate FFTs as outlined earlier in equations (5.51) and (5.52). Only one set of exponential factors was stored in block RAM, as the terms for assembling the second quarter of the output spectrum are identical to the first except for $-j$, -1, and j terms. Multiplying by $-j$, -1, and j were implemented as combinations of negation and switching the real and imaginary parts of the given complex number (indicated by putting $-j$, -1, and j next to the wires in the diagram). After each quarter spectrum was computed (i.e. $k = \{0, 1, ..., N/4 - 1\}$ and $k = \{N/4, N/4 + 1, ...N/2 - 1\}$), the magnitude estimator (section 5.8) was used to compute the magnitude of the complex numbers. Then an accumulator was used to convert each of the two 1 point per clock cycle streams into 2 point per 2 clock cycle streams before writing the result to a FIFO (FIFO.0 for the first quarter spectrum, FIFO.1 for the second quarter spectrum). Then the re-ordering logic (bottom right) read out of FIFO.0 until the entire first quarter spectrum was extracted (using an internal counter to keep count), and then from FIFO.1 until the entire second quarter spectrum was extracted. The resulting output contained the re-ordered half spectrum in chunks of 2 data points.

mately, the most efficient approach depends on the available resources on the FPGA and the available FPGA developers skill set. Since the current FPGA design described in this work has significant free block RAM, the scale-by-replication approach may provide a more optimal alternative.

Although the FPGA processing codes are well evaluated in terms of functionality and processing speed, signal quality has not been fully evaluated and compared with floating point methods on the host computer. Further effort will be needed to fully characterize signal quality in terms of quantities such as signal to noise ratio (SNR). Depending on the result, various parts of the processing pipeline may need to be modified to increase precision or to decrease precision (for example, 39-bits after the FFT may be excessive).

