A NEW TECHNIQUE FOR 3D MODELING OF WATER SURFACES USING A GEIGER MODE RECEIVER

A Thesis
Presented to
The Academic Faculty

by

Nicholas J. Guida

In Partial Fulfillment
of the Requirements for the Degree
Masters of Science in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
August 2015

A NEW TECHNIQUE FOR 3D MODELING OF WATER SURFACES USING A GEIGER MODE RECEIVER

Approved by:

Dr. Gisele Bennett, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Thomas Michaels
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Grady Tuell, Co-Advisor
Georgia Tech Research Institute – Electro Optical Systems Laboratory
Georgia Institute of Technology

Dr. Gregory Durgin
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Jennifer Michaels
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: June 3rd, 2015

ACKNOWLEDGEMENTS

I would first like to thank my advisor, Dr. Gisele Bennett, for serving as my thesis advisor and providing me the opportunity to work as a graduate research assistant at GTRI EOSL. I greatly appreciate the time and effort she has given me while at EOSL and in preparation for my thesis.

Next, I would like to thank my co-advisor, Dr. Grady Tuell, for bringing me onto his NOAA/Dewberry project within EOSL and providing invaluable knowledge and advice for my work and thesis. I am thankful for his patience and support in helping me better understand airborne lidar bathymetry.

I would like to thank Domenic Carr and Matthew O'Shaugnessy, colleagues of mine at EOSL, who I worked with closely during my time in the lab. They both provided many key components to the overall research effort and helped me immeasurably throughout the project. It was a pleasure to work with them both and know that any future project related research will be continued by their extremely capable skills.

I would like to thank Ryan James, Chris Valenta, and Robert Ortman, three colleagues of mine at EOSL, for providing help with electronics and setup of the laser pathfinder. Their work greatly aided in the deployment and utilization of all necessary hardware for this research.

I would like to thank Eric Brown, a colleague who specializes in mechanical engineering at EOSL, for creating the necessary scaffolding for the pathfinder, providing critical implementation components and details for testing, and allowing usage of the water tank in the Georgia Tech mechanical engineering building. This research and

experimentation would not have been possible without this great deal of effort and I am sincerely grateful for your time and energy.

I would like to thank all other EOSL colleagues who I have worked with in the past and present, many of whom helped me with specific problems related to different projects. The friendliness, care, patience, and intelligence exuded by everyone made my time at EOSL pleasant and educational. For nearly the past 2 years, I couldn't have asked for a better work environment or people.

I would like to thank all my friends and family who have helped motivate me throughout my time here at Georgia Tech. Despite myself being further away from home, their interaction and communication has never ceased and has kept me focused on achieving this degree.

Finally, I would like to thank my parents and brother who have supported, cared, and loved me for the past 23 years. Their endless support and love has allowed me to reach new heights as a person in every aspect of my life. I would not be the person I am today without their knowledge, wisdom, guidance, and care they have all provided me and am thankful every day for the integral role they all play in my life. I love them all very dearly and hope to continually make them proud.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

cm	Centimeter
kHz	kilohertz
l_w	Path length in water
l_A	Path length in air
m	Meter
ns	Nanosecond
Nd:YAG	Neodymium-doped Yttrium Aluminum Garnet
Nd:YVO4	Neodymium-doped Yttrium Orthvanadate
ps	Picosecond
s	Second
V_b	Breakdown voltage
ΔV	Bias Voltage
ϕ_A	Off-Nadir Angle

LIST OF ABBREVIATIONS

ALB	Airborne Lidar Bathymetry
APD	Avalanche PhotoDiode
CMOS	Complementary Metal-Oxide Semiconductor
CZMIL	Coastal Zone Mapping and Imaging Lidar
DCR	Dark Count Rate
DSM	Digital Surface Model
EOSL	Electro-Optical Systems Laboratory
FPA	Focal Plane Array
GmAPD	Geiger-Mode Avalanche PhotoDiode
GPU	Graphics Processing Unit
GTRI	Georgia Tech Research Institute
GUI	Graphical User Interface
LiDAR	Light Detection and Ranging
MAE	Mean Absolute Error
MLA	Micro Lense Array
MSE	Mean Squared Error
NED	Northing, Easting, Down
NIR	Near Infrared
NLM	Non-Local Means
NOAA	National Oceanographic Atmospheric Association
PCL	Point Cloud Library
PDE	Probability Detection Efficiency

PMT	PhotoMultiplier Tube
PRR	Pulse Rate Repetition
ROIC	Read Out Integrated Circuit
SPD	Symmetric Positive Definite

SUMMARY

This paper represents my work in supporting Georgia Tech Research Institute's (GTRI) Electro-Optical System Laboratory (EOSL) Dewberry/National Oceanographic and Atmospheric Association (NOAA) project. My contributions assist current EOSL research and development in efforts to improve accuracy and speed when computing seafloor coordinates from bathymetric Light Detection and Ranging (LiDAR) data.

The objective of my research was to construct a 3D digital surface model (DSM) in real-time of a sea surface using airborne LiDAR bathymetry (ALB) and a high resolution Geiger Mode Avalanche PhotoDiode (GmAPD) Near Infrared (NIR) receiver. These digital surface models would provide important sea surface topography information which could be used for additional processing calculations, specifically air-water interface refraction corrections in an ALB system for sea floor coordinate computations. Such a technique must be fast enough to allow for real time data processing of vast amounts of incoming information, yet reliable in its modeling to produce a precise representation of the sea surface. This thesis details the process of taking raw GmAPD data and creating a water DSM using a highly sensitive and spatially dense 32 x 32 array GmAPD NIR camera and NIR 1064nm Nd:YAG laser. The benefit of utilizing this DSM for an ALB system is discussed and shown numerically. DSM model accuracy was assessed through comparisons with flat target and calm water conditions with additional future opportunity for wavy surface reconstruction.

On a final note, this research had the secondary goal of 3D modeling the water in real time – as that is one significant aspect of EOSL's research program in bathymetric LiDAR. Real time processing provides the capability to detect errors and gaps in data immediately or ability to transmit only seafloor coordinates, rather than raw data, when deploying ALB systems on unmanned aerial vehicles (UAV). That being said, the technique for 3D modeling presented in this research does not inherently need to be done in real time. If done in post-processing, more sophisticated and computationally intensive algorithms might result in better accuracies.

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

In an Airborne Bathymetric LiDAR system, the accuracy of a ray traced sea floor coordinate depends upon on the refraction of the incident beam at the air-water interface [1][2]. While the equation is a simple calculation (Snell's Law), the angle of incidence to be refracted must first be calculated. This can be easily determined when the water surface is flat. However, a more complicated sea surface structure changes both the incidence and refracted angles [3]. Correcting for this uneven water surface becomes central to improving the accuracy of the ray traced computations [1][2]. This thesis attempts to establish a new technique for 3D modeling water surfaces using a mock ALB system that could provide additional useful information for such correction applications.

1.2 Origin and History

ALB is a technique used to survey and map coastal waters and littoral zones. Since the 1980's, ALB has been used extensively thanks to the pioneering work of Gordon [4], Guenther [3], and the advantages the technology has when compared to traditional methods. One of the uses of ALB is mapping sea floor coordinates and underwater object detection [5][6], which requires an accurate understanding and modeling of the air-water interface [7]. Light refraction of the incident beam at the water surface is a vital parameter to compensate for in order to adequately ray trace to the sea floor [1]. However, sea surface structure changes the entry angle of the beam, making it more difficult to reliably know the direction of the in-water (defined as the area between

the water surface and sea floor) pointing vector. Illustrated in Figure 1.1 is an example of the error in seafloor coordinates arising from beam steering when the entry angle differs from an assumed flat water surface by some angle α.

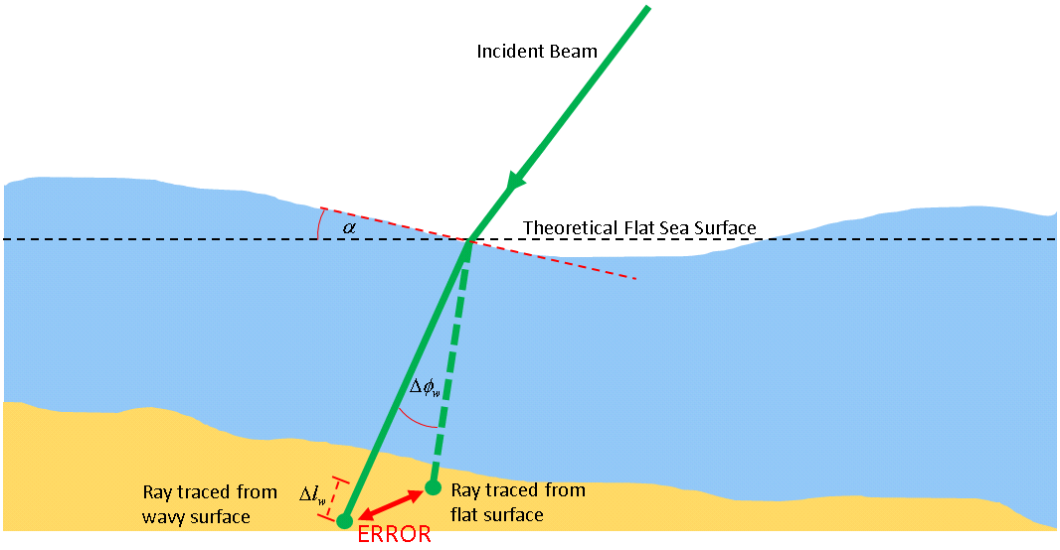

Figure 1.1: Error of sea floor mapping when not accounting for wavy water surface [1]

The magnitude of this error is a function of the in-water path length (l_w) and the angular discrepancy (α) between the assumed entry angle and the actual entry angle of the incident beam [1]. Figure 1.2 graphically shows this error for different angular discrepancies and in-water path lengths.

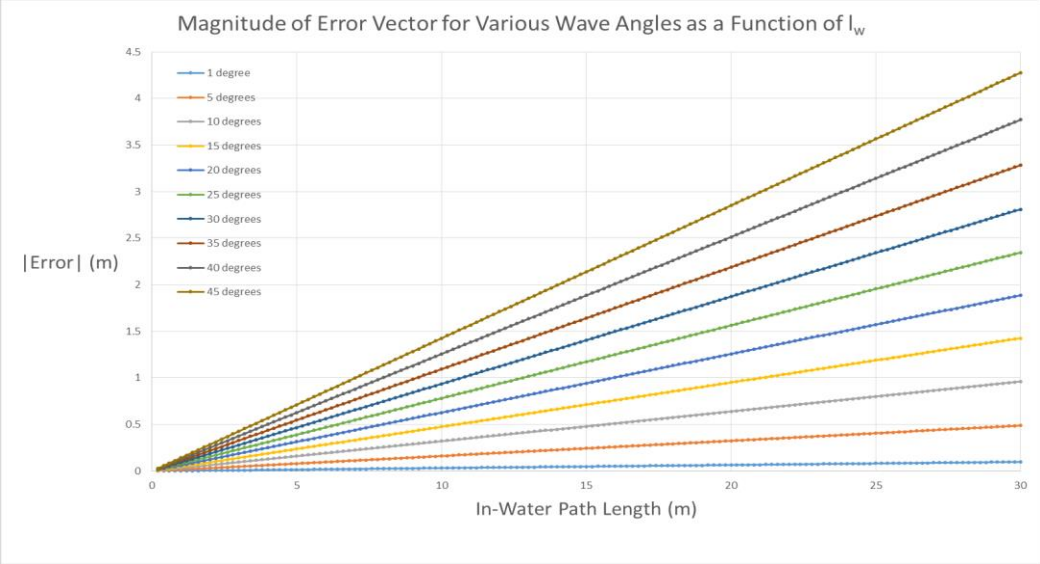

Figure 1.2: Magnitude of error vector at various wave angles as a function of water depth [1]

While this error is documented in the ALB community [6][7], exact measures taken for refraction corrections with respect to an uneven sea surface in commercial systems are seldom discussed in literature. One method that has been mentioned for correction is a calculation of the mean water level from a data collection followed by a subtraction of the calculated mean from the varying sea surface heights [6][8][9]. This produces the sea surface structure which can then be used for refraction correction. An example of a system implementing this method is the Hawkeye III system, manufactured by AHAB of Jonkoping, Sweden [10][11]. Figure 1.3 shows a sample LiDAR sea surface data from the Hawkeye III [10], where the x axis (northings distance) represents the space in front of and behind the aircraft. Their processing of this data using the previously described method produces 25cm variation on the sea floor when the wave height is 1m [10]. Moreover, they conclude that this error could be improved to near zero if each local data patch had their own mean instead of using a global mean [10].

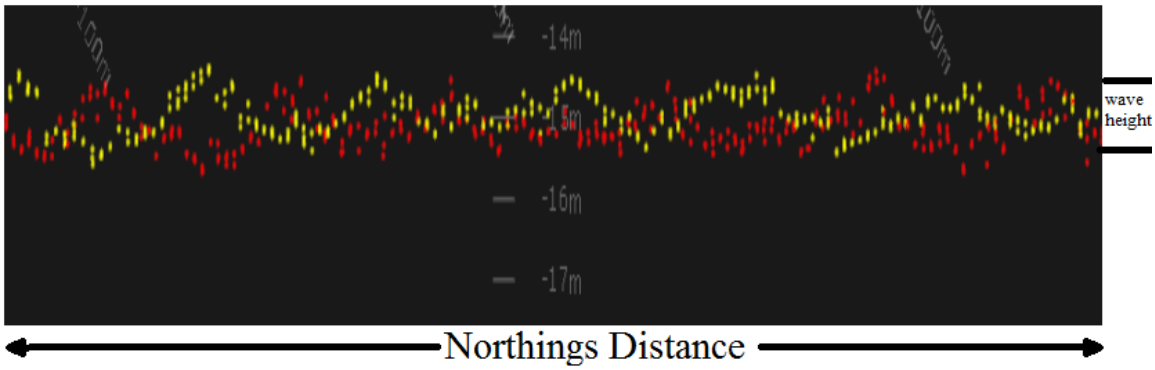

Figure 1.3: LiDAR data of water level from the Hawkeye III system [10]

One way to improve sea surface modeling would be through use of a high spatial resolution receiver. This increased spatial resolution would allow for densely packed points on the sea surface providing for the computation of planar surfaces to be used in the refraction correction. High-resolution 3D images of the sea surface can be produced with a LiDAR employing a Geiger-Mode Avalanche PhotoDiode (GmAPD) detector: a newly emerging technology that possesses many attractive qualities for ALB use. In Chapter 2, details about each of these advantageous characteristics is discussed. One of the most beneficial to the field of ALB and refraction correction is the several orders of magnitude improved spatial resolution [12][13][14] over that achieved with a LiDAR using a single-element detector [5][14]. This increased resolution is a key component in achieving the required point density for accurate DSM representations used in this research.

1.3 Overview of Research

Chapter 1 is an introduction to the current problem at hand and a brief introduction of the GmAPD receiver, which is the key piece of technology utilized in this

paper to create the water DSMs. Chapter 2 is a literature review on each of the components that would typically be required to effectively construct water DSMs: ALB system, GmAPD receiver, and data processing algorithms. A brief overview of ALB is introduced to familiarize nomenclature and system details related to current deployed systems. The overview will also outline the current spatial resolution limitations of ALB receivers. This leads into a discussion on the emergence of the GmAPD receiver with a more thorough look into its potential benefits towards water DSM construction and resulting refraction correction in an ALB system. Additional details regarding relevant denoising filters and data structures related to the support of the GmAPD data are also discussed. Chapter 2 concludes with the computation of the important water surface normal vectors from the DSM and how this ultimately impacts the accuracy of sea floor ray tracing calculations.

Chapter 3 elucidates on the methodology behind the experiment. It establishes the parameters of the setup and expected outputs of the experiment for analysis of the results. Chapter 3 also briefly details calibration tests that were required for the system and the methods used for determining the accuracy of the DSM surface normals and corrected refracted beam. Chapter 4 outlines the results of the experiments using statistical analysis between measured and actual DSMs. Chapter 5 is a summary of my contributions to the thesis.

CHAPTER 2

BACKGROUND

2.1 Airborne LiDAR Bathymetry (ALB) Overview

This section provides a brief overview of ALB, including a description of water surface detection and current receiver spatial limitations. This overview provides background needed for my research to establish 3D modeling of the water surface.

2.1.1 ALB Overview

ALB is a widely used tool for surveying and mapping coastal and littoral areas. This process typically involves an aircraft flying several hundred meters above the surface using a timed, pulsed, and scanned laser. These laser pulses hit a target or surface and reflect back into an optical receiver containing a photodetector, usually an APD or PMT operated in linear mode. A few different transmitter wavelengths can be used, but the most common are 532nm (green) and 1064nm (NIR) [6][8][15]. These wavelengths are generated using a Nd:YAG or Nd:YVO_4 1064nm NIR laser with the 532nm green signal being produced through frequency doubling of the 1064nm NIR signal. Returned energy from the two wavelengths is typically captured into separate receiver channels. Each channel possesses an important property related to ALB: the 1064nm channel has little water penetration and thus gives a return which is a good representation of the surface, while the 532nm channel has penetration into the water [6]. These properties are shown in Figure 2.1, which depicts how an array-based detector for the NIR might be used simultaneously with a single element detector for the green light [19].

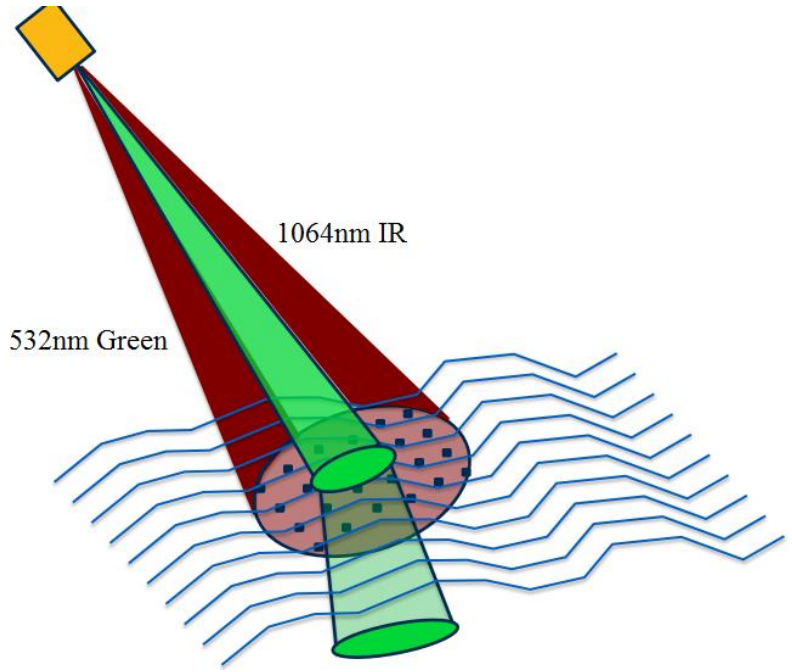

Figure 2.1: Illustration of green beam water surface penetration and NIR beam reflection [19]

Combining the data returned from these two separate wavelengths and taking the difference between return times provides the in-water path length, which can be converted to depth through simple geometry [6][15]. The precision of these bathymetric measurements varies according to many variables outside the scope of this paper [16], but average accuracy can expected to be within sub meter [5][11] [17]. The pulse repetition rates (PRR) of an ALB system for the bathymetric channels are in the 10-100 kHz range [5][11][17] [18], meaning that up to the PRR is the theoretical maximum amount of data that could be collected per second per available receiving channel.

In addition to these parameters, ALBs have an off-nadir angle (ϕ_A) and scanning angle (θ) used to sweep the beam and generate area coverage when in flight [6]. The off-nadir angle is the angle off from straight down, while the scanning angle is the angle off from straight forward. Both are shown in Figure 2.2.

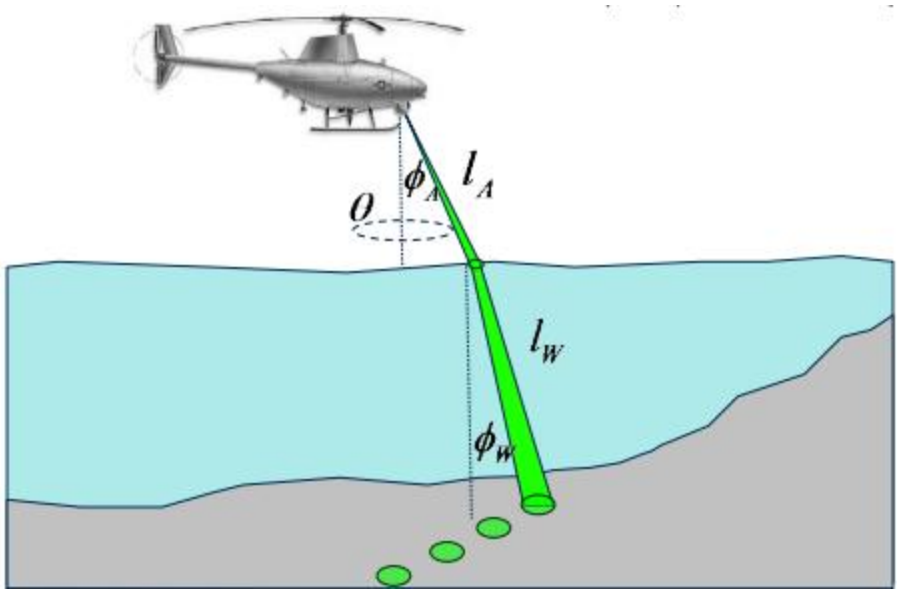

Figure 2.2: Off-nadir and scanner angles [19]

Off-nadir angles tend to be between 15-20 degrees which provide economical swath widths (roughly equal to one half of the aircraft altitude) while still providing good probability of separating the sea surface and water column returns in the system's signal processing software [3][6]. Larger angles tend to add various timing errors induced through the more complicated geometry [6].

Scanning patterns vary depending on the ALB system of the aircraft [3][6], although common patterns are circular, arc, and rectangular. A sample circular pattern is shown in Figures 2.2 and 2.3.

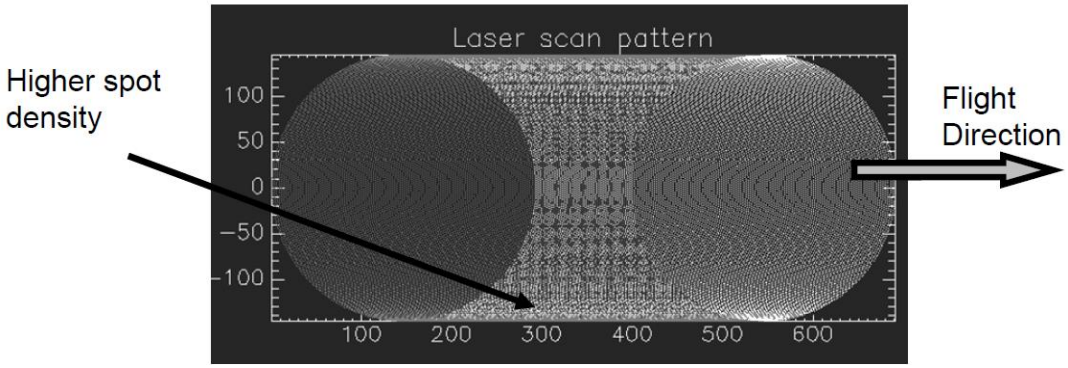

Figure 2.3: Scanning pattern and density of points for typical ALB system [20]

Scanning rates also vary and are dependent on the scan pattern of the ALB system [20]. The combination of the pulse repetition rate of the laser, the off-nadir angle, scanning rate, receiver density, scanning pattern, and forward velocity of the aircraft define the sampling density of the system [6] [16] (points per square meter). Most modern systems have a sampling density between 0.4-1.2 points/m^2 over water with higher densities over land [11][17].

The receiving components of most modern ALBs require the ability to detect and analyze a reflected return signal from the initial transmitted laser pulse. In order to detect the generally weak returns, systems use amplified PMTs or linear APDs to detect the signal and increase the signal to noise ratio (SNR) [3][6][15][16]. Once a return has been received, an analog to digital converter (digitizer) is used to convert the analog waveform to a digitized sample. Available sampling rate and bandwidth dictate accuracy and performance of the digitizer, and therefore having a capable and robust system generally requires having a sophisticated digitizer [21]. Figure 2.4 shows a typical return signal from the sea surface and floor.

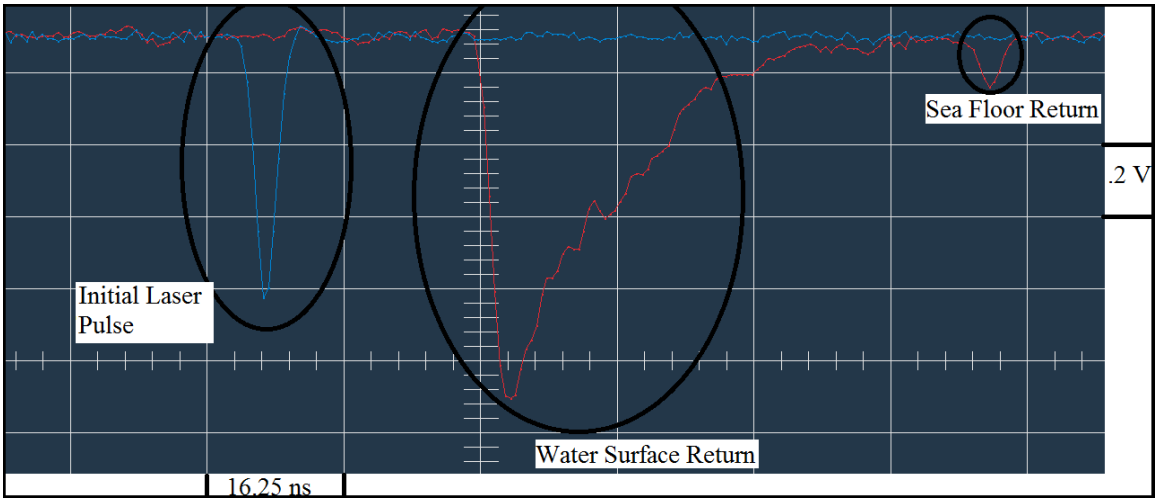

Figure 2.4: Sample waveform return from an ALB system

The current data collections span several hours [6][18], and terabytes of information are collected [18] over distances which span tens to hundreds of kilometers [6][7]. Once finished and landed, data is taken off the aircraft and post processed to improve its accuracy [6][16].

2.1.2 Water Surface Detection

A near infrared (NIR) wavelength is typically used to detect the water surface in an ALB system. At this wavelength (usually 1064nm), light is strongly attenuated by the water. Therefore, the returned energy originates at or very near the water's surface [6]. Detection of NIR is similar to green, in that it involves use of linear APDs that are sensitive to NIR wavelength [3][6][15][16]. In order to obtain a return signal, there must be some minor structure on the water surface in order to generate facets perpendicular to the NIR beam [3][6][8]. Often produced by wind, these waves cause a small part of the NIR beam to reflect back to the receiver for detection. Floating particulates on the surface and water column serve a similar purpose, although these may negatively affect water

depth penetration from the green beam channel [3][6][8]. These requirements cause issues in completely calm and/or clear water with no particulates or facets on the surface [3][6][8]. In these circumstances, the energy reflected from the sea surface may be so small that the surface is not detected, making it impossible to accurately separate ranges measured by the green beam into their in-air and in-water components. Moreover, particulates that exist between the water and air (e.g fog) can cause inaccurate depth measurements due to early returns or significant beam spreading [6]. Rectifying these problems entails imaging other nearby areas or waiting for appropriate environmental conditions to appear [6].

2.1.3 ALB Receiver Sample Density Limitations

A limitation of some current ALB receivers is the lack of high spatial sampling density. Sample density is a function of many inter-related factors, such as scanner rate, laser pulse repetition rate, scanning pattern, environmental conditions, the number of transmitted beams, and the number of detectors [6][16]. While both scanner rate and pulse repetition rate can be increased, they come with tradeoffs. Increasing the former results in increased wear and tear on the scanner and reduced lifetime expectancy, whereas increasing the latter results in lower energy per pulse if the average power of the laser is constant [20]. Use of a circular scanning pattern yields high spatial resolution and certain advantages in calibration, but introduces other complexities. A scanning pattern or reduced off-nadir angle that increases area overlap could be used for improvement, but this comes at the cost of longer flight times to survey the area, increasing cost. Environmental factors are outside the control of an ALB system, and water surfaces

inherently have more limited reflectivity than topographic areas, limiting the amount of possible returns.

The final contributing factor is the number of detectors used in the receiver, a design factor which is more costly to implement in analog receivers due to requiring extra hardware (digitizers, etc) for each of the receiver channels and higher overall complexity for the system. CZMIL, for example, had a state of the art fiber-optic coupled receiver comprised of nine detectors [20] (seven shallow green, one deep green, one IR). Shown in Figure 2.5 is CZMIL's segmented detector for the green beam.

