Real-Time-Capable Algorithms for SPAD Array Flash LiDAR Data of Targets Collected in Clear and Obscured Environments

Man Ching Joyce Mau

A thesis submitted for the degree of
Doctor of Philosophy of
The Australian National University

August 2025

Except where otherwise indicated, this thesis is my own original work.

Man Ching Joyce Mau
August 25, 2025

To the incredible opportunity for deep thought and discovery.

Acknowledgments

I would like to thank my primary supervisor Professor Jochen Trumpf for his excellent supervision during my PhD candidature. His consistent guidance and mentoring have been critical in helping me work through this PhD. I would also like to thank my co-supervisor Adjunct Professor Dennis Delic, who has worked really hard to provide me with incredible opportunities and experiences during my PhD candidature. His passion for the technology has been inspiring and kept me motivated during this PhD. I would like to thank my co-supervisor Adjunct Associate Professor Vladimyros Devrelis as well, for his patience in listening to my day-to-day issues. His guidance and expertise have really helped shaped the research undertaken in this PhD.

I would like to acknowledge all members of the Emerging Sensors Discipline at DST Group for their contributions and support in writing this thesis. I would like to thank the many industry partners I got to work with during my PhD. Without their support in my other work commitments, I could not have completed this PhD.

I would like to thank Dr. Saeed Afshar for sharing a copy of his offline implementation (not included in this thesis) of the random feature-detecting network. His generous lessons on coding and algorithm development at the beginning of my research career will stay with me. I would like to thank Dr. Rudi Lussana as well for assisting me with developing the live data acquisition module that is used as part of the embedded software implementation of the random feature-detecting network. I would like to also thank Aaron Panella who was with Aurizn (formerly elmTEK) for assisting me with optimising the embedded software implementation of the random feature-detecting network. In addition, I would like to thank the Australian Institute of Marine Science (AIMS), for providing access to their indoor saltwater tank and assistance with the experiment set-up. In particular, I would like to thank Tom Barker, Craig Humphrey, David Crossman, Melanie Olsen and Dr. Lyndon Llewellyn.

Last but not least, I would like to thank my husband, Adam Cannizzaro, who has provided me immense support in my personal life, allowing me time to rest and recharge so I can put in my all for this PhD. I would also like to thank my family and in-laws for their encouragement and support throughout this PhD. And finally my friends, who have brought me so much joy and laughter as well as mentoring.

Third party editorial advice has been provided by Dr Liz Charpleix AE from Iris42 Editing Services. Her service included copy-editing and proofreading this thesis, as per ANU's Guideline: Higher Degree by Research—Editing of Theses.

During my own editing of this thesis, I also used ChatGPT (versions 3.5, 4o and 4o mini) for partially improving the clarity of expression, grammar and punctuation. All suggestions were carefully reviewed and edited by me, ensuring that the final

content reflects my own understanding and scholarly effort.

This research is supported by the Commonwealth of Australia as represented by the Defence Science and Technology Group of the Department of Defence. The research presented in Chapter 4 is also co-funded by NATO SPS project 984840.

Abstract

This thesis focuses on real-time-capable algorithms that can perform single-target discrimination from 3D data collected using a 32×32 Single Photon Avalanche Diode (SPAD) rectangular planar array 3D flash direct Time-of-Flight (dToF) Light Detection and Ranging (LiDAR) system. Different algorithms are presented for data collected in different environments.

For data collected in air, a real-time target classification algorithm is implemented on the Nvidia Jetson TX2 module to classify live data of four different model airplanes. The algorithm achieves an overall classification accuracy of 99.35% and an F1-score of 0.99 when using 64 convolutional filters. The algorithm has an execution time of 65.07 ms and its power usage with a live data acquisition module is 5.1 W.

For data collected in obscured environments, the focus of the presented algorithms shifts from target classification to depth image reconstruction as a necessary step towards target classification. In particular, the presented algorithms process collected data to reconstruct images for subsequent target detection.

Two image-reconstruction algorithms, the detection threshold and median algorithm and the histogram averaging algorithm, are applied to three separate datasets of different targets collected in different real and simulated natural water environments. The detection threshold and median algorithm can reconstruct simple shapes obscured by water turbidity levels up to 3.43 attenuation lengths (ALs). The minimum overall shape classification accuracy is 95% when using the images reconstructed by this algorithm for different sediment-only conditions, which are up to a sediment concentration of 5.62 mg/L (3.29 ALs for 532 nm wavelength). The histogram averaging algorithm performs better for above-water imaging, where it reconstructs a 4.5 m submerged Secchi Disc (SD) in clear water conditions in the Defence Science and Technology Group (DSTG) indoor freshwater tank. However, it can only reconstruct a white disc submerged down to 2 m at the Port Adelaide River.

Image reconstruction and target classification are demonstrated for fog-obscured data. The range gate and process algorithm reconstructs range-gated data of frigate silhouettes at a fog level of 3.69 ALs (at 532nm) and its images produce 93% overall silhouette classification accuracy. Without range-gating, the finite mixture model algorithm is used instead to reconstruct images of simple shape obscured by a fog level of up to 1.14 ALs. These images provides 100% overall shape classification accuracy. At a higher fog level of 2.28 ALs (at 532nm), the Fourier descriptor algorithm achieves better global localisation accuracy than the previous algorithm. A simple conversion of the algorithm's binary images results in reconstructed images. Target classification is possible using the shape uniqueness of Fourier descriptors.

At the conclusion of this thesis, future research directions are suggested to overcome the limitations of the presented results.

Contents

List of Figures

List of Tables

Acronyms

ADMM Alternating Direction Method of Multipliers

AIMS Australian Institute of Marine Science

AL attenuation length

ALs attenuation lengths

APDs Avalanche Photodiodes

CMOS Complementary Metal-Oxide-Semiconductor

CNN Convolutional Neural Network

CNNs Convolutional Neural Networks

COTS commercial off-the-shelf

CPU central processing unit

CUDA Compute Unified Device Architecture

CWT Continuous Wavelet Transform

DAE depth absolute error

DSTG Defence Science and Technology Group

dToF direct Time-of-Flight

EM Expectation-Maximization

FFD Frigates Fog Dataset

FFT Fast Fourier Transform

FOV field-of-view

FPGA Field-Programmable Gate Array

GLRT Generalised Likelihood Ratio Test

GM-APD Geiger-Mode Avalanche Photodiode

GMMs Gaussian Mixture Models

GPU graphical processing unit

GPUs graphical processing units

GUI Graphic User Interface

IID independent and identically distributed

IOPs Inherent Optical Properties

iToF indirect Time-of-Flight

KDE Kernel Density Estimator

LiDAR Light Detection and Ranging

LM-APD Linear-Mode Avalanche Photodiode

MAP Maximum a-Posteriori

MCP Micro-Channel Plate

MLE Maximum Likelihood Estimation

M2R3D Median-based Multi-scale Restoration of 3D images

PCA Principal Component Analysis

PMT Photo-Multiplier Tube

PSNR Peak Signal-to-Noise Ratio

RADAR Radio Detection And Ranging

RARE relative average ranging error

RMSE root mean square error

ROV Remotely Operated Vehicle

ROVs Remotely Operated Vehicles

SD Secchi Disc

SFD Shapes Fog Dataset

SM Streaming Multiprocessor

SMs Streaming Multiprocessors

SNR Signal-to-Noise Ratio

SoM System-on-Module

SONAR Sound Navigation and Ranging

SPAD Single Photon Avalanche Diode

SPADs Single Photon Avalanche Diodes

SRE Signal-to-Reconstruction Error

SWaP Size, Weight and Power

SWIR Short-Wave Infrared

TCSPC Time-Correlated Single-Photon Counting

ToF Time-of-Flight

TR Target Recovery

UAV Uncrewed Autonomous Vehicle

UAVs Uncrewed Autonomous Vehicles

Chapter 1

Introduction

Protecting a military unit from hostile threats is essential for the safety of personnel, the security of equipment, and the success of the mission. For this reason, the identification of hostile threats needs to be performed quickly and at a distance. Rapid assessment of the military operation areas assists the military unit to react in real time to hostile threats. Performing surveillance and reconnaissance from a distance allows the military unit to identify and locate hostile threats without compromising its own position and safety.

One way of conducting remote surveys is to use a distributed network of small and mobile platforms such as Uncrewed Autonomous Vehicles (UAVs), satellites or Remotely Operated Vehicles (ROVs) and then transmit all their information to one central node. The military can survey safely from a distance by placing the central node in a safe location far away from potential hostile threats. However, such a system architecture requires a high data-transmission bandwidth. This poses several challenges for data security and transmission reliability.

Edge computing is a computing paradigm that brings computation and data storage closer to where data is generated, specifically in applications where real-time processing of data is required [3]. In the context of a distributed network, data is generated by the sensor on board the distributed node's platform. The application of edge computing means the sensor data is processed on-board in real time at the distributed node, instead of being transmitted before being processed centrally. This provides significant reduction in response time and transmission bandwidth [3]. One possible approach to on-board processing is to employ an embedded board.

One implementation of the edge computing paradigm is performing autonomous target discrimination. Target discrimination is defined as "A process in which an object is assigned to a subset of a larger set of objects based on the amount of detail perceived by the observer" [68]. The mobile platform can transmit target-specific information instead of a stream of data to the central node. This reduction in transmission information changes the data transmission from a long-term data rate scheme to an occasional small data package transmission. This reduces the susceptibility of the transmission to being compromised.

For imaging targets in a scene, 3D imaging is generally preferred over 2D because it allows targets to be distinguished based on their distances, making it easier to segment them from other clutter or obscurants [40]. Light Detection and Ranging

Figure 1.1: A long tunnel filled with fog. The fluorescent light and green laser on the right illustrates how light is scattered and absorbed in the fog.

(LiDAR) is a widely used 3D imaging technique in remote sensing [110, 107, 157]. It offers some advantages over other 3D imaging solutions for long range imaging in challenging conditions, such as fog or smoke in the air or turbidity in natural water environments, where obscurants can hinder image clarity and degrade overall image quality. For fog obscured conditions (such as the scene shown in Figure 1.1), Radio Detection And Ranging (RADAR) is a common sensing technique [139]. However, for long-distance sensing scenarios such as remote surveillance on an Uncrewed Autonomous Vehicle (UAV), LiDAR offers finer distance and spatial resolution imaging than RADAR [139]. For natural water environments, Sound Navigation and Ranging (SONAR) is traditionally used for sensing underwater targets at fine distance resolution and at long ranges. However, when it comes to conducting remote sensing water environments from air, LiDAR is more suitable because more light can travel through the air-water interface compared to sound waves. For these reasons, LiDAR is an excellent candidate as a complementary sensor to enhance existing remote sensing capability.

This thesis presents a collection of real-time-capable algorithms that can perform static single-target discrimination using 3D data collected from a Complementary Metal-Oxide-Semiconductor (CMOS) silicon 32×32 Single Photon Avalanche Diode (SPAD) rectangular planar array within a 3D flash direct Time-of-Flight (dToF) LiDAR system. For simplicity, this thesis will henceforth refer to this type of system as the SPAD array flash LiDAR system. In this thesis, real-time-capable means that the algorithm has been demonstrated to require an execution time that either matches or is less than the time to acquire the amount of data required to execute one instantiation of the algorithm.

In this thesis, the real-time capability of the presented algorithms are demonstrated via either a real-time implementation or computational demand analysis. A

real-time implementation of an algorithm means the algorithm finishes its execution before the next batch of SPAD LiDAR frames is acquired. A computational demand analysis examines the computational operations required by an algorithm in order to demonstrate that the algorithm is real-time capable when optimised in its implementation. The algorithms' implementation is assumed to be optimised for execution on an embedded board that contains a graphical processing unit (GPU), such as the Nvidia Jetson TX2 module.

The imaging system and algorithms are chosen with the low Size, Weight and Power (SWaP) requirement in mind, ensuring they can be mounted on a mobile platform for real-time target discrimination on a distributed node. The imaging system, being a SPAD array flash LiDAR system, can be engineered to have a low SWaP requirement. The presented real-time-capable algorithms is intended for execution on an embedded board, which has low SWaP. In addition, because the algorithms have been designed to have low computational demand, the implementation can be designed to execute on an embedded board using low amounts of power.

The functionality of the presented algorithms is demonstrated using data collected from different types of environments that are commonly encountered by the military, which includes air, natural water, and fog. Different algorithms are used to process data collected from different environments. The imaged scenes are mostly in indoor laboratory locations, and in one instance, the imaged scene is a real-world natural water location. The data is collected with various SPAD array flash LiDAR systems.

The visible range is selected as the laser wavelength for all imaging presented in this thesis. This is because this wavelength range can be used for imaging through various different environments, including fog and natural waters. While Short-Wave Infrared (SWIR) exhibits less attenuation through fog compared with the visible range [158, 157], SWIR radiation cannot penetrate natural water environments.