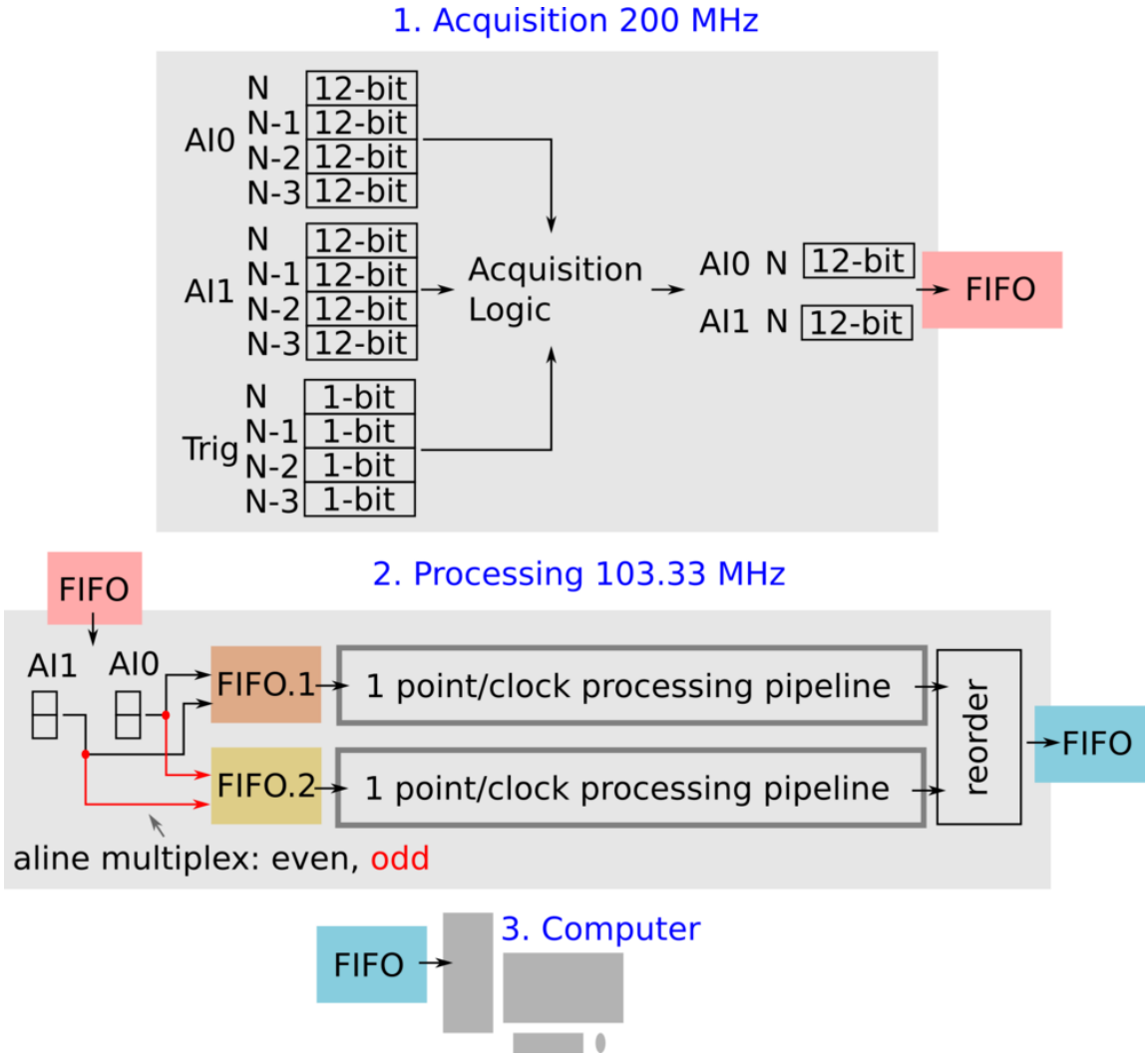

Figure 5.37: Possible alternative pipelining approach, similar to that described in [33]. Instead of processing one 2 point per clock cycle stream of data per analog input channel, two simultaneous 1 point per clock cycle per input channel streams could be processed. This would lessen the complexity of the linear interpolation and FFT processing stages, but increase block RAM usage and latency. In this processing scheme, FIFOs would be needed (FIFO.1 and FIFO.2) to buffer the input 2 point per clock cycle data streams. Even numbered sweeps would be input into FIFO.1, and odd numbered sweeps would be input to FIFO.2. Then each processing pipeline would extract and process data at 1 point per clock cycle. Additional logic would be needed to ensure that output data was transferred in a continuous way to the host computer (as the two processing pipelines would eventually begin to output valid data simultaneously). An approach similar to the re-ordering logic from the 400 MHz FFT (figure 5.36) could be used for this purpose. Alternatively, output data from even and odd numbered alines could be sent to the host computer in an interleaved fashion, and then the host computer could de-interleave the data (lessening the burden on the FPGA).

6. CONCLUSION AND FUTURE WORK

This chapter gives a condensed summary of the conclusions and future plans from each chapter. The reader is urged to refer to the relevant chapter(s) for more detailed information.

6.1 Conclusion

A novel real time FD-FLIM system was developed and its ability to classify CD68 regions in postmortem human coronary arteries was evaluated against histology as the gold standard, and an existing TD-FLIM system. The FD-FLIM system was then combined with an existing SS-OCT system to allow for simultaneous and automatically co-registered FD-FLIM and SS-OCT imaging. A relatively large database of 150 postmortem artery segments from 43 human subjects was imaged with the combined OCT and FLIM systems during the course of this work. Finally, a real time SS-OCT FPGA processing algorithm along with alternative higher speed implementations was detailed. Although processing of SS-OCT data on an FPGA is not new, the authors of other works rarely give sufficient information necessary for others to replicate their work or to compare performance against their work. The FD-FLIM and SS-OCT FPGA processing algorithms described here provide necessary details for other FPGA programmers to make use of this work either as a starting point for their own code or as a reference.

6.2 Future Work

The FD-FLIM system described here could be improved by reducing its cost and by improving its sensitivity. Sensitivity could be increased by optimizing the excitation waveform. Cost could be decreased by optimizing the FPGA DFT algorithm, by using lower cost non-National Instruments FPGAs that have more resources, and by using a lower cost

light source. However, using an FPGA that is not from National Instruments will require further FPGA programming. This is because the majority of the current FPGA acquisition and processing codes are written in National Instruments's proprietary LabVIEW FPGA language (2014 version). Currently, LabVIEW FPGA cannot be legally compiled into VHDL or Verilog for use on other vendor's FPGAs.

Further classification studies could be performed with the current OCT and FLIM database of 150 postmortem human coronary arteries. This work investigated the CD68 classification performance of the FD-FLIM and TD-FLIM systems. But the collagen and elastin content of coronary arteries is also used to characterize atherosclerotic plaques. The current MOVATS stained histology slides could be analyzed by expert readers and used as the gold standard for more classification analyses. The SS-OCT data could also be included in the classification analyses to potentially increase accuracy, sensitivity, and specificity.

REFERENCES

[1] J. R. Lakowicz, *Principles of fluorescence spectroscopy.* New York : Springer, [2006], 2006.