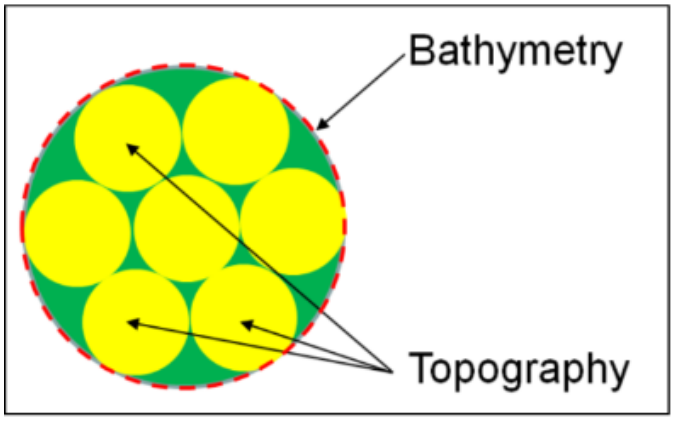

Figure 2.5: Czmil Imaging LiDAR with seven distinct green channels [20]

The logistics behind adding all the extra processing, power, bulk, cost, and cooling made having a large number of independent receivers unfeasible. However, with the advent of GmAPD arrays, the density of these points could be improved by implementing a sea-surface detection receiver based on a GmAPD camera. This additional density could theoretically improve the accuracy of the water DSM due to a larger number of representative surface points. From this, a ray tracing model utilizing a more precise water surface refraction correction can be implemented to achieve more

accurate sea floor coordinate computations. For this paper, we focus solely on the water DSM construction while analytically showing the potential accuracy improvement of refraction correction techniques when exploiting the added spatial resolution of the GmAPD receiver.

2.2 Emergence of Geiger-Mode Avalanche PhotoDiode (GmAPD) Receivers

While past and current topographic LiDAR and ALB systems use APDs or PMTs for their receivers, a new technology called GmAPD has recently been used in receiver designs for topographic LiDARs. The GmAPD is a modified form of a linear APD having slightly different voltage characteristics but significantly higher receiver sensitivity [12]. While a linear APD operates below the breakdown voltage of the diode, GmAPD operates above the breakdown voltage by some bias ΔV [12]. This bias voltage makes a GmAPD receiver sensitive enough for a single photon to cause a reaction producing detectable currents [12][22]. For a typical topographic LiDAR and ALB system, this incredible sensitivity allows for higher flying altitudes [22][23][24] (increases coverage due to higher swath widths [6] and use of higher laser pulse frequencies, but is only applicable for topographic LiDAR since ALB requires measurement of green light), much lower laser energies [12][22][25][23][26], and reduced hardware requirements which lessen the overall weight and cost of the system [12][22][23][14]. Moreover, in NIR wavelengths, GmAPD's high sensitivity offers a higher probability of detection efficiency (PDE) than the PMTs and APDs commonly used in ALB [27]. GmAPDs are currently manufactured to operate in two useful wavelength ranges, including 1064nm (NIR) and ~1500nm [28][29] (short wavelength infrared, used heavily in optical communications). Possibly the biggest advantage,

however, is the massive improvement in spatial resolution over current APDs and PMTs without having to sacrifice, size, weight, or power [14][30]. This increase in resolution is achieved by packing many GmAPD into a focal plane array (FPA) which does not require additional detection hardware. While still a relatively nascent technology, GmAPD looks to contain a lot of promise in the field of ALB.

2.2.1 Origin and History of GmAPD Receivers

The GmAPD receivers were developed and first described in the late 1990's by researchers at MIT's Lincoln Laboratory, with the following seminal papers being released in the early 2000's [12][31]. Through pairing of traditional Geiger-Mode Avalanche Photodiodes and CMOS timing circuitry, a GmAPD receiver was created allowing for single photon detection at sub nanosecond scales [26]. These receivers have been used and tested in 3D imaging topographic LiDAR systems at Lincoln Laboratory [13], although commercial availability was restricted until the late 2000's when Princeton Lightwave Inc (PLI) and Spectrolab (Boeing) began manufacturing commercial products. The most common resolution currently for a receiver is a 32x32 pixel array [25], although higher density implementations are also being sold [26].

2.2.2 GmAPD Process

A traditional PhotoDiode is a reverse biased device which converts light into current [12]. This reverse bias voltage creates an electric field in the p-n junction of the PhotoDiode, causing electrons to be confined to the n side and holes to the p side [12]. Incoming absorbed photons of sufficient energy create electron-hole pairs which drift to their respective sides, resulting in the flow of current. Figure 2.6 depicts this process.

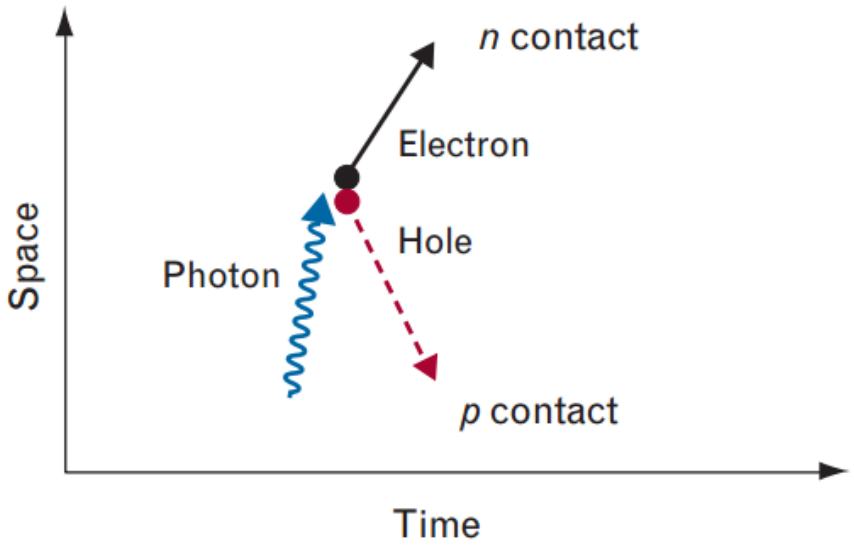

Figure 2.6: Electron-hole pair formed by incident photon [12]

Two common varieties of the PhotoDiode are the GmAPD and linear APD. Both operate at a significantly higher reverse bias voltage than a typical PhotoDiode, creating a stronger electric field in the device. This bolstered electric field can provide the generated electron-hole pair with additional energy, causing each electron and hole to accelerate and impact with the crystal lattice structure. This collision forms additional electron-hole pairs [12] that can undergo an identical collision process to produce more pairs. This activity is known as impact ionization and is what ultimately leads to the 'avalanche' creation of electron-hole pairs.

The key difference between GmAPDs and linear APDs is the amount of reverse bias voltage applied. Linear APDs operate with a reverse bias voltage below breakdown V_b, meaning that the avalanche will eventually die out [12] naturally and allows for linear APDs to act as gain amplifiers for an incoming signal. On the other hand, GmAPDs operate at a reverse voltage above V_b by some additional bias voltage ΔV [12]. While

armed above V_b, electron-hole pairs are created faster than they can be destroyed, producing an exponentially increasing amount of current. This can be seen in Figure 2.7.

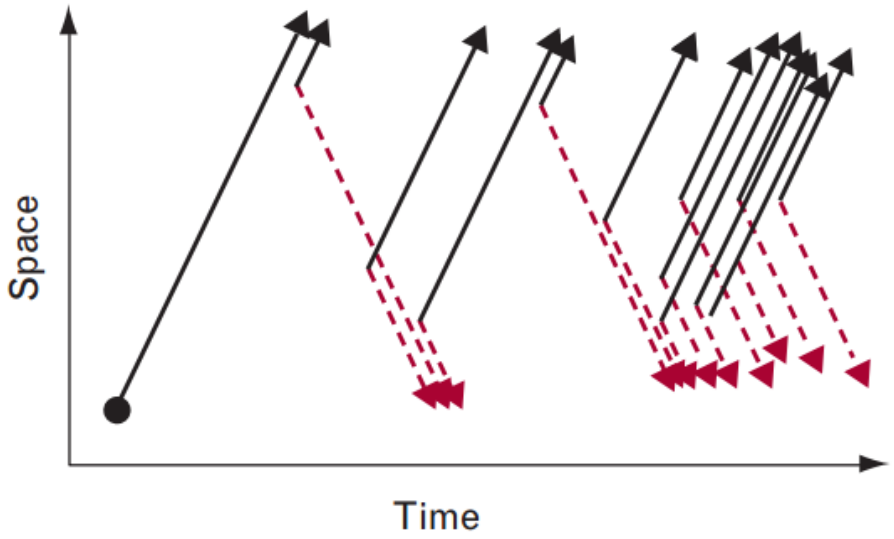

Figure 2.7: Geiger mode APD avalanche reaction forming electron-hole pairs faster than can be collected [12]

This rapidly increasing current means additional circuitry needs to be in place that can accurately and quickly detect the avalanche, record it, and then quench it [12]. This is achieved through threshold detection, a CMOS timing circuit, and a quenching circuit.

Threshold detection is a simple process, with a comparator and a preset threshold used to provide a purely digital output [25][29][32] of when the current exceeds the threshold. This output gets recorded by a CMOS timing circuit which stops a counter that was originally started when the GmAPD was first armed [29][33]. The quenching circuits come in two varieties: passive and active [12][29][30]. Passive quenching circuits use something akin to a series resistor with the diode to increase the voltage drop as current rises, leading to less voltage for the high electric field. This drop in voltage in the high

electric field continues until an eventual steady state condition is reached at the breakdown voltage V_b [12][29]. The problem with this passive quenching method is the long dead time that exists when attempting to rearm the GmAPD [12][29]. This long dead time is a result of the RC time constant of the quenching resistor and device capacitance and leads to a slower overall rearm time (and thus theoretical max frame rates) of the camera. On the other hand, active quenching circuits typically involve use of a shunting switch which immediately quenches the avalanche and restores the voltage to V_b, allowing for quick rearming [12][25][34]. This fast quench and rearming time is crucial to attain high frame rates, as one of the biggest drawbacks for a GmAPD is its inability to detect additional photons until it has been quenched and rearmed [25][29]. Figure 2.8 illustrates the I-V characteristics of the GmAPD and how the quenching and thresholding detection works.

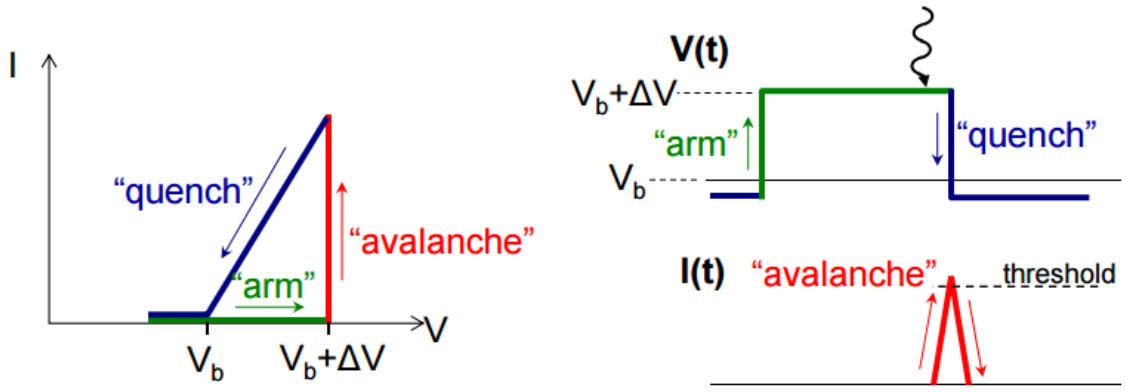

Figure 2.8: I-V Characteristics of a GmAPD and threshold/quenching properties [35]

Finally, the concept of gain is absent in GmAPDs, as an avalanche is either triggered or it isn't; a binary situation [12]. This process has the benefit of making detection and readout an inherently noiseless operation [25][30][33] and completely

eliminates the need for added digitizers and processing hardware that are required for PMTs or APDs operated in the linear mode. This is the main reason why it is much simpler to create and utilize an array of these GmAPDs in a small, low power, and low weight package.

2.2.3 GmAPD Detection and Noise Characteristics

GmAPDs are characterized by several parameters which help illustrate the overall performance of the receiver. These parameters are Photon Detection Efficiency (PDE), Dark Count Rate (DCR), and Timing Jitter (TJ), all of which tend to be interrelated and require additional hardware considerations for suppression or augmentation [23][35]. Due to the manufacturing process of a GmAPD FPA, these performance parameters are per pixel dependent [25], with pixels in the center having higher PDEs and associated DCRs while edge pixels have lower PDEs and DCRs [23]. This non uniformity can be a nuisance since imaging behavior will vary depending on whether the object is directly in the middle of the scene or not. Furthermore, while GmAPDs offer PDE advantages over APDs and PMTs at NIR wavelengths [27], it can come at the expense of higher DCR.

2.2.3.1 Photon Detection Efficiency

PDE is a straightforward metric that is equal to the probability that an incident photon of sufficient energy (around 1.03 eV for 1064nm receivers [25]) triggers an avalanche [12]. A perfect solution would have a unity PDE, yet PDE values for modern GmAPDs range between 20-50% [25][32]. PDE can be increased by increasing the reverse bias voltage above breakdown, essentially increasing the electron-hole pairs generated when an incident photon is present. However, this comes with the negative effect of increasing sensitivity to DCR, or false positives, which have a variety of causes

[23][32][30]. A balancing act is generally required for finding the best tradeoff between PDE and DCR, although environmental variables can sometimes warrant an increase in PDE at the expense of DCR. For example, imaging with a bright solar background means that PDE can increase up to the point where DCR is equal to the background noise [32]. In contrast, nighttime imaging benefits from lower DCR and hence lower PDE [32]. Figure 2.9 shows the DCR of pixels in a GmAPD array as a function of that pixel's PDE. Note the tradeoff that increasing PDE has with added DCR.

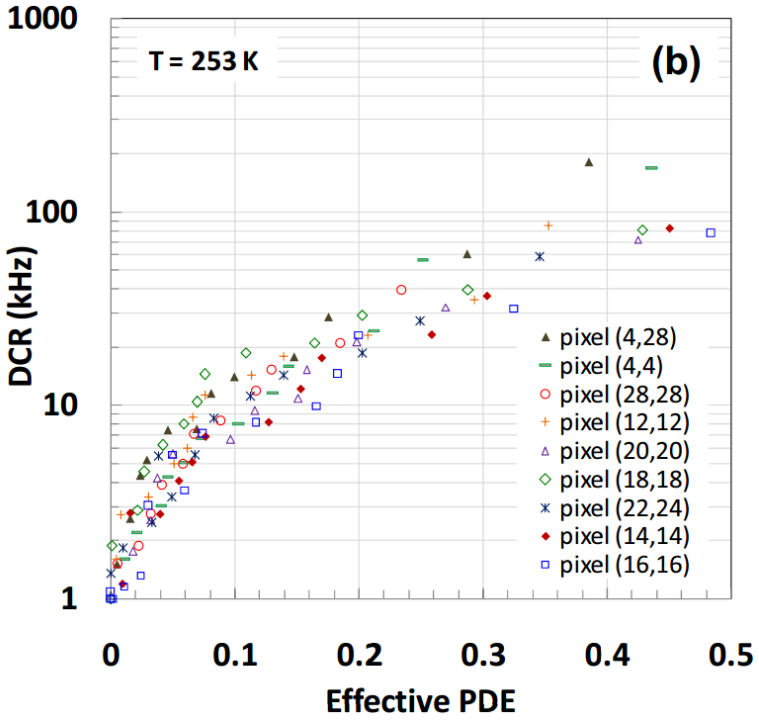

Figure 2.9: PDE vs DCR increasing relationship and pixel non uniformity at 253K temperature in a typical PLI GmAPD receiver [25]

2.2.3.2 Dark Count Rate

DCR is the measure of triggered avalanches detected when no incident photon was present [12][25][23]. This is a false positive count that is effectively the noise in a

GmAPD receiver that can generally be modeled as a Poisson process. It must be reduced to as little as possible without having to make significant PDE or range gate (the period of time where ΔV is applied, called frame or frame period hereafter) sacrifices. DCR can be further broken down into several different components.

2.2.3.2.1 Thermal Generation

Excess heat in the circuitry and wafer design defects can cause unwanted thermally generated dark counts in the GmAPD [23][29][35]. Reducing this heat involves having a refined manufacturing process to minimize defects and reducing the temperature of the circuitry to as little as possible, typically 250K through use of a thermal electric cooler (TEC) [25]. Thermal generation is the main contributor to DCR at ambient temperatures [35], and therefore cooling the system is an extremely important step to improve the overall DCR of the receiver.

2.2.3.2.2 Crosstalk

Crosstalk represents the chance that an avalanche occurring in one pixel causes an avalanche in a neighboring pixel despite an incident photon not being present in the latter pixel [23][32][34]. This probability stems from hot carrier luminescence at a rate of one photon per 10^5-10^6 carriers that flow through the avalanche [32][34]. Steps to reduce this phenomena include etching trenches in between pixels to reduce direct line of sight emission probabilities [32] and lowering the reverse bias voltage to reduce the number of electron-hole pairs being generated [12]. However, crosstalk is generally low in modern day GmAPD receivers, with the probability that a cross talk event takes place at a nearest neighbor pixel being less than 1.6% [34].

2.2.3.2.3 Band-to-Band and Trap-Assisted Tunneling

Trap-Assisted Tunneling (TAT) and Band-to-Band Tunneling (BBT) are examples of quantum tunneling that happen inside the circuitry and are the dominant forms of DCR at low temperatures (<220K) [35][28][36][34][29]. TAT is tunneling based on defects within the material and is much more likely to occur than BBT, which is just ordinary quantum tunneling between the absorption layer and multiplication layer [35] [28]. These tunneling effects can be reduced through creation of a gap between the photon absorption layer and the multiplication avalanche layer [25][30]. This gap reduces the electric field in the absorption layer to prevent dark carriers from tunneling, while keeping the electric field high in the multiplication layer for optimal avalanche conditions [25][36][30]. Further improvements in the manufacturing process have helped lower rates of tunneling effects through reduction of defects in the material [36][29].

2.2.3.2.4 Afterpulsing

Afterpulsing occurs when charge carriers get trapped in an atomic defect site in the multiplication region of a GmAPD during an avalanche [37][38][29]. At some time later, these carriers become free and can cause another avalanche without an incident photon being detected. This high correlation with previous avalanche trap sites makes afterpulsing highly predictable [23]. One way to correct this involves implementing a 'hold-off' time after an avalanche has triggered to allow for trapped carriers to become free and drift away from the multiplication region [37][38]. However, this incurs a longer

dead period where incident photons cannot be detected and therefore reduced frame rates. Other methods involve reducing the amount of charge trapped per avalanche by either having very short frame periods or implementing faster quenching circuitry to stifle the avalanche more quickly [37][29]. Despite this issue, GmAPD detectors manufactured by PLI are still capable of attaining ~200 kHz frame rates through a combination of longer hold off periods (which double as a frame readout period) and fast active quenching circuitry [25][32]][23]. This should be adequate for virtually all current ALB systems given current PRR of between 10 kHz – 100 kHz [5][11][17] [18].

2.2.3.3 Timing Jitter

Timing Jitter (TJ) is the amalgamation of timing delays which take place in detection timing [12][32][23]. These timing delays are stochastic processes which are affected by the GmAPD receiver design. The first of such delays is the randomness in photon detection, where the photon could be detected at the leading, middle, or trailing edge of the pulse [12][32]. Second, there is a finite period of time where the photon must drift from the absorption layer to the multiplication layer where the avalanche can begin. This is largely dependent on the design of the GmAPD and other physical properties [12]. A final source of timing jitter is the avalanche build up time, with faster build ups being detected sooner than slower build ups [12]. Current implementations are capable of keeping the total jitter time to be <200 picoseconds [38][23] through sophisticated circuitry and sufficient ΔV. This is an important parameter to minimize when attempting to increase frame rates and decrease frame length of the receiver.

2.2.4 NIR GmAPD Receiver Design

The favorable physical characteristics of an individual GmAPD are what permit for the creation of a high resolution GmAPD receiver. Organizing these GmAPDs into an array structure and bonding it with other hardware and circuitry creates a receiver which has a spatial resolution many times higher than traditional NIR receivers.

The physical design behind the GmAPD receiver is an important characteristic to understand in order to fully grasp how an incident photon ultimately generates an integer time of flight (TOF) value. Note: Design details here are based off of PLI's cameras which may differ from other manufacturer cameras.

2.2.4.1 GmAPD Schematic

GmAPDs are composed of several different layers which are all important for proper detection of a photon. For NIR wavelengths, a quaternary InGaAsP absorber is used for optimal 1064nm incident photon absorption [25][34]. The creation of an electron-hole pair in this absorption layer causes impact ionization in the separate secondary InP layer called the multiplication region. It is here where the avalanche takes place, and a macroscopic pulse is generated which will need to be detected through additional thresholding circuitry. These GmAPDs are packed onto a PhotoDiode Array (PDA, referred to as Focal Plane array (FPA) hereafter) with an active region diameter of 34 micrometers and pixel pitch of 100 micrometers [25]. Figure 2.10 shows a cross section of the FPA substrate.

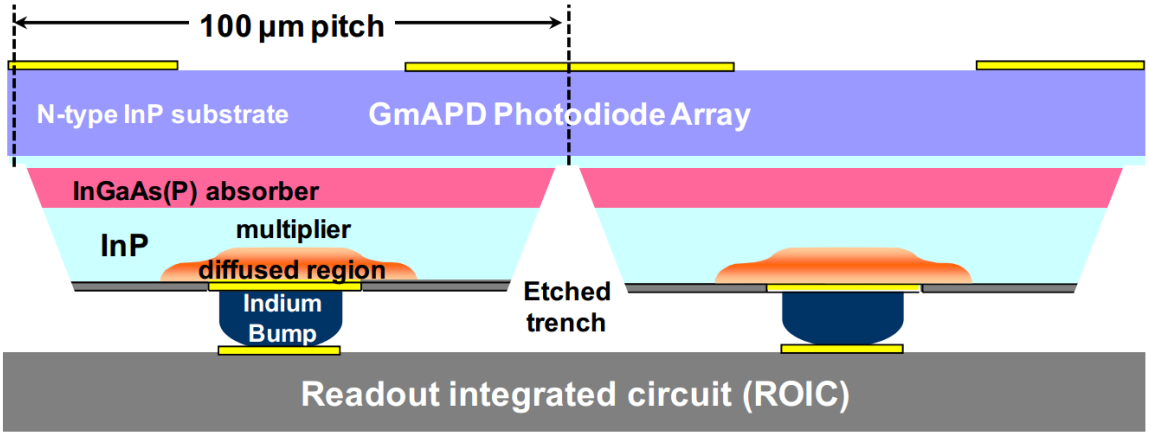

Figure 2.10: Cross section of GmAPD FPA substrate [25]

2.2.4.2 CMOS Readout Integrated Circuit

Converting the recorded photon avalanche period to a real binary number involves use of the CMOS Readout Integrated Circuit (ROIC) [25][33]. When the circuit becomes armed, the voltage is raised by some ΔV above V_b and counters begin for each pixel. A thresholding circuit is included on a per pixel basis for detection of an avalanche. When an avalanche is detected, the active quenching circuit removes the excess ΔV and the counter is stopped. After the designated frame time, all non-avalanched pixels record a terminal count value and a frame readout of all pixel values is initiated on a per row basis [25][33]. The vast majority of current GmAPD receivers operate in what's called "Range-Gating" mode, wherein each pixel can only fire once per frame period. Because of this limitation, intensity images must be created through multiple frame data. While another mode exists that allows for each pixel to reset after an avalanche event without waiting for the entire frame to reset (called 'Free Running Mode') [33], it is still undergoing further research and isn't available on our PLI camera.

2.2.4.3 Microlens Array

The Microlens Array (MLA) is placed on the front of the receiver to improve the optical fill ratio of the GmAPD array. Since the ratio of the actual active diameter region of the GmAPD to the total pixel region is ~ 9% (34 micrometer diameter active region and 100 micrometer pixel pitch), the MLA is required to help ensure incident photons are steered towards the active region. This addition helps improve the optical fill ratio to ~75% [25][32] and consequently increases the chance of an incident photon being absorbed.

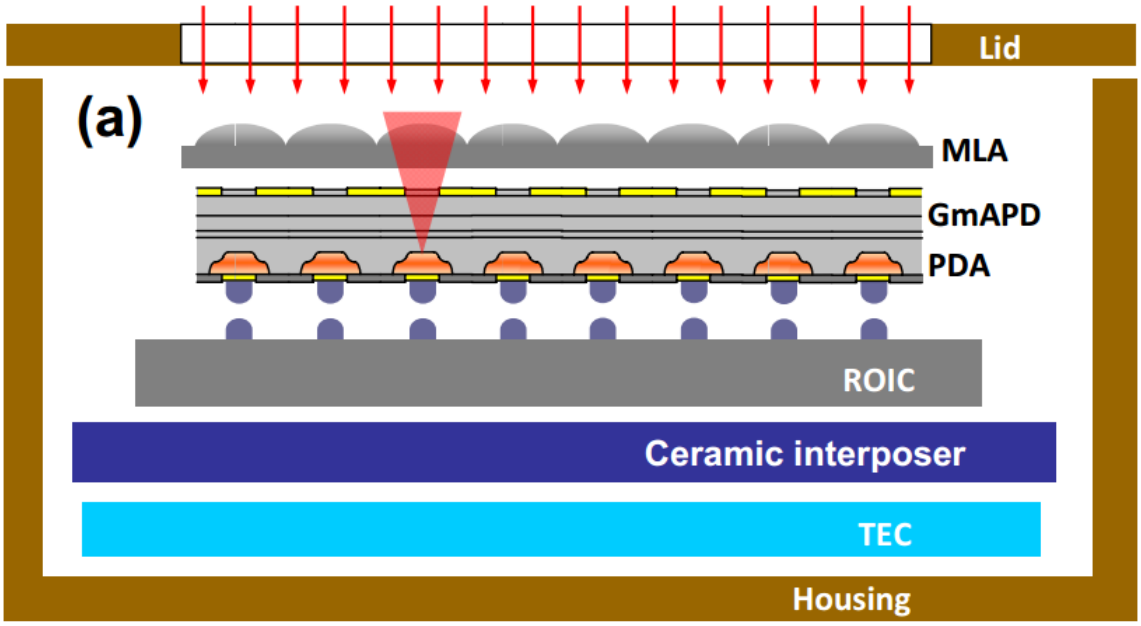

Figure 2.11: cross section of entire PLI GmAPD camera chip [25]

2.2.4.4 Thermal Electric Cooler, Ceramic, and Housing

With the MLA + FPA + ROIC all integrated on top of one another, the final optical hardware can be integrated. A ceramic interposer is placed on the bottom of the ROIC to allow for electronic wiring of the internals to the rest of the system [25]. This is then placed on top of a thermal electric cooler (TEC) which cools the array to roughly

250K. Finally, it is all placed inside a protective housing. A full cross section diagram is shown in Figure 2.11.

2.2.4.5 System I/O and other components

An additional controller is used for generating clock pulses and handling the data that is collected from the internal optics and packaging it into a standard format. PLI designed a specialized FPGA controller [25] for this task which is required for receiver functionality. This FPGA formats the data and transmits it over a standard high speed CameraLink protocol which is used as the primary communication interface between the computer and the receiver. Based off the internals of the camera and the high speed CameraLink interface, framerates of up to ~200 kHz [25][32][34] are achievable for real time data acquisition and saving.

2.2.5 GmAPD Receiver Output

The final data output of the GmAPD receiver is a frame readout of integer values corresponding to a TOF (also called time bin) value for each pixel. A frame usually lasts for a few microseconds. Each frame is further divided into thousands of sub frames which represent the different time intervals during the frame. For example, for the PLI camera used in this research, each frame lasts 2 microseconds and each sub frame is a 250 picosecond window within that 2 microseconds. Therefore, a total of 8,000 possible sub frames are possible. Each sub frame number corresponds to a binary TOF value indicating whether or not a pixel within the FPA fired during that sub frame (illustrated in Figure 2.12).