Despite LiDAR having advantages for long range imaging in obscured environments such as fog and natural waters, obscurants still pose a challenge for accurately localising targets and producing well-reconstructed 3D images using LiDAR data. In particular, the quality of LiDAR images degrades due to the light scattering in fog [183] and water [88]. This makes it difficult to perform target discrimination using the collected LiDAR data. In this thesis, SPAD is selected as the detector in the LiDAR system because it has single photon sensitivity. This means it's able to detect the sparse number of photon returns from the obscured target.

Target discrimination can be divided into several different levels: detection, orientation, classification, recognition and identification. The level of target discrimination depends on the amount of detail it provides about the target, where a higher level of target discrimination provides more detail. Currently, there are no consistent definitions of these terms in the literature. Therefore, I present my definitions of these terms in Figure 1.2, which I use for this thesis. These definitions are inspired by the different definitions of these terms provided by the authors in [126] (as cited in [163]), [83], and [17]. The figure also contains an example of each target discrimination level in the context of discriminating a car target.

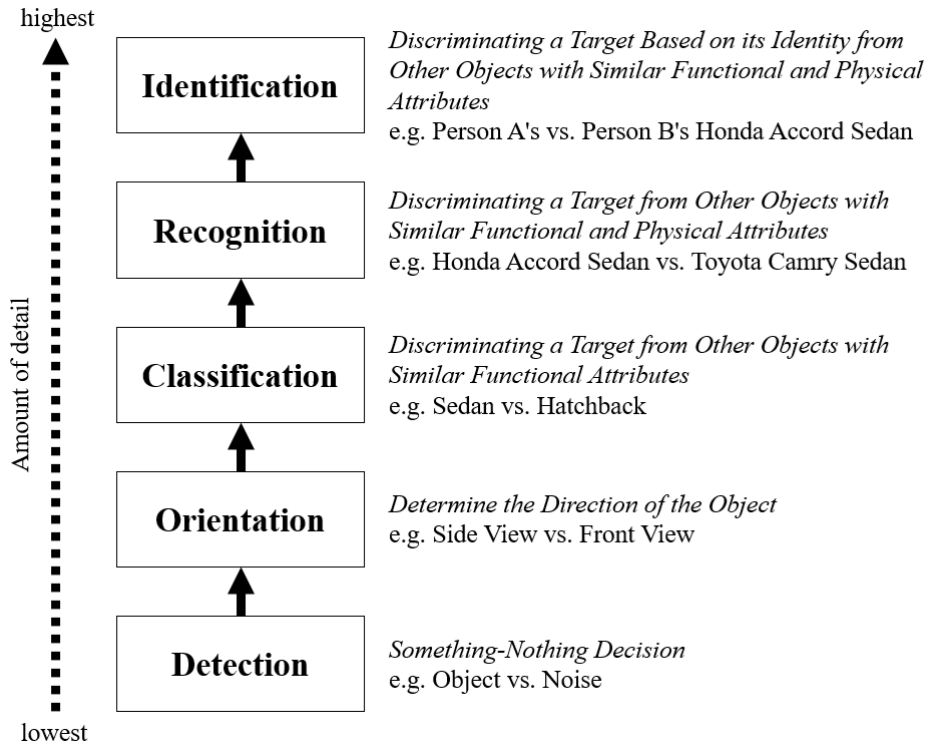


Figure 1.2: Different levels of target discrimination.

The application of the algorithms presented in this thesis can be divided into two main focus areas: target detection and classification. In particular, this thesis only considers scenarios in which target detection and classification operate on images. In order for these tasks to be successful, the target needs to be discernible in the images. This requires effective image reconstruction of the data collected by the SPAD array flash LiDAR system. This thesis focuses only on the reconstruction of depth images and not the reconstruction of reflectivity images. However, a simple alteration can be applied to many of these algorithms to enable them to reconstruct reflectivity images in addition to depth images.

While an accurately reconstructed image is not a prerequisite for target detection and classification, it does reduce classification's computational complexity compared with a poorly reconstructed image. The increased complexity in target detection and classification of poorly reconstructed images can hinder real-time execution and limit the feasibility of using the algorithm for edge computing implementation on a distributed network.

Machine learning is a possible approach to target detection and classification without the use of images. It is possible to use machine learning to distinguish targets by training its system with a large amount of SPAD LiDAR data. The system may be able to observe tiny differences in the dToF measurements between an object and noise (for detection), or associated with different types of targets (for clas-

sification). However, these types of machine learning systems that operate on raw data instead of images, such as neural networks, require a larger amount of training data compared with machine learning systems that process images. Therefore, this approach is impractical for military applications due to the scarcity of adversary target images, especially the absence of SPAD LiDAR data. While simulated data can be obtained, it may be difficult to determine the features of adversary targets (such as easily customisable commercial off-the-shelf (COTS) UAVs), which can be fast-evolving during conflict.

Rather than focusing on determining target detection and classification algorithms that are capable of accurately processing inaccurate reconstructed images of obscured targets, this thesis concentrates on algorithms that accurately reconstruct images of these targets. By doing so, the reconstructed image can contain a faithful representation of the target at the additional price of computation, thereby enhancing the accuracy of subsequent existing target detection and classification operations, without adding further computation steps.

Following this introduction, Chapter 2 reviews existing work that performs target discrimination using SPAD LiDAR data. Chapter 3 then presents the motivation and background of SPAD array flash LiDAR systems, along with a detailed description of the specifications of the SPAD array flash LiDAR systems used to test the presented algorithms in this thesis. The remainder of Chapter 3 provides definitions and explanations of key terms used throughout the thesis.

In this thesis, the presented algorithms have either a focus in target detection or classification, depending on the type of environment from which the data is collected.

For data collected in air, Chapter 4 presents a real-time algorithm that uses most of its processing for target classification. Image reconstruction and target detection are conducted in the initial stage of the algorithm, serving the purpose of checking there is a target in the image before it is evaluated for classification. This algorithm is implemented, together with a live SPAD LiDAR data acquisition module, into an embedded software program for execution on a COTS embedded board called the Nvidia Jetson TX2 module. The classification and hardware performances of the embedded software program are evaluated with live data of model airplanes in an indoor laboratory environment.

For algorithms related to processing data collected in obscured environments such as fog and natural waters, the algorithms perform image reconstruction to create a clear representation of the obscured target for detection. If a target is discernible in a reconstructed image, then the image can be used for target detection. For some of these algorithms, target classification performance is evaluated over these reconstructed images as well.

For data collected in natural water environments, Chapter 5 examines the performance of two basic image-reconstruction algorithms—the detection threshold and median algorithm, and the histogram averaging algorithm—under varying turbidity and depth conditions. The explanation of these algorithms are accompanied by a computational demand analysis to demonstrate that these algorithms are real-time-capable. Target classification performance is also evaluated for reconstructed images

of data collected in varying amounts of sediments in saltwater. In addition, this chapter details experiments conducted in both controlled laboratory settings and a real-world natural water site. For some of these experiments, a novel imaging set-up is used to conduct above-water imaging into the water.

For data collected in fog-obscured environments, Chapter 6 presents three different image-reconstruction algorithms: range gate and process algorithm, finite mixture model algorithm, and Fourier descriptor algorithm. The SPAD array flash LiDAR data used for evaluating these algorithms are collected from an indoor laboratory environment and artificial fog is generated using a fog machine and water-based liquid. To demonstrate the real-time capability of these algorithms, computational demand analyses are provided.

The reconstructed images from the first two algorithms are also tested separately for target classification. Reconstructed images from the third algorithm, the Fourier descriptor algorithm, are not tested for target classification because it is developed to reconstruct images of a single type of target. However, this algorithm could be easily extended for target classification in the future. Similar to the algorithms presented in Chapter 5, a discussion is provided to demonstrate that these algorithms are real-time-capable.

When target classification is demonstrated for some of these image-reconstruction algorithms in Chapters 5 and 6, it is apparent that the algorithms can reconstruct a clear enough representation of the obscured target that it can be used for higher levels of target discrimination. The image-reconstruction algorithms presented in this thesis are agnostic to the geometry of the target. Therefore, these image-reconstruction algorithms can be used to reconstruct other types of targets in the future, in which their images can be used for target classification.

All of the SPAD array flash LiDAR systems images used in this thesis are 32×32 in pixel size. Currently, researchers in the SPAD community are investigating the creation of SPAD arrays with larger numbers of SPAD detectors [162, 43, 116]. The presented algorithms are scalable to enable the classification or reconstruction of images collected by flash LiDAR system with larger arrays of Single Photon Avalanche Diodes (SPADs). However, further work will be required to confirm their processing times for larger image spatial resolutions, to confirm that they can execute in real-time.

1.1 Contributions

My novel contributions in this thesis to the field of employing SPAD LiDAR for imaging applications are detailed in the three lists below. Each list represents contributions related to data collected in a different environment. A reference is added to the start of the dot point if a paper has been published related to the contribution. These papers are listed in Section 1.2 below.

Data collected in air:

- [102, 178] - In Chapter 4, a real-time target classification algorithm for live SPAD

LiDAR data is implemented for execution on an Nvidia Jetson TX2 module, which is a COTS board. Part of the implementation leverages the on-board GPU. To the best of my knowledge, at the time of publication of [102], this marks the first implementation of such an algorithm on an embedded board for the real-time target classification of data collected by a SPAD array flash LiDAR system. Subsequently, related literature has emerged, detailed in Section 2.5 of Chapter 2.

Data collected in natural waters:

- In Chapter 5, the imaging in this chapter are taken by a SPAD array flash LiDAR system at the Port Adelaide River (Section 5.6), which is an estuary consisting a mixture of fresh and saltwater, and the Australian Institute of Marine Science (AIMS) indoor saltwater tank containing real sediments and chlorophyll (Section 5.4) from the ocean. The imaging set-up conducted at AIMS has been published in [101]. To the best of my knowledge, this is the first time natural water environments or lab-simulated natural water environments with real marine constituents have been used for SPAD LiDAR imaging. The details of the imaging setup at the Port Adelaide River are in Section 5.2.4 and the details of AIMS indoor saltwater tank is in Section 5.2.2. As a consequence of this novel type of imaging environment, several other contributions have been made:
  - Section 5.2.3 of Chapter 5 presents a novel imaging set-up for above-water SPAD LiDAR imaging at the Defence Science and Technology Group (DSTG) indoor freshwater tank, which is a laboratory-controlled water tank. This novel imaging set-up is used for the outdoor natural water imaging at the Port Adelaide River presented in Section 5.2.4.
  - Section 5.6 of Chapter 5 demonstrates that the histogram averaging algorithm is able to reconstruct discernible images of a submerged target in up to a depth of 2 m at the Port Adelaide River, which had a water turbidity level of 3.4m Secchi Disc (SD) depth at the time of data collection.
  - Section 5.5 of Chapter 5 demonstrates that the histogram averaging algorithm is able to reconstruct discernible images of a target in a clear freshwater tank submerged at water depths up to 4.5 m.
  - Section 5.4.1 of Chapter 5 demonstrates that the detection threshold and median algorithm is able to reconstruct discernible images of a target submerged in a saltwater tank containing a mix of sediment and/or chlorophyll that is equivalent to a turbidity level of up to 3.43 attenuation lengths (ALs) with respect to 532 nm.
  - [101] - In Chapter 5, Section 5.4.3 demonstrates that the images reconstructed by the detection threshold and median algorithm can produce an overall classification accuracy of 95% for data collected at the AIMS indoor saltwater tank in sediment-only conditions with sediment concentrations up to 5.62 mg/L (equivalent to 3.29 ALs with respect to 532 nm).

Data collected in fog:

- [99] - In Chapter 6, Section 6.3 presents an image-reconstruction algorithm called the finite mixture model algorithm, which is an algorithm developed during my PhD candidature. Its reconstruction ability is demonstrated in Section 6.5.2.1. The algorithm operates by using a probability distribution to model the likelihood of each possible dToF measurement over a batch of frames. In the literature, to be discussed in Section 2.3 of Chapter 6, a common approach involves removing the dToF measurements of detected photons reflected from fog before modelling those related to the target. However, this algorithm diverges from this convention by concurrently modelling dToF measurements of detected photons reflected from both the fog and the target, employing a mixture of lognormal and Gaussian probability distributions. This algorithm stands in contrast to existing modelling approaches, in which a gamma probability distribution is typically used for modelling the dToF measurements of detected photons reflected from fog. The divergence arises from the application of a phenomenological approach to the dToF measurements rather than modelling the dToF measurements with the physics of photon propagation and detection.