[2] L. Marcu, "Fluorescence lifetime techniques in medical applications," *Annals of biomedical engineering*, vol. 40, no. 2, pp. 304–331, 2012.

[3] J. Park, P. Pande, S. Shrestha, F. Clubb, B. E. Applegate, and J. A. Jo, "Biochemical characterization of atherosclerotic plaques by endogenous multispectral fluorescence lifetime imaging microscopy.," *Atherosclerosis*, vol. 220, pp. 394–401, 2012.

[4] W. Drexler and J. G. Fujimoto, *Optical coherence tomography: technology and applications*. Springer Science & Business Media, 2008.

[5] K. C. Schuermann and H. E. Grecco, "flatFLIM: enhancing the dynamic range of frequency domain FLIM," *Optics express*, vol. 20, no. 18, pp. 20730–20741, 2012.

[6] P. Vitta, I. Reklaitis, and A. Žukauskas, "Frequency-domain fluorometry in the presence of high in-phase background," *Measurement Science and Technology*, vol. 23, no. 3, p. 035502, 2012.

[7] B. Yuan, S. R. McClellan, B. F. Al-Mifgai, E. A. Growney, and O. A. Komolafe, "A cost-efficient frequency domain fluorescence lifetime measurement system," *American Journal of Physics*, vol. 78, no. 1, pp. 28–34, 2010.

[8] A. D. Elder, J. H. Frank, J. Swartling, X. Dai, and C. F. Kaminski, "Calibration of a wide-field frequency-domain fluorescence lifetime microscopy system using light emitting diodes as light sources," *Journal of Microscopy*, vol. 224, no. 2, pp. 166–180, 2006.

[9] M. Booth and T. Wilson, "Low-cost, frequency-domain, fluorescence lifetime confocal microscopy.," *Journal of Microscopy*, vol. 214, no. 1, pp. 36–42, 2004.

[10] Q. S. Hanley, V. Subramaniam, D. J. Arndt-Jovin, and T. M. Jovin, "Fluorescence lifetime imaging: multi-point calibration, minimum resolvable differences, and artifact suppression.," *Cytometry*, vol. 43, no. 4, pp. 248–260, 2001.

[11] J. C. Chan, E. D. Diebold, B. W. Buckley, S. Mao, N. Akbari, and B. Jalali, "Digitally synthesized beat frequency-multiplexed fluorescence lifetime spectroscopy," *Biomedical optics express*, vol. 5, no. 12, pp. 4428–4436, 2014.

[12] B. A. Feddersen, D. W. Piston, and E. Gratton, "Digital parallel acquisition in frequency domain fluorimetry.," *Review of Scientific Instruments*, vol. 60, no. 9, p. 2929, 1989.

[13] R. Liu, Z. Zhao, L. Zou, Q. Fang, L. Chen, A. Argento, and J. F. Lo, "Compact, non-invasive frequency domain lifetime differentiation of collagens and elastin," *Sensors and Actuators B: Chemical*, vol. 219, pp. 283–293, 2015.

[14] R. A. Colyer, C. Lee, and E. Gratton, "A novel fluorescence lifetime imaging system that optimizes photon efficiency.," *Microscopy Research And Technique*, vol. 71, no. 3, pp. 201–213, 2008.

[15] T. Iwata, T. Taga, and T. Mizuno, "Fpga-based photon-counting phase-modulation fluorometer and a brief comparison with that operated in a pulsed-excitation mode," *Optical Review*, vol. 25, no. 1, pp. 94–101, 2018.

[16] J. R. Lakowicz, "Principles of fluorescence spectroscopy.," 2006.

[17] R. A. Colyer, C. Lee, and E. Gratton, "A novel fluorescence lifetime imaging system that optimizes photon efficiency," *Microscopy research and technique*, vol. 71, no. 3, pp. 201–213, 2008.

[18] H. D. Vishwasrao, A. A. Heikal, K. A. Kasischke, and W. W. Webb, "Conformational dependence of intracellular nadh on metabolic state revealed by associated fluorescence anisotropy," *Journal of Biological Chemistry*, vol. 280, no. 26, pp. 25119–25126, 2005.

[19] J. Park, J. A. Jo, S. Shrestha, P. Pande, Q. Wan, and B. E. Applegate, "A dual-modality optical coherence tomography and fluorescence lifetime imaging microscopy system for simultaneous morphological and biochemical tissue characterization.," *Biomedical Optics Express*, vol. 1, no. 1, pp. 186–200, 2010.