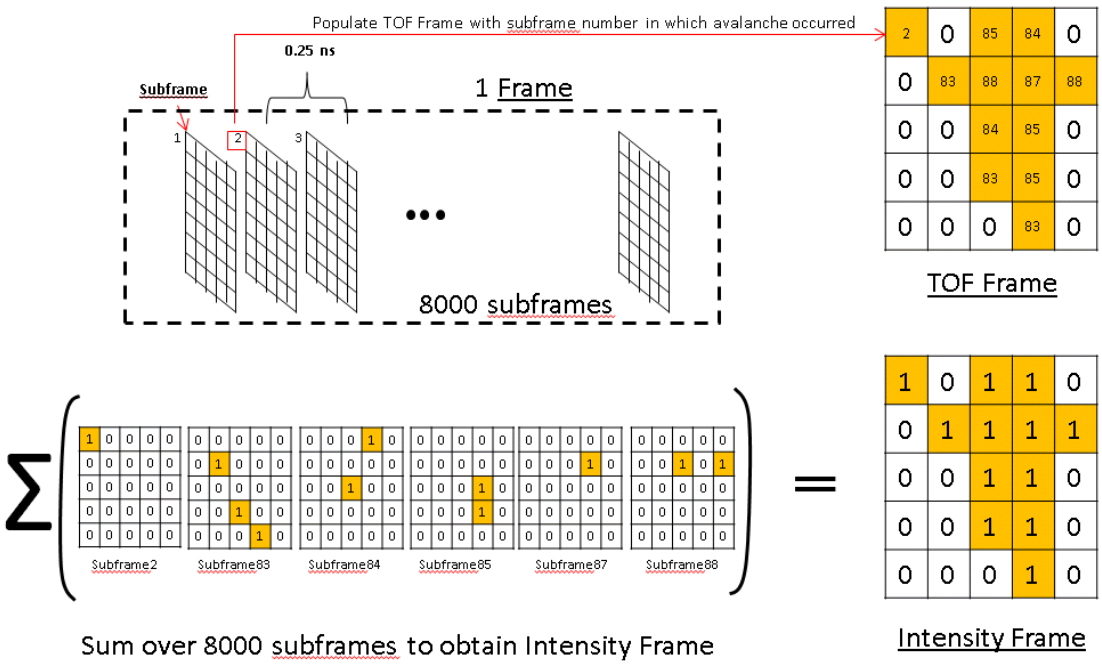

Figure 2.12: Summation of subframes to create intensity and TOF frames

In practice, a GmAPD-based LiDAR is operated by generating a frame for each pulse of the laser. Multiple pulses consequently generate multiple frames, and these frames can be accumulated in data processing software. In this scenario, the intensity of a pixel is the summation of all the hits that occurred in multiple frames. The maximum intensity can only be equal to the number of frames, as a pixel can only fire once per frame [34]. We normalized this intensity for clarity and future computations described in Chapter 3. The normalized intensity (hereafter called intensity) equation can be outlined in Equation 2-1.

$$I(i,j) = \frac{1}{f}\sum_{n=1}^{f} H(i,j)\,, \qquad H(i,j) = \begin{cases} 1, & TOF_n(i,j) > 0 \\ 0, & otherwise \end{cases} \tag{2-1}$$

where,

$I(i,j)$	=	is the normalized intensity of pixel (i,j)
H(i,j)	=	is the binary value for pixel (i,j)
f	=	is the number of frames to be integrated over
$TOF_n(i,j)$	=	is the TOF value for a pixel (i,j) in frame n

To help reduce noise, a threshold is introduced to eliminate pixels that don't meet an empirically derived intensity over the frame integration period. This threshold prevents spurious DCR counts on pixels that otherwise would have had no intensity.

The final frame readout is the combination of all the sub frames. Each pixel has an associated integer TOF value. This TOF value can then be mapped to an in-air path length (l_a) using a simple Equation 2-2:

$$l_a = \frac{TOF(i,j) * t_{SF} * c}{2} \quad (2\text{-}2)$$

where,

l_a	=	is the in air path length distance
$TOF(i,j)$	=	is the TOF value reported from the camera for pixel (i,j)
t_{SF}	=	is the sub frame timing window in seconds
c	=	is the speed of light in meters per second

From Equation 2-2, it's easily shown that one time bin value corresponds to ~ 3.75cm distance. Since one time bin represents a 250 picosecond window, it is evident how quickly results can become noisy. Timing jitter alone can correspond to a 7.5 centimeter ranging ambiguity before even considering DCR or environmental noise. It is therefore paramount to reduce or suppress as many sources of DCR as possible when using a GmAPD camera in order to obtain reliable and accurate results.

2.2.6 GmAPD Frame Averaging

As shown in Equation 2-1, the only way to accumulate an intensity image for a pixel is to integrate over a number of frames, producing what we refer to as an image.

However, integrating over these frames also yields multiple TOF values for the given pixels with hits. Therefore, since we have multiple TOF data for a given pixel over some frame integration period, we can add together all the TOF data for a given pixel and divide by the pixel's non-normalized intensity to get its average TOF value for that image, as represented in Equation 2-3.

$$\overline{TOF}_f(i,j) = \frac{\sum_{n=1}^{f} TOF_n(i,j)}{\sum_{n=1}^{f} H(i,j)}, \qquad H = \begin{cases} 1, & TOF_n(i,j) > 0 \\ 0, & otherwise \end{cases} \tag{2-3}$$

where,

$\overline{TOF}_f(i,j)$	=	Average TOF value for pixel (i,j) over *f* number of frames
$H(i,j)$	=	is the binary value for pixel (i,j)
$TOF_n(i,j)$	=	TOF value for pixel (i,j) in frame n
f	=	number of frames to be averaged

This frame averaging provides an additional benefit: it smooths the noise by reducing depth variance. This effect can be readily seen in Figure 2.13, which shows resulting GmAPD point clouds when averaging different numbers of frames from TOF data ($\overline{TOF}_1$, $\overline{TOF}_{23}$, $\overline{TOF}_{200}$, $\overline{TOF}_{1000}$, $\overline{TOF}_{5000}$, and $\overline{TOF}_{10000}$) gathered from a hard flat background. $\overline{TOF}_1$ is a sparse surface coordinate matrix due to only a single frame being used, indicating that not every pixel registered a hit in that frame. Moreover, $\overline{TOF}_{23}$ was measured since 23 frames represents the minimum number of frames required for every pixel to register a hit and thus fully saturate the surface coordinate matrix. The point cloud data is being viewed from a side angle, with color corresponding to depth away from the receiver.

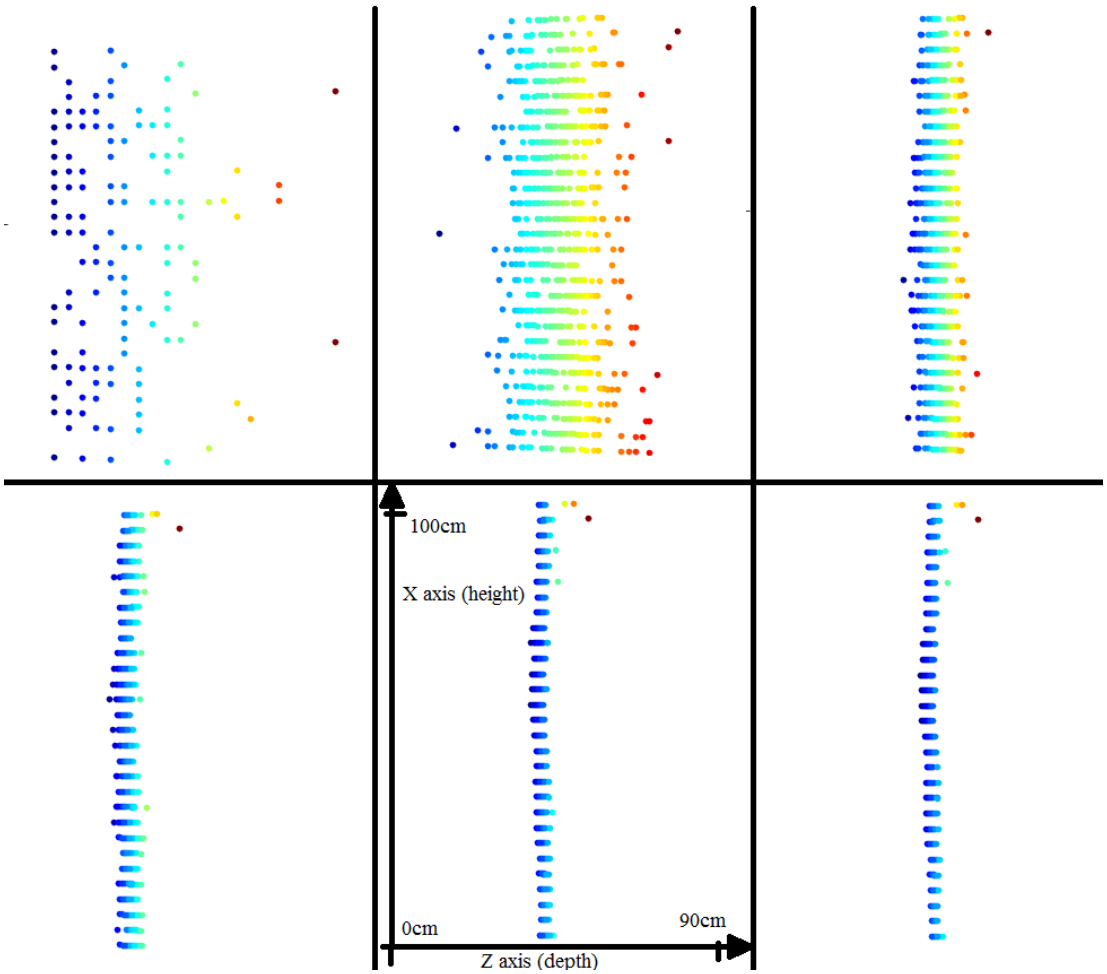

Figure 2.13: (From top to bottom, left to right) Results of averaging 1 frame, 23 frames, 200 frames, 1000 frames, 5000 frames, and 10000 frames on a flat hard target. σ = 15.1cm, 8.2cm, 2.9cm, 1.5cm, 1.1cm, and 1.0cm respectively.

The degraded accuracy of smaller averaged $\overline{TOF}$ surface coordinates when compared to larger $\overline{TOF}$ surface coordinates stems from the variation that a single time bin displacement has on the average range measurement (Equation 2-2). These results also show the advantage of having fast frame rates on the receiver. The setup used in this research only allows for 2 kHz PRR of the laser, almost 1/100 of the theoretical maximum the camera allows. Figure 2.13 illustrates that improving this frame rate would yield a marked increase in the accuracy of all range measurements.

2.2.7 Advantages and Disadvantages of GmAPD Receivers

A consolidated list of the advantages and disadvantages are listed in Table 2.1.

Table 2.1: Advantages and disadvantages of GmAPD Receivers

Advantages	Disadvantages
High detection efficiency	Current models only allow a pixel to have one avalanche per frame period
Very fast photon timing detection	High noise
Low power	Manufacturing defects affect performance parameters
Fast frame rates / Data acquisition	Non uniform pixel performance
Lower required laser energy	
Enables higher altitude flying	
Dramatically improved spatial resolution	

2.3 Point Cloud Noise Reduction and Smoothing Filters

Despite attempts to reduce noise in the receiver, system noise sources are intrinsic to nearly every imaging medium. Because each pixel in the FPA is single photon sensitive, environmental and multiple scattering of light can generate sufficient photons to trigger pixels within subframes between the detector and the target of interest. This noise is not only undesirable, but also unavoidable. With GmAPD images, these noise sources cause blurring and reduce the clarity of the scene or targets of interest. This noise is apparent even when imaging a hard, flat background in a static environment (same conditions as Figure 2.13) as shown in Figure 2.14.

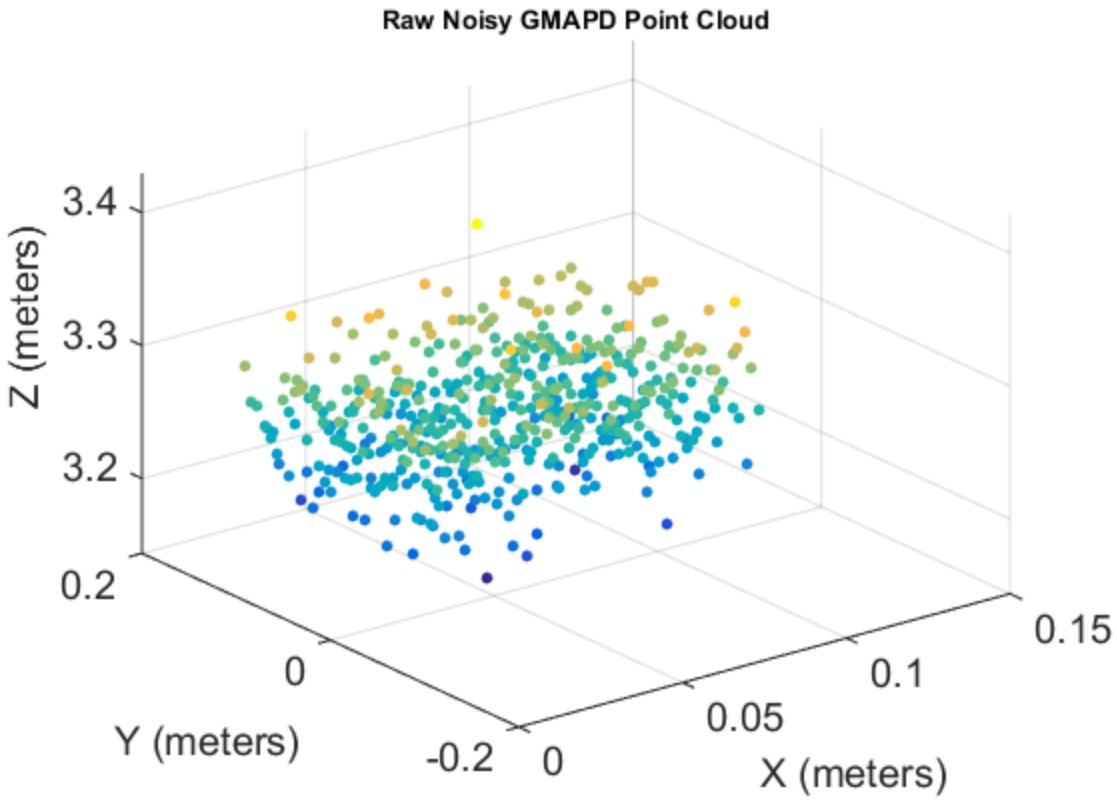

Figure 2.14: Noisy $\overline{TOF}_{200}$ point cloud data from hard flat target. Std σ = 2.83cm. Color represents z-axis height.

It is therefore important to remove as much of this noise as possible, and this is typically done in post processing. This is achieved using some form of smoothing or denoising filter which attempts to separate the noisy parts from the source. Given the many different types of available filters that aim to accomplish this goal, choosing the correct one depends on the data and processing time needed. Research has been done on the topic of 'coincidence processing' [39][40], an area which focuses on utilizing multiple beam return data for analysis and filtering. This paper uses aspects of coincidence processing for solving the specific problem of fitting many NIR LiDAR points to a water surface coupled with refinement via filtering algorithms.

For the development of real time processing systems, selecting the best quality filter meeting the speed requirements is a non-trivial task. Yet, technological maturity within the image processing field has led to fast and qualitatively impressive filters [41] which can meet the required description. Two of these filters which are used within this paper are implementations of the median filter and a fast non-local means filter [42].

2.3.1 Median Filter

The median filter is a nonlinear smoothing technique [43] used for digital image processing since the 1970s [44]. It is a very simple yet powerful filter which is highly effective at removing outliers (‘salt and pepper noise’) [43] while conserving edges within an image. It is often used as a preprocessing filter [45] to removing extreme impurities before applying a more robust weighted filter which would otherwise smear the extreme error over other values. Figure 2.15 illustrates a 3x3 window median filter over a grid of 4x4 data, where each center pixel is replaced by the median value of the window.

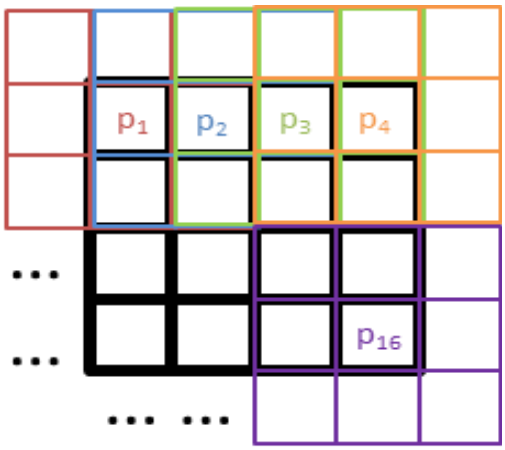

Figure 2.15: Sliding window median filter

The basic mathematic principles behind the filter are simple, and the equation for a 2 dimensional median filter is easily represented as Equation 2-4:

$$C_{i,j} = Median\left(w_{i,j}(U)\right) \tag{2-4}$$

where,

$C_{i,j}$	=	is the corrected value at pixel location i,j
$w_{i,j}$	=	is the rectangular window centered at pixel location i,j
U	=	is the noisy image

Edges are padded symmetrically to ensure proper edge computations, and variable window sizes lead to different degrees of smoothing.

In the research discussed in this paper, a median filter was applied immediately to the collected data to remove outliers. Since we are implementing real time processing, we use a small (n=3) window size to achieve higher speeds and avoid excessively smoothing the data. Median filtering can become very computationally expensive since it is consistently grabbing a set of data, sorting it, and selecting the median. This operation is often repeated thousands of times over the course of an image, requiring optimization in order to avoid sluggish performance. While such optimizations exist [46], one that is amenable to our data is the use of a sliding window and insertion operations [43]. This benefit exploits the fact that the median window replaces the previous row (or column) with the next row (or column), but all the remaining data stays the same. Therefore, only the new data needs to be inserted into the already sorted order which takes considerably less time than resorting unsorted data. This leads to drastically improved median filtering times which were sufficient for a real time application.

An example of its application is shown in Figure 2.16, where we show the median filter of window size 3x3 applied to the previous noisy GmAPD point cloud in Figure 2.14. Notice the improved standard deviation when compared to the original noisy point cloud.

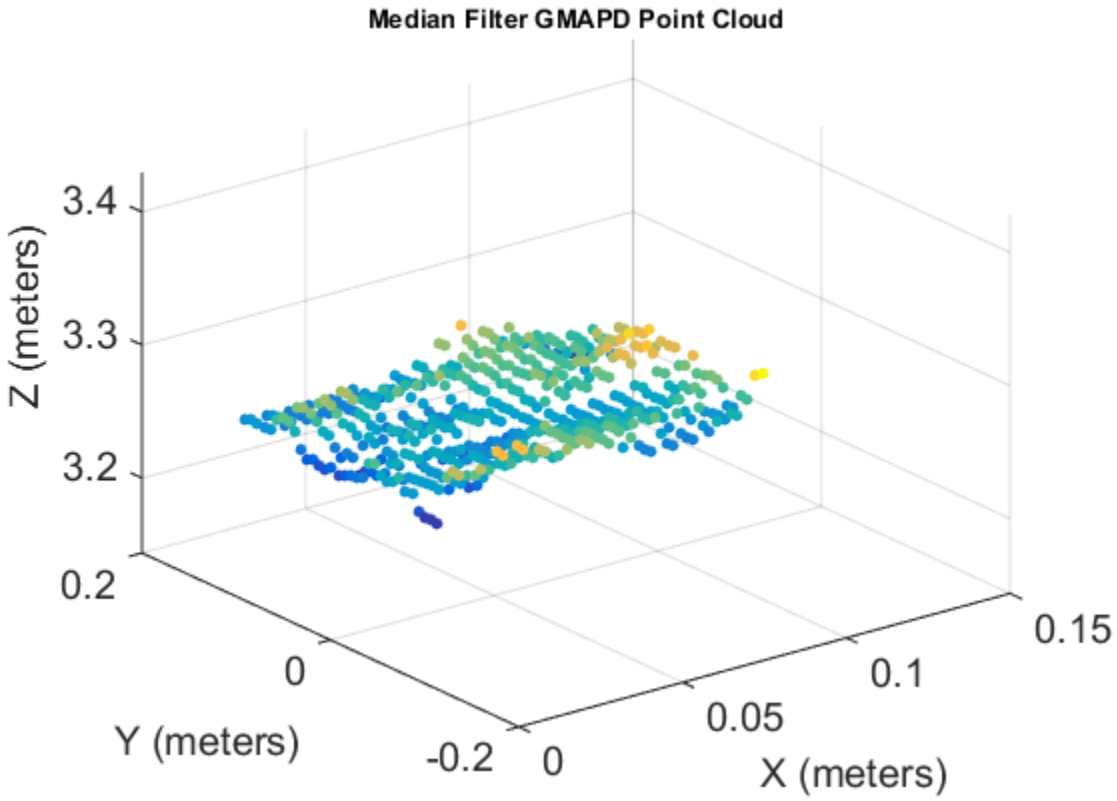

Figure 2.16: Median filtered noisy $\overline{TOF}_{200}$ data from Figure 2.14. Std σ = .8cm. Color represents z-axis height.

2.3.2 Fast Non-Local Means Filter

In addition to the preprocessing median filter, a more robust filter is also implemented to drastically reduce the noise in an image. There are literally hundreds of unique variations of sophisticated denoising algorithms [47][48][49][50], many of which work better with certain types of data relative to others. After empirical testing of a few

filters, a fast variation of a traditional Non-Local Means (NLM) filter [51] was selected [42].

NLM filter was first implemented a decade ago [51] and takes advantage of traditional Gaussian weights and pixel neighborhood means. Defining an image as a discrete regular grid Ω, we can formulate an equation for the NLM filter for restored image u at site $s \in \Omega$ as Equations 2-5, 2-6, and 2-7 [51]:

$$u(s) = \frac{1}{Z(s)} \sum_{t \in N(s)} w(s,t)v(t)\,;\;\; Z(s) = \sum_{t \in N(s)} w(s,t), \tag{2-5}$$

$$w(s,t) = \; g_h\left(\sum_{\delta \in \Delta} G_\sigma(\delta)(v(s+\delta) - v(t+\delta))^2\right), \tag{2-6}$$

$$g_h(x) = \; e^{\left(-\frac{x^2}{h^2}\right)}, \tag{2-7}$$

where,

$v(\cdot)$	=	is the noisy image at some site
$Z(s)$	=	is the normalization constant for site s
$N(s)$	=	is the set of neighboring sites of s
$w(s,t)$	=	is the non-negative weights between sites s and t
G_σ	=	is the Gaussian kernel with variance σ^2
Δ	=	is the discrete patch region containing neighboring sites δ
g_h	=	is the continuous non-increasing function
h	=	is the level of filtering applied

To summarize the above equations, the NLM filter restores an image by computing a weighted average of pixel values while considering spatial and intensity equivalences between pixels. This equivalence is calculated between equally sized patches since the patches adequately capture the local structure of their area. The algorithm searches over the square searching window N centered at site s and performs

the comparison to each square patch *t* inside the search window, computing weights based off of how similar the two patches are. While NLM has a very good image denoising quality overall when compared to other filters, empirical results showed it being too slow for the real time processing requirement of this research.

However, a faster implementation of the NLM filter was proposed in [42], which sped up the calculations without a noticeable impact on quality. The authors developed a modified version of Equations 2-5, 2-6, and 2-7 [42]:

$$u(s) = \frac{u_1(s)}{Z(s) + M(s)}; \qquad u_1(s) = u_2(s) + M(s)v(s), \tag{2-8}$$

$$M(s) = \max(M(s), w(s)); \qquad Z(s) = \sum_{k=0}^{n-1} w(k) \tag{2-9}$$

$$u_2(s) = \sum_{k=0}^{n-1} w(k)u_2(k + d_x) \tag{2-10}$$

$$w(s) = g_h\left(S_{d_x}(s + P) - S_{d_x}(s - P)\right) \tag{2-11}$$

$$S_{d_x}(n) = \sum_{k=0}^{n-1} (v(k) - v(k + d_x))^2, \;\; d_x \in [-K, K]^d \tag{2-12}$$

$$g_h(x) = e^{-\frac{x^2}{h^2}}, \qquad g_h(0) = 1, \qquad \lim_{x\to\infty}\left(g_h(x)\right) = 0 \tag{2-13}$$

where,

$v(\cdot)$	=	is the noisy image at some site
$Z(s)$	=	is the normalization constant for site *s*, initialized to 0
$u_1(s)$	=	is the final weighted value at site *s* before normalization scaling
$u_2(s)$	=	is the iterated weight value at site *s*, initialized to 0
K	=	is the half radius of the search window. Entire window $[-K, K]^d$
d	=	is the dimension of the signal
n	=	is the number of pixels in image
P	=	is the half patch size
$M(s)$	=	is the maximum weight value for each site *s*, initialized to 0
$w(\cdot)$	=	is the non-negative weights

S_{d_x} = is the discrete integration of the squared difference
d_x = is the translation vector,
g_h = is the continuous non-increasing function
h = is the level of filtering applied

The innovation in the algorithm leading to the increased speed comes from the method of calculating the weights. In the original NLM filter, the weight values are calculated between site *s*, its corresponding neighboring site *t*, and every distance δ within the patch window. This approach leads to a time complexity of $O(n(2K + 1)^d(2P + 1)^d)$, where n is the number of pixels, K is the half window width of the neighbor search, P is the half window width of the patch search, and d is the dimensionality. The faster NLM method introduced in [42] improves upon this by calculating the weights at just the ends of fixed half width patch offset instead of iterating through the entire patch. Since the improvement only requires 2^d patch operations for every comparison between sites *s* and *t*, the time complexity is $(n(2K + 1)^d 2^d)$, which is significantly faster than the traditional NLM method. The improvement in image quality achieved by combining this filter with the median filter is show in Figure 2.17, which is the result of the fast NLM filter applied to the data shown in Figure 2.14 and 2.16.

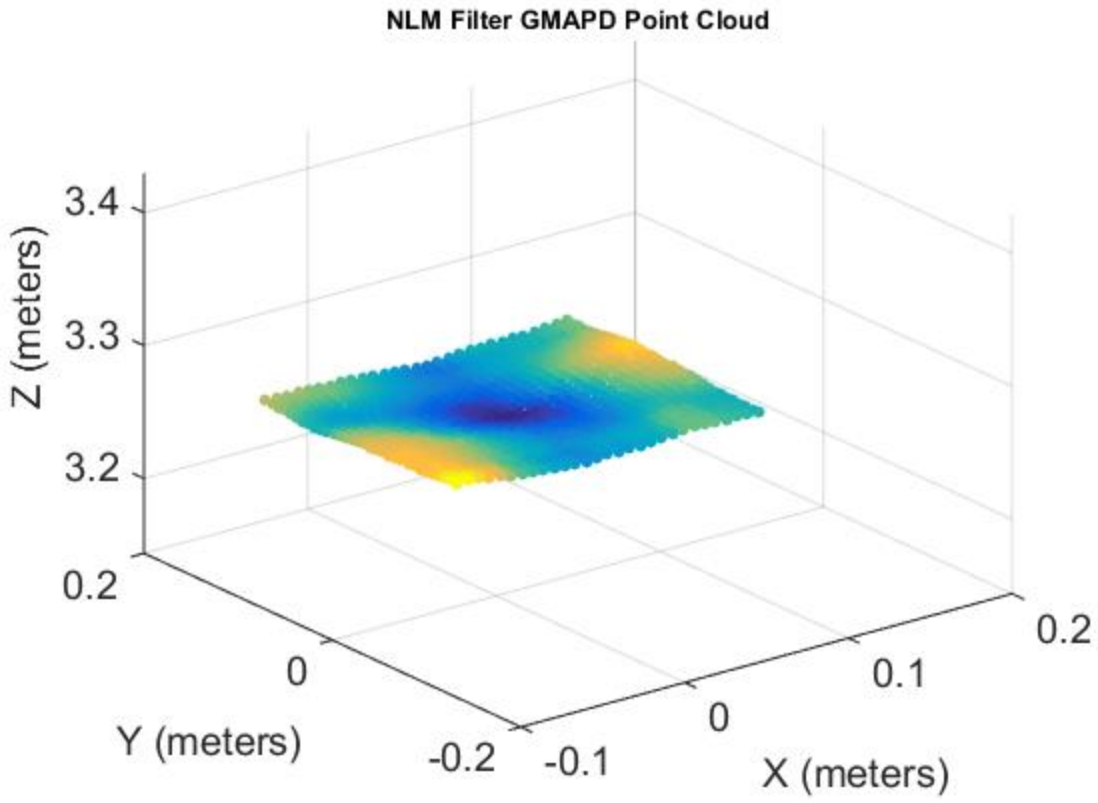

Figure 2.17: NLM filter over median filtered $\overline{TOF}_{200}$ data from Figure 2.16. Std σ = .3cm. Color represents z-axis height.

With the median and NLM filter, we achieved adequate smoothing and denoising of the data for real time processing.

2.4 Digital Surface Model: Data Structures

Correctly modeling point cloud data is important for having an accurate visual representation. Certain important features of the data may be hidden or deemphasized when an inappropriate data structure is used to model it [52]. Moreover, certain structures may work well with one particular set of data and then fail with another set of data [53]. Identifying these important parameters intrinsic to the data and collection methods is vital when selecting an appropriate data structure for a DSM. A DSM is a visually modeled representation of coordinate data [53][54], created to help aid analysis and viewing of

large amounts of information. It has many applications in Geographic Information Systems (GIS) and Geomatics [52][53][54]. While collection of the data is usually dependent on the system (e.g LiDAR, photogrammetry, etc) the modeling of that data is user controlled. A few different options exist for the structure, but the two most currently used are raster grids and triangulated irregular networks (TIN) [52]. Each has its own advantages and flaws which makes appropriate selection a matter of the data and user preference.