- [104] - In Section 6.4 of Chapter 6, the Fourier descriptor algorithm is developed for the global localisation of a target. This is a novel algorithm developed during my PhD candidature. I introduce the concept of Fourier shape descriptors, an image feature technique, to localise geometric targets from SPAD LiDAR data. To the best of my knowledge, the closest the SPAD LiDAR literature has come to using image features involves applying spatial correlation between local neighbouring pixels, either as additional regularisation terms (e.g. in [60] as part of their depth reconstruction optimisation problem) or for post-reconstruction image processing (e.g. inpainting by [127]). Unlike these existing algorithms, which exploit correlation between local neighbouring pixels, the developed Fourier descriptor algorithm uses Fourier shape descriptors to analyse all the pixels globally. It obtains information about the target's shape by analysing the entire image before calculating a Fourier shape descriptor. Although this Fourier descriptor algorithm is demonstrated for target localisation only, it has potential applications for future image reconstruction and target classification. Image reconstruction can be easily computed from the binary image created as part of the algorithm processing, at the target's localised depth determined by the algorithm. Target classification is possible because different geometric shapes correspond to unique Fourier shape descriptors. Hence, target classification is achieved by comparing the target's corresponding Fourier shape descriptor with the different descriptors associated with possible shapes of the target. To the best of my knowledge, this is the first instance of the introduction of an image feature technique, such as the Fourier shape descriptor, for the localisation, reconstruction and classification of targets imaged by a SPAD LiDAR system.

- [100, 99] - In Chapter 6, target classification of reconstructed images from SPAD LiDAR measurements of targets obscured by fog is investigated. Specifically, target classification accuracy is examined for images reconstructed from two algorithms: the range gate and process algorithm (detailed in Section 6.5.1.2) and the finite mixture model algorithm (discussed in Section 6.5.2.3). The test for the first algorithm focuses on classifying two different frigate silhouettes, while the test for the second algorithm is to classify three different simple geometric shapes. To the best of the author's knowledge, there exists limited literature on target classification of SPAD LiDAR images of targets obscured by fog.

1.2 Publications

These papers are published and related to my PhD research. I am the lead author in all of these papers.

- [102] - **J. Mau**, S. Afshar, T. J. Hamilton, A. van Schaik, R. Lussana, A. Panella, J. Trumpf, and D. Delic, "Embedded implementation of a random feature detecting network for real-time classification of time-of-flight SPAD array recordings," in Proc. SPIE, vol. 11005, Baltimore, USA, May 2019, p. 1100505, doi: 10.1117/12.2517875.

- [101] - **J. Mau**, V. Devrelis, G. Day, J. Trumpf, and D. Delic, "Impact of water quality on Single Photon Avalanche Diode direct time-of-flight imaging," in Proc. Global Oceans 2020: Singapore – U.S. Gulf Coast, Biloxi, MS, USA, 2020, pp. 1–8, doi: 10.1109/IEEECONF38699.2020.9389293.

- [100] - **J. Mau**, V. Devrelis, G. Day, G. Nash, J. Trumpf, and D. Delic, "Through thick and thin: Imaging through obscurant using SPAD array," in Proc. IEEE Sensors, Rotterdam, Netherlands, 2020, pp. 1–4, doi: 10.1109/SENSORS47125.2020.9278706.

- [99] - **J. Mau**, V. Devrelis, G. Day, J. Trumpf, and D. V. Delic, "The use of statistical mixture models to reduce noise in SPAD images of fog-obscured environments," in Proc. SPIE, vol. 11525, Online Only, 2020, p. 115250P, doi: 10.1117/12.2580251.

- [104] - **J. Mau**, J. Trumpf, G. Day, and D. Delic, "An image feature-based approach to improving SPAD Flash LiDAR imaging through fog," in Proc. SPIE, vol. 12274, Berlin, Germany, Nov. 2022, p. 1227405, doi: 10.1117/12.2633941.

I am a co-author in the following book chapter:

- [178] - F. Zappa, F. Villa, R. Lussana, D. Delic, **M. C. J. Mau**, J.-M. Redouté, S. Kennedy, D. Morrison, M. Yuce, T. Alan, T. Hamilton, S. Afshar, "Microelectronic 3d imaging and neuromorphic recognition for autonomous UAVs," in Advanced Technologies for Security Applications: Proceedings of the NATO

Science for Peace and Security'Cluster Workshop on Advanced Technologies', 17-18 September 2019, Leuven, Belgium, C. Palestini, Ed. Dordrecht, The Netherlands: Springer, 2020, ch. 17, pp. 185–194, Accessed: February 15, 2024, doi: https://doi.org/10.1007/978-94-024-2021-0_17. [Online]. Available: https://link.springer.com/chapter/10.1007/978-94-024-2021-0_17

I am involved in the following papers. I am acknowledged but not an author:

- [136] - A. D. Ruvalcaba-Cardenas, T. Scoleri, and G. Day, "Object classification using deep learning on extremely low-resolution time-of-flight data," in Proc. Digit. Image Comput.: Techn. and Appl. (DICTA), Canberra, Australia, Dec. 2018, pp. 1–7, doi: 10.1109/DICTA.2018.8615877. [Online]. Available: https://ieeexplore.ieee.org/document/8615877/

- [117] - G. Nash and V. Devrelis, "Flash LiDAR imaging and classification of vehicles," in Proc. IEEE Sensors, Rotterdam, Netherlands, Oct. 2020, pp. 1–4, doi: 10.1109/SENSORS47125.2020.9278655.

These papers were also published during my PhD candidature, but are outside the scope of my PhD research. I am an author in all of these papers:

- [42] - V. Devrelis, **J. Mau**, G. Day, and D. Delic, "Estimation of underwater horizontal visibility for divers and ROVs," in Proc. Global Oceans 2020: Singapore – U.S. Gulf Coast, Biloxi, MS, USA, 2020, pp. 1–6, doi: 10.1109/IEEECONF38699.2020.9389402.

- E. G. S. Rugai, B. J. Slimming, D. Beniwal, **J. Mau**, G. Day, V. Devrelis, and D. Delic. "Development of an experimental setup for measuring ocean property changes resulting from turbulence," in Proc. SPIE, vol. 12543, Florida, United States, June 2023, p. 125430B, doi: https://doi.org/10.1117/12.2663302.

- B. J. Slimming, D. Beniwal, V. Devrelis, **J. Mau**, and D. Delic. "The non-acoustic signatures of underwater vehicles," in Proc. SPIE, vol. 12543, Florida, United States, June 2023, p. 1254306, doi: https://doi.org/10.1117/12.2663304.

Chapter 2

Related Work

In the research field of utilising direct Time-of-Flight (dToF) measurements from Single Photon Avalanche Diode (SPAD) Light Detection and Ranging (LiDAR) systems for target discrimination purposes, the SPAD LiDAR community primarily focuses its efforts on target detection. There is less literature addressing the utilisation of dToF measurements for higher levels of target discrimination that provide a higher amount of detail about the target (i.e. orientation, classification, recognition, identification).

Most SPAD LiDAR literature related to target detection develops techniques to perform image reconstruction from dToF measurements. Some image-reconstruction algorithms reconstruct only depth images, while others reconstruct depth as well as reflectivity images. This review focuses on algorithms that reconstructs depth images. For algorithms where reflectivity images are also reconstructed, that portion of the algorithm is omitted if the reflectivity values are not essential for reconstructing depth images. This is because reflectivity images are beyond the scope of this thesis.

The quality of the reconstructed depth images is often discussed in terms of whether the target is discernible by visual inspection, which can be interpreted as a performance assessment for target detection. Currently, there are no consistent quantitative metrics for assessing the localisation accuracy of the reconstructed dToF values in reconstructed depth images.

The literature related to image reconstruction can be categorised based on the environmental conditions under which dToF measurements are collected by a SPAD LiDAR system. Various environments are of interest to the SPAD LiDAR community, and this chapter provides an overview of the literature for each of these environments. Firstly, Section 2.1 provides an overview of literature conducted in environments with no obscurants. Then, Section 2.2 addresses environments partially occluded by objects such as foliage and camouflage nets. This is followed by a discussion of image reconstruction in environments obscured by scattering mediums, including fog or smoke in Section 2.3 and turbid water in Section 2.4.

Subsequently, after discussing literature related to image reconstruction, Section 2.5 presents an overview of SPAD LiDAR literature that conducts higher levels of target discrimination. Here, some higher levels of target discrimination algorithms utilise reconstructed reflectivity images in addition to reconstructed depth images. Discussion of reflectivity images are only included if it is an essential part of the tar-

get discrimination algorithm. Finally, the chapter concludes with a table organising the literature discussed by category. The table is based on the different imaging environments, level of target discrimination, type of SPAD LiDAR system, and whether the literature demonstrates the algorithm to be real-time capable, which includes the algorithm having been implemented in real time, or the algorithm has only been demonstrated to operate in an offline computational environment.

The literature in the field of SPAD LiDAR imaging continues to expand at a rapid pace. The cut-off date for inclusion in the discussion of this chapter is 31 May 2024, as all of the research presented in this thesis had concluded before this date.

2.1 Environments with no Obscurants

For algorithms that reconstruct images from dToF measurements obtained in environments with no obscurants, most literature can be categorised based on the type of SPAD LiDAR system that is used to collect these measurements. These systems fall into two distinct categories: SPAD scanning LiDAR, which encompasses the use of single SPADs and linear SPAD arrays, and SPAD array flash LiDAR, which solely utilises rectangular arrays of SPAD. A more comprehensive explanation of the differences between these configurations is provided in Section 3.2 of Chapter 3. These categorisations are essential as they reflect the distinct research focuses within the SPAD LiDAR community for each configuration.

2.1.1 SPAD Scanning LiDAR

The image-reconstruction literature on SPAD scanning LiDAR systems primarily focuses on of dToF measurements collected by a single SPAD detector, with an emphasis on long-range targets situated in the order of kilometres. Several studies demonstrate the quality of the reconstructed images from dToF measurements collected by these single SPAD scanning LiDAR systems to provide centimetre-resolution images of targets. One such study shows reconstructed images of dToF measurements of a target at a distance of up to 201.5 km [81]. An extended review of the image reconstruction literature for long-range targets imaged by SPAD LiDAR systems (including systems other than scanning types) is included in [54].

A major challenge in real-time display of SPAD scanning LiDAR imagery is the lengthy data acquisition time required by the imaging system to accumulate sufficient photon counts for effective image reconstruction [124], especially to achieve centimetre depth resolution [106]. For instance, reconstructing an image of a 201.5-km target [81] requires a total data acquisition time of approximately 8.5 hours due to the long acquisition time per pixel. This extended acquisition time prompts the literature to explore methods for reducing data acquisition times.

One common approach to reducing data acquisition time is the development of algorithms capable of reconstructing images from data collected by a SPAD scanning LiDAR that have sparse photon returns [175, 81, 80, 36, 15, 30, 56, 31, 16, 14]. These algorithms aim to use fewer photon counts for image reconstruction, thus requiring

less data, which reduces acquisition time. However, despite the reduced acquisition time, image-reconstruction algorithms that utilise this type of data often have long execution times (in the order of 10–100 seconds), rendering them ineffective for real-time display of SPAD scanning LiDAR imagery [158]. Nevertheless, recent advancements have demonstrated real-time reconstruction of images of moving targets [36], showing promising progress in reducing acquisition time.

Another approach to reducing acquisition time is to optimise the data acquisition protocol of single SPAD scanning LiDAR systems, where the authors in [73, 55] offer two different protocols. Alternatively, instead of using a single SPAD detector, one study uses linear SPAD arrays instead in the SPAD scanning LiDAR system, which reduces acquisition time by acquiring data for multiple pixels simultaneously and requiring less scanning [85].

Beyond reducing acquisition times, the image reconstruction literature explores improving the depth resolution of dToF values in reconstructed images of SPAD scanning LiDAR system. The authors in [74, 146] demonstrate that the depth resolution can be enhanced to sub-picosecond timing precision.

Another research area in the literature focuses on improving the quality of reconstructed images in environments where the dToF measurements have a low Signal-to-Noise Ratio (SNR) in addition to having sparse photon returns [172, 171, 121, 46, 127]. The algorithms developed for dealing with this type of dToF measurements can be applied in scenarios where a SPAD LiDAR system with a low-powered laser images long-range outdoor targets under bright sunlight, where sunlight-generated noise may overwhelm the sparse photon returns [127]. Additionally, this type of algorithm can be applied for image reconstruction of targets obscured in scattering environments such as fog and smoke. The literature related to SPAD LiDAR image reconstruction in scattering environment is discussed in Section 2.3.

Recently, a paper by authors in [32] has proposed a real-time image-reconstruction algorithm that processes single SPAD scanning LiDAR data. However, when it comes to evaluating on empirical data, it has only been demonstrated for a short-range target at an imaging distance of 2 m.

2.1.2 SPAD Array Flash LiDAR

Image reconstruction of dToF measurements collected by SPAD array flash LiDAR systems is also a subject of investigation in the literature. Similar to SPAD scanning LiDAR systems discussed earlier, image reconstruction has been demonstrated for this system's dToF measurements of long-range targets [87, 173, 176, 84, 29, 174]. An extended review of image reconstruction literature for long-range targets imaged by all types of SPAD LiDAR systems is included in [54].