[20] H. Lemmetyinen, N. V. Tkachenko, B. Valeur, J. ichi Hotta, M. Ameloot, N. P. Ernsting, T. Gustavsson, and N. Boens, "Time-resolved fluorescence methods (IUPAC Technical Report).," *Pure & Applied Chemistry*, vol. 86, no. 12, pp. 1969–1998, 2014.

[21] J. V. Morris, M. A. Mahaney, and J. R. Huber, "Fluorescence quantum yield determinations. 9, 10-diphenylanthracene as a reference standard in different solvents," *The Journal of Physical Chemistry*, vol. 80, no. 9, pp. 969–974, 1976.

[22] D. A. Barrow and B. R. Lentz, "The use of isochronal reference standards in phase and modulation fluorescence lifetime measurements," *Journal of biochemical and biophysical methods*, vol. 7, no. 3, pp. 217–234, 1983.

[23] J. A. Jo, J. Park, P. Pande, S. Shrestha, M. J. Serafino, J. J. R. de Jimenez, F. Clubb, B. Walton, L. M. Buja, J. E. Phipps, M. D. Feldman, J. Adame, and B. E. Applegate, "Simultaneous morphological and biochemical endogenous optical imaging of atherosclerosis," *European Heart Journal - Cardiovascular Imaging*, 2015.

[24] S. Shrestha, M. J. Serafino, J. Rico-Jimenez, J. Park, X. Chen, S. Zhaorigetu, B. L. Walton, J. A. Jo, and B. E. Applegate, "Multimodal optical coherence tomography and fluorescence lifetime imaging with interleaved excitation sources for simulta-

neous endogenous and exogenous fluorescence," *Biomedical Optics Express*, vol. 7, no. 9, pp. 3184–3197, 2016.

[25] K. Martinez and J. Cupitt, "Vips-a highly tuned image processing software architecture," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 2, pp. II–574, IEEE, 2005.

[26] H. Y. Lee, P. D. Raphael, J. Park, A. K. Ellerbee, B. E. Applegate, and J. S. Oghalai, "Noninvasive in vivo imaging reveals differences between tectorial membrane and basilar membrane traveling waves in the mouse cochlea," *Proceedings of the National Academy of Sciences*, p. 201500038, 2015.

[27] S. Lee, M. W. Lee, H. S. Cho, J. W. Song, H. S. Nam, D. J. Oh, K. Park, W.-Y. Oh, H. Yoo, and J. W. Kim, "Fully integrated high-speed intravascular optical coherence tomography/near-infrared fluorescence structural/molecular imaging in vivo using a clinically available near-infrared fluorescence–emitting indocyanine green to detect inflamed lipid-rich atheromata in coronary-sized vessels," *Circulation: Cardiovascular Interventions*, vol. 7, no. 4, pp. 560–569, 2014.

[28] H. S. Cho, S.-J. Jang, K. Kim, A. V. Dan-Chin-Yu, M. Shishkov, B. E. Bouma, and W.-Y. Oh, "High frame-rate intravascular optical frequency-domain imaging in vivo," *Biomed. Opt. Express*, vol. 5, pp. 223–232, Jan 2014.

[29] J. Bec, H. Xie, D. R. Yankelevich, F. Zhou, Y. Sun, N. Ghata, R. Aldredge, and L. Marcu, "Design, construction, and validation of a rotary multifunctional intravascular diagnostic catheter combining multispectral fluorescence lifetime imaging and intravascular ultrasound.," *Journal Of Biomedical Optics*, vol. 17, no. 10, p. 106012, 2012.

[30] M.-J. Bertrand, P. Lavoie-L'Allier, and J.-C. Tardif, "Near-infrared spectroscopy (nirs): A novel tool for intravascular coronary imaging," in *Developments in Near-*

Infrared Spectroscopy, InTech, 2017.

[31] S. Kim, P. D. Raphael, J. S. Oghalai, and B. E. Applegate, "High-speed spectral calibration by complex fir filter in phase-sensitive optical coherence tomography," *Biomed. Opt. Express*, vol. 7, pp. 1430–1444, Apr 2016.

[32] R. G. Lyons, *Understanding digital signal processing. 3rd ed.* Upper Saddle River, NJ : Prentice Hall, [2011], 2011.

[33] V. Bandi, J. Goette, M. Jacomet, T. von Niederhäusern, A. H. Bachmann, and M. Duelk, "Fpga-based real-time swept-source oct systems for b-scan live-streaming or volumetric imaging," in *Optical Coherence Tomography and Coherence Domain Optical Methods in Biomedicine XVII*, vol. 8571, p. 85712Z, International Society for Optics and Photonics, 2013.