2.4.1 Raster Grids

The most commonly used DSM data structure is the raster grid, which is a grid of rectangular tiles spanning in the x and y direction with elevation data assigned to each tile [52][53]. Raster grids are easy to interpret and are computationally efficient to process since they have known dimensions and static tile size, akin to digital photo image processing. Moreover, raster grids can accurately represent many of the features provided by the data such as different types of surfaces, slopes, and elevations. When taking an aerial photo, a raster grid is a common choice for an overlay using pixel intensity as elevation [53]. This gives a direct one to one correspondence of data and data structure which can then be further processed. A sample look at the raster grid DSM structure can be seen in Figure 2.18.

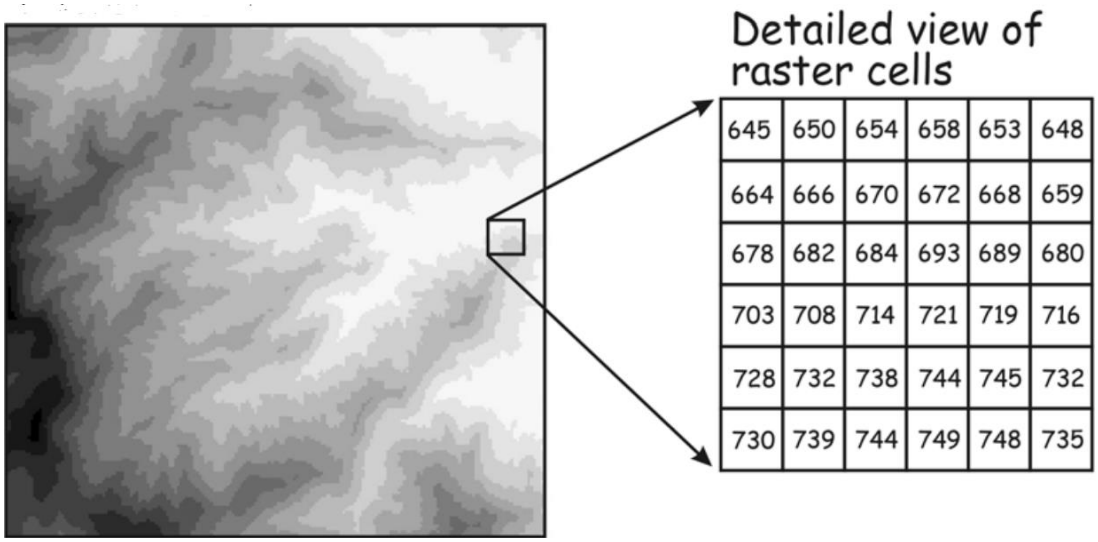

Figure 2.18: Raster grid example with zoomed in grid values [53]

However, unless used with a compression scheme, raster grids do have drawbacks which other methods (such as the TIN) can remedy. One issue with raster grids is its data redundancy over non changing terrain [52][53]. All the data represented by the tiles is stored even if a large patch of adjacent tiles contains similar or identical data. In this case, larger storage capabilities and additional processing are required though no additional detail is provided by the model. Furthermore, using a raster grid to model relief features of terrain data is difficult due to the static spacing between points. Essentially, areas that would require more data to visually indicate a depression or elevation get the same treatment as a completely flat piece of terrain, creating a more smoothed look over natural ridges [52][53]. Despite these limitations, the flexibility, natural data layout, and programmatic simplicity of raster grids is what has led it to be the predominant form of data structure used in current applications.

2.4.2 Triangulated Irregular Networks

Triangulated irregular networks help to eliminate some of the drawbacks which are presented by the raster grids. TINs are constructed by drawing triangles between (x,y,z) data points where each triangle aims to minimize the difference between the three angles making up the triangle [55]. In other words, it tries to reduce the number of long, skinny triangles and maximize the number of near equilateral triangles. The process of creating these triangles is through Delaunay Triangulation (DT) [56], and is a relatively computationally expensive algorithm to perform with large data sets. Nonetheless, the resulting triangulation produces a reduced set of points and corresponding triangles which reflect the correct elevation metrics of the data [52][53]. Flat terrain can be represented by a single triangle (three points) which can span very large distances, whereas areas of sharp elevation change can have numerous triangles (many data points) to reflect the rapid elevation dichotomies. This ensures that only the data that is needed is stored, and that stored data is a very precise representation of the actual image. Figure 2.19 shows what a triangulated network looks like with points on surface.

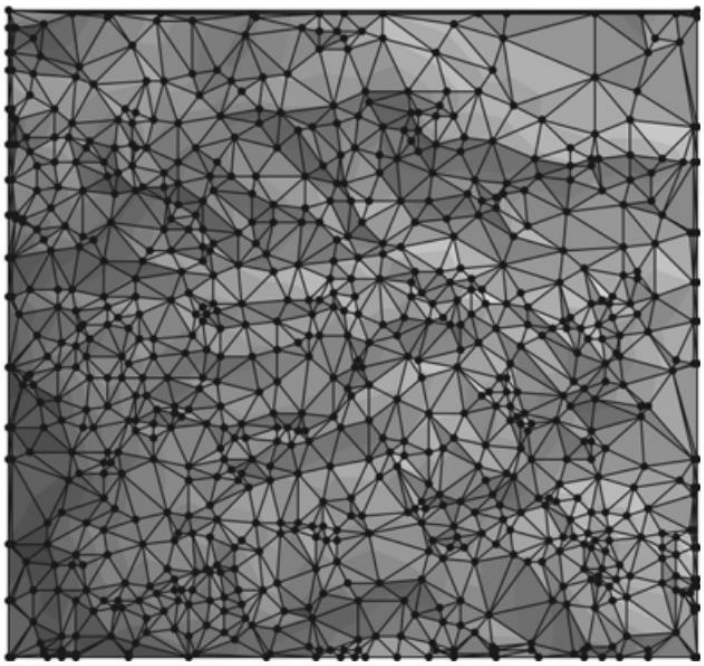

Figure 2.19: Triangulated Irregular Network of points [53]

While the TIN seems to fix the issues inherent to raster grids, the process of triangulation from the original data is non-trivial and increases with the number of points. Moreover, further post processing of TIN data is much more complex programmatically as one has to iterate over various triangles and points instead of a more logical 2D grid in the raster case [52][53]. Still, TINs provide more clarity at a reduced storage cost when compared to raster grids and are used extensively in the field of GIS.

For the GmAPD, raster grids are a byproduct of the outputted format from the GmAPD being in a square array. This output structure, however, prevents exploiting any of the benefits TINs provide in their unequally distributed points. Theoretically, a TIN could be computed from this raster grid after it is made, and was originally attempted at the beginning phases of this research. However, this consumed too much processing time

for limited gain. Therefore, raster grids were the implemented DSM data structure for this research.

2.5 Ordinary Least Squares Regression Planes and Normal Computations

The final step in correcting for sea surface structure is the computation of the vector normal to the calculated surface. The accuracy of the normal vector is inarguably the most important factor related to this research, as it is the main component used in correcting for the refracted beam inside the water column [1][2] (this is shown mathematically at the end of this section). However, before a normal can be computed, a surface must first be fit to the data. Since the GmAPD camera already outputs a raster grid of points, a simple but effective ordinary least squares (OLS) regression plane can be constructed for surface fitting [57]. Once this is established, normal vectors are provided automatically through the plane equation coefficients and the surface computations will be completed. A regression plane is calculated for each pixel in the GmAPD raster grid, with the regression plane serving as a convolution kernel window centered on the pixel it is computing a normal vector for. This provides additional accuracy for each normal vector since they are each the center of their respective neighborhood being fitted to a plane.

2.5.1 Regression Planes

Regression planes are the three dimensional variant of regression curves. A plane can be described as in Equation 2-14.

$$d = ax + by + cz \tag{2-14}$$

where ,

d = is the perpendicular distance away from origin
a, b, c = are the components of the normal vector to the plane

x, y, z = are the coordinates of a point on the plane

Rearranging Equation 2-14 to get it as a function of the independent variables x and y yields Equations 2-15, 2-16, 2-17, and 2-18.

$$z = Ax + By + C \quad (2\text{-}15)$$

$$A = -\frac{a}{c}; \quad B = -\frac{a}{c}; \quad C = \frac{d}{c} \quad (2\text{-}16;2\text{-}17;2\text{-}18)$$

Setting the constant *c* to -1 can be done to simplify the calculations and indicate which direction we want the z component of the normal vector to point. We choose -1 because for our research, the positive z direction is downward. Therefore, it makes sense visually to think of the normal vectors pointing upwards off the water surface into the negative z direction, thus necessitating a -1 for *c*. This establishes our normal vector to the plane to be Equation 2-19.

$$V_{Norm} = [A \quad B \quad -1] \quad (2\text{-}19)$$

where,

A = is the x component of the normal vector
B = is the y component of the normal vector
-1 = is the z component of the normal vector

From here, regression planes aim to minimize some metric between the two independent variables (x and y) and the one dependent variable (z), shown in Figure 2.20.

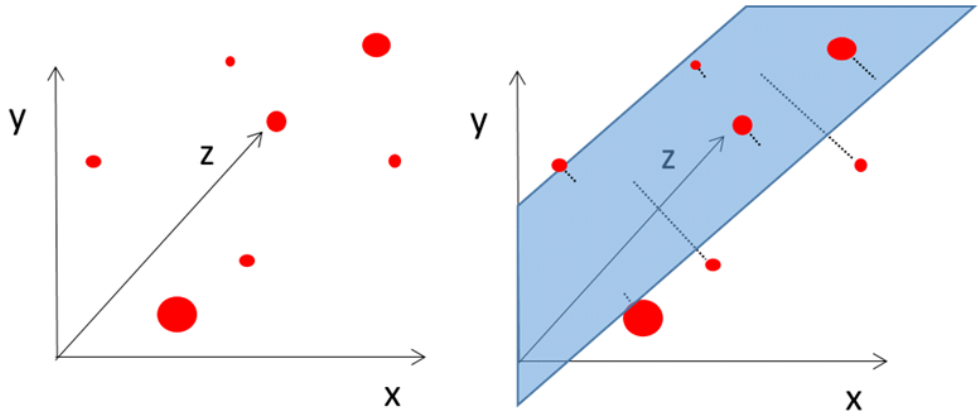

Figure 2.20: Regression plane for points in three dimensions. Larger points are closer along depth axis

While many different minimizations exist, one of the most widely used is the OLS regression [57]. The OLS regression minimizes the sum of the squared distances between the independent variables and the dependent variable. This can be conveniently shown using linear algebra matrix representations. Rewriting Equations 2-15, 2-16, 2-17, and 2-18 into a matrix representation of a system of equations creates the following Equations 2-20, 2-21, 2-22, and 2-23 .

$$Z = Xv; \quad Z = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}; \quad X = \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix}; \quad v = \begin{bmatrix} A \\ B \\ C \end{bmatrix}; \qquad \text{(2-20;2-21;2-22;2-23)}$$

where,

Z	=	is the vector of z dependent variables
X	=	is the matrix of dependent variables
v	=	is the vector of coefficients
n	=	is the number of points being regressed

Multiplying both sides of the equation by X^T (transpose of X) yields the following set of Equations 2-24 and 2-25.

$$(X^T X)v = X^T Z \tag{2-24}$$

$$\begin{bmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} y_i^2 & \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} y_i & \sum_{i=1}^{n} 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} x_i z_i \\ \sum_{i=1}^{n} y_i z_i \\ \sum_{i=1}^{n} z_i \end{bmatrix} \tag{2-25}$$

where,

x_i	=	is the x value at point i
y_i	=	is the y value at point i
z_i	=	is the z value at point i
A, B, C	=	are the solution coefficients
n	=	is the number of points being regressed

An important detail about the multiplication of $X^T X$ is that X is of full column rank (i.e each column is linearly independent of one another). This means that $X^T X$ produces a symmetric positive definite (SPD) square matrix of size 3x3. This property will become very important when solving this system of equations.

The final system of equations can be formed by rearranging Equation 2-24 and solving for *v*, becoming Equation 2-26.

$$A_{mat} v = b \Rightarrow v = {A_{mat}}^{-1} b \tag{2-26}$$

where,

A_{mat}	=	is $(X^T X)$ from Equation 2-24
v	=	is the solution coefficient vector
b	=	is $(X^T Z)$ from Equation 2-24

Because A_{mat}, is SPD, Cholesky decomposition can be used to solve for *v* without explicitly solving the inverse ${A_{mat}}^{-1}$. This leads to not only faster performance, but also a more numerically stable computation. Solution coefficients *A*,*B*, and *C* are computed and plugged into Equation 2-19 for the normal vector to the plane and into Equation 2-15 to compute a new z-value for the point.

2.5.2 Normal Vector Ray Trace Equation

A brief overview of the importance of the normal vector will be explained by using Glasner's ray trace equation [2]. A quick look at the underlying equation will make it apparent why the computed normal vector is vital to the overall accuracy of ALB sea floor measurements. This effect can also be seen in Figure 2.21 using the same type of coordinate system presented in this research (forward-starboard-down).

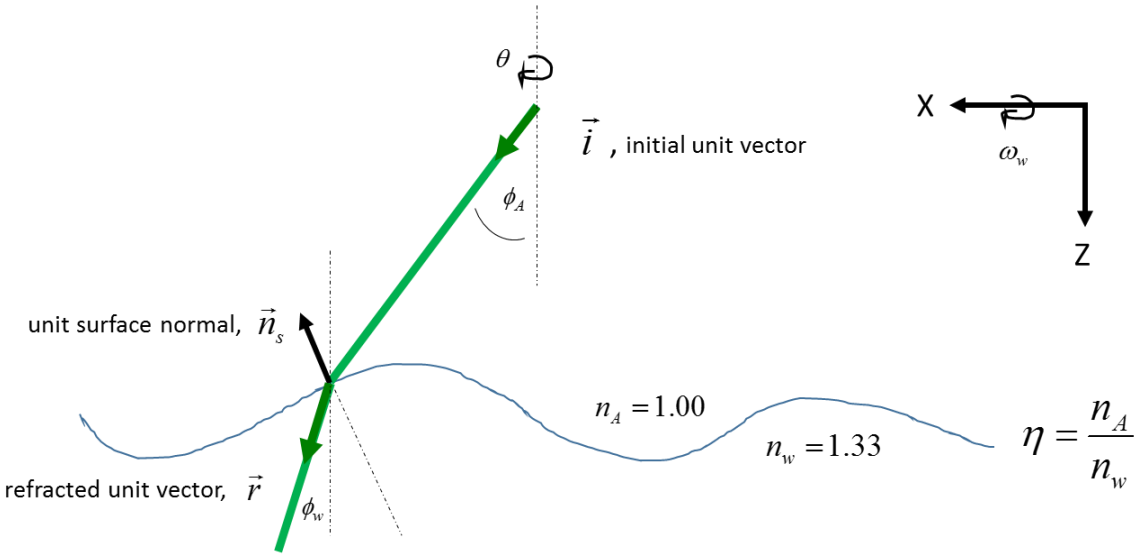

Figure 2.21: Effect the normal vector has on refraction angle [1].

Glasner's method to calculate the refracted vector in the water column can be seen in Equation 2-27 [2].

$$\vec{r} = \eta\vec{\imath} + \left(\eta k - \sqrt{1 + \eta^2(k^2 - 1)}\right)\vec{n}_s \quad (2\text{-}27)$$

where,

$\vec{r}$	=	is the refracted vector in the water
$\vec{n}_s$	=	is the normal vector to the surface
$\vec{\imath}$	=	is the pointing vector from the incident beam
k	=	is the dot product between $-\vec{n}_s$ and $\vec{\imath}$
η	=	is the refractive index ratio n_{air}/n_{water}

Equation 2-27 indicates that the refracted vector is a linear combination of the incident beam vector and the computed surface normal vector. The importance of the surface normal vector increases the wavier the water surface is, leading to huge propagation errors if uncorrected (uncorrected meaning just using the incident beam's off-nadir angle) as shown previously in Figure 1.2. Creating an accurate water DSM and applying the appropriate normal vector corrections to minimize this error is the main application behind this research.

2.6 Summary

Chapter 2 presented a foundation of the ALB system and identified the current spatial density limitation present in most receivers. It also briefly touched on the current standard of accounting for refraction correction over bathymetric data. This transitions into the emergence of the GmAPD receiver which has many potential benefits for an ALB system, with the key one of improved spatial resolution being highlighted as the catalyst for this research. Described are the techniques used for the processing of the dense data output of the GmAPD camera and further formatting into an appropriate DSM data structure. Finally, Chapter 2 ends with the creation of the DSM and accompanying

normal vectors which could be used for refraction correction in sea floor ray tracing equations.

Chapter 3 will establish the methodology behind the experiment, illustrating and explaining the setup and parameters of the testing environment. It will apply and relate to information that has been described in Chapter 2 while defining additional implementation and calibration steps for the experiment.

CHAPTER 3

METHODOLOGY

3.1 Hardware and Software

Development, deployment, and setup of the required hardware and software used for this research was an important stage of the research in order to accurately mimic real life ALB conditions. The accuracy of the setup is directly linked to the reliability of the digital model since TOF data needs to ultimately be mapped to real 3D coordinates. Everything from the positioning of the laser down to the placement of wave generator required precision in order to ensure a consistent testing environment. The setup and data collection involved numerous researchers to have a valid experiment. The overall goal of the experiments was to create a water DSM using the GmAPD receiver and then compute sea floor coordinates with the DSM providing refraction correction information. My main contributions to these experiments and corresponding focus of this thesis are the processing of the GmAPD data and computation of the 3D points clouds and surface normal vectors in real time.

3.1.1 Water Tank

The testing took place in a 6.2m meter deep tank filled with water located in the Love Building on the campus of Georgia Tech. Grates above the water tank allowed for movement in and around the area, and an overhead crane allowed for setup of the wave generator device. When no fan or wave generator device is active, the water is completely calm and allows for perfect still water imaging. Normal water clarity was crystal clear, and therefore vacuums were run prior to testing in order to kick up some dust from the

bottom of the tank to better replicate real life water conditions and generate volume scattering with the water column to be measured by the LiDAR's green beam. Figure 3.1 shows the view from the back of the water tank and pathfinder.

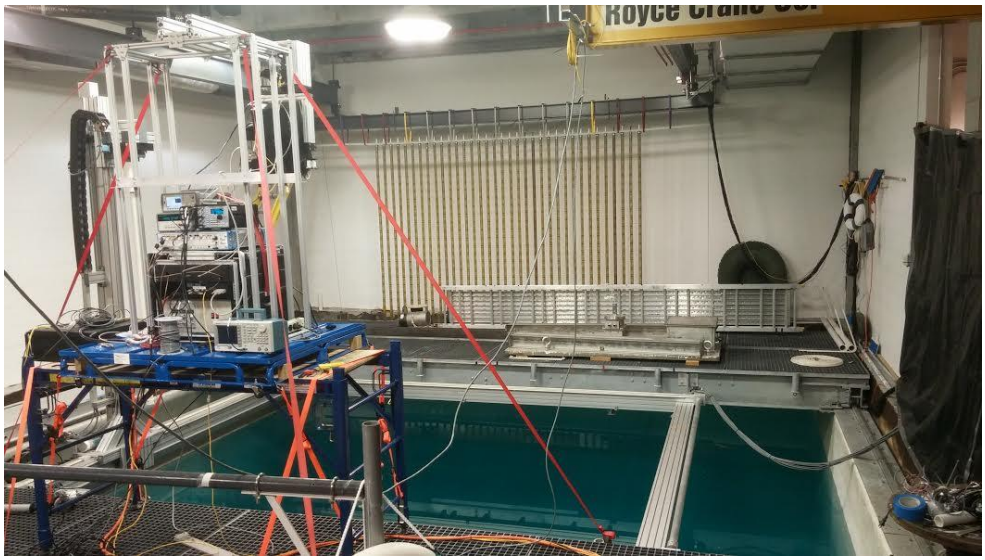

Figure 3.1: Image of experimental water tank and LiDAR in Love Building at Georgia Tech

3.1.2 GTRI Pathfinder LiDAR

The GTRI Pathfinder LiDAR is a hybrid waveform-resolved and GmAPD LiDAR under development within GTRI's Electro-Optical Systems Laboratory (EOSL). Approximately 25 employees have been engaged in designing and building it. The Pathfinder has separate green and NIR transmitters (and corresponding beam expanders), as shown in Figure 3.2. It rests on a cantilever attached to temporary scaffolding which is supported by the water tank grates and tension cables. The laser is a solid state frequency-doubled Nd:YAG producing co-aligned NIR and green beams. The 2 beams are purposefully separated external to the laser allowing implementation of different divergence angles for the 2 components. The laser has a pulse repetition rate of 2 kHz

and is oriented 19.5 degrees off-nadir. There are additional positional and angular offsets for the NIR GmAPD receiver and transmitter which are mounted outside the main shell housing the green beam and transmitter. The GmAPD receiver is a PLI manufactured engineering grade camera which has about 150 dead pixels concentrated mainly in the bottom three rows of the FPA. Because of this, the GmAPD array size was reduced from 32 rows x 32 columns to 29 rows x 32 columns. The pathfinder's height from the water is roughly 3.1m with a hypotenuse of roughly 3.3m. This short in-air path length required introducing beam expanders for the 2 components so that the physical size of the footprints at the water's surface would be sufficient to study the effects of water surface waves on the in-water pointing vector. Overhead and side views are shown in Figure 3.3 with the coordinate axes for our experiment.

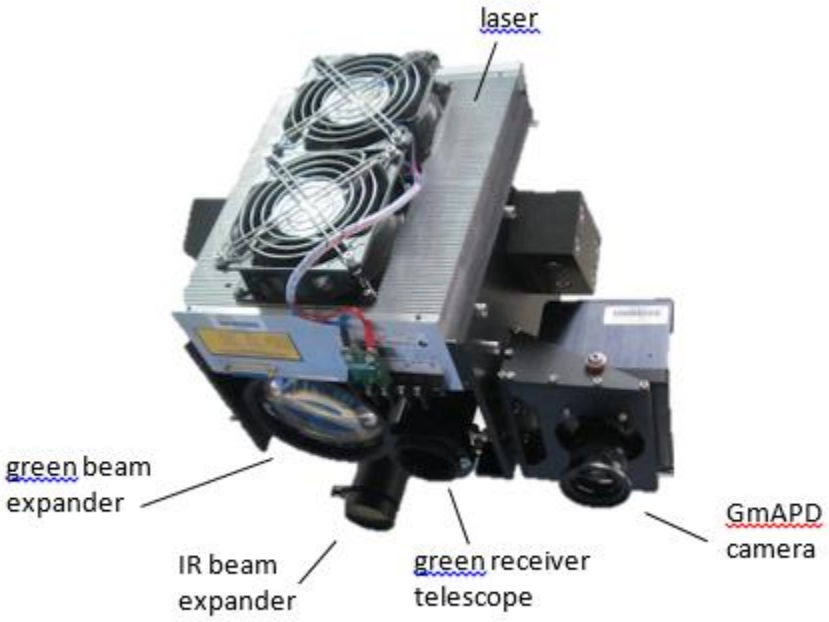

Figure 3.2: Picture of the laser pathfinder and components

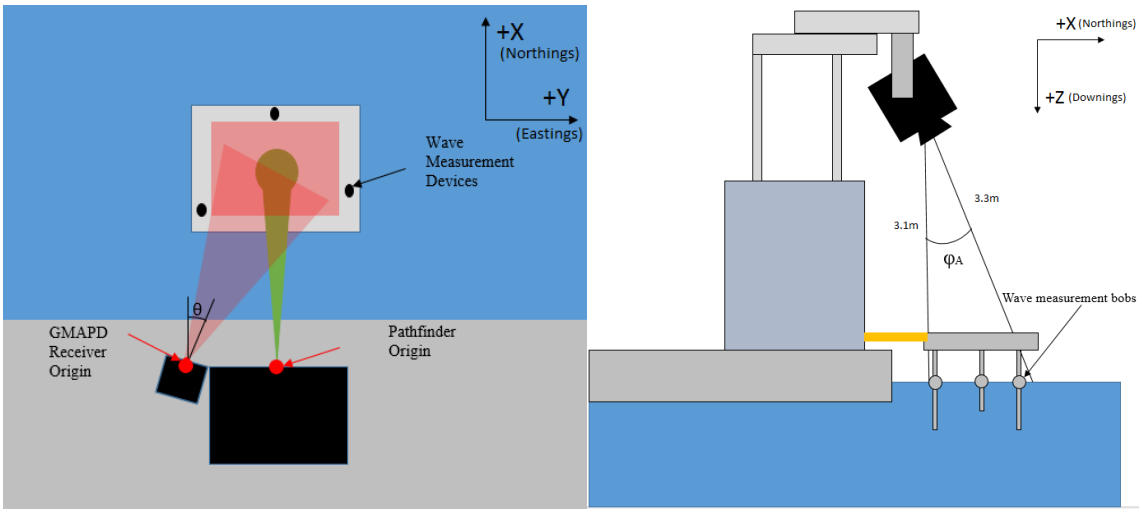

Figure 3.3: Overhead and side views of the experimental setup with scan angle θ, off nadir angle ϕ_A, and coordinate axes

The transmitters for both the NIR and green are collocated with the receivers for our given imaging distance, but each device transmits and receives from their own location within the pathfinder. This setup represents a biaxial system, where the transmitters and receivers are disjoint from one another, as shown in Figure 3.4. Commercially deployed ALBs are coaxial, having their transmitter and receivers in a single telescope. While a coaxial arrangement is more difficult to produce (and therefore expensive), it doesn't require as much geometric calibration and alignment compared to a biaxial system.

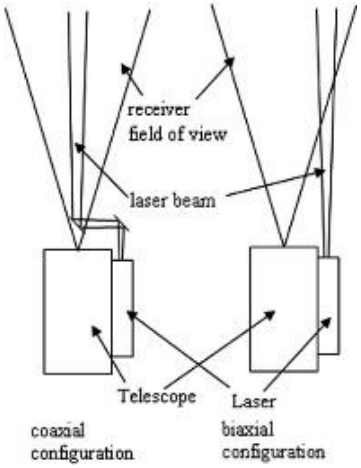

Figure 3.4: Coaxial versus Biaxial arrangement

3.1.3 Imaging Area

The imaging area for our experiment is a spot on the surface of the water directly in front of where the laser is aimed. Our GmAPD receiver has an imaging dimensionality of roughly 20cm x 18cm due to the beam expander on the pathfinder, meaning there is about a .75cm distances between points on the surface of the water. The green beam is collocated with the IR beam and is in the center of the IR beam footprint, making up roughly ¼ of our imaging area. While the entire IR beam data is collected, the ray tracing and refraction correction only occurs for the centered green beam laser. Since the green laser is ¼ of our imaging area and centered in the middle of our IR beam, we will use the middle 16 x 16 block of pixels (256 total) of the GmAPD receiver for ray tracing computations.

In addition to the properties of the laser, there is a steel plate suspended near our imaging area for holding wave height measurement devices in fixed positions. The steel plate has three additional cut outs around the edge for the auxiliary wave height measurement devices. This plate and wave measurement devices were originally intended

to be placed around our beam, but the wood was causing imaging issues with the receiver. These can all be seen in Figure 3.5.

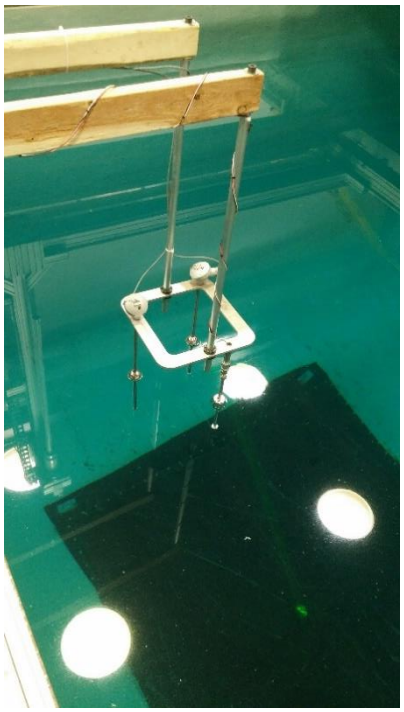

Figure 3.5: Imaging area suspended in front of LiDAR pathfinder setup

3.1.4 Wave Height Measurement Devices

Also shown in the Figure 3.5, we used the wave measurement devices to quantify the wave height over our imaging area. Two of the wave height measurement devices are 20 inches long with 1/5 inch resolution and sampling rate of 1 kHz, whereas the third one is 12 inches long with ¼ inch resolution and identical sample rate. These devices are connected to a LabJack which measures the voltage based off the position of the center bob. These voltages are linearly related to the bob distance and allows us to measure the wave heights as a function of voltage.

3.1.5 Wave Generator

A steel container partially filled with water was used as a wave generator to produce consistent low frequency waves. The container weighed roughly 400 lbs. when empty and around 800 pounds when filled 1/3 the way with water. The wave generator was suspended using a pulley mechanism from an overhead crane which prevented it from moving much side-to-side. An additional rope was attached to the pulley mechanism that could be pulled to raise and lower the container into and out of the water. After a period of raising and dropping the wave generator, the surface of the water obtained significant wave height on the order of 4 – 8 inches peak to trough. While a relatively crude implementation, it proved an effective mechanism for the purposes of our experiments. Figure 3.6 shows the wave generator in the corner of the tank.