A primary challenge in the reconstructed images from SPAD array flash LiDAR system is its lower pixel resolution compared to SPAD scanning LiDAR systems. The main approach to resolving this issue is in the development of Complementary Metal-Oxide-Semiconductor (CMOS) silicon SPAD planar arrays, where there is significant research in increasing the number of SPAD detectors (pixels) within a

microchip to enhance spatial resolution [34]. The advancement of SPAD planar array research only experienced a notable acceleration in the early 2000s when it became possible to use standard CMOS silicon technologies for its fabrication [25]. The initial CMOS silicon SPAD planar array had a resolution of 4x8 [129, 34].

Two major challenges impede the creation of large planar arrays. Firstly, a higher power consumption is necessitated for transmitting digital signals to all pixels' timing circuitry in a large planar array of Single Photon Avalanche Diodes (SPADs) [34]. Secondly, when accommodating a significant number of SPAD detectors, the pixel area, initially dedicated solely to SPAD detectors, must now also house digital circuitry for SPAD operation [34]. This diminishes the fill factor of the SPAD detector in the pixel area, thereby reducing the probability of photon detection by the SPAD. For further exploration and discourse on CMOS silicon SPAD planar array development, comprehensive reviews can be found in [166, 122, 34, 50].

In addition to the pursuit of developing planar arrays with increased numbers of SPAD detectors, the SPAD LiDAR community also explores other avenues to increase the pixel resolution of SPAD array flash LiDAR images. One line of effort investigates conducting scanning with the planar array in the image plane to enhance pixel resolution [135, 87, 168, 169, 93]. Another line of effort is in increasing pixel resolution via image reconstruction [115, 59, 160, 22, 133, 158], which is a well-researched problem in computer vision [158]. In particular, the authors in [115] have demonstrated a real-time-capable super-resolution image-reconstruction algorithm, where the algorithm's execution time matches or is less than the acquisition time of the input batch of frames.

Despite the lower pixel resolution, SPAD array flash LiDAR systems boast faster data acquisition times than SPAD scanning LiDAR systems [153]. Therefore, when it comes to achieving real-time display of a SPAD array flash LiDAR imaged scene, data acquisition is not creating the bottleneck; it is the image reconstruction instead [153]. Performing fast image reconstruction becomes more difficult with the development of higher resolution SPAD arrays, which generate larger amounts of output data with each laser pulse cycle, adding to the processing load for image reconstruction algorithms [94]. Therefore, real-time image reconstruction of data collected by a SPAD array flash LiDAR system is an area of research in the SPAD LiDAR community.

Among the few studies that have investigated real-time image reconstruction of SPAD array flash LiDAR dToF measurements, most of them demonstrate that their algorithms are real-time-capable. These real-time-capable algorithms have not been demonstrated via real-time implementation, as their execution times are typically measured offline using collected dToF measurements. The algorithm proposed in [79] demonstrates real-time capability, utilising a graphical processing unit (GPU) for parallelisation and achieving high reconstruction rates for dynamic scenes. The algorithm can produce 500 or 1,000 reconstructed images per second, depending on the scene's complexity. Regardless of the scene, the algorithm is able to reconstruct one image from the amount of data that can be collected over the time of one execution of the algorithm.

Another approach to enabling real-time image display of data collected by a SPAD

array flash LiDAR system is to engineer the SPAD array camera to output a more compact form of raw data. This reduction in data volume may alleviate the computational burden on image reconstruction algorithms, enabling faster execution. For instance, the authors in [63] designed a SPAD chip that outputs either a null result or the timing measurement of a detected photon. This compact form of data output has facilitated real-time and real-time capable imaging of underwater targets in existing studies [94, 124]. The algorithms employed in these studies overcome challenges associated with dToF measurements collected from an underwater environment and they will be discussed in detail in Section 2.4.

Recently, the authors in [123] proposed a real-time capable image reconstruction algorithm which is demonstrated to create a reconstructed image from the same SPAD LiDAR data collected by the authors in [51]. The SPAD LiDAR data is collected by a reconfigurable 256×256 SPAD chip. The chip is different to the traditional SPAD planar arrays; its details and differences are presented at the beginning of Section 2.5.1. Their proposed algorithm consists of image reconstruction and de-noising. For image reconstruction, it uses Maximum Likelihood Estimation (MLE) to create the depth and reflectivity images from the collected data. For de-noising, the authors adapt the multi-scale approach proposed by the authors in [58]. The adaptation is to change the algorithm from requiring a histogram as input to only requiring one depth and one reflectivity image. The multi-scale approach uses the multi-scale versions of these depth and reflectivity images to reject outliers and fill holes in these images. The algorithm is implemented using Compute Unified Device Architecture (CUDA), a parallel programming language, and C++ on a GPU. It is demonstrated to be real-time capable when using 3×5 sized filters to reconstruct images with 64×64 pixels. Its execution time is 0.6 ms and the data acquisition time is 2ms (deduced from 500 fps).

Detailed discussions of image reconstruction algorithms for dToF measurements collected by SPAD LiDAR systems in other types of environments are provided in subsequent sections, focusing on addressing challenges introduced by occlusions or obscurants.

2.2 Partially Occluded Environments

The SPAD LiDAR community has shown interest in image reconstruction of dToF measurements of targets in partially occluded environments. Examples of these are semitransparent materials such as windows [144] and distributed reflective media such as foliage [87] and camouflage net [61]. In these environments, multiple surfaces may reflect photons to the same SPAD pixel, creating multiple peaks in the histogram of dToF measurements [170]. For instance, a study has demonstrated image reconstruction of a single SPAD scanning LiDAR's dToF measurements of a mannequin through a partially reflective material similar to a window [144]. Another study has demonstrated image reconstruction of a similar single SPAD scanning LiDAR system's dToF measurements of a mannequin placed behind a wooden fence [170].

Limited literature exists on reconstructing images of SPAD LiDAR dToF measurements collected through foliage. In one study [87], the authors presents an image of an obscured mannequin head that is reconstructed from dToF measurements obtained by a SPAD array flash LiDAR system. However, there is no standardisation for measuring the density of foliage, making direct comparisons challenging. Another study [64] used a single SPAD scanning LiDAR system to collect dToF measurements of a person behind foliage.

For camouflage net, the authors in [153] presents a real-time-capable image-reconstruction algorithm, which reconstructs scenes with moving people behind the net using a SPAD array flash LiDAR system. The algorithm reconstructs an image every 20 ms for dynamic scenes, with each reconstruction using a batch of 3,008 frames, which also takes 20ms to acquire. This synchronous processing renders the algorithm real-time-capable. Additionally, the reconstructed images exhibit an increased resolution of 96x96, achieved through the application of super-resolution as part of the reconstruction process. Leveraging parallelisation computational techniques, the algorithm is implemented on a GPU to enable a short execution time.

Offline image reconstructions of stationary targets behind camouflage net are reported in [155, 152, 61], with various reconstruction algorithms employed and their offline reconstruction performances compared. These algorithms' performance are reported for reconstruction of dToF measurements of mannequin heads or a person, situated within 1 m of the camouflage net. The authors in [61] proposed the L_{21}+TV approach while the authors in [155] utilised the L_{21}+TV approach, and the authors in [152] proposed the ManiPoP approach. In [152], the authors concluded that ManiPoP provided superior reconstruction compared to L_{21}+TV, with performance assessed based on the visual quality of the reconstructed image.

The next section explores the reconstruction of SPAD LiDAR dToF measurements collected through fog or smoke, which are obscurants potentially occluding the target.

2.3 Imaging Through Fog and Smoke

A branch of SPAD LiDAR literature focuses on reconstructing images of targets obscured by scattering environments such as fog [158] and smoke [147]. These environments exhibit high levels of scattering, which adversely affects the propagation of LiDAR laser pulses [157], resulting in sparse photon returns and lower SNR in the reconstructed image [60]. Image reconstruction of dToF measurements collected through fog and smoke is particularly relevant for improving remote sensing, as conventional cameras struggle to produce clear images in such environments without extensive processing [157]. A significant review is be provided in this section on this topic because it is central to the contributions of this thesis.

Currently, there is limited image-reconstruction literature for dToF measurements collected through smoke using SPAD LiDAR systems. Existing research reconstructs dToF measurements of an indoor scene obscured by white canister smoke [157] and

measurements of outdoor targets obscured by piezo-ignited smoke [147], where outdoor targets are obscured by piezo-ignited smoke. Additionally, the authors in [181] reconstructed images of smoke-obscured targets in an indoor lab environment, but details on smoke generation are unclear. Existing image-reconstruction literature related to smoke-obscured targets employs similar reconstruction techniques for dToF measurements as image-reconstruction literature for fog-obscured targets, such as the algorithms investigated by [157]. Therefore, this section discusses image reconstruction techniques for dToF measurements collected through smoke and fog collectively.

2.3.1 Performance Metrics

In the current literature, the design of image-reconstruction algorithms is independent of the SPAD LiDAR system configuration. Reconstruction performance is primarily assessed based on the ability to use the reconstructed image to visually distinguish targets. Authors often describe reconstruction capability in terms of the maximum number of attenuation lengths (ALs) at which the target is discernible in the reconstructed image. Number of ALs is a measure dependent on wavelength, as demonstrated by the authors in [158]. The calculation of this value is described in detail in Section 3.6.1 of Chapter 3. Most literature present the number of ALs for a maximum of two different wavelengths, with one of these wavelengths being the one used by the SPAD LiDAR system in the experiments reported by the literature. This value cannot be calculated for an alternative wavelength based solely on the number of ALs provided for a specific wavelength, as the calculation requires either measuring the power or knowing the beam attenuation coefficient of that wavelength in the specific imaging environment.

Quantifying the quality of reconstructed images lacks a standardised metric and often relies on pixel-wise assessments. One common approach is to evaluate the percentage of pixels within a predefined threshold distance from the ground truth value. This is employed in different formats in [158] (using depth absolute error (DAE)) and [183] (employing Target Recovery (TR)). Another approach is a geometric metric that involves computing the root mean square error (RMSE) between pixel values and their corresponding ground truth values. It is referred by different names; [86] discusses it as the average error range while [181, 183] refers to it as relative average ranging error (RARE). Signal processing perspectives are also utilised to quantify the reconstruction quality, with metrics such as Signal-to-Reconstruction Error (SRE) applied in [157] and Peak Signal-to-Noise Ratio (PSNR) employed in [70]. Additionally, for algorithms employing statistical modelling for a collection of dToF measurements, some produce an uncertainty map to provide the standard deviation of the fitted statistical distribution [158].

2.3.2 Modelling each Pixel's dToF Measurements with a Gamma Probability Distribution

A prevalent approach in reconstructing images of obscured targets is to model each pixel's distribution of dToF measurements as a per-pixel probability distribution. The model in each pixel is used to inform the new depth value for the reconstructed image.

Many studies utilise a gamma distribution to model the probability distribution of a pixel's dToF measurements associated with photons detected from fog or smoke [139, 86, 181, 183, 182]. There are various algorithms within this modelling framework to extract the target's location at each pixel, forming the dToF value for the reconstructed image at the corresponding pixel. The most common approach involves estimating a probability distribution from the dToF measurements in a pixel and fitting a gamma distribution to it using MLE. Subsequently, the gamma distribution is subtracted from the estimated probability distribution. A Gaussian distribution is then fitted to the residual probability distribution, with its mean representing the new location of the target, which becomes the dToF value of the pixel in the reconstructed image [139, 86, 181].

The physics model of photon propagation and interaction with atmospheric particles provide a rationale for the use of gamma distributions to model the dToF measurements of detected photons reflected from fog or smoke. A histogram of dToF measurements can be interpreted as the probability of a certain number of photon detections at various distances. The major factor that influence the probability of photons from each distance to reach the SPAD camera after multiple scattering in the obscurant is the probability of photon transmission through fog or smoke [181]. Specifically, a photon's transmission time can be modelled as the sum of scattering event times, with the probability of an individual scattering event being modelled by an exponential distribution [181]. The probability of photons transmitting through fog or smoke is the sum of the probabilities of their total number of scattering events [181]. Since the sum of any number of independent exponential random variables is gamma distributed, the probability of photons transmitted through fog or smoke follows a gamma distribution [181]. Therefore, the probability distribution of dToF measurements of detected photons reflected from fog or smoke can be modelled by a gamma distribution. A more detailed mathematical derivation explaining why the gamma distribution is used can be found in [181]. Despite the authors in [181] focusing on smoke in their modelling, the derivation applies to fog as well because the equation for modelling scattering is the same, except for differences in the attenuation coefficient values.