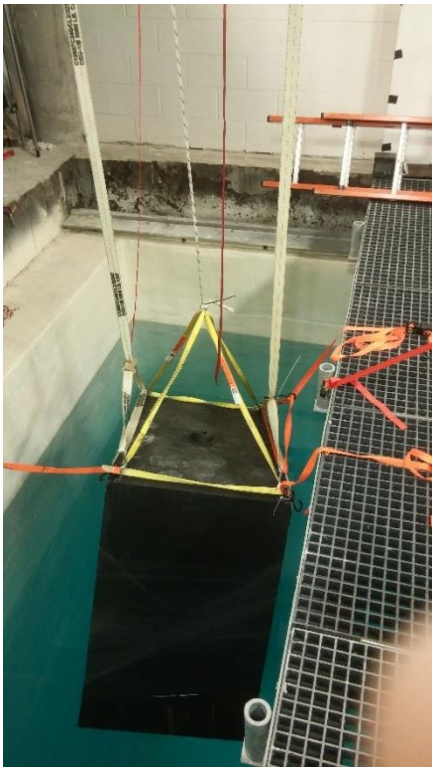

Figure 3.6: Wave generator

3.1.6 Underwater Camera

An underwater camera was centered and angled downward at 40.92 degrees over the bottom of the tank. A black tarp was placed on the bottom of the white pool floor for easy identification of the laser beam steering. This was a Point Grey Gigabit-Ethernet camera which provided raw gray scale images of roughly 30 FPS. When paired and synced with our other imaging hardware, we can see the exact location of the sea floor beam. Using a perspective transform of the image and known coordinates of the corners of the tarp, a calculation can be made to find the coordinates of the green beam in our global coordinate system. Figure 3.7 shows the underwater tarp from overhead and underwater Point Grey camera view.

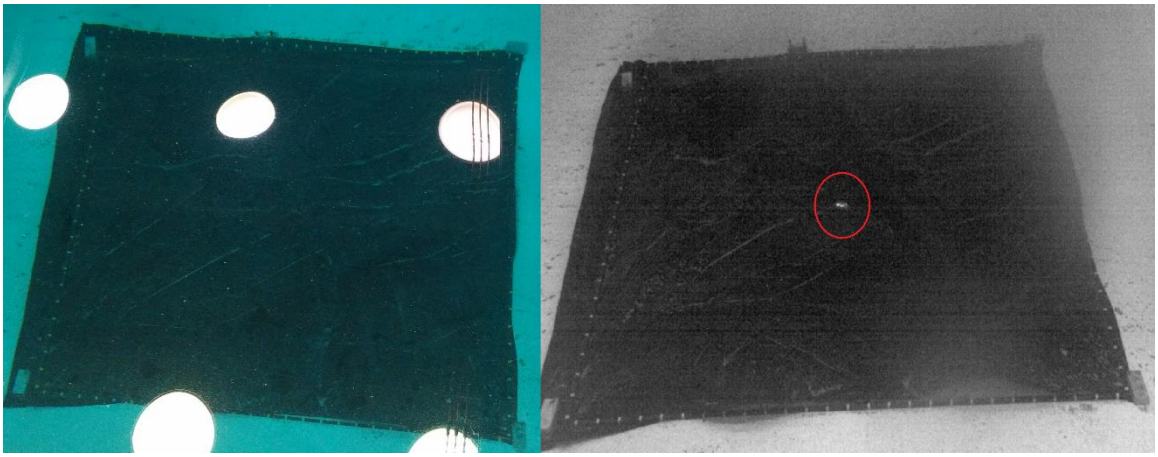

Figure 3.7: Overhead and underwater view of black tarp. Green beam location is circled in underwater view. The white circles are the reflection of the overhead lights from the water's surface.

3.1.7 Point Cloud Library Visualizer

Point Cloud Library (PCL) is a C++ library used for plotting 3D points for visualization purposes. Our implementation was to have PCL running in its own separate

program and have our main software send the computer surface coordinates and surface normal vectors to it for display. The visualizer has two halves: the left half shows all the coordinates in the X-Y plane, whereas the right half shows all the coordinates in the X-Z plane and displays the normal for the collocated green beam pixels. This gave real time imaging and visualization to what was being detected and processed by the receiver and accompanying algorithms for accuracy assessment purposes. Figure 3.8 shows the point cloud and normal vectors on the water's surface as computed and visualized in real-time with PCL.

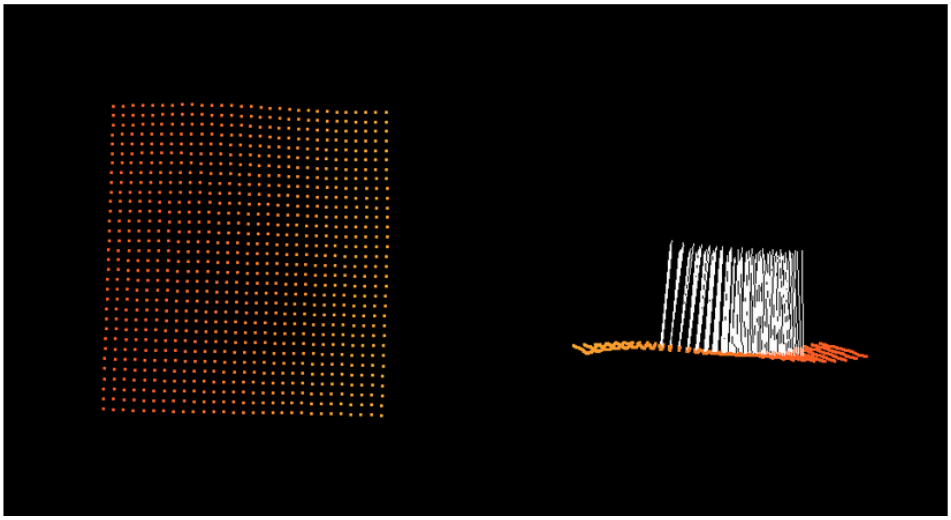

Figure 3.8: PCL illustration of $\overline{TOF}_{200}$ point cloud and green beam normal vectors. Imaging area is about 20cm x 18cm with approximately 0.75cm distance between points

3.1.8 Software GUI

New software for configuring the hardware, visualizing the data in real time, and saving the data was created specifically for this project. The GUI was created and evolved through contributions of many researchers, including myself, for interaction with the laser pathfinder and associated hardware. It also allows for real time monitoring of

the generated point clouds, green beam surface normals, GmAPD Receiver intensity image, and underwater camera. When an experiment ended, it saved all the data collected in a compact data format (HDF5) for easy accessibility and parsing. Figure 3.9 shows a picture of the complete GUI.

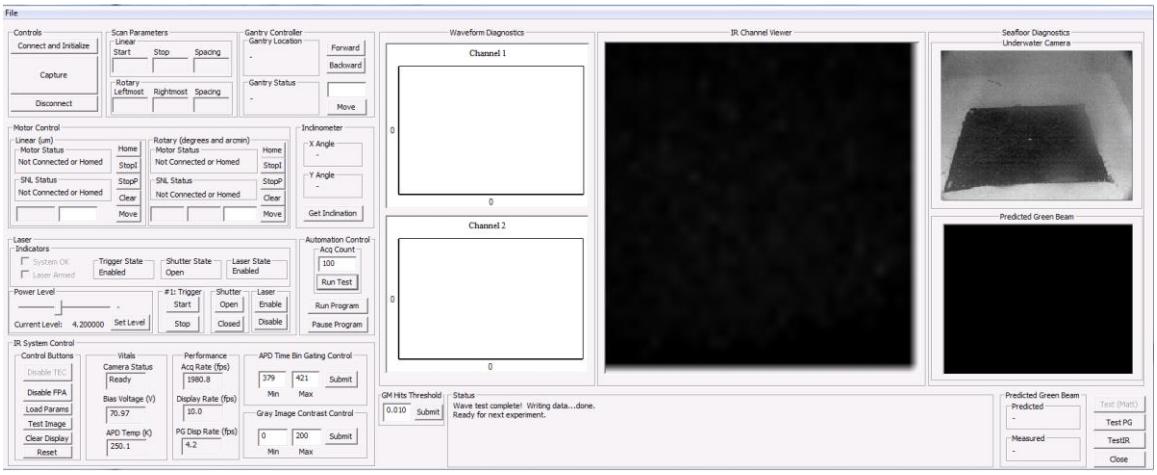

Figure 3.9: Software GUI to image intensity, underwater camera, calculated green beam location, green channel waveform returns, and overall system control

3.1.9 Other Electronics

There are many other electronics required for the setup and deployment of this system. Function generators were used for synchronization and trigger signals for the laser and GmAPD camera. LabJacks were used for wave measurement height voltage readouts and laser control. PMTs and a digitizer were coupled with the pathfinder for green beam waveform quantizing and resolving. These produced in air path lengths and in water path lengths that were used for the uncorrected incident beam ray tracing calculations. Finally, a computer with adequate hard drive storage and processing speed was used for running the required software, saving the data, and displaying the information for us. A timing diagram is shown in Figure 3.10 which illustrates the systems process.

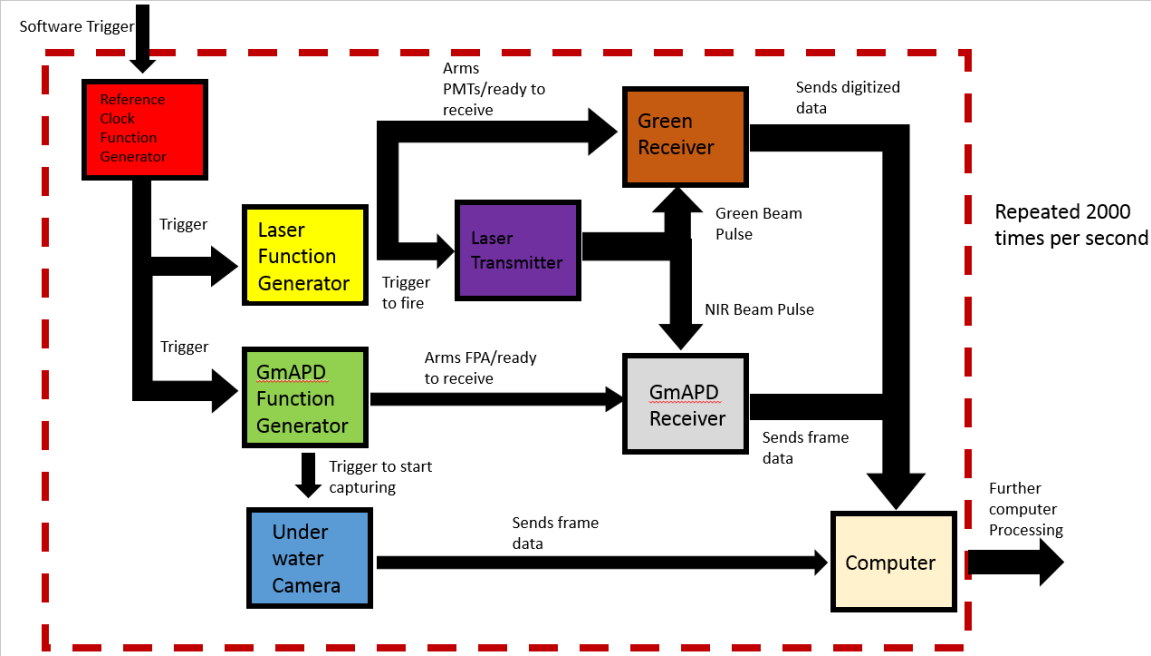

Figure 3.10: Timing diagram for the hardware triggering and component interaction

3.2 Experiment Parameters

Table 3.1 provides a summary of the relevant experimental parameters for our test setup.

Table 3.1: Experimental parameters for the research

Name	Value	Description
Length of one acquisition	100 ms	Amount of time surpassed to accumulate enough data to compute coordinates and surface normals
Acquisitions per Experiment	100	Total number of acquistions during a single experiment
Length of one experiment	10 seconds	Total length of time of an experiment
Number of Experiments	34	Total number of experiments run for data collection
Pulse Repetition Rate	2 kHz	Pulse rate of transmitters and receivers
GmAPD Receiver Frames per Acquisition	200	Number of averaged frames to create a single image
GmAPD Array Size	32x32 pixels	Size of GmAPD plane array for imaging

Green Beam Footprint	16x16 pixels	Size of green beam footprint, centered in the GmAPD footprint
NIR Beam Size	20cm x 18 cm	NIR beam size on water surface
Water Camera FPS	30	Frame rate of water camera
Wave Measurement Sampling Rate	1 kHz	Rate of quantized wave height measurements
Testing DCR	<100Hz	Final DCR of GmAPD receiver under testing conditions
Testing PDE	30%	Final PDE of GmAPD receiver under testing conditions
Laser Voltage	4.2V	Voltage of the LiDAR transmitter
NIR Optical Density Filters	1	Number of OD filters used for NIR transmitter

Every tenth of a second (100ms), an acquisition is taken. This composes of 200 frames of GmAPD data averaged into a single image, taking every third frame of the water camera data, and taking every 100th wave height measurement data. The water camera and wave height measurement images are used as the truth data, whereas the GmAPD image gets processed for sea surface DSM and normal calculations.

The GmAPD receiver DCR was computed by setting the experimental range gate and collecting data when no transmit beam was present, resulting in <100Hz DCR. The PDE was provided by PLI and can be changed via a sensitivity setting on the camera. This PDE setting was deemed the optimal point by PLI and our testing with respect to further DCR tradeoffs.

3.3 Sea Surface Characterization

Accurately characterizing the sea surface is pivotal to this effort since we want to assess how accurately a GmAPD receiver can model the water surface and then correct for it. There is surprisingly little literature regarding how to accurately capture wave structure, and most attempts simply use buoy GPS data over a period of time or take the

mean water level in post processing. Due to this difficulty, this research was only focused on the flat condition experiments and the process of converting the raw GmAPD data into a useful water DSM. This provides the foundation for future experiment analysis involving the collected wavy water data in the event better sea surface characterization techniques can be applied. To capture the wave height, we have three wave measurement height devices positioned in a triangular setup around near our imaging area as shown in Figure 3.5. While the data is not directly used in this research, these wave height devices can be used for future processing of the wavy water experiments. A sample readout of the wave height is shown in Figure 3.11.

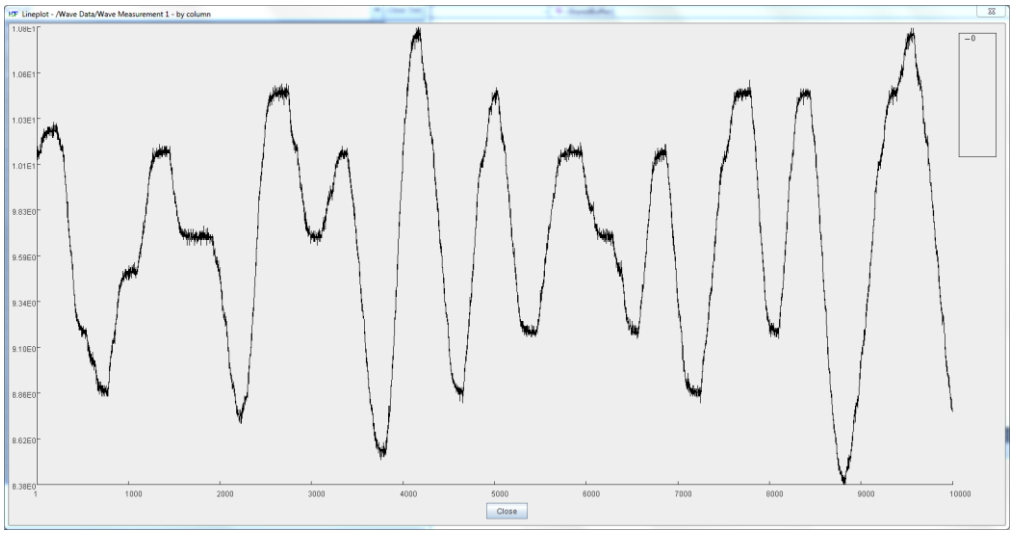

Figure 3.11: Sample wave measurement readout from one device

These devices were chosen since the waves being generated are low frequency with widths that extend beyond the range of the wave measurement devices. It would not be satisfactory for smaller, capillary waves since each wave measurement device could

be registering separate waves at distinct time instants. Nonetheless, it added some useful information for future water detailing.

3.4 System Calibration

Before testing could begin, certain calibrations to the overall system or individual parts had to be conducted in order to achieve accurate results. Each calibration is described in the following subsections.

3.4.1 Laser Drift

Through experimental testing, an interesting result consistently appeared which wasn't immediately obvious until data collection. When the laser is first fired, the LiDAR's measured range is too long by about 10 time bins (~.38m). This bias lasts for thirty seconds to one minute before going sharply in the opposite direction and such that the LiDAR's ranges are too short. After about six minutes of consistent firing, the ranging system stabilizes as shown in Figure 3.12.

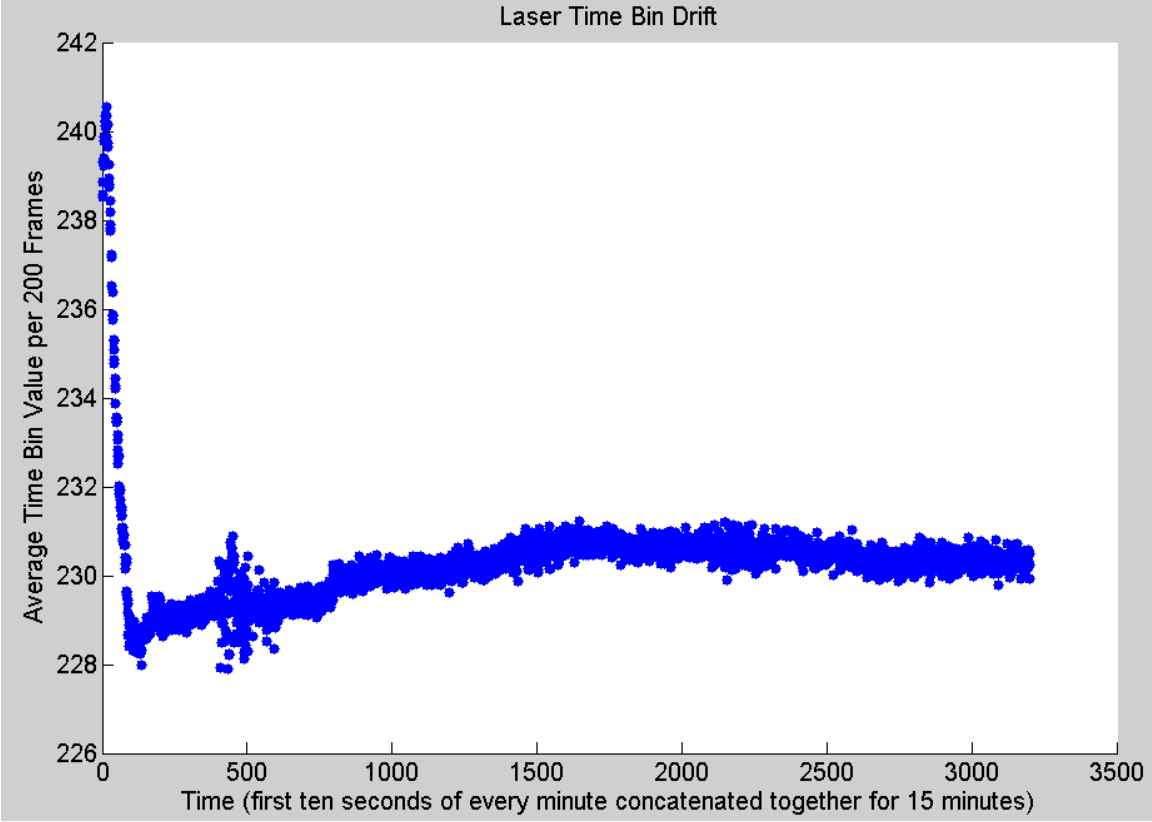

Figure 3.12: Graphical overview of the average time bin count per 200 frames over the course of 15 minutes.

From Figure 3.12, it is clear the system requires a warm-up time of several minutes to accurately record. The exact reason for this behavior is not yet known, but it was hypothesized that the temperature of the laser was the most likely cause. Regardless, it was an issue which had to be accounted for during testing.

3.4.2 Time Bin Gating

Knowing the approximate TOF gate times for your camera at a given distance facilitates producing accurate and noise free results. Since GmAPD cameras are extremely sensitive and vulnerable to errant dark counts, gating the range of acceptable time bin values removes a large percentage of potential noise and outliers. Therefore,

before collecting experimental data at the water tank, 20,000 frames were collected of raw time bin data and used to plot the histogram shown in Figure 3.13. From this histogram, the appropriate window was selected based off a time bin range which captured the most data (47 time bins) centered at the peak count.

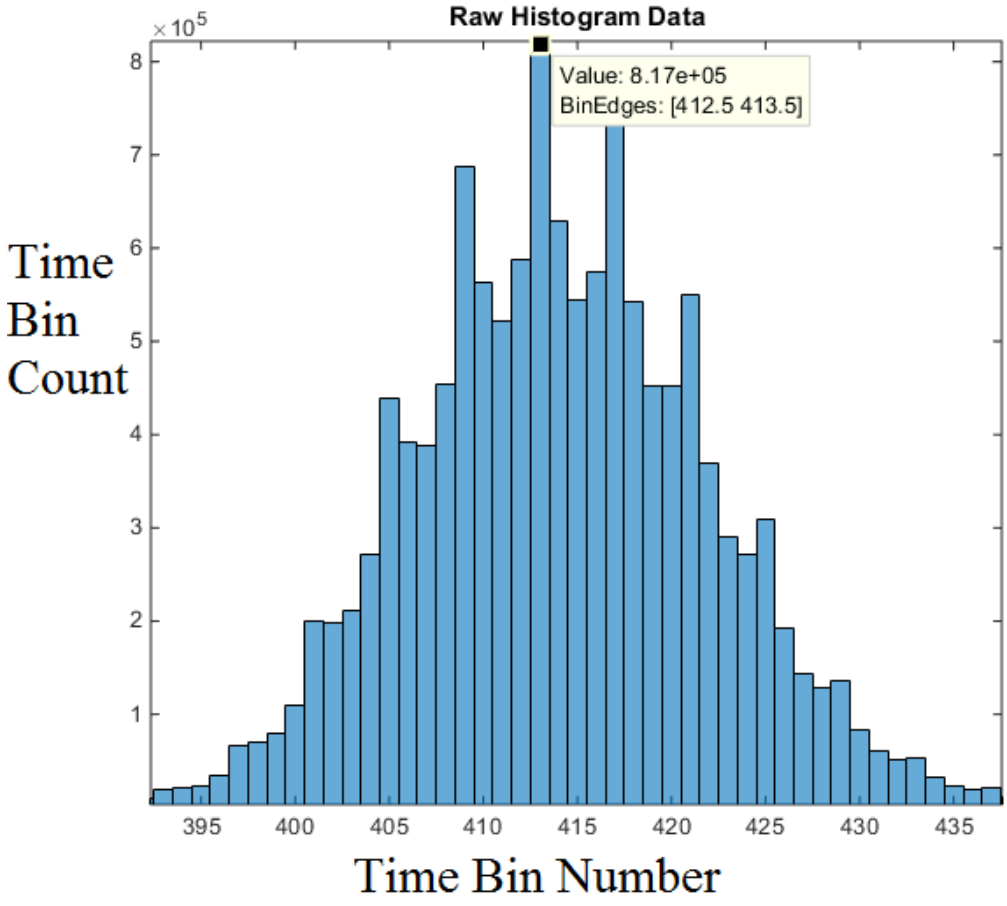

Figure 3.13: Raw histogram time bin count from 20,000 frames of data taken over the water tank and centered at time bin 413. This led to a time bin gate width of 390-436.

This time bin gate was used for the entire testing period. To illustrate the importance of this procedure, Figures 3.14, 3.15, and 3.16 show the effects that no time bin gating, improper time bin gating, and correct time bin gating have on $\overline{TOF}_{200}$ LiDAR point cloud coordinates when computed at a distance of 3.3m to a hard flat target with a time

bin center of 413 (identical conditions as Figures 2.13 and 2.14). Standard deviations and mean squared error (MSE) are shown for the z (depth) coordinate.

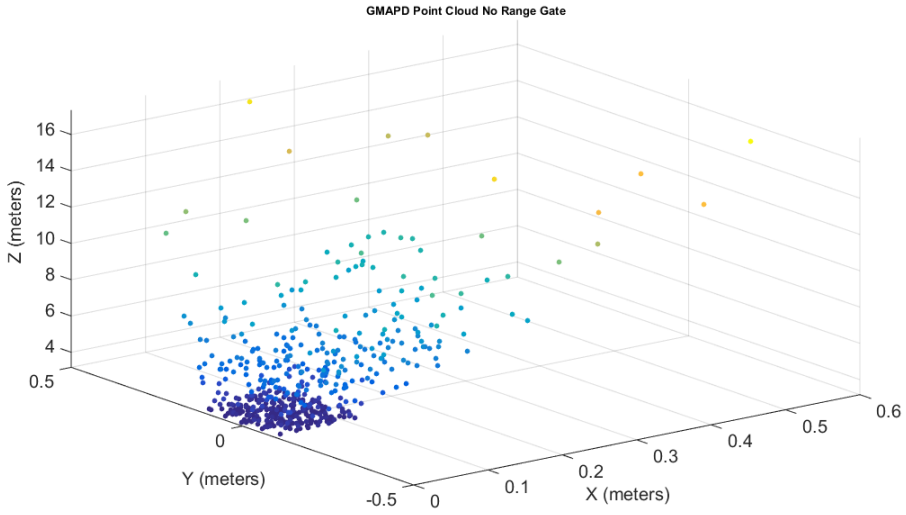

Figure 3.14: No time bin gating for $\overline{TOF}_{200}$. Std σ = 2.39m, MSE = 9.73 m^2

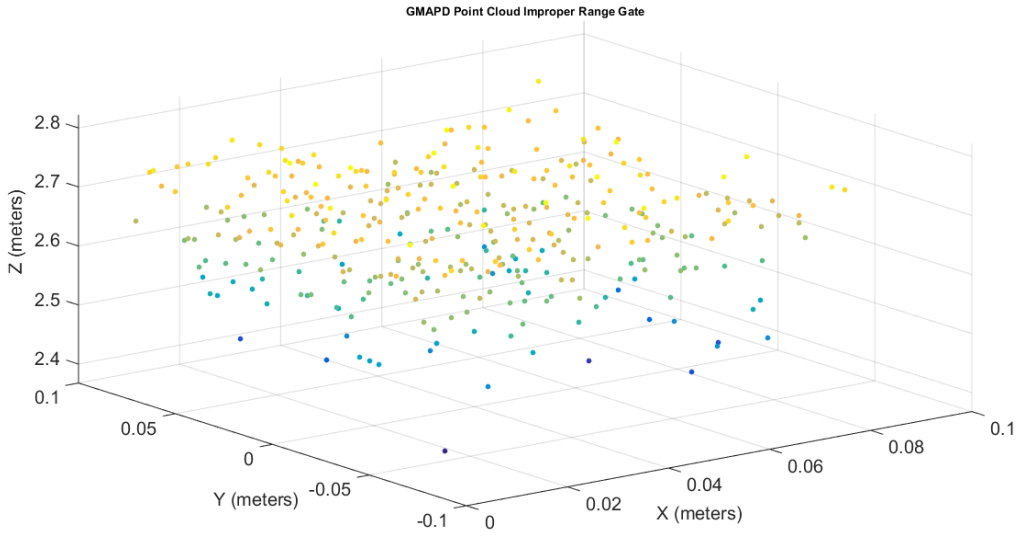

Figure 3.15: Improper time bin gating (time bin center selected at 400 instead of actual 413) for $\overline{TOF}_{200}$. Std σ = 6.6cm, MSE =40 cm^2

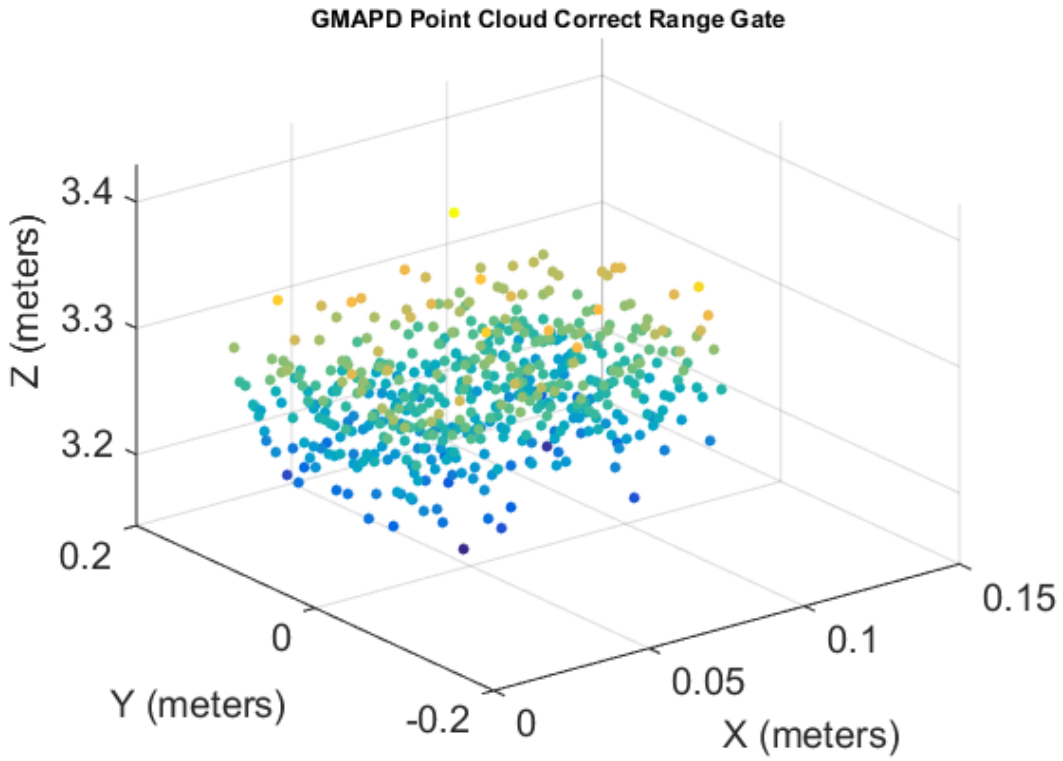

Figure 3.16: Proper time bin gating (time bin center 413). Std σ = 2.8cm, MSE =.001 cm^2. Very similar results to $\overline{TOF}_{200}$ coordinates from Figure 2.13 and 2.14

It can be clearly seen that if no time bin gating is used, the computed point cloud has higher standard deviation and mean square error. Improper time bin gating remedies the standard deviation, but keeps the large MSE (i.e the target will be either much further or closer depending on which direction the time bin gate is shifted from the correct center time bin). The correct time bin gate has good standard deviation and nearly perfect MSE, showing that a GmAPD can achieve great results if proper time bin gating is performed.