The differences between image-reconstruction algorithms that use the probability of dToF measurements play a crucial role in their performance and applicability. The first application of this approach by the authors in [139] laid the groundwork, utilising MLE for fitting gamma and Gaussian distributions. The authors in [181] found this algorithm found this algorithms from [139] can reconstruct up to 3.6 ALs (based on 1,064 nm) of smoke, which is higher than the equivalent number of ALs

found by the authors in [139]. However, the algorithm is not real-time capable due to its execution time being longer when compared with the data acquisition time. The authors in [86] used a similar approach. Rather than employing the approach of [139] in estimating two different parameters called shape and rate parameters, the authors in [86] estimated only the shape parameter, leveraging the attenuation coefficient of fog to calculate the rate parameter. However, the authors in [181] found that this algorithm from [86] can only reconstruct up to 1.2 ALs (based on 1,064 nm) in smoke, which is at a lower fog level than the algorithm in[139]. The authors in [181] refines this algorithm from [86] by integrating Continuous Wavelet Transform (CWT) for rate parameter estimation, achieving enhanced reconstruction capability up to 3.6 ALs in smoke at 1,064 nm. Despite its advancements, the smoke generation process is unclear from [181].

In [183], the authors presents a refined approach to the utilisation of gamma distribution fitting by performing the process three times. MLE is first applied to fit a gamma distribution to the dToF measurements (which they refer to as the original histogram data), aiming to approximate the fog distribution and use the residual dToF measurements to calculate a preliminary depth estimation of the target. These preliminary estimations serve multiple purposes, including background noise suppression and correction of the dToF measurements for the SPAD's pile-up effect. The second fitting of a gamma distribution uses MLE for better estimation. Then, the least squares method is used for the third fitting of a gamma distribution. This iterative refinement process enables the reconstruction of a target up to 1.4 km away under 1.7 km visibility in outdoor fog conditions, showing the algorithm's robustness and effectiveness in challenging real-world scenarios.

The authors in [184] employ the algorithm described in [183] within a multi-scale framework. This approach combines histograms of groups of pixels' dToF measurements into a single histogram of dToF measurements to create superpixels for each dataset at each level of the multi-scale framework. The number of pixels in each group is determined by the number of pixels that constitute a superpixel at each level. The algorithm developed by the authors in [184] involves three steps applied to each dataset at every level of the multi-scale framework.

Firstly, the algorithm from [183] is applied to obtain the target's depth and reflectivity, as well as the dToF measurements' associated histogram peak's echo pulse width. Secondly, the output from the previous step is analysed to identify spatial similarity features, which are then used to denoise the reconstructed image at each level. Finally, object edges are detected in each image at each level to further denoise the reconstructed images.

The algorithm proposed in [184] demonstrates its capability to reconstruct scenes similarly to [183], with the target being up to 1.4 km away under outdoor fog visibility of 1.7 km. It reconstructs an image with only 800 frames, a significant reduction from the 20,000 frames required in [183], while maintaining similar image quality. This demonstrates that less data acquisition time is required to collect enough frames for image reconstruction using the algorithm proposed by the authors in [184].

In [182], an alternative approach to utilising the gamma distribution is explored.

The authors apply Principal Component Analysis (PCA) and K-clustering to the residual probability distribution after fitting and subtracting the gamma distribution. This algorithm aims to distinguish between different photon detections reflected from various sources such as the target (ship), sea surface, and secondary reflections from both the sea surface and ship. The authors hypothesised that the distribution of dToF measurements of secondary reflections might exhibit a similar probability distribution shape to the dToF measurements from the ship, hence the need for a more sophisticated algorithm to differentiate them. Despite not accounting for fog in their image-reconstruction algorithm, the authors demonstrated this algorithm to reconstruct images of ships that are mildly obscured by sea fog. However, no quantitative measures are provided for the fog level in the scene during imaging.

2.3.3 Modelling each Pixel's dToF Measurements with a Poisson Probability Distribution

The preceding discussion in this section has primarily focused on modelling the probability distribution of each dToF measurement across a batch of frames using a gamma distribution. An alternative approach commonly adopted for image reconstruction involves utilising a Poisson distribution.

The Poisson distribution is derived from a different perspective on photon transmission physics. The gamma distribution models the number of photons from the sum of photon scattering events. In contrast, the Poisson distribution models the number of photon returns directly, where the distribution's mean is defined to be the theoretical number of photons returns that the SPAD camera registers as a function of depth [143].

Using the Poisson distribution for modelling the number of photon returns is commonly employed for analysing dToF measurements collected in environments without obscurants. In such scenarios, where it is assumed that only one surface is present per pixel, the theoretical count of detected photons is computed as the product of the single surface's reflectivity and the SPAD LiDAR's system impulse response at the depth of the surface, with the addition of the background noise and dark counts of the SPAD detector [127, 143, 15].

For scenarios involving multiple surfaces per pixel, the Poisson distribution's mean is defined as the summation of the products of each potential surface's reflectivity and the SPAD LiDAR's system impulse response at the corresponding depth of the surface, along with the background noise and dark counts of the SPAD detector [61, 152, 60].

When utilising the Poisson distribution to model the probability distribution of dToF measurements across a batch of frames acquired in environments obscured by fog, the Poisson distribution's mean parameter is adjusted to incorporate the light propagation characteristics through fog. In the literature, two distinct adaptation approaches have been identified.

In the approach outlined in [137], an adaptation is made to the mean parameter of the Poisson distribution and it is defined as the sum of the mean count of the

target signal and the background noise. The target signal mean is further defined as the convolution of the laser pulse profile with two different terms, each representing the fog and the target's reflectivity and depth, respectively.

The first term encapsulates the fog, expressed as the product of the fog's attenuation factor and a unit step function denoting the depths at which the fog is present. The unit step function is employed under the assumption of a uniform fog distribution. The second term represents the target, calculated as the product of the target's attenuation factor and the Dirac delta function localised at the depth value of the target.

By employing MLE, they fit only the fog portion of the Poisson model to a set of dToF measurements, where they approximate the Poisson distribution as a Gaussian distribution for easier computation. The computation is further aided by an Expectation-Maximization (EM) algorithm to expedite the process. Subsequently, the estimated fog model is subtracted from a separate probability distribution derived from the histogram of the same set of dToF measurements. The Generalised Likelihood Ratio Test (GLRT) is then applied to calculate a likelihood value for each bin of the residual histogram, determining which bin contains the target. [137]

The authors in [137] verified the algorithm's capability to reconstruct images of fog-obscured targets, although the precise measure of the fog level remains unclear. The algorithm is capable of reconstructing one pixel every 50 ms. However, the SPAD scanning LiDAR system necessitates 64 ms per pixel for acquisition. Consequently, although the reconstruction algorithm operates faster than data acquisition, the imaging system's scanning speed renders it impractical for real-time imaging applications.

The second distinct application of the Poisson distribution's mean parameter is proposed for the Median-based Multi-scale Restoration of 3D images (M2R3D) algorithm detailed in [158]. In the M2R3D algorithm, the mean of the Poisson distribution follows a similar definition to the scenario discussed earlier where there is only a single surface without obscurants. Specifically, it is determined as the product of the single surface's reflectivity and the SPAD LiDAR's system impulse response at the depth of the surface. However, instead of incorporating the background noise and dark counts of the SPAD detector, the noise is adjusted to be either the background noise or the noise from detected photons reflected from fog, depending on which value is greater.

The noise attributed to fog is characterised by the tail end of an exponential distribution. This is feasible because the dToF measurements collected by the SPAD LiDAR system in [158] utilises a range gate, which restricts dToF measurements to only a narrow depth range of 3 m. Since the targets being imaged are located at a minimum distance of 50 m away, the SPAD primarily detects photons reflected from further into the fog, rather than the entire fog span along the imaging range.

To fit the model to the data, the authors employ a Bayesian framework with prior distributions of depth, reflectivity and hyperparameters. These hyperparameters are used to preserve the edges of the target in the reconstructed image. The optimisation problem is represented using a Maximum a-Posteriori (MAP) method, incorporating

multi-scale information into the likelihood function. For computational efficiency, a coordinate descent algorithm is utilised.

With this approach, the algorithm demonstrates the ability to reconstruct images for dToF measurement collected in up to 5 ALs of oil-based vapor, with respect to a wavelength of 1,550 nm. Moreover, the algorithm is implemented on a GPU and utilises parallel computing techniques, enabling real-time reconstruction of frames of a moving target obscured by oil-based vapor up to 3 ALs (based on 1,550 nm). These frames are captured using a SPAD array flash LiDAR system. Additionally, the authors show that the algorithm can reconstruct images offline at a rate of 10 images per second, demonstrating real-time capability.

2.3.4 Use of Prior Distributions in a Bayesian Model

The ManiPoP algorithm, introduced in [152], employs a Bayesian model for fitting the data. While not originally intended for reconstructing fog- or smoke-obscured targets, this algorithm was adapted by [60] to reconstruct images of dToF measurements collected in a water-based fog scenario at 2.8 ALs, with respect to a wavelength of 1,550 nm. This algorithm does not use the Poisson distribution's mean to account for effects from fog.

The ManiPoP algorithm structures the reconstruction problem as a Bayesian model and incorporates prior distributions. These prior distributions capture two key properties of the data. Firstly, one prior distribution accounts for spatial correlations between pixels in the image to enhance their depth values. It assumes that pixels belonging to the same surface are generally close in distance range. Secondly, another prior distribution accounts for light propagation properties, assuming that depth values collected for the same pixel should be segregated into groups based on their distances. This assumption arises from the algorithm's design for imaging multiple surfaces, thereby anticipating photon returns from different surfaces to occur at varying depths. To solve the optimisation problem, the algorithm also employs a MAP method, similar to [158]. It leverages the Bayesian framework to estimate the most likely depth values given the observed data and prior distributions.

2.3.5 Use of Regularisation Terms in Optimisation Problems

In contrast to modelling light propagation properties using known probability distributions, an alternative approach is to incorporate regularisation terms directly into the optimisation problem. This approach does not involve estimating the parameters of probability distributions but rather focuses on estimating the depth of the target directly for image reconstruction. This is achieved by designing the optimisation problems to incorporate a Poisson distribution model, which models the number of photons detected at each possible dToF value for each pixel. The Poisson distribution used here is for the case where there are no obscurants, so it only accounts for the depth and reflectivity of the target as well as background noise and dark counts of the SPAD detector. Multiple algorithms employ this technique [60, 61, 56], where the

regularisation terms used in the optimisation problem are what differentiate these algorithms from each other. These regularisation terms are designed to capture the known light propagation properties and characteristics of SPAD LiDAR data.

One such algorithm utilising this approach is the M-NR3D algorithm, as described in [60], which is specifically designed for reconstructing images of fog-obscured targets. Additionally, TV-l_{21} [61] and RDI-TV [56] algorithms, though not originally intended for fog or smoke-obscured targets, have been adapted for this application in [60] and [157]. All of these three algorithms utilise Alternating Direction Method of Multipliers (ADMM) to compute the solution to the optimisation problem.

In M-NR3D, two regularisation terms are employed to account for reflectivity and sparse photon returns [60]. TV-l_{21} utilises regularisation terms for spatial correlation of pixels and to handle the large number of empty bins when the dToF measurements are arranged into a histogram, due to the SPAD camera having a fine depth resolution. By contrast, RDI-TV incorporates total variation (TV) and sparsity of the discrete cosine transform (DCT) coefficients as regularisation terms.

The M-NR3D algorithm, specifically designed for fog-imaging applications, has demonstrated the best performance out of these three algorithms, capable of reconstructing targets obscured by up to 3.8 ALs (with respect to 1,550 nm) in water-based fog [60]. Similarly, TV-l_{21} has been assessed in the same study, demonstrating its ability to reconstruct targets obscured by up to 3.1 ALs (based on 1,550 nm) in water-based fog [60]. RDI-TV has been evaluated in [157], proving its effectiveness in reconstructing targets within similar conditions, up to 3.1 ALs (based on 1,550 nm) of water-based fog or glycol-based smoke.

Out of the three algorithms, only the execution time of RDI-TV has been measured. Its execution time is in the range of tens of seconds, suggesting its impracticality for real-time applications [157].

2.3.6 Use of Convolutional Kernel in Signal Processing Approach

Treating the effect of fog or smoke on light propagation as a convolutional kernel on the system's impulse response presents another approach to reconstructing images of obscured targets [161]. In this algorithm, the histogram of dToF measurements is regarded as a signal of total number of photons detected for each possible distance (i.e. possible dToF measurement), and the authors assume the measured signal is a convolution of the impulse response of their employed SPAD LiDAR system with a power propagation loss model of light in fog. In the power propagation loss model, the authors assume the fog is uniformly spanned across a pre-defined depth range. The impulse response is assumed to be the signal containing the dToF measurements of an unobscured target by their employed SPAD LiDAR system. The aim of this approach is to determine the impulse response signal, which provides the reconstructed dToF value of the target. The authors employed two approaches to compute it. One is employing deconvolution calculations and the other is utilising morphological filters presented by the authors in [72]. The maximum obscurant level that these algorithms can reconstruct images at is unclear from [161].