3.4.3 Time Calibration

Raw time bin values recorded by the GmAPD do not accurately report the true distance to an imaged object, and like all LiDAR systems, a GmAPD-based system must

be carefully calibrated to account for timing delays. Based on Equation 2-2, a time bin value of 400 would correspond to a distance of 15m to the water's surface in our experiments, but the hypotenuse distance to the water in our test setup is 3.3m. This ranging bias comes from the timing delay induced by cable lengths and other internal mechanisms that drastically affect the accuracy of the time bin count. A mere one nanosecond extra added in timing delay adds four time bins (15 centimeters, Equation 2-2) to the distance. Fortunately, determining this range bias is very straight-forward. Using a solid black background target, we computed average time bin values for the entire FPA at eight known distances and fit a linear regression to these data to compute slope and bias terms. An important parameter for the equation was that we only used TOF data for pixels that were very low intensity ($.005<I(i,j)<.02$). The reason for this constraint was to remove other bias induced by range walk, explained in the next section. The linear fit to our data is shown in Figure 3.17.

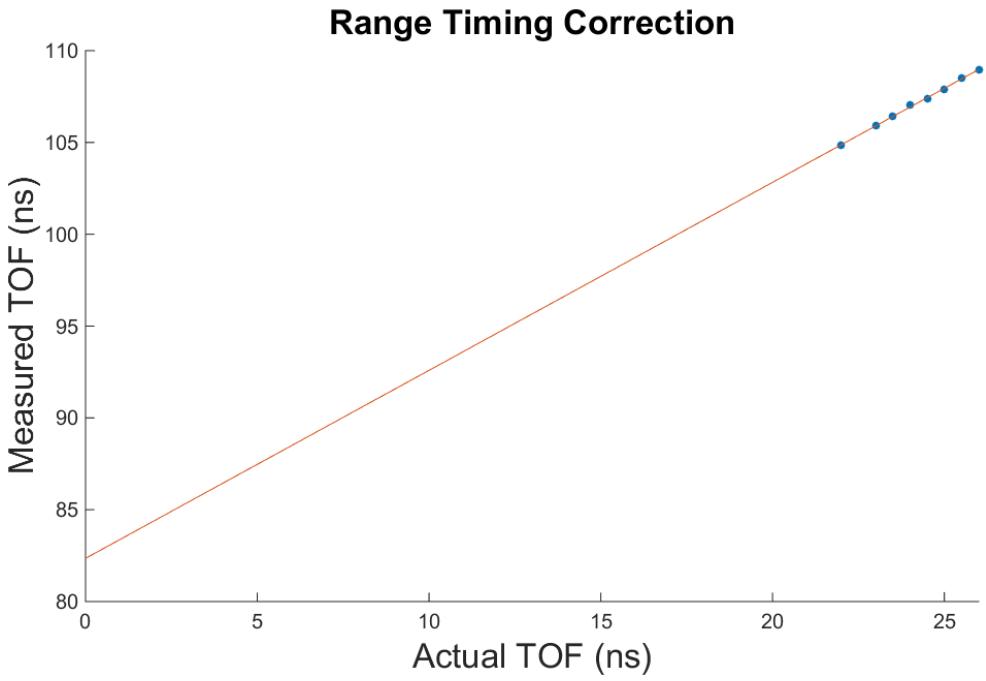

Figure 3.17: Linear regression used to estimate timing delay

And it's represented by Equations 3-1 and 3-2:

$$TOF_m^d = R_0 TOF_A^d + R_1 \tag{3-1}$$

$$TOF_m^d = \begin{cases} \frac{1}{t}\left(\sum_{\substack{0\le i<32\\0\le j<32}}\left(\frac{1}{f}\sum_{n=0}^{f} TOF_n^d(i,j)\right)\right), & .005 < I(i,j) < .02 \\ 0, & otherwise \end{cases} \tag{3-2}$$

where,

TOF_m^d	=	is the mean measured TOF for distance d in nanoseconds
R_0	=	is the slope, calculated to be 1.02379235588971
TOF_A^d	=	is the actual time of flight for distance d in nanoseconds
R_1	=	is the y intercept, calculated to be 82.3525317669176
f	=	is the number of frames to be integrated
$TOF_n^d(i,j)$	=	is the TOF value in frame n at pixel (i,j) for distance d
$I(i,j)$	=	is the intensity (Equation 2-1) at pixel (i,j)

Rearranging Equation 3-1 and solving for actual distance TOF_A yields the modified Equation 3-3:

$$TOF_A^d = \frac{TOF_m^d - R_1}{R_0} \tag{3-3}$$

Equation 3-1 states that for each given distance we measured, calculate the mean TOF value for all the mean TOF pixel values if and only if the associated intensity value for the pixel falls between the designated boundaries. Doing this for each of the eight distances and plotting the line results in the regression line seen in Figure 3.17.

3.4.4 Range Walk

An interesting phenomenon recently discussed in the literature [58] [59] is the issue regarding GmAPD pixel range walk, or intensity error. Essentially, the more intense a pixel is (the more hits it has over some frame integration period), the smaller the time bin values are (i.e. it appears closer). This has the effect of making brighter objects appear closer, creating a discernable error in images with varying intensities. This can be immediately seen in Figure 3.18, where five pieces of painters tape were placed perfectly flat on a black background, and the resulting point cloud data from reported time bins showed them as raised bumps.

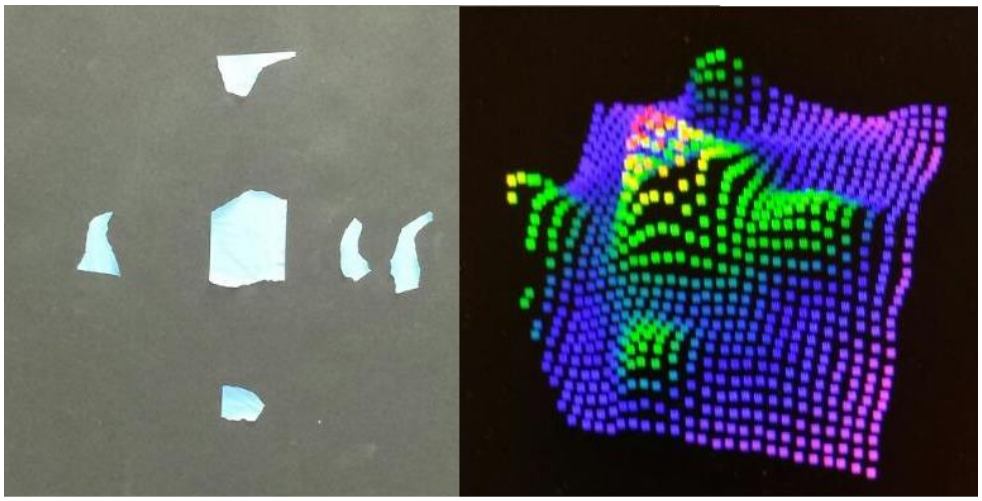

Figure 3.18: Painter's tape on a black background and the resulting point cloud. Higher intensity points were 5-10cm closer than background

Through knowledge of GmAPDs and recent literature on the topic published in [58] and [59], it became clear that the bias came from the reduced time it took for an avalanche within a GmAPD to start when more incident photons were present, as shown

in Figure 3.19. While these timing differences are on the scale of hundreds of picoseconds, it's enough of a difference to skew the pixel by a few time bins and make an object appear closer.

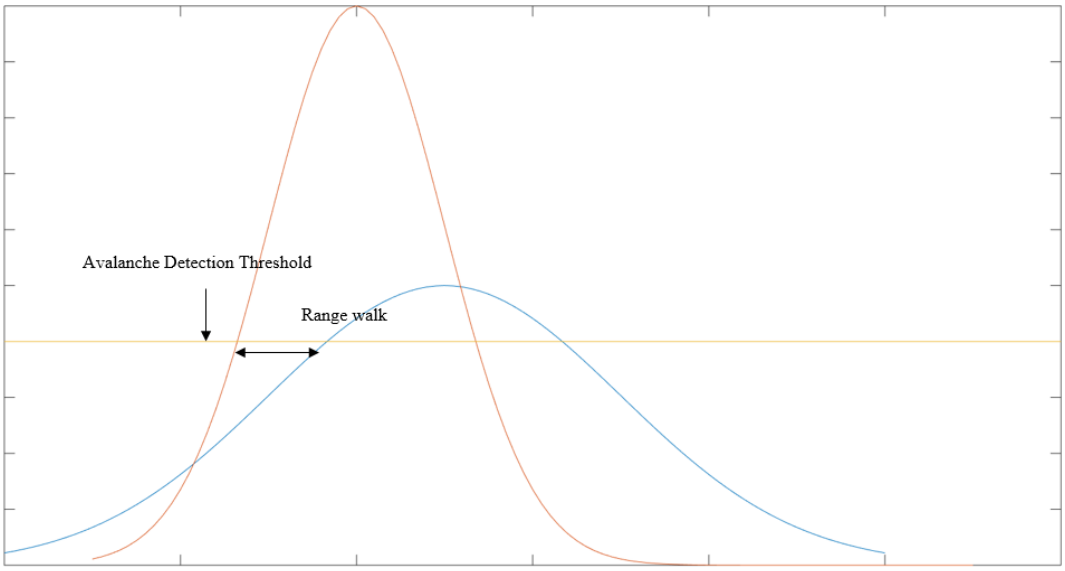

Figure 3.19: Illustration of range walk due to faster avalanche resulting from higher incident photons

To solve this issue, we used a similar setup presented in [58] and [59] to create a calibration curve for correction. We plotted the intensity of a pixel (Equation 2-1) versus it's time error T_e(i,j), which is calculated below in Equation 3-4:

$$T_e^d(i,j) = \left(\frac{1}{f}\sum_{n=1}^{f} TOF_n^d(i,j)\right) - TOF_m^d \tag{3-4}$$

where,

$T_e^d(i,j)$= is the time error in nanoseconds for distance d at pixel (i,j)
TOF_m^d = is the mean measured TOF for distance d in nanoseconds
$TOF_n^d(i,j)$ = is the TOF value in frame n at pixel (i,j) for distance d
f = is the number of frames to be integrated

$I(i,j)$ = is the intensity of a pixel (i,j) between [0 1]
H = is binary 0 or 1 dependent on if there was an avalanche for a frame

The time error calculated in Equation 3-4 shows how far away (in nanoseconds) the average TOF value for some pixel is from the already exact TOF value in computed with Equation 3-3. For lower intensities, this time error is nearly zero which is as expected since we use near zero intensities as our accurate reference point. However, as intensities increase, the relationship between $T_e^d(i,j)$and *I(i,j)* grows exponentially, as shown in the Figure 3.20.

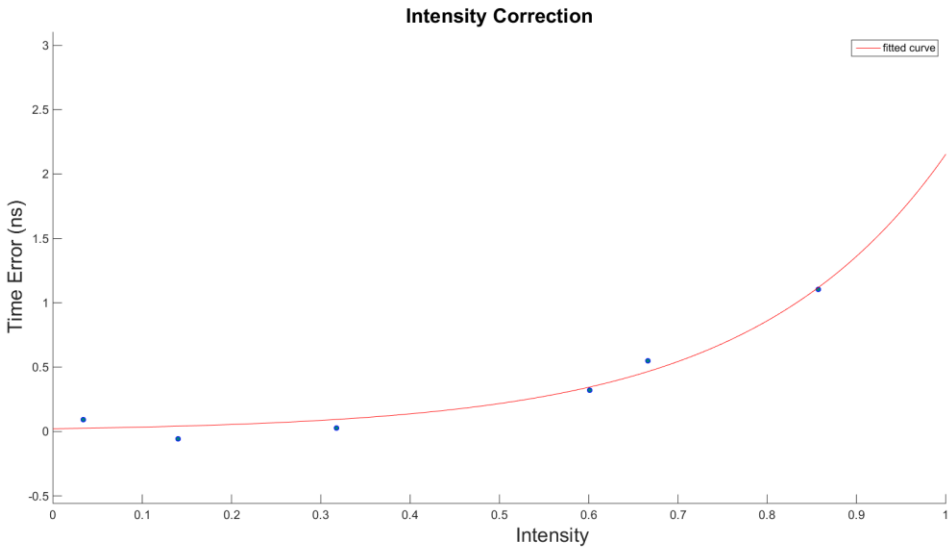

Figure 3.20: Exponential fit for intensity vs time error curve

Using an exponential regression fit, we can calculate $T_e^d(i,j)$for a pixel from its intensity. The line in Figure 3.20 is described as Equation 3-5:

$$T_e^d(i,j) = Ae^{(B*I(i,j))} \quad (3\text{-}5)$$

where,

$T_e^d(i,j)$= is the time error in nanoseconds for distance d at pixel (i,j)
$I(i,j)$ = is the intensity of pixel (i,j) (Equation 2-1) in the image
A = is the base coefficient, calculated to be 0.021900186808850
B = is the exponent coefficient, calculated to be 4.588548204216641

The final important point to be made about range walk is the choice of using lower intensity as the reference versus higher intensity. The logic behind it is rather simple: intensities which reached unity or near unity exhibited extremely high variance in time error behavior. This error could be caused by having one unity intensity pixel receive many more incident photons than another unity intensity pixel, causing additional range walk bias. However, since they are either already at or near unity, selecting the pixel which better represents a more 'accurate' unity intensity would become very difficult. With low intensity pixels, there was no such variance in time errors for a given intensity except at extremely low ($I(i,j) < .005$) intensities, likely due to DCR. This is easily bypassed through use of a simple threshold operation per image integration period as described in Section 2.2.5, allowing for a smooth and consistent performance without additional guesswork.

3.5 Initialization

As a final step before real time processing can begin, a coordinate system needs to be established which relates all the components of the system together. Moreover, once this coordinate system is established, the appropriate rotations and offsets need to be applied to the receiver components before the collection can begin. When all of this is in place, the data that is received can be accurately mapped to a consistent coordinate space for further imaging and processing.

3.5.1 Global Coordinate System

ALB systems generally produce data in a 3D Cartesian local geodetic frame (LGF) using Northing, Easting, and Down (NED) coordinates. The setup for this research mimicked that coordinate system by establishing a LGF reference frame within the water tank. Our global origin was a point in front of the pathfinder LiDAR and imaging area, as shown in Figure 3.21 and Table 3.2.

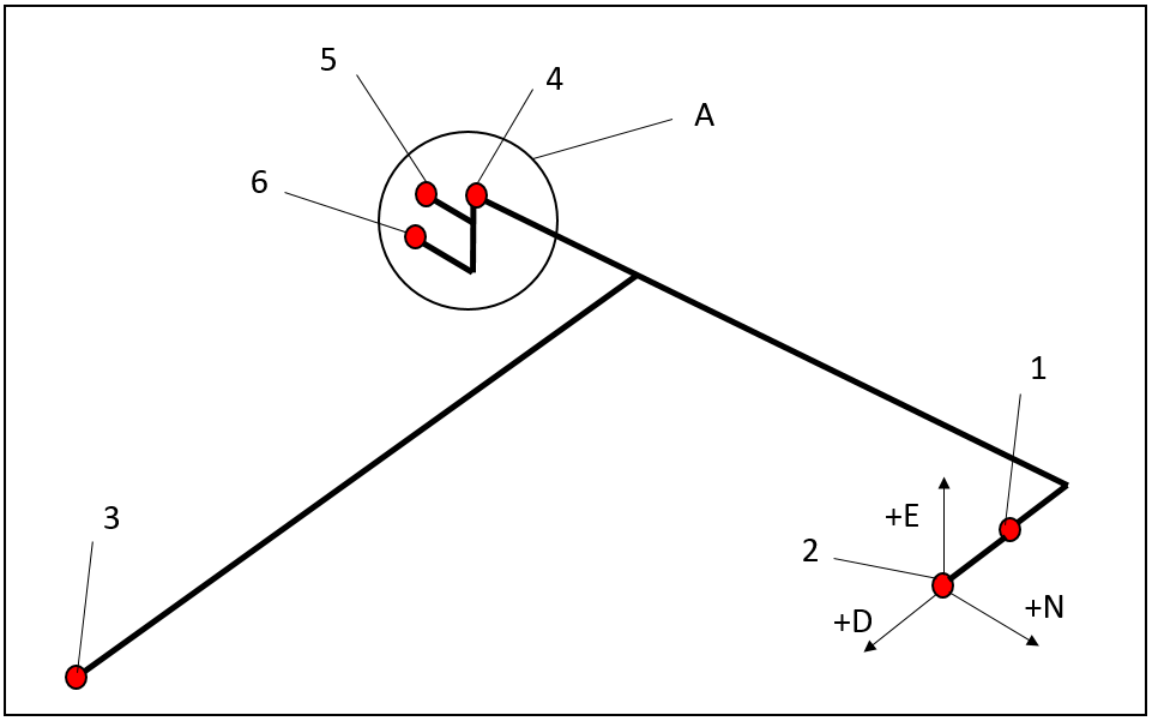

Figure 3.21: Global coordinate system depiction with axes

Table 3.2: Figure 3.21 point descriptions and LGF coordinate positions

Label	Name	LGF Coordinates (N,E,D)
A	Pathfinder Housing	N/A
1	Global Origin	(0, 0, 0)m
2	Water Surface	(0, 0, .81915)m

3	Grid Origin (also water tank floor)	(-2.843, 0, 6.963)m
4	LiDAR Fixture	(-3.909, 0, -2.41)m
5	LiDAR Origin	(-4.019, -.0983, -2.309)m
6	GmAPD Receiver Origin	(-4.013, -.3376, -2.292)m

The choice of the LGF origin was selected due to its close alignment with the pathfinder in the pointing axis, its static location, and precision of its marking. A plum bob and laser distance meter were used to set the global origin, which ended up being at a steel beam nearly perpendicular to the pointing axis (the x-axis). All further calculations would now be translated into the global coordinate system.

The procedure used to map the imaging area into the LGF is described in Equations 3-6, 3-7, and 3-8 [1].

$$\begin{bmatrix} N \\ E \\ D \end{bmatrix}_{LGF} = \left(R_z(\kappa) R_Y(\varphi) R_x(\omega) \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{IBF} \right) + \begin{bmatrix} N_T \\ E_T \\ D_T \end{bmatrix} \tag{3-6}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{IBF} = R_z(\Delta\kappa) R_Y(\Delta\varphi) R_x(\Delta\omega) \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{SBF} + \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} \tag{3-7}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{SBF} = R_z(\theta) R_Y(\phi_A) \begin{bmatrix} N_x(i,j) \\ N_y(i,j) \\ N_z(i,j) \end{bmatrix} l_a \tag{3-8}$$

where,

$LGF_{N,E,D}$	=	are the NED local geodetic frame points on the surface
N_T, E_T, D_T	=	are the NED coordinates of receiver
κ	=	is the aircraft heading
φ	=	is the aircraft roll
ω	=	is the aircraft pitch
$IBF_{X,Y,Z}$	=	are the inertial body frame points of the pathfinder
$\Delta\kappa$	=	is the bore sight alignment angle in κ
$\Delta\varphi$	=	is the bore sight alignment angle in φ
$\Delta\omega$	=	is the bore sight alignment angle in ω
ΔX	=	is the inertial to sensor body frame x-lever arm
ΔY	=	is the inertial to sensor body frame y-lever arm

ΔZ	=	is the inertial to sensor body frame z-lever arm
$SBF_{X,Y,Z}$	=	are the sensor body frame points of the GmAPD receiver
θ	=	is the scanning angle
ϕ_A	=	is the off-nadir angle
l_A	=	is the in-air path length
R_x, R_Y, R_z	=	are the rotation matrices about the x, y, and z axes
N_x, N_Y, N_z	=	are the normal vector components for the image space

This procedure describes the computation of LiDAR coordinates as a series of coordinate transformations. Mapping to NED coordinates requires first mapping the local sensor body frame of the GmAPD receiver into the inertial body frame of the LiDAR. From there, you map those coordinates into the LGF coordinates using the positioning offsets of the pathfinder relative to the global origin. In our experiment, most of the rotation angles are zero since this is a fixed location without variation in roll or heading. Removing the zero values modifies Equations 3-6, 3-7, and 3-8 into a much simpler form shown in Equation 3-9.

$$\begin{bmatrix} N \\ E \\ D \end{bmatrix}_{LGF} = \left(\left(R_z(\theta) R_Y(\phi_A) \begin{bmatrix} N_x(i,j) \\ N_y(i,j) \\ N_z(i,j) \end{bmatrix} l_a \right) + \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} \right) + \begin{bmatrix} N_T \\ E_T \\ D_T \end{bmatrix} \tag{3-9}$$

Table 3.3 lists the measured values for ΔX, ΔY, ΔZ, N_T, E_T, and D_T for our experiment setup.

Table 3.3: Measured positional offsets

Name	Value
$\Delta \boldsymbol{X}$	.0157sin(ϕ_A)m
$\Delta \boldsymbol{Y}$	-.23932m
$\Delta \boldsymbol{Z}$	.0157cos(ϕ_A)
$\boldsymbol{N_T}$	-4.018915m
$\boldsymbol{E_T}$	-0.098298m

$\boldsymbol{D_T}$	-2.3086314m

The last two remaining missing parameters are the normal vector components of the image space coordinates and the in-air path length (Equation 2-2). The latter is calculated by our GmAPD receiver and accompanying processing algorithms, whereas the former is precomputed as a look up table which will be described in the next section.

3.5.2 Image Space Coordinates

Image space coordinates are the x and y positions of the pixels on the FPA of our camera relative to the principal point (the intersection of the optical axis with the FPA), and a z coordinate defined as the focal length of the camera (f). By normalizing the coordinates by f and scaling by the in-air path length, we can compute object space coordinates on the surface of the water. Calculating the image space points for every pixel on the array is done beforehand since the FPA does not increase or decrease in size throughout the experiment. Figure 3.22 depicts this image space and object space connection.

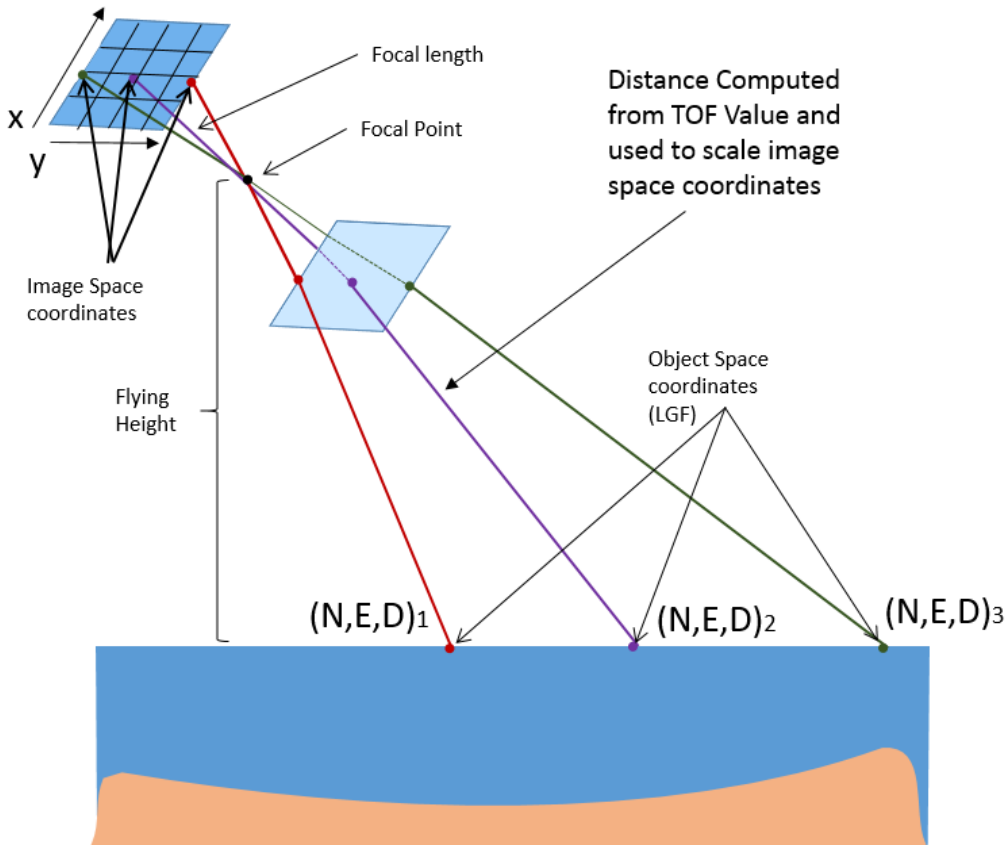

Figure 3.22: FPA image space coordinates being scaled to object space coordinates on the water surface

The 3D camera used in this research has an FPA consisting of a 32 x 32 array of GmAPDs with 100 micrometer pitch, 3.2 millimeter array height and width, and the GmAPD pixel lies directly in the center of each region. Since the sensor body frame origin is the lens directly in front of the array, the z component of the coordinates is simply the focal length *f*. We iterate through every pixel within the FPA and calculate each pixel's image space (x,y,z) coordinate, as shown in Equation 3-10

$$\begin{bmatrix} x_{(i,j)} \\ y_{(i,j)} \\ z_{(i,j)} \end{bmatrix} = \begin{bmatrix} p(15.5 - i) \\ p(j - 15.5) \\ f \end{bmatrix}, \qquad 0 \leq i < 32, \qquad 0 \leq j < 32 \tag{3-10}$$

where,

$x_{(i,j)}, y_{(i,j)}, z_{(i,j)}$	=	are the x, y, and z coordinate at pixel location (i,j)
f	=	is the focal length of the receiver
p	=	is the pitch between pixels

A modification to this formula for this research was to divide out the focal length and scale everything accordingly, leading to larger x, y, and z values for better clarity and mathematical precision. This gives the following modified Equation 3-11

$$\begin{bmatrix} x_{(i,j)} \\ y_{(i,j)} \\ z_{(i,j)} \end{bmatrix} = \begin{bmatrix} \frac{p(15.5 - i)}{f} \\ \frac{p(j - 15.5)}{f} \\ 1 \end{bmatrix}, \qquad 0 \leq i < 32, \qquad 0 \leq j < 32 \tag{3-11}$$

The focal length of the system is 50 millimeters, so dividing by this gives us much closer to integer values than what would otherwise have been used. The final step for the look up table is to obtain the normalized vector for the coordinate so that when the in-air path length is used to scale it, it maps directly to some point on the surface. Note: scaling by the in-air path length provides the coordinates in the sensor body frame and not the ultimate local geodetic frame. After scaling, Equation 3-9 shows the appropriate rotation and offset correction that needs to be applied in order to translate the sensor body frame coordinates into NED coordinates.