A similar convolutional kernel approach is adopted in [82], where spatial and temporal effects are represented. Although not specifically designed for reconstructing obscured targets, this approach is applied in [70] for reconstructing a building at 13.4 km in up to 2.7 ALs (with respect to 1,550 nm) of environmental fog. The authors in [70] claim the reconstruction performance should be readjusted to 5.2 ALs (with respect to 1,550 nm) to account for the effect of geometric attenuation. The algorithm's model consists of a 3D matrix containing all pixels' reflectivity and depth values, which is convolved with spatial and temporal kernels [70]. An inverse deconvolution process is applied to determine the dToF values for the reconstructed image.

2.3.7 Machine Learning Algorithms

Machine learning algorithms have been explored for reconstructing images of obscured targets. Specifically, a non-local network for reconstructing images, originally proposed by [121], has been applied in [70] to reconstruct dToF measurements of obscured targets. The network accounts for long-range temporal and spatial correlations to distinguish between objects and noise as part of its image reconstruction process. The network is trained on simulated dToF measurements of unobscured indoor targets by [121] and tested on empirical dToF measurements of outdoor obscured targets by [70]. Although not initially designed for reconstructing fog-obscured targets, the authors in [70] claim that the reconstruction performance of this algorithm is also 2.7 ALs (with respect to 1550nm) with an additional 2.5 ALs from geometric attenuation, using the same imaging scene for testing [82]. Moreover, the network is real-time capable if implemented on a GPU, as demonstrated by [70].

2.3.8 Unmixing Algorithm

All of the discussed algorithms utilise the distribution of dToF measurements for image reconstruction. However, the unmixing algorithm by [127], which has been applied for fog and smoke imaging in [157, 60], does not use distributions as part of its processing. Instead, it processes a list of dToF measurements arranged in order of detection by the SPAD camera. This algorithm employs an iterative cycle of processing to ensure there are enough dToF measurements for depth estimation. Depth is estimated by solving a constrained MLE problem using the list of photon detection times.

The iterative cycle involves windowing to remove dToF measurements that do not belong to a potential cluster of dToF measurements and reflectivity estimation. A variation of superpixels is used to add dToF measurements to pixels that do not have enough photon counts for depth estimation. This superpixels process repeats until all pixels have enough dToF measurements for depth estimation or there are not enough neighbouring pixels available to utilise for adding more dToF measurements to the pixel.

Moreover, the unmixing algorithm incorporates depth information from other

pixels and performs inpainting after depth estimation is calculated. Although not designed for the reconstruction of fog-obscured targets, it has been demonstrated to work in different types of fog-obscured environments [157, 60]. Its highest reconstruction performance is 3.8 ALs (with respect to 1,550 nm) of water-based fog [157]. However, its execution time for reconstructing one image has been measured in [157] to be in the order of seconds (over different numbers of frames, ranging between 156 to 46,800), making it not real-time capable.

In summary, most of the image-reconstruction algorithms proposed in the literature have not been demonstrated to be able to real-time-process live feeds of SPAD LiDAR data. However, a few studies [158, 70] have demonstrated algorithms that are real-time capable. This means each of the algorithms' execution time is equal to or lower than their corresponding SPAD LiDAR system's data acquisition time, such that the algorithm is able to produce reconstructed images in real time if fed with a live stream of frames.

In the next section, literature related to image reconstruction in turbid water is discussed.

2.4 Imaging Through Turbid Water

Turbid water presents another challenging environment for conducting image reconstruction of dToF measurements collected through it using a SPAD LiDAR system. Similar to fog and smoke, turbid water also causes scattering, leading to sparse photon returns [94]. In addition, there is a lower number of detected photons reflected from the target compared with the background [124], which in turn lowers the SNR for the reconstructed image.

Despite turbid water sharing similar scattering properties with fog and smoke, image-reconstruction literature related to turbid water in the SPAD LiDAR community appears to be distinct from that related to fog and smoke. There has been limited literature that applies image-reconstruction algorithms for dToF measurements from fog or smoke to dToF measurements from turbid water, and vice versa.

To the best of my knowledge, the current literature predominantly focuses on the reconstruction of images derived from dToF measurements acquired within controlled freshwater tank environments, particularly under simulated turbid conditions. The common practice entails the utilisation of Maalox as a standard chemical agent for emulating turbidity [89, 88, 92, 95]. Maalox is chosen in the literature because the authors in [88] shown that water with 0.003% of Maalox has a similar attenuation length (AL) spectrum (over wavelengths between 500–900 nm) as 15-hours-settled sea water.

The current image-reconstruction literature related to reconstructing dToF measurements of water environments are evaluated with data collected by different SPAD LiDAR systems. This includes single SPAD scanning LiDAR systems [88, 89, 91, 57, 90], linear SPAD array scanning LiDAR system [92], and SPAD array flash LiDAR system [91, 95, 94, 138]. Traditionally, dToF measurements are collected with these

systems positioned external to the water tank while the target remains submerged. However, recent advancements have led to submerging these systems within the water tank for collecting dToF measurements of submerged targets [94].

Similar to image-reconstruction algorithms created for fog or smoke environments, image-reconstruction algorithms for turbid water environments are also assessed in terms of the maximum number of ALs at which the target is discernible in the reconstructed image. Section 3.6.1 of Chapter 3 describes how the number of ALs can be calculated. The number of ALs is a measure dependent on wavelength, as demonstrated by [88]. In the existing literature, most related literature employs a wavelength in the visible light range for the SPAD LiDAR system. Most literature related to imaging in turbid water present the number of ALs for the wavelength used by the literature's employed SPAD LiDAR system. The AL value cannot be calculated for an alternative wavelength based on the number of ALs provided for a specific wavelength, as the calculation requires either measuring the power or knowing the beam attenuation coefficient of that wavelength in the specific imaging environment.

Currently, image reconstructions can achieve a quality where the target is discernible for up to 9.2 ALs (with respect to 690 nm) with single SPAD scanning LiDAR systems [89], up to 8.3 ALs (with respect to 670 nm) with linear SPAD array scanning LiDAR systems [92], and up to 7.5 ALs (with respect to 532 nm) with SPAD array flash LiDAR systems [94].

For real-time image reconstruction, the literature proposes algorithms capable of real-time processing of dToF measurements of underwater targets collected by a SPAD array flash LiDAR system [94, 124], as mentioned at the end of Section 2.1.

In [124], the authors demonstrate the real-time capability of their algorithm for reconstructing dynamic scenes with targets moving in water. They employ a 128×128 SPAD array flash LiDAR system to capture the scene. Leveraging parallelised computational techniques and GPU implementation, the algorithm achieves execution times ranging from 1 to 6.3 ms on simulated data of a 128×128 SPAD array flash LiDAR system, where the time variation depends on the number of employed filters. Processing a batch of 64 frames at a time, the execution time is shorter than the data acquisition time of 128 ms.

Real-time image reconstruction of live SPAD array flash LiDAR dToF measurements has been demonstrated by [94], which, to the best of the author's knowledge, is one of the few studies that demonstrate real-time reconstruction of live SPAD LiDAR data. In [94], the authors develop a 192×128 SPAD array flash LiDAR system designed for underwater imaging. By porting existing image-reconstruction algorithms onto a GPU for real-time processing, they evaluate three different algorithms: cross-correlation, RT3D [153], and ensemble method [44]. All three algorithms can be parallelised. Cross-correlation and ensemble method can naturally be adapted for parallel computation due to their pixel-wise processing approach, while RT3D incorporates point cloud denoising techniques from the computer vision community to enable parallel computation. These algorithms are reported by the authors in [94] for their real-time reconstruction performance of dynamic scenes, including a T-

connector submerged in water rotating in two different orientations. The algorithms' execution times across different levels of obscurants in the water (ranging from 0.5 to 6.6 ALs) varies between 0.91 to 1.27 ms for cross-correlation, 28.14 to 34.67 ms for RT3D, and 29.47 to 35.23 ms for the ensemble method. The authors in [94] found that higher obscurant levels resulted in reduced execution times. Importantly, all these execution times are below the 50 ms data acquisition time for 50 frames, enabling real-time reconstruction at a speed of 10 frames per second using the built SPAD array flash LiDAR system.

The literature overview related to image reconstruction in the SPAD LiDAR community concludes here. The subsequent section discusses literature proposing classification algorithms for SPAD LiDAR data.

2.5 Higher Levels of Target Discrimination

Besides conducting image reconstruction for target detection, algorithms for higher levels of target discrimination (i.e. orientation, classification, recognition, identification) are developed for data collected by SPAD LiDAR systems. For these higher levels of target discrimination, the current literature mainly focuses on single targets imaged by SPAD LiDAR systems without the presence of obscurants.

The primary performance metric used in the literature for these higher levels of target discrimination algorithms is overall accuracy. However, for target classification algorithms that specifically uses machine learning, some authors include additional performance metrics adopted from the machine learning community. These metrics include precision, recall, and F1-measure, which are employed in [136, 113]. A background explanation of these target classification metrics is provided in Section 3.7 of Chapter 3.

The design of these higher levels of target discrimination algorithm remains independent of the SPAD LiDAR system configuration. Typically, these algorithms begin by collecting dToF measurements from the target, then the measurements are processed for image reconstruction to generate depth and, in some cases, reflectivity images. This is followed by image processing techniques to transform these depth and reflectivity images into discernible representations of the target, where the target can be detected in these images. Subsequently, these images are used for high levels of target discrimination.

When it comes to reconstructing images of unobscured targets for higher levels of target discrimination, the widely used basic technique of the TCSPC is applied to reconstruct images. In [117], only reconstructed depth images are utilised for classification, by using the TCSPC technique. The reconstructed dToF value is only used in the reconstructed depth image only if the number of photons detected exceeds a certain threshold. Each reconstructed depth image is generated from 15–50 frames [117]. Alternatively, some approaches involve SPAD LiDAR systems outputting an array of histograms of dToF measurements for each pixel, with TCSPC applied on-chip [134]. In such scenarios, depth and reflectivity images can be automatically

generated from these histograms.

One classification approach bypasses TCSPC to reconstruct depth images, directly utilising SPAD LiDAR data to produce a discernible image. This involves aggregating all dToF measurements from all pixels into a histogram to discern the distances of the target, background, and foreground. Subsequently, these distance values are utilised during the image-processing step to reconstruct a discernible depth image of the target for classification [136].

The approach proposed by the authors in [142] diverges from utilising reconstructed images for classification. Instead, it explores an event-based 8×8 SPAD LiDAR system, and demonstrates classification capability on a set of simulated 8×8 event-based SPAD LiDAR data of digits 0–9. The classification algorithm for this data type employs a neuromorphic system [142].

Various image-processing techniques are applied to make the target more discernible in the reconstructed image before it is used for target classification. In [117], the author employs coordinate transformation on the reconstructed depth images to generate images of the same targets but captured from a different angle. Additionally, silhouette images are generated from these transformed depth images to provide additional data for classification [117]. The authors in [136] apply various image-processing techniques, including a shot noise removal algorithm, to create a single 2D binary mask of the target from the reconstructed depth image [136]. Another approach involves employing neural networks. In [134], the author trains a network to enhance the resolution of a set of 4×4 histogram data into a set of 32×32 histogram data for classifying different human poses.

For techniques that perform higher levels of target discrimination, machine learning is a prevalent approach [136, 134]. In [136], authors evaluated a 2D image Convolutional Neural Network (CNN) and a 3D image CNN, achieving 95% average precision and recall for categorising three different targets (airplane, chair, and Uncrewed Autonomous Vehicle (UAV)). Each target is imaged under various indoor and outdoor conditions, with the data having an image resolution of 64×64 pixels. However, the extensive data processing involved to create a discernible image for target classification renders this approach non-real-time capable. Another machine learning approach is presented in [134], where it conducts classification of different human poses, using a neural network trained on 32×32 pixel-wise histogram data (from previous image processing) and reflectivity data collected by the same 4×4 SPAD sensor. The author integrates this neural network with the image-processing network for training, creating three networks based on the number of people's poses being estimated (up to 3) and training them with either 7,000 or 9,500 frames, depending on the number of people. Additionally, the authors demonstrate real-time pose estimation, with the total execution time (including image processing and pose estimation) approximately 0.134 seconds per frame, although it is unclear for which of the three networks this measurement is taken.

In contrast, a different machine learning approach is tested in [117]. PCA is utilised to extract target features, and then the authors investigated three different target classification algorithms: Gaussian Naïve Bayes classifier, nearest-neighbour

classifier, and CNN. The CNN achieves the highest classification accuracy of 86.3%.

The closest example of higher levels of target discrimination of obscured targets imaged by a SPAD LiDAR system is [182]. In [182] (discussed in Section 2.3, the authors utilise PCA and K-clustering algorithms to classify different waveform returns received from the scene. This classification is tested to differentiate detected photons reflected from the ship compared with secondary reflections of the ship from the sea surface or direct reflections from the sea surface.

The authors in [66] developed a machine learning algorithm that deviates from this traditional structure. The authors utilise a deep learning network to perform target discrimination on the SPAD LiDAR data directly, specifically to determine target orientation and perform target classification. The input data to the deep learning network is in the form of histograms of dToF measurements. The authors train a deep learning network using two different synthetic SPAD LiDAR datasets: one comprising 45 different classes of targets, and the other consisting of 24 different view angles of a target "bunny". The deep learning network's classification accuracy of two targets (jet vs bunny) imaged at 3 m varies between 67% and 100%. For evaluating the algorithm in determining the orientation of the target, the algorithm's accuracy in determining different imaged perspectives of a bunny in the laboratory ranges from 64.58% to 100%. Both classification and orientation accuracies depend on the different SNR levels of the collected data. When used to determine different orientation of a flying drone imaged outdoors at a distance of 200 m, the classification accuracy is 79.20%.

2.5.1 Algorithms using Data from a Reconfigurable 256×256 SPAD Chip

Recently, there has been an influx of new algorithms for higher levels of target discrimination [114, 112, 113, 140] designed for targets imaged by a SPAD LiDAR system [53] that uses a reconfigurable 256×256 SPAD chip [69]. This chip can be configured to detect multiple photon events per laser pulse cycle, marking a departure from the conventional SPAD LiDAR setup where only the first photon detection is registered per laser pulse cycle [51, 113]. The reconfigurable nature of the chip allows it to group its 256×256 SPAD pixels into 4×4 groups, effectively converting the detector into a 64×64 array of macropixels that can detect multiple photons in one laser pulse cycle. Photon detection from each SPAD detector in the 4×4 macropixels is combined to provide a single histogram of dToF measurements, thereby generating a multi-event histogram for each of the 64×64 macropixels [69, 51]. This capability to detect multiple photons in one laser pulse cycle introduces significant differences in the histogram's behaviour, especially in high ambient light conditions [51]. The new higher-levelled target discrimination algorithms based on machine learning neural networks take advantage of this unique feature of the reconfigurable SPAD chip, representing a departure from the traditional structure discussed in the previous section for the higher levels of target discrimination for SPAD LiDAR data.

In addition to multi-event histogramming, this reconfigurable 256×256 SPAD chip introduced by [69] can be reconfigured for photon counting mode. In this mode,

it generates a 256×256 reflectivity image. In addition, this chip can seamlessly switch between this photon-counting mode and the timing frames (which produces multi-event histograms) for high-speed imaging. This means reflectivity and depth images can be obtained concurrently by using this chip. This is demonstrated by the authors in [51, 52] with imaging events such as a fast-rotating fan, man juggling a ball indoors and outdoors, an apple shattered by a hammer [51], balloon bursting and a ball thrown into a plate of milk [52].

The algorithms proposed by the authors in [140] for target identification and determining the target's orientation take advantage of this mode-switching capability to access both photon counting and multi-event histogram information. Specifically, the proposed neural network requires only the multi-event histogram of dToF measurements and the reflectivity images as input, eliminating the need for creating a reconstructed depth image to conduct target discrimination. This improves computation efficiency.

Another feature of this 256×256 SPAD chip is its ability to be configured to utilise only half of its SPAD pixel rows. Consequently, the chip operates with a reduced array size of 256×128 SPADs instead of its full 256×256 array. This configuration results in images having 64×32 macropixels instead of 64×64, where each macropixel still consists of 4×4 SPADs [112, 113]. Some classification algorithms [112, 113] exploit this feature to enhance computational speed, as less time is spent on data throughput. This represents an effective compromise, trading off reduced image resolution and field-of-view (FOV) for faster classification speed.

These recent higher-levelled target discrimination algorithms have found applications in various domains, including classifying hand gestures under different lighting conditions and background objects [112], human activity recognition (e.g. running, waving, jumping, standing up) [114], identifying drone type, and determining its orientation [140], and classifying six different targets (e.g. bucket, chair, duck, football, box, statue) [113]. Some algorithms [112, 140] have achieved overall accuracy above 90% when utilising both the multi-event histogram and reflectivity data. The authors in [112] claim their network's execution time is under 25 ms for each frame, with image acquisition time in the millisecond range. Therefore, the proposed classification regime by [112] qualifies as a real-time-capable system, meaning it has not demonstrated real-time classification for live feed of images, but has the capability to do so.

Alternatively, [113] found that using solely the multi-event histogram data resulted in better overall accuracy, albeit the improvement being relatively small compared with using both the multi-event histogram and reflectivity values. Therefore, the decision to use reflectivity values will depend on the specific context in which the network is deployed. The use of reflectivity values may lead to a smaller neural network in scenarios where the number of bins in the structure of the histogram data exceeds 16 bins [113]. The authors claim that the execution time of their network, with or without using reflectivity values, is less than 1 ms per frame. Even when including the data acquisition time from the SPAD LiDAR system, the total data acquisition and execution time remains in the millisecond range. Consequently, this

classification network and the SPAD LiDAR system proposed in [113] constitute a real-time-capable classification system.

Recently, the authors in [114] presented a real-time-capable classification system that uses solely the multi-event histogram data. This system utilises a deep learning network to classify seven different classes of human activity, such as crouching, idle, waving, and running. The network consists of two parts: the first part segments the human figure from a 64×32 SPAD LiDAR reconstructed depth image, while the second part classifies the human activity.

Synthetic SPAD LiDAR data, in the format of dToF measurements instead of multi-event histograms, is used to train the network. The segmentation part of the network is trained on 80,000 examples of reconstructed depth images from scenes containing one or two humans and various objects at different locations, with 10% of the data used for validation. For the classification part, 7,600 reconstructed depth image sequences are used, with each sequence showing a human performing one class of activity. Similarly, 10% of this set is used for validation. The deep learning network is trained and tested on a desktop computer with GPU assistance and processes a set of 32 frames for each classification.

The empirical data configures the 256×256 SPAD chip to collect data at 64×32 pixel resolution. The imaging scene consists of a human and other objects in an outdoor environment at a stand-off distance of 30–40 m. The data acquisition speed is 50 frames per second. The deep learning network is tested on 1,237 batches of 32 frames, achieving a weighted average accuracy of 89%. Some misclassifications occur due to some sets of frames containing the human transitioning between two classes of activities. The network processes data at 66 frames per second with GPU assistance, making it real-time capable. However, this is contingent on each block of 32 frames being resampled from 64 raw frames, with the first half of the 64 raw frames extracted from the latter half of the previous block of 64 raw frames.

This concludes the overview of the related literature for the application of dToF measurements from SPAD LiDAR systems for reconnaissance purposes. A summary of the literature discussed in this chapter is presented in Table 2.1 in Section 2.6 below.

2.6 Chapter Summary

An overview of the related work discussed in this chapter is provided in Table 2.1. It organises the literature discussed by category. The table is based on the different imaging environments, level of target discrimination, type of SPAD LiDAR system, and whether the literature demonstrates the algorithm to be real-time capable, which includes the algorithm having been implemented in real time, or the algorithm has only been demonstrated to operate in an offline computational environment. The cells highlighted in blue in Table 2.1 are the research areas directly contributed to by this thesis. Entries in bold represent my own published articles, as listed in Section 1.1 of Chapter 1.

Table 2.1: Overview of literature reviewed in this chapter.

Type of Target Discrimination	**Image Reconstruction for Target Detection**		**Target Orientation, Classification, Recognition, Identification (Section 2.5)**	
	offline	**real-time or real-time capable**	**offline**	**real-time or real-time capable**
No Obscurants (Section 2.1)	*SPAD array flash LiDAR:* [87, 173, 176, 84, 65, 128, 28, 174, 29, 115, 93, 160, 22, 59, 63] *single SPAD scanning LiDAR:* [175, 172, 171, 32, 179, 81, 82, 80, 36, 120, 106, 27, 105, 75, 76, 74, 146, 73, 55, 81, 80, 36, 15, 30, 56, 31, 16, 14, 46, 121, 127, 170, 144, 152, 13] *linear SPAD array scanning LiDAR:* [85]	[36, 79, 123]	[66, 117, 142, 136, 140]	**[102, 178]** [114, 134, 112, 113]
Foliage (Section 2.2)	[87, 64]			
Camouflage Net (Section 2.2)	**[100]** [155, 152, 61]	[153]		
Smoke (Section 2.3)	[147, 181, 157, 184]			
Fog (Section 2.3)	[139, 86, 183, 182, 184, 137, 158, 60, 60, 157, 161]	**[100, 99, 104]** [158, 70]	[182]	**[100, 99]**
Water with artificial obscurants (Section 2.4)	[88, 89, 91, 57, 90, 92, 95, 138]	[94, 124]		
Natural Waters (Ocean or Rivers) or water with natural water obscurants		**[101]**		**[101]**

Chapter 3

Background

This chapter introduces the key concepts used in this thesis. It starts with an explanation and motivation for using the various elements in a Single Photon Avalanche Diode (SPAD) array flash Light Detection and Ranging (LiDAR) system for collecting 3D empirical data reported in this thesis (Section 3.1). Next, the operation of the SPAD array flash LiDAR systems is explained (Section 3.2). Section 3.3 covers Time-Correlated Single-Photon Counting (TCSPC), a fundamental technique underlying all the algorithms presented in this thesis. The technical specifications of the SPAD array flash LiDAR systems are provided in Section 3.4. Then, Section 3.5 clarifies the version of the Beer-Lambert Law employed in this thesis to calculate the beam attenuation coefficient. Metrics for measuring the level of obscurant in water and fog are outlined in Section 3.6, including the number of attenuation lengths (ALs) (Section 3.6.1), Secchi Disc (SD) depth (Section 3.6.2), and visibility (Section 3.6.3). Finally, Section 3.7 presents the performance metrics for evaluating different target classification algorithms. While SI units are used throughout the thesis, other units may occasionally appear in some equations.

3.1 Motivation for Using a SPAD Array Flash LiDAR System for 3D Imaging

All data that are presented in this thesis are collected by a Complementary Metal-Oxide-Semiconductor (CMOS) silicon 32×32 SPAD rectangular planar array within a 3D flash direct Time-of-Flight (dToF) LiDAR system. This thesis refers to this type of system as SPAD array flash LiDAR system.

All imaging in this thesis takes place in the visible range. Despite SWIR exhibiting less attenuation through fog compared with the visible range [158, 157], SWIR radiation cannot penetrate natural water environments. Therefore, the laser wavelengths reported in this thesis are in the visible range because of its broad applicability for imaging through fog and natural waters.

LiDAR obtains 3D measurements of objects using the Time-of-Flight (ToF) technique. There are two types of ToF techniques: direct Time-of-Flight (dToF) and indirect Time-of-Flight (iToF) techniques [38]. The distinction between these two

techniques is seen in how the arrival time of the return light signal is recorded. In dToF, an accurate timer measures the ToF from the emission of the laser pulse to its detection upon return. Figure 3.1 in Section 3.2 illustrates a schematic representation of the laser pulse propagation path for a dToF system. By contrast, iToF calculates the ToF by measuring the phase shift of the return signal relative to the emitted signal [38].

dToF is used for the LiDAR systems in this thesis. This is because the range accuracy of iToF measurements diminishes with range and sunlight, whereas dToF measurements maintain high accuracy for imaging distances extending up to hundreds of kilometres, even in the presence of sunlight [113].

The LiDAR imaging results that are reported in this thesis rely on a SPAD camera. This is because SPAD's single photon sensitivity makes it a strong candidate for sensing through obscurants, which is in the imaging scene of a majority of the empirical SPAD LiDAR data presented in this these. Its extremely high sensitivity also means that the LiDAR system can use a laser with a lower Size, Weight and Power (SWaP) requirement [120]. The SPAD camera itself can be engineered to have a low SWaP requirement as well, making the entire SPAD array flash LiDAR system suitable for smaller platforms. A lower laser power also makes the system eye-safe and operational in low-light conditions [26]. To further reduce noise in these images, the SPAD cameras reported in this thesis uses optical filters such that the SPAD sensor only detects light with a wavelength close to the laser's wavelength.

There are other single photon detection technologies such as Photo-Multiplier Tube (PMT) and Micro-Channel Plate (MCP), but they are not used because SPAD cameras are much smaller in size, making them more suitable for mounting onto a mobile platform for a distributed network surveillance application. Both PMTs and MCPs use a tube to collect photons and convert them into electrons using different processes. A PMT uses one large tube [149] while an MCP uses a plate that contains many tiny tubes [5]. In contrast, the SPAD camera reported in this thesis is a small unit containing multiple SPAD detectors, a Field-Programmable Gate Array (FPGA) board and a breakout board. The small size of the SPAD camera makes it suitable to be mounted on a mobile platform for the distributed network application discussed in this thesis introduction.