Normalizing the coordinate vector is simple, shown in Equations 3-12, 3-13, 3-14, and 3-15 (*z(i,j)* is replaced with 1 for simplicity).

$$N = \sqrt{{x_{(i,j)}}^2 + {y_{(i,j)}}^2 + 1}, \qquad 0 \leq i < 32, \qquad 0 \leq j < 32 \tag{3-12}$$

$$N_x(i,j) = \frac{x_{(i,j)}}{N}, \qquad N_y(i,j) = \frac{y_{(i,j)}}{N}, \qquad N_z(i,j) = \frac{1}{N}, \qquad \text{(3-13;3-14;3-15)}$$

where,

N_x, N_Y, N_z = are the normal vector components for pixel (i,j)
N = is the norm calculated for pixel (i,j)

These normal vector components for each coordinate are stored in a lookup table and are used for all future coordinate computations.

3.6 Real Time Processing

With the initialization of the coordinate system and image space coordinates established, the actual processing procedure can begin. Experiments last 10 seconds and run through 100 acquisitions, with each acquisition processing lots of different information and performing thousands of operations. For my research, the real time processing can be broken up into three distinct components: collecting the data, processing the data, and then displaying and saving the data.

3.6.1 Collecting the Data

The first part of data processing involves the collection of the data using the GmAPD receiver and doing some preprocessing. Range gating of the receiver is combined with the thresholding and averaging of the accumulated frame data that is described in Section 2.2.5. This immediately reduces the noise characteristics of the system by eliminating many of the DCRs that appear due to methods outlined in Section 2.2.3. Once this data has been collected and preprocessed, it is passed on to the main computing portion of the code for further processing.

3.6.2 Processing the Data

The processing of the data follows a defined structure in order to ensure accurate results are achieved. This processing step can be broken down into three additional phases: correction, filtering, and regression plane normal computations. These three phases take place ten times a second for ten seconds, so the computations in addition to the transition between phases needs to be quick in order to meet real time processing needs. Computations were done using C++ code and aided immensely through the help of open linear algebra packages Armadillo[60] and OpenBLAS[61], without which this processing would never have been able to perform in real time. This sequence is shown in Figure 3.23.

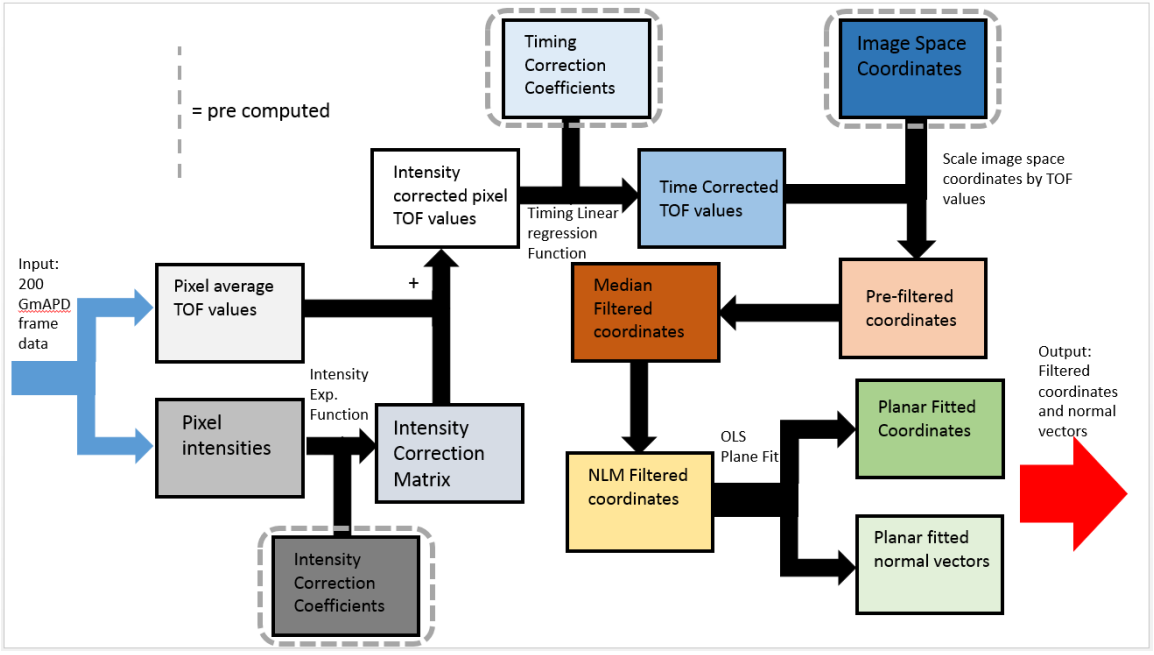

Figure 3.23: Processing diagram for incoming GmAPD frame data

3.6.2.1 Correction

This phase is the correcting of the averaged time bin data. The TOF data needs to be intensity corrected (Equation 3-3), range corrected (Equation 3-5), and then finally image space corrected (Equation 3-9). It is imperative that any bad values which are detected in one stage of correction do not further propagate to later stages or other processing phases. This could be disastrous for the NLM filter or regression planes which rely on near neighbor computations. Proper value checking is implemented in the code in order to adequately detect and nullify illegitimate values.

3.6.2.2 Filtering

This phase applies the filtering methods outlined in Section 2.3 to the now corrected data. The small windowed median filter, typically of size 3x3 or 5x5, is applied first in order to remove the extremes without performing significant smoothing. This prepares the data for the fast NLM filter which drastically improves the image by removing lots of the noise by using a 15 x 15 search window and patch size of 7. This two-step filtering process consumes the majority of the processing time, yet it is a vital step in order to obtain a smooth and accurate image.

3.6.2.3 Regression Plane Normal Computations

The final processing phase is the localized regression plane for each coordinate in the array. The regression plane window size used is 3x3 or 5x5 to prevent over smoothing of the normal calculation while maintaining some spatial locality. Once the regression plane is computed for each coordinate, the normal vectors are given by the solution coefficients and stored into a results array structure which will be written to a file and sent to PCL for display.

3.6.3 Displaying and Saving the Data

Once all the computations have been done and stored into their results structure, the data can be sent to PCL for visualization purposes and to the HDF5 file manager for writing. PCL visualizes all the points in the FPA while displaying the 256 green beam pixels (referred to as beamlets) normal vectors. This provides a quick and easy visual output so that results can be immediately known without having to open up and process the saved HDF5 file. Moreover, it also acts a quick warning indicator of potential malfunctions since the data will not display correctly (or at all) if there are underlying issues affecting some part of the real time processing.

3.7 Error Characterization

To assess the results and potential improvement caused by the proposed methodology in this paper, there has to be a metric to characterize the error between what DSM is being computed and what the actual surface resembles. Two such error analyses are proposed to check the validity and utility of the research. These are: (1) measuring the variance, bias, and mean squared error (MSE) between the computed GmAPD surface and the theoretical flat plane, and (2) comparing the error between the normal vector computation of the GmAPD surface and the theoretical perfectly vertical normal vectors.

3.7.1 Variance, Bias, and MSE

When constructing the DSM, coordinate error can exist in three major ways: variance, bias, and MSE. These error metrics can be applied to a surface generated from a single acquisition (a holistic measure of surface accuracy) or on a per pixel basis across multiple acquisitions (individual pixel variation), shown in Equations 3-16, 3-17, 3-18, 3-19, 3-20, and 3-21.

$$MSE_S^n = (Bias_S^n)^2 + Var_S^n \quad (3\text{-}16)$$
$$Bias_S^n = E[D_n] - D_{true} \quad (3\text{-}17)$$
$$Var_S^n = E[(D_n - E[D_n])^2] \quad (3\text{-}18)$$

$$MSE_{P(i,j)} = Bias_{P(i,j)}{}^2 + Var_{P(i,j)} \quad (3\text{-}19)$$
$$Bias_{P(i,j)} = E\left[D_{i,j}\right] - D_{true} \quad (3\text{-}20)$$
$$Var_{P(i,j)} = E\left[\left(D_{i,j} - E\left[D_{i,j}\right]\right)^2\right] \quad (3\text{-}21)$$

where,

D_n	=	are the computed down values for a surface from acquisition *n*
$D_{i,j}$	=	are the computed down values for pixel *(i,j)* from all acquisitions
D_{true}	=	is the actual down value
MSE_S^n	=	is the MSE for a surface from acquisition *n*
$Bias_S^n$	=	is the bias for a surface from acquisition *n*
Var_S^n	=	is the variance for a surface from acquisition *n*
MSE_P	=	is the MSE for pixel *(i,j)* across all acquisitions
$Bias_P$	=	is the bias for pixel *(i,j)* across all acquisitions
Var_P	=	is the variance for pixel *(i,j)* across all acquisitions

The first error is the bias, or how far off in depth distance (Down) the computed surface is from the theoretical flat surface. This error is likely the result of imprecise calibration measurements, specifically the time calibration, due to distances used for the linear regression values being not exact. It would be easy and simple to reduce this error with more accurate pre-experiment measurements. The surface bias shows the depth error on average between the computed surface and the theoretical surface, whereas the per pixel bias indicates how far off a pixel at some location is from the theoretical surface when averaged across all acquisitions.

The second error is the variance of the depth coordinate. Surface variance indicates how far off from perfectly flat the computed surface structure is. Per pixel

variance shows the variance of a single pixel across all acquisitions. This error can be the result of many different variables, including filtering, range gating, pixel intensity, improper geometric calibration/offsets, and innate GmAPD technology limitations. All of these factors must be considered and accounted for when attempting to correct for this variance error.

The third is the MSE, which is the error combination between the bias and variance. This will show the total error for each pixel and surface in the constructed water DSM.

Finally, Equations 3-16 – 3-18 showed that MSE_S, $Bias_S$, and Var_S are all computed for a single surface that is generated from one acquisition. Since many acquisitions are taken during the experimentation, dozens of different error values are calculated for the multitude of acquisition surfaces. We aggregate the absolute value of these errors and report the average MSE, bias, and variation across all acquisitions, shown in Equations 3-22, 3-23, and 3-24.

$$\overline{MSE_S} = \frac{1}{a}\sum_{n=1}^{a}|MSE_S^n| \tag{3-22}$$

$$\overline{Bias_S} = \frac{1}{a}\sum_{n=1}^{a}|Bias_S^n| \tag{3-23}$$

$$\overline{Var_S} = \frac{1}{a}\sum_{n=1}^{a}|Var_S^n| \tag{3-24}$$

where,

$\overline{MSE_S}$ = is the average MSE across all acquisitions
$\overline{Bias_S}$ = is the average bias across all acquisitions
$\overline{Var_S}$ = is the average variance across all acquisitions
a = is the number of acquisitions

3.7.2 Normal Vector Error

The secondary error metric will be the comparison between the computed normal vectors at the surface of the water and the theoretical perfectly vertical normal vector. Each pixel's normal vector and a surface normal vector (the average of all pixel normal vectors) will be compared separately to the theoretical normal. The angular deviation equation used to compare these normal vectors will be the standard vector cosine Equation shown in 3-25 and 3-26.

$$\theta_S^n = \cos^{-1}\left(\frac{\vec{T}\cdot\vec{S}_n}{\|\vec{T}\|\|\vec{S}_n\|}\right)\left(\frac{180}{\pi}\right) \tag{3-25}$$

$$\theta_{P_{(i,j)}} = \cos^{-1}\left(\frac{\vec{T}\cdot\vec{P}_{(i,j)}}{\|\vec{T}\|\|\vec{P}_{(i,j)}\|}\right)\left(\frac{180}{\pi}\right) \tag{3-26}$$

where,

θ_S^n	=	is the average angular deviation in degrees for acquisition surface *n*
$\theta_{P_{(i,j)}}$	=	is the angular deviation in degrees for pixel *(i,j)* for all acquisitions
$\vec{T}$	=	is the perfectly vertical normal vector [0 0 -1]
$\vec{S}_n$	=	is the average surface normal vector for acquisition surface *n*
$\vec{P}_{(i,j)}$	=	is the pixel surface normal vector for pixel *(i,j)*

Like Equations 3-22 – 3-24, Equation 3-25 will be averaged across all acquisitions from our experiments to produce a single error metric, shown in Equation 3-27.

$$\bar{\theta}_S = \frac{1}{a}\sum_{n=1}^{a}\theta_S^n \tag{3-27}$$

where,

$\bar{\theta}_S$ = is the average surface angular deviation across all acquisitions
a = is the number of acquisitions

3.8 Summary

Chapter 3 provided a thorough understanding of the testing environment and methodology that was used for this research. It further explained essential calibration steps which were required for the system and provided progression through the differing processing steps from data collection to output. It ended with a brief description of the error analysis techniques that would be incorporated for measuring the overall accuracy of the water DSM. Chapter 4 will provide the results and analysis for the experiments.

CHAPTER 4

RESULTS

4.1 Flat Hard Target

It is important to first see how the system is capable of performing in a best case scenario situation. This entails imaging a hard flat black target, performing the necessary processing steps, and then showing the resulting normal vectors and errors. The conditions and data used for this experiment are identical to the previous flat hard target cases shown in Figures 2.13, 2.14, 2.16, 2.17 and 3.14-3.16. Table 4.1 shows the parameters of the experiment and Figure 4.1 shows a single acquisition $\overline{TOF}_{200}$ GmAPD surface data on top of a theoretical perfect flat plane placed at the exact imaging distance. Figure 4.2 shows a rotated view of Figure 4.1 with surface normal vectors also displayed, and Table 4.2 shows the resulting errors.

Table 4.1: Hard Target Parameters

Name	Value
Distance	3.3m
PRR	2 kHz
Number of Acquisitions	50
Frames Per Acquisition	200
Acquisition Length	100ms
Off-Nadir Angle (ϕ_A)	0 degrees

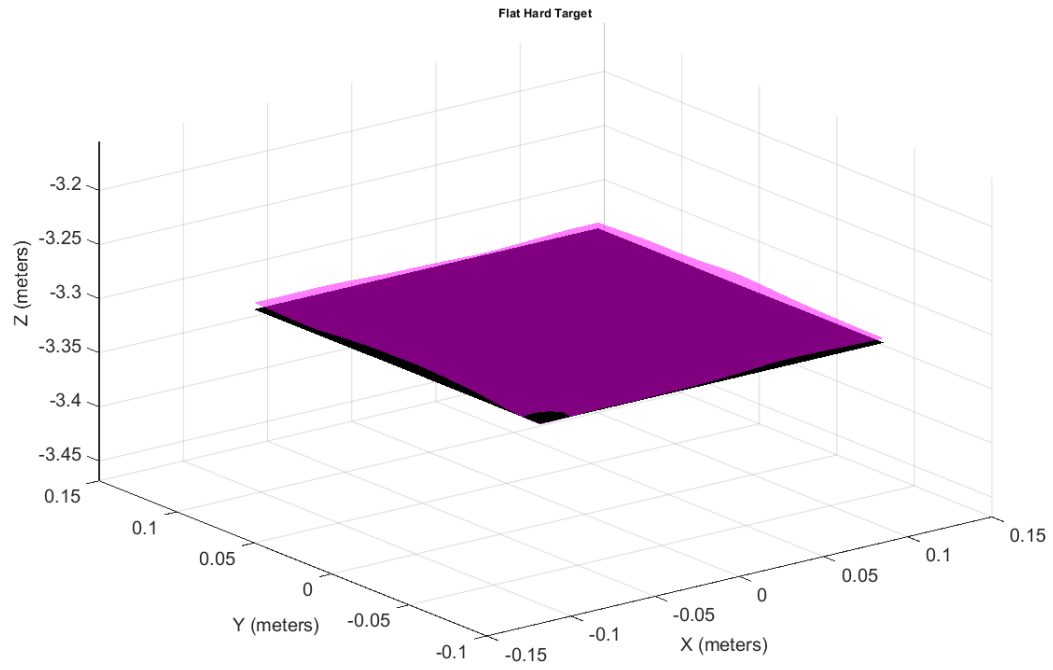

Figure 4.1: Flat surface (black) superimposed beneath GmAPD flat hard target surface (pink) generated from single acquisition $\overline{TOF}_{200}$ surface coordinates.

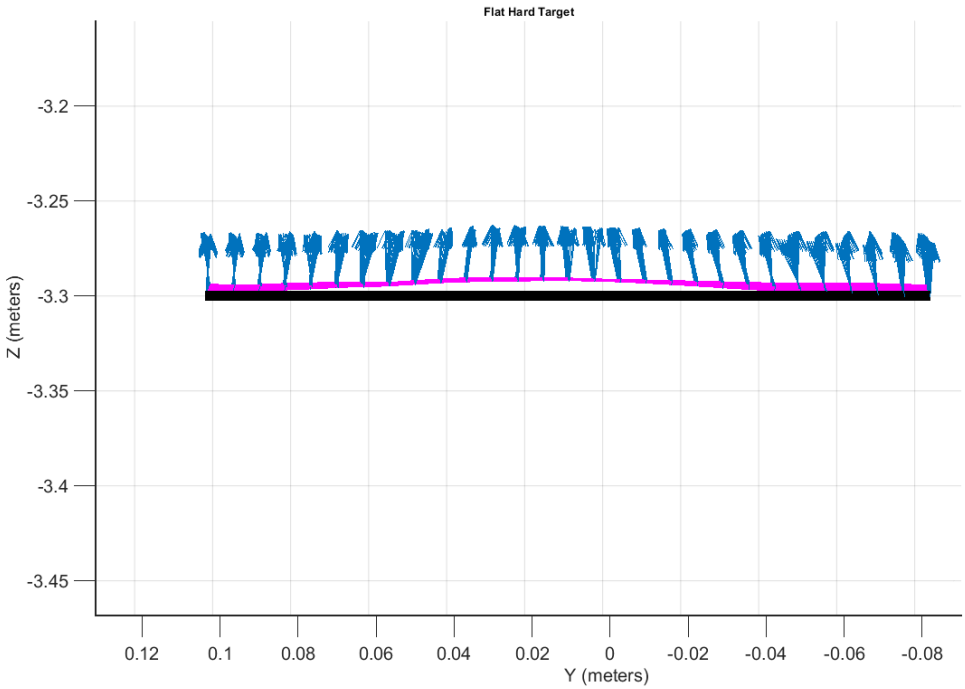

Figure 4.2: Side view of superimposed plane, GmAPD surface, and normal vectors from a single acquisition $\overline{TOF}_{200}$ flat hard target surface coordinates.

Table 4.2: Hard Target Average Surface Errors

Name	Value
$\overline{Bias_S}$	-.007m
$\overline{Var_S}$	.00001387m^2
$\overline{MSE_S}$	.00008233m^2
$\overline{\theta}_S$	5.54 degrees

As shown in Figures 4.1, 4.2, and Table 4.2, the results are generally excellent for coordinates computed from a flat hard target. Very little error in elevation with minor angular error for the normal vectors. The target did appear slightly closer than 3.3m, but that could be due to minor geometric calibration error.

The per pixel error metrics show additional pixel variability, illustrated in Figures 4.3, 4.4, 4.5, and 4.6.

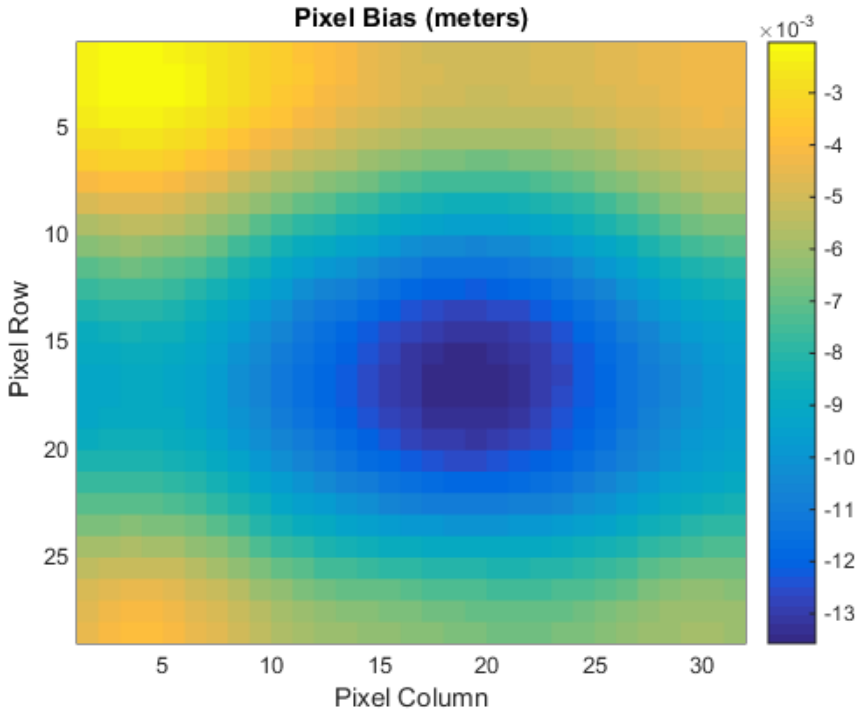

Figure 4.3: Per pixel bias ($Bias_{P(i,j)}$) across 50 acquisitions for flat hard target

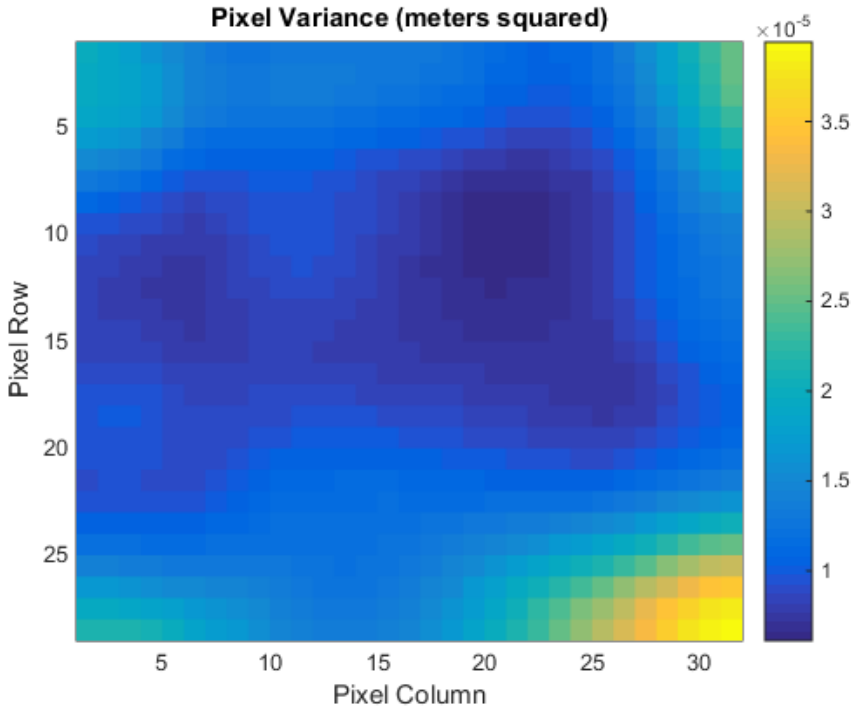

Figure 4.4: Per pixel variance ($Var_{P(i,j)}$) across 50 acquisitions for flat hard target

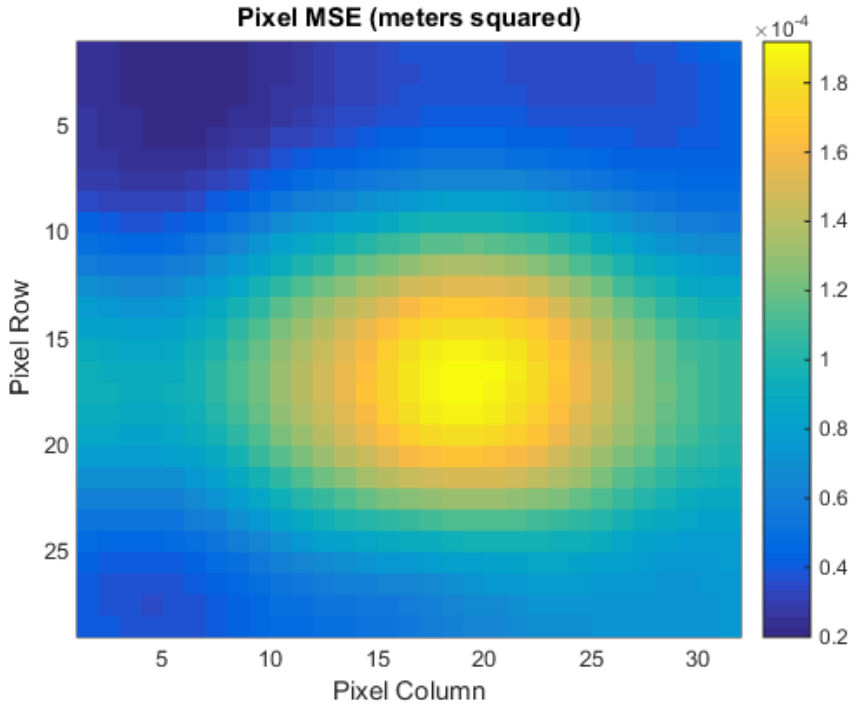

Figure 4.5: Per pixel MSE ($MSE_{P(i,j)}$) across 50 acquisitions for flat hard target

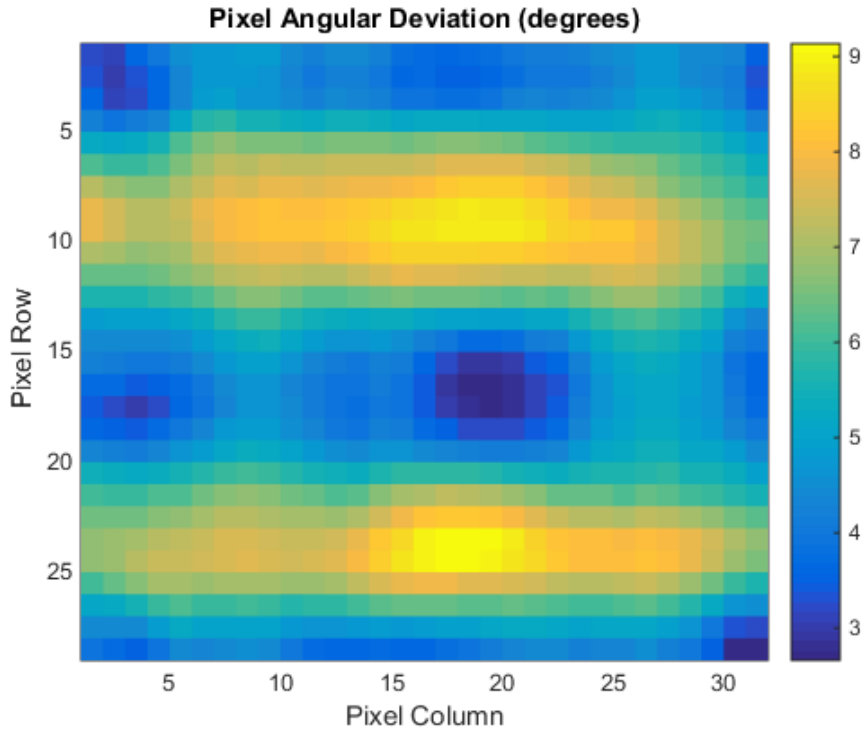

Figure 4.6: Per pixel angular deviation $\theta^{D}_{P(i,j)}$ across 50 acquisitions for flat hard target

The per pixel performance shows equally impressive results, similar overall to the surface average statistics. However, it is apparent that similar patterns form in the FPA for most of the metrics, potentially indicating the area on the FPA where the laser was centrally located and/or further showing the overall non uniformity of the FPA. With this best case scenario in mind, we will now observe how imaging calm water in a mock ALB system compares.

4.2 Calm Water

Four calm water experiments were conducted in the water tank. Without any waves or fans running, the pool is completely flat and allowed us to test the different behavior exhibited by water versus a hard target. Four experiments were run on the calm

water, with each experiment composing of 100 acquisitions. Table 4.3 shows the parameters of the four experiments.

Table 4.3: Calm Water Parameters

Name	Value
Hypotenuse Distance	3.3m
Vertical Distance	3.1m
PRR	2 kHz
Experiments	4
Acquisitions Per Experiment	100
Frames Per Acquisition	200
Acquisition Length	100ms
Off-Nadir Angle (ϕ_A)	19.5 degrees

Figures 4.7 and 4.8 show a surface of a single acquisition ($\overline{TOF}_{200}$). It is placed over the theoretical flat plane of the water surface. Table 4.4 indicates the error parameters across all acquisitions.

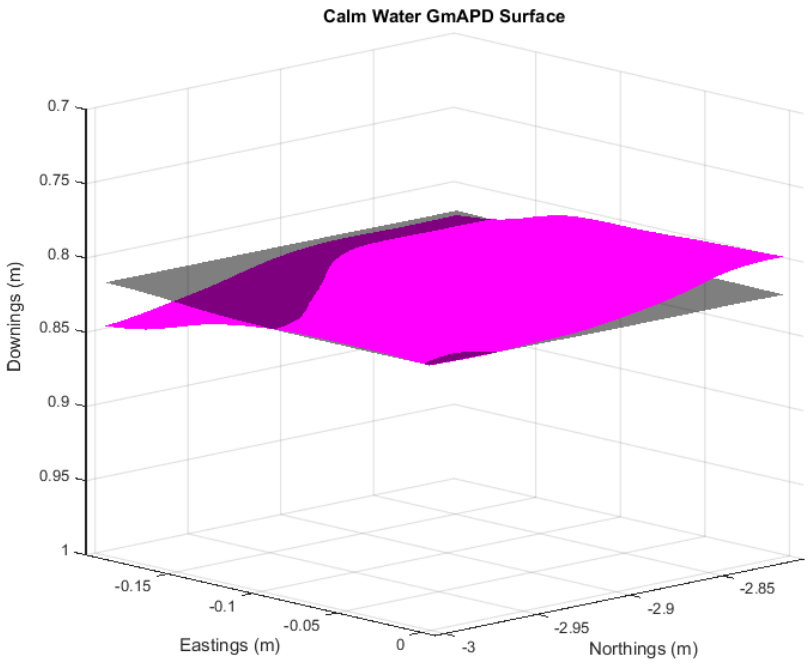

Figure 4.7: Flat surface (black) superimposed on GmAPD water surface (pink) generated from single acquisition $\overline{TOF}_{200}$ surface coordinates.

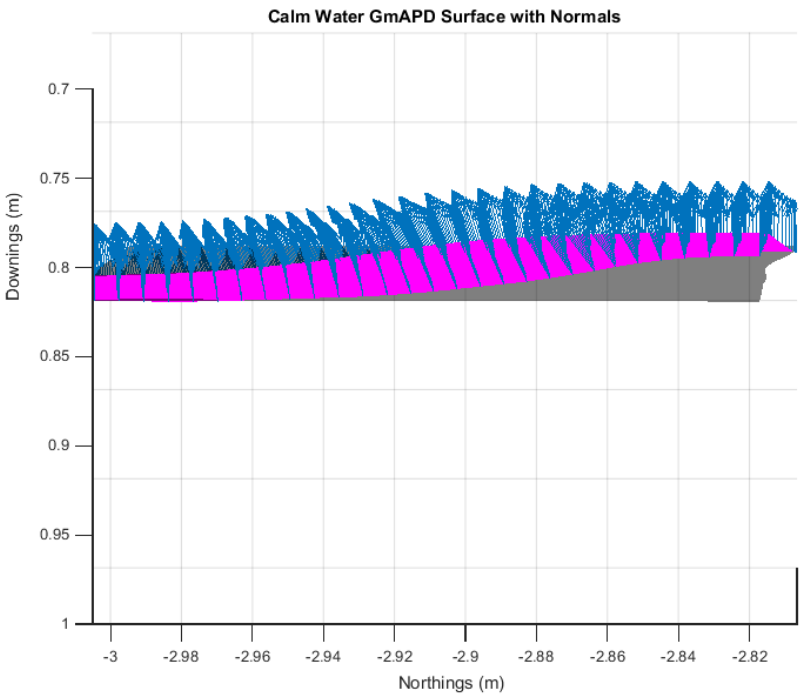

Figure 4.8: Side view of superimposed plane, GmAPD surface, and normal vectors from a single acquisition $\overline{TOF}_{200}$ calm water surface coordinates.

Table 4.4: Calm Water Surface Error

Name	Value
$\overline{\boldsymbol{Bias_S}}$	.00612m
$\overline{\boldsymbol{Var_S}}$	.000134m^2
$\overline{\boldsymbol{MSE_S}}$	.00024m^2
$\overline{\boldsymbol{\theta_S}}$	11.196 degrees

The error is slightly worse in the calm water case when compared to the flat hard target case. Overall average surface bias is similar to the hard target case, with variance and MSE each being around an order of magnitude worse (although still very small). The average normal vector is roughly two times worse than the flat target case, indicating that the surface construction of the water was not as accurate as the flat hard target. This could be due to the added geometric offsets that may have been improperly measured or a result of the intrinsic reduced reflectivity from the water surface.

The same analysis is performed on a per pixel basis, with Figures 4.9-4.12 showing bias, variance, MSE, and angular deviation for each pixel in the GmAPD FPA.

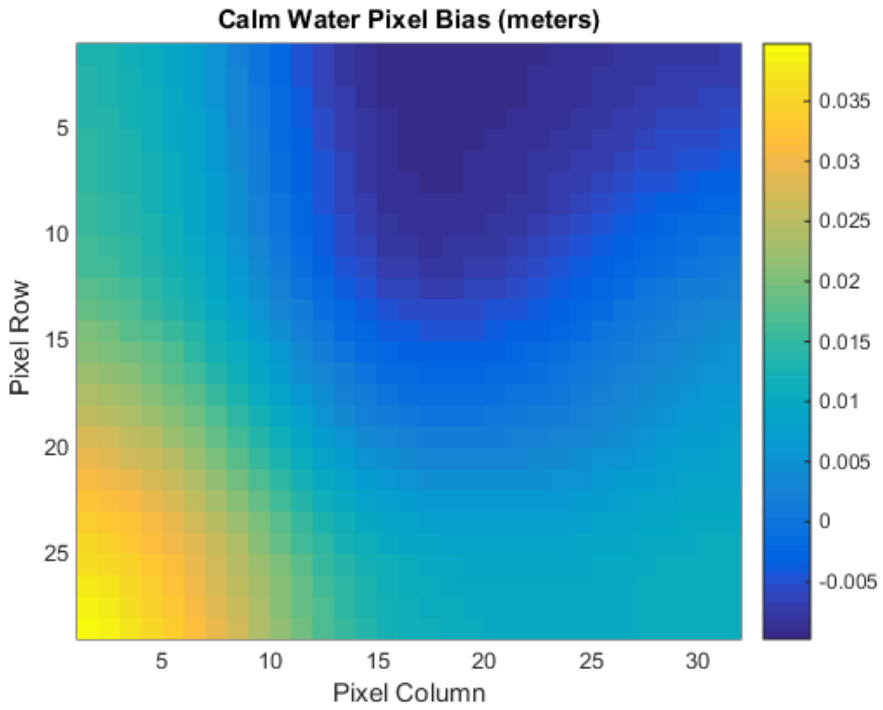

Figure 4.9: Per pixel bias ($Bias_{P(i,j)}$) across 400 acquisitions for calm water

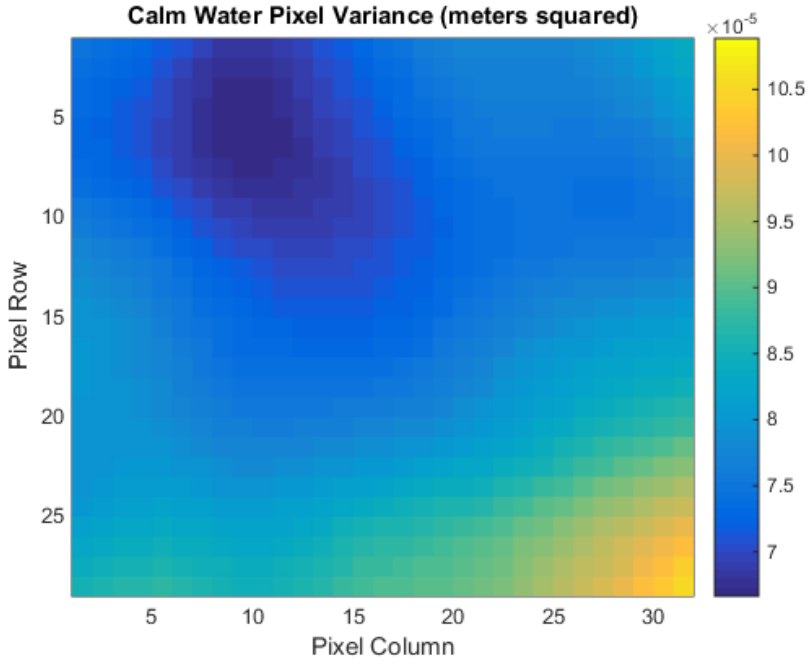

Figure 4.10: Per pixel variance ($Var_{P(i,j)}$) across 400 acquisitions for calm water

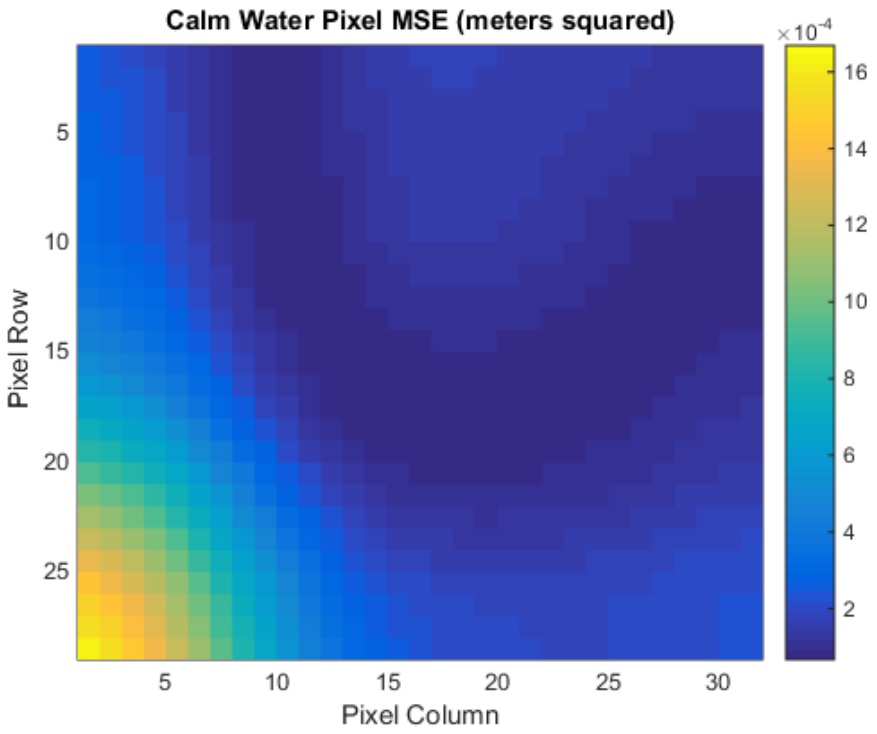

Figure 4.11: Per pixel MSE ($MSE_{P(i,j)}$) across 400 acquisitions for calm water

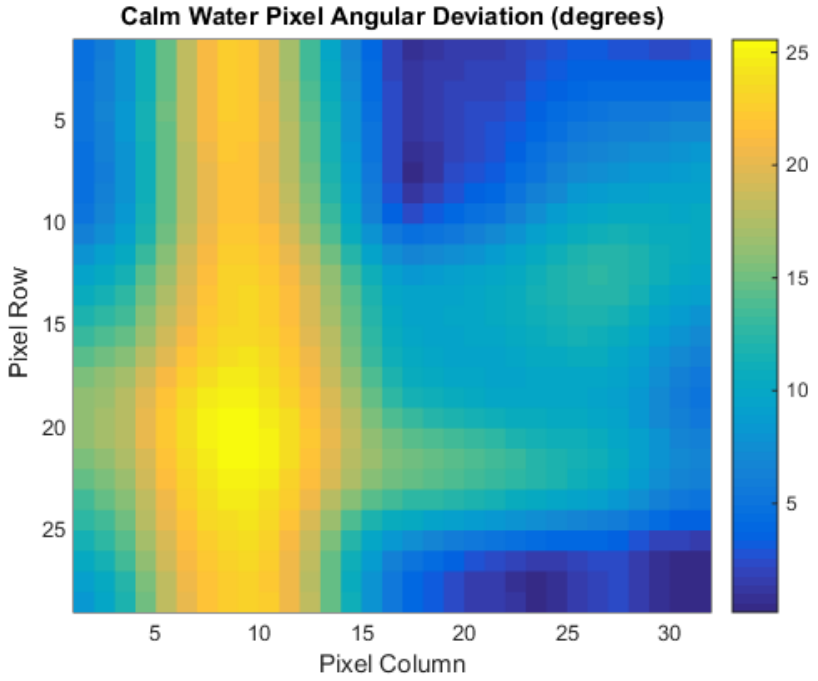

Figure 4.12: Per pixel angular deviation $\theta^{D}_{P(i,j)}$ across 400 acquisitions for calm water

Figures 4.9-4.12 indicate that performance varies considerably over the entirety of the FPA, likely indicating the necessity to further refine our ranging and intensity calibrations. Even with the poor uniformity, the overall accuracy of the water DSM is acceptable with an area of weakness that includes the average normal vector for the surface and on a per pixel basis. Certain areas (hypothesized to be where the beam is actually located) of the FPA have normal vectors are 25 degrees off the theoretical normal vector, leading to potentially unacceptable behavior if a ray tracing algorithm was used that utilized those normal vectors for correction. Even with some of these limitations, these results show a lot of promise for future water DSM construction and indicate that the added spatial resolution of the GmAPD receiver can be a boon for accurate imaging.

4.3 Summary

This section showed the results from a flat hard target acquisition and the four different calm water experiments conducted involving a total of 401 acquisitions of data. The results indicated that the GmAPD camera performed well when imaging a flat hard target and calm water, although the former was better in all error metrics when compared to the latter. The per pixel performance was shown to vary wildly depending on the location within the FPA, possibly indicating that additional per pixel corrections need to be done. Finally, surface normal vectors were adequate in some pixels but off by tens of degrees in others. This would lead to errors in potential ray tracing computations if they

were used for refraction correction. Correcting these will be central to further improving future sea floor imaging accuracy.

CHAPTER 5

CONCLUSION

5.1 Summary of Contributions

In performing the work to complete my research on creating a water DSM, I was primarily responsible for handling raw GmAPD data and performing all necessary processing steps to produce the final DSM. This involved developing and implementing all required real time correction and filtering code from scratch, culminating in a software backbone that can take in varying raw GmAPD data and output corresponding coordinates and normal vectors. This capability allows for any future GmAPD processing work to be done easily and provides flexible parameter adjustment for further fine tuning. Additionally, I modified existing PCL software for visualization, creating a split screen software visualizer that depicts coordinate and normal vectors in real time. This provides an additional piece of information for quick examination and error checking without needing to post process the data. I was also responsible for creating an interface between our GUI and the underwater camera so that parameters and data could be seamlessly communicated and used for additional ray trace processing software. All of these individual parts were then incorporated with other colleague's software to produce the final GUI. Two other colleagues and I were also responsible for the overseeing and collection of the data at the water tank.

Finally, the work done in this thesis represents EOSL's first foray into the field of GmAPD receivers; an area which is beginning to receive much attention, research, and development. Recent commercial availability has allowed for GmAPD technology to

play a significantly larger role in future systems, and this thesis' work directly reflects EOSL's progress and standing as being at the forefront of this field. Having this in house software capable of processing GmAPD data signifies the deep and thorough understanding our group has of GmAPD technology and provides additional credibility for future related work.

REFERENCES

[1] Tuell, G., Carr, D., Guida, N., and O'Shaughnessy, M., "Procedures and Algorithms for Raytracing Lidar Measurements Through An Irregular Sea Surface," NOAA Technical Report, (2015).

[2] Glassner, A. S., An introduction to Ray Tracing. Academic Press, (1989).

[3] Guenther, G.C., "Airborne Laser Hydrography: System Design and Performance Factors," NOAA Professional Paper Series, National Ocean Service 1, (1985).

[4] Gordon, H.R., "Interpretation of Airborne Ocean Lidar: Effects of Multiple Scattering," Applied Optics, vol. 21(16), pp. 2996-3001, (1982).

[5] Yang, E. and LaRocque, P., "Small Object Detection Using SHOALS Bathymetric Lidar," International Hydrographic Review, pp. 24-36, (2010).

[6] Guenther, G.C., Cunningham, A.G., LaRocque, P.E., and Reid, D.J., "Meeting the Accuracy Challenge in Airborne Lidar Bathymetry," Proceedings of 20th Workshop on Lidar Remote Sensing of Land and Sea, pp.1-27, (2000).

[7] Mandlburger, G., Pfennigbauer, M., and Pfeifer, N., "Analyzing Near Water Surface Penetration in Laser Bathymetry – A Case Study at the River Pielach," Remote Sensing and Spatial Information Sciences, Vol. II, pp.175-180, (2013).

[8] Guenther, G.C., LaRocque, P.E., and Lillycrop, W.J., "Multiple surface channels in SHOALS airborne lidar," Proc. SPIE, Ocean Optics XII, vol. 2258, (1994).

[9] Thomas, R.W.L. and Guenther, G.C., "Water Surface Detection Strategy for an Airborne Laser Bathymeter," Proc. SPIE, Ocean Optics X, vol. 1302, (1990).

[10] "AHAB Bathymetric LiDAR Systems", http://www.bshc.pro/media/documents/lidar2014/04_140527_AHAB_Bathymetric_LiDAR.pdf (Accessed February 17th, 2015).

[11] "HawkEye III", http://www.airbornehydro.com/hawkeye-iii (accessed February 17th, 2015).

[12] B.F. Aull, A.H. Loomis, D.J. Young, R.M. Heinrichs, B.J. Felton, P.J. Daniels, and D.J. Landers, "Geiger-Mode Avalanche Photodiodes for Three-Dimensional Imaging," Linc. Lab. J., Vol. 13, Number 2, pp.335-350, (2002).

[13] Marino, R.M. and Davis, W.R., "Jigsaw: a Foliage-Penetrating 3D Imaging Laser Radar System," Linc. Lab. J., Vol. 15, Number 1, pp.23-36, (2005).

[14] Kim, S., Lee, I., and Kwon, Y.J., "Simulation of a Geiger-Mode Imaging LADAR System for Performance Assessment," Sensors, Vol. 13, pp.8461-8489, (2013).

[15] Wu, J., Shih, P.T., Chen, J., and Chen, J., "The Display and Analysis of Bathymetric Lidar Waveform," Asian Conference on Remote Sensing, (2010).

[16] Carr, D.A., "A Study of the Target Detection Capabilities of an Airborne Lidar Bathymetry System," Georgia Institute of Technology, (2013).

[17] Tuell, G.H, Barbor, K., and Wozencraft, J.M., "Overview of the Coastal Zone Mapping and Imaging Lidar (CZMIL): A New Multi-sensor Airborne Mapping Systems for the U.S. Army Corps of Engineers," Proc. SPIE, Algorithms and Technologies for Multispectral, Hyperspectal, and Ultraspectral Imagery XVI, vol. 7695, (2010).

[18] Payment, A., Feygels, V., Fuchs, E., and Tuell, G.H., "Proposed Lidar Receiver Architecture of the CZMIL System", Proc. SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVI, vol. 7695, (2010).

[19] Tuell, G., "More Thoughts on Total Propagated Uncertainty for Bathymetric Lidar," SHOALS USACE Workshop, Mobile, AL (2014), http://shoals.sam.usace.army.mil/Workshop_Files/2014/Day2_pdf/1400_Tuell.pdf.

[20] Fuchs, E., and Mathur, A., "Utilizing Circular Scanning in the CZMIL system", Proc. SPIE, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVI, vol. 7695, (2010).

[21] Tuell, G., James, R., Ortman, R.L., Valenta, C.R., Carr, D., and Zutty, J., "Qualification Testing of the Agilent M9703A Digitizer for use in a Bathymetric LIDAR," IGARSS, (2014).

[22] Yuan, P., Sudharsanan, R., Bai, X., McDonald, P., and Labios, E., "Single photon sensitive Geiger-mode LADAR cameras," Proc. SPIE, Biosensing and Nanomedicine V, vol. 8460, (2012).

[23] Itzler, M.A., Krishnamachari, U., Entwistle, M., Jiang, X., Owens, M., and Slomkowski, K., "Dark Count Statistics in Geiger-Mode Avalanche Photodiode Cameras for 3-D Imaging LADAR," IEEE, Quantum Electronics, vol. 20, no. 6, (2014).

[24] Hiskett, P.A. and Lamb, R.A, "Design considerations for high-altitude altimetry and lidar systems incorporating single-photon avalanche diode detectors," Proc. SPIE, Advanced Photon Counting Techniques V, vol. 8033, (2011).

[25] Itzler, M.A., Entwistle, M., Owens, M., Patel, K., Jiang, X., Slomkowski, K., and Rangwala, S., "Geiger-mode avalanche photodiode focal plane arrays for three-dimensional imaging LADAR," Proc. SPIE, Infrared Remote Sensing and Instrumentation XVIII, vol. 7808, (2010).

[26] Itzler, M.A., Entwistle, M., Krishnamachari, U., Owens, M., Jiang, X., Slomkowski, K., and Rangwala, S., "SWIR Geiger-mode APD detectors and cameras for 3D imaging," Proc. SPIE, Advanced Photon Counting Techniques VIII, vol. 9114, (2014).

[27] Williams, G.M., "GHz-rate Single Photon Sensitive Linear Mode APD Receivers," Proc. SPIE, Quantum Sensing and Nanophotonic Devices VI, vol. 7222, (2009).

[28] Donnelly, J.P., Duerr, E.K., McIntosh, K.A., Dauler, E.A., Oakley, D.C., Groves, S.H., Vineis, C.J., Mahoney, L.J., Molvar, K.M., Hopman, P.I., Jensen, K.E., Smith, G.M., Verghese, S., and Shaver, D.C., "Design Considerations for 1.06-m InGaAsP–InP Geiger-Mode Avalanche Photodiodes," IEEE, Quantum Electronics, vol. 42, no. 8, (2006).

[29] Jiang, X., Itzler, M.A., Ben-Michael, R., and Slmokowski, K., "InGaAsP–InP Avalanche Photodiodes for Single Photon Detection," IEEE, Quantum Electronics, vol. 13, no. 4, (2007).

[30] Leitner, T., Feiningstein, A., Turchetta, R., Coath, R., Chick, S., Visokolov, G., Savuskan, V., Javitt, M., Gal, L., Brouk, I., Bar-Lev, S., and Nemirovsky, Y., "Measurements and Simulations of Low Dark Count Rate Single Photon Avalanche Diode Device in a Low Voltage 180-nm CMOS Image Sensor Technology," IEEE, Electron Devices, vol. 60, no. 6, (2013).

[31] Albota, M.A., Aull, B.F., Fouche, D.G., Heinrichs, R.M., Kocher, D.G., Marino, R.M., Mooney, J.G., Newbury, N.R., O'Brien, M.E., Player, B.E., Willard, B.C., and Zayhowski, J.J., "Three-Dimensional Imaging Laser Radars with Geiger-Mode Avalanche Photodiode Arrays," Linc. Lab. J., Vol. 13, Number 2, pp.351-370, (2002).

[32] Itzler, M.A., Entwistle, M., Owens, M., Chen, J., Xudong, J., Patel, K., Slomkowski, K., and Rangwala, S., "Single-Photon 3-D Imaging LADAR Camera Based on Geiger-Mode Avalanche Photodiodes," Optronics in Defense and Security, (2012).

[33] Frechette, J.P., Grossmann, P.J., Busacker, D.E., Jordy, G.J., Duerr, E.K., McIntosh, K.A., Oakley, D.C., Bailey, R.J., Ruff, A.C., Brattain, M.A., Funk, J.E., MacDonald, J.G., and Verghese, S., "Readout Circuitry for Continuous High-Rate Photon Detection with Arrays of InP Geiger-Mode Avalanche Photodiodes*," Proc. SPIE, Advanced Photon Counting Techniques VI, vol. 8375, (2012).

[34] Itzler, M.A., Entwistle, M., Jiang, X., Patel, K., Slomkowski, K., Koch, T., and Rangwala, S., "InP-based Geiger-mode avalanche photodiode arrays for three-dimensional imaging at 1.06 μm," Proc. SPIE, Advanced Photon Counting Techniques III, vol. 7320, (2009).

[35] Itzler, M.A., "Geiger-Mode Avalanche Photodiodes for Near-Infrared Photon Counting," CLEO/QELS Presentation, (2007).

[36] Ramirez, D.A., Hayat, M.M., Karve, G., Campbell, J.C., Torres, S.N., Saleh, B.E.A., and Teich, M.C., "Detection Efficiencies and Generalized Breakdown Probabilities for Nanosecond-Gated Near Infrared Single-Photon Avalanche Photodiodes," IEEE, Quantum Electronics, vol. 42, no. 2, (2006).

[37] Jensen, K.E., Hopman, P.I., Duerr, E.K., Dauler, E.A., Donnelly, J.P., Groves, S.H., Mahoney, L.J., McIntosh, K.A., Molvar, K.M., Napoleone, A., Oakley, D.C., Verghese, S., Vineis, C.J., and Younger, R.D., "Afterpulsing in Geiger-mode avalanche photodiodes for 1.06 μ m wavelength," Applied Physics Letters 88, 133503, (2006).

[38] Itzler, M.A., Jiang, X., and Entwistle, M., "Power law temporal dependence of InGaAs/InP SPAD afterpulsing," Journal of Modern Optics Vol. 59, No. 17, pp.1472–1480, (2012).

[39] Fouche, D.G., "Detection and false-alarm probabilities for laser radars that use Geiger-mode detectors" Optical Society of America, Applied Optics, Vol. 42, No. 27, (2003).

[40] Zhang, Z., Zhao, Y., Zhang, Y., Wu, L., and Su, J., "A real-time noise filtering strategy for photon counting 3D imaging lidar," Optical Society of America, Optics Express, Vol. 21, No.8, (2013).

[41] Buades, A., Coll, B., and Morel, J.M., "A review of image denoising algorithms, with a new one," SIAM Journal on Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal, 4 (2), pp.490-530, (2005).

[42] Darbon, J., Cunha, A., Chan, T.F., Osher, S., and Jensen, G.J., "Fast Nonlocal Filtering Applied to Electron Cryomicroscopy," IEEE, Bioedical Imaging: From Nano to Macro, pp.1331-1334, (2008).

[43] Aarias-Castro, E. and Donoho, D.L., "Does Median Filtering Truly Preserve Edges Better Than Linear Filtering?", The Annals of Statistics, Vol. 37, No. 3, pp.1172–1206, (2009).

[44] W. K. Pratt, "Median filtering," in Semiannual Report, Image Processing Institute, Univ. of Southern California, pp. 116-123, (1975).

[45] Fuguo, D., Hui, F., and Da, Y., "A Novel Image Median Filtering Algorithm based on Incomplete Quick Sort Algorithm," International Journal of Digital Content Technology and its Applications, Vol. 4, No. 4, (2010).

[46] Perreault, S. and Hebert, P., "Median Filtering in Constant Time," IEEE, Image Processing, Vol. 16, Issue 9, (2007).

[47] Dauwe, A., Goossens, B., Luong, H.Q., and Philips, W., "A Fast Non-Local Image Denoising Algorithm," Proc. SPIE, Image Processing: Algorithms and Systems VI, vol. 6812, (2008).

[48] Huang, Y., NG, M.K., and Wen, Y., "A Fast Total Variation Minimization Method for Image Restoration," Multiscale Model Simul, Vol. 7, No. 2, pp.774-795, (2008).

[49] Shin, D., Kirmani, A., Goyal, V.K., Shapiro, J.H., "Photon-Efficient Computation 3D and Reflectivity Imaging with Single-Photon Detectors," IEEE, International Conference on Image Processing, (2014).

[50] Lee, C., Lee, C., and Kim, C., "MMSE Nonlocal Means Denoising Algorithm for Poisson Noise Removal," IEEE, International Conference on Image Processing, pp.2561-2564, (2011).

[51] Buades, A., Coll, B., and Morel, J., "A non-local algorithm for image denoising," IEEE, Computer Vision and Pattern Recognition, Vol.2, pp.60-65, (2005).

[52] Rahman, A.A., "Digital Terrain Model Data Structures," Buletin Ukur, Vol. 5, No.1, pp.61-72, (1994).

[53] Bolstad, P., GIS FUNDAMENTALS. Elder Press, 4th Edition, (2012).

[54] Smith, M.J., Edwards, E.P., Priestnall, G., and Bates, P., "Exploitation of New Data Types to Create Digital Surface Models for Flood Inundation Modelling," Flood Risk Management Research Consortium, Research Report UR3, (2006).

[55] Lee, J., "Comparison of existing methods for building triangular irregular network models of terrain from grid digital elevation models," International Journal of Geographical Information Systems, Vol. 5, No. 3, pp.267-285, (1991).

[56] Su, P. and Drysdale, R.L.S., "A Comparison of Sequential Delaunay Triangulation Algorithms," Computational Geometry, Vol. 7, Issues 5-6, pp.361-385, (1997).

[57] Eberly, D., "Least Squares Fitting of Data," http://www.geometrictools.com/Documentation/LeastSquaresFitting.pdf (Accessed December 5th, 2014).

[58] He, W., Sima, B., Chen, Y., Dai, H., Chen, Q., and Gu, G., "A correction method for range walk error in photon counting 3D imaging LIDAR," Optics Communications, Vol. 308, No. 1, pp.211-217, (2013).

[59] He, W., Chen, Y., Miao, Z., Chen, Q., Gu, G., and Dai, H., "Range Walk Error Correction using prior modeling in Photon Counting 3D Imaging Lidar," Proc. SPIE, Laser Sensing and Imaging and Applications, vol 8905, (2013).

[60] Armadillo, http://arma.sourceforge.net/, (Last Accessed March 14th, 2015).

[61] OpenBLAS, http://www.openblas.net/, (Last Accessed March 14th, 2015).