SPAD is also referred to as Geiger-Mode Avalanche Photodiode (GM-APD), which represents one of the two types of Avalanche Photodiodes (APDs), the other being Linear-Mode Avalanche Photodiode (LM-APD). APDs are semiconductor devices capable of generating an electrical current upon photon impact [39]. Comprising a semiconductor with a pn-junction functioning as a diode [39], the distinguishing factor between GM-APD and LM-APD lies in their electrical biasing. GM-APD operates at a voltage exceeding breakdown, while LM-APD operates slightly below breakdown [39]. Consequently, GM-APD is used in the research reported in this thesis because of its single photon sensitivity; where it can generate a detectable electronic current upon impact by a single photon. In contrast, LM-APD requires a minimum of 50–100 photons to produce a detectable current [39]. For a comprehensive analysis of APDs and a detailed comparison between GM-APD and LM-APD, refer to [39]. In

the context of this thesis, the single-photon sensitivity of GM-APD renders it more suitable for imaging through obscurants.

Single Photon Avalanche Diodes (SPADs) can be used as a single detector, a linear array of detectors or in a rectangular array of detectors. For the arrays, each SPAD detector corresponds to each pixel in the image. The SPAD LiDAR data in this thesis are collected with a rectangular array because their measurements can form a reconstructed image faster than the other types of detectors. Rectangular array SPADs can be used in a flash LiDAR configuration, where the scene is flood-illuminated and the array detects the first photon return for each pixel. This means all pixels are able to make a dToF measurement with one laser pulse cycle. In contrast, single and linear array SPADs perform imaging in a scanning LiDAR configuration, where the dToF measurement for a single pixel (for a single SPAD) or a line of pixels (for a linear array SPAD) is only available after multiple laser pulse cycles. The use of more than one laser pulse cycle means it takes a longer time for a scanning LiDAR system to collect dToF measurements for all pixels. In addition, the use of scanning or beam steering in LiDAR systems for single or linear array SPADs causes problems such as mechanical wear, vibration, and/or motion blur [12, 19, 38].

The SPADs that are discussed in this thesis are SPAD rectangular planar arrays fabricated using the CMOS process on silicon. This is because SPADs fabricated from silicon have been demonstrated to effectively detect photons in the visible range [39, 167]. Also, CMOS processes offer cost-effective production of SPAD planar arrays [130].

Most importantly, the CMOS fabrication process facilitates the integration of digital circuitry alongside SPAD detectors on the same planar microchip [130, 25, 34]. This integration allowed for the incorporation of digital circuitry capable of photon counting or timing directly on the chip [25]. The inclusion of timing circuitry on the chip eliminates the need for an additional timing module in SPAD array flash LiDAR systems. Typically, an electronic module is utilised to measure the time interval between laser emission and SPAD camera photon detection, as demonstrated in [106, 88]. With the addition of a timing module on the chip, the SPAD camera can directly connect to the laser, ensuring synchronisation between SPAD's detection window and laser pulse emission. This configuration is demonstrated in [165, 158]. This reduces the SPAD array flash LiDAR system to a smaller size.

For the SPAD cameras that are reported in this thesis, the photon timing circuits are integrated on the same planar chip as the SPAD detector. Section 3.4 provides details of the different SPAD array flash LiDAR systems that are reported in this thesis.

In the next section, an explanation is provided for the operation of the SPAD array flash LiDAR systems that are reported in this thesis.

3.2 SPAD Array Flash LiDAR Systems

All imaging that is reported in this thesis is conducted using a CMOS silicon 32×32 SPAD rectangular planar array within a 3D flash dToF LiDAR system operating in the visible wavelength. This thesis refers to this type of system as SPAD array flash LiDAR system. Figure 3.1 shows a schematic of the components of such a system and how it operates for imaging a target. Even though different systems are reported in this thesis, they all contain these components or some variation of this configuration.

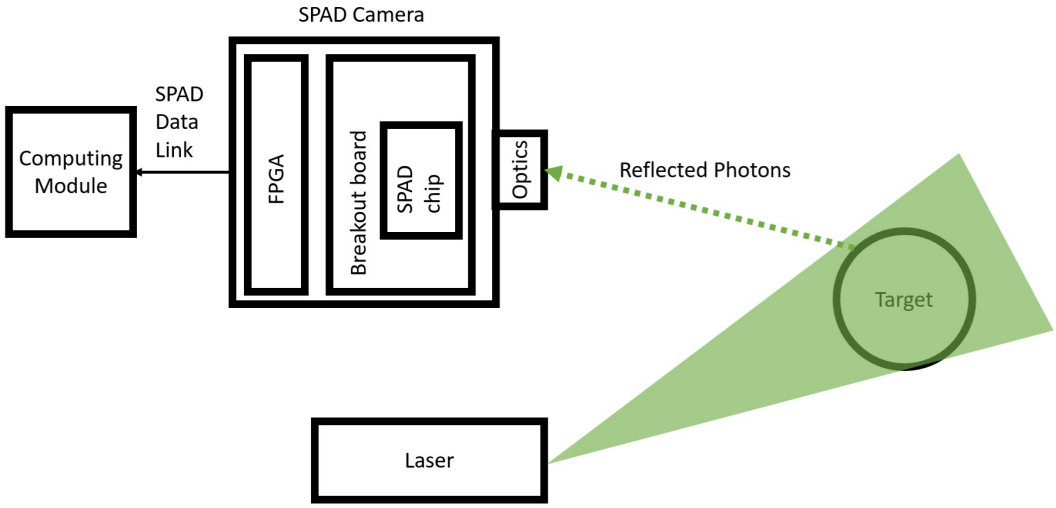


Figure 3.1: A schematic of a bistatic SPAD array flash LiDAR system that measures dToF of a target. The green triangle represents a laser pulse's flood illumination of the target. The dotted arrow represents the trajectory of any reflected photons that can reach the SPAD camera. The sizes of each component are not representative of their sizes relative to each other.

The LiDAR configuration used in this thesis is bistatic, where the SPAD camera and laser are aligned side by side and positioned at an equal distance from the target.

The SPAD camera consists of several components: the FPGA, breakout board, SPAD microchip and optics. The SPAD microchip is mounted on a breakout board, which is connected to an FPGA. The breakout board is an electronic board used to enable the FPGA to connect to the pins of the SPAD microchip. The SPAD detectors are manufactured onto the SPAD microchip. For the SPAD cameras that are reported in this thesis, the photon timing circuits are integrated on the same planar chip as the SPAD detectors.

The authors in [50] provide a flow chart on the typical electronic signal sequence for a SPAD LiDAR to measure a detected photon's dToF [50, Figure 6]. These authors in [50] also discusses the typical timing artefacts caused by the limitations from the SPAD detectors and associated electronics. In addition, these authors in [50] present a review of various electronic implementations of this sequence, which they categorise based on the extent of on-chip signal sequence implementation.

The FPGA downloads the data from the SPAD microchip and transfers it to the computing module. It also is responsible for operating the LiDAR system, where it sends a repetitive laser driving signal to enable the laser to send pulses to the target. Along with this signal, it also sends a corresponding signal to the SPAD microchip to enable on-chip dToF measurements.

The computing module in Figure 3.1 represents where the dToF measurements are processed. In this thesis, the computing module is either a PC or an embedded board called the Nvidia Jetson TX2 module.

The components of the SPAD array flash LiDAR system are fixed and do not move during imaging. Figure 3.1 shows that during imaging, the laser flood illuminates the target (i.e. flash). Then each SPAD detector of the SPAD microchip records the dToF measurement of the first photon it detects. All dToF values in this thesis are presented in terms of clock cycles.

Figure 3.2 shows a photo of one of the SPAD array flash LiDAR systems that is reported in this thesis. The photo shows the laser and SPAD camera mounted together on the black plate, where the SPAD camera is the silver container below the plate and the laser's components are all mounted above the plate. Details of the different SPAD array flash LiDAR systems that are reported in this thesis are discussed in Section 3.4.

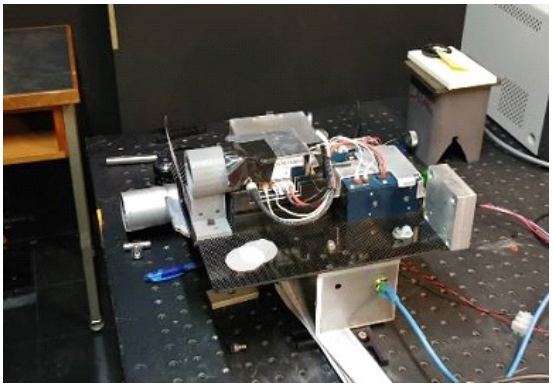


Figure 3.2: DST built SPAD array flash LiDAR system in the laboratory (System C in Table 3.1).

A common method to determine the depth of the target from collected dToF measurements is to first aggregate them into a histogram of dToF values using the TCSPC technique. Section 3.3 provides more detail of this technique and how its output is used for depth estimation.

3.3 Time-Correlated Single-Photon Counting (TCSPC)

The TCSPC technique is a common method used by the SPAD LiDAR community to aggregate dToF measurements obtained by a SPAD LiDAR system. The TCSPC technique employs statistical sampling to measure the dToF of photons based on their

arrival times relative to a synchronisation signal [38]. In the context of this thesis, the synchronisation signal corresponds to the emission time of the laser pulse. By accumulating the dToF measurements of detected photons, TCSPC organises these measurements into a histogram [38]. This histogram is referred to as the histogram of dToF measurements throughout this thesis. Initially developed for fluorescence decay analysis [21], TCSPC was first adapted for SPAD imaging as outlined in [96] and [97].

The effectiveness of the TCSPC technique relies on several assumptions. It requires low signal intensity, high repetition rate, and the rarity of multiple signal occurrences within a single period [20]. These characteristics make it particularly suitable for application in SPAD LiDAR systems employing low-powered lasers. Consequently, such SPAD LiDAR systems can generate fine depth-resolution 3D images despite their low power output [153]. Additionally, the TCSPC technique offers picosecond-level timing resolution for dToF measurements, facilitating detailed surface reconstruction even across the long ranges that SPAD LiDAR systems are capable of imaging, spanning from hundreds of metres to kilometres [153, 25]. The synergy between the high depth resolution achieved through TCSPC and the low power demands inherent in SPAD LiDAR systems, coupled with their extensive imaging range capability, positions the SPAD LiDAR system as a highly promising option for remote sensing applications.

The resulting histogram from TCSPC allows for high-resolution depth information to be extracted. Various algorithms exist for depth estimation from the aggregated histogram, with the cross-correlation approach being a commonly employed fundamental technique [95, 157].

In the SPAD LiDAR community, the set of dToF measurements aggregated by the TCSPC method can be for photons detected at the pixel level or over the entire array of a SPAD camera. When dToF measurements are examined for each pixel, their aggregation is usually related to image reconstruction. For example, TCSPC and cross-correlation can be iteratively applied on each pixel of the SPAD camera to reconstruct an image [38]. When dToF measurements over all pixels are aggregated into a histogram using TCSPC, it is usually conducted in order to use the histogram for a rough depth estimation of the target.

The next section provides the specifications of different SPAD array flash LiDAR systems used in this thesis.

3.4 SPAD Array Flash LiDAR System Specifications

The details of different SPAD array flash LiDAR systems that are reported in this thesis are provided in Table 3.1. Appendix A presents additional details about Systems B and C's laser. The timing corrections and minimum operating distances for Systems B and C are mentioned in the following subsections. The timing corrections affect the calculation of distance measurements from the dToF measurements when used to discuss algorithm performance in this thesis.

Table 3.1: Summary of systems parameters.

System Component	System Parameter	System A: Chapter 4	System B: Chapter 5 (For Imaging at the AIMS Indoor Saltwater Tank)	System C: Chapter 5 (For DSTG Water Tank Imaging and Outdoor Field Imaging) and Chapter 6
SPAD camera	Pixel Array Size	32×32 [177]	32×32	
	Fill Factor	3.14% [165]	3.142%	
	Excess Bias Voltage	4-6 V [164]	29.1 V	
	Median Dark Count Rate	100 Hz [177]	2000-6000 cps	
	Photon Detection Efficiency	N/A	12% at 532nm, V_{ex}=2.3 V, Dark Count Rate=210 cps	
	Internal timing clock frequency	160 MHz [177]	300 MHz (externally driven)	
	Microlens	N/A	100 µm diameter	
	Lens used with SPAD camera	Navitar NMW-12WA [7] lens	Pentax TV Zoom lens 12.5 to 75 mm with F number of 1 to 1.8	
	Filter used with SPAD camera	Thorlabs 660 nm filter [4]	2nm narrow bandwidth filter at 532nm (Edmund Optics part no. 68-970)	
	Bin width	390ps [177]	3.33 ns (equivalent to 0.5 m in air)	
Laser	Model	Coherent CUBE diode laser (name: CUBE 660-100C) [2]	Bright Solutions	Aréte
	Illumination Wavelength	660 nm	532 nm	532 nm & 1064 nm (only 532 nm is used in this work)
	Pulse Energy	0.67 mJ [2]	10 µJ	40 mJ (48% at 532 nm)

	Pulse Width	17 ns	1.3 ns	5 ns
	Jitter	N/A	Order of microseconds	1 ns
	Laser Repetition Rate	150 Hz [2]	25-100 Hz	25 Hz
	Diffuser used with laser	N/A	N/A	drafting paper
	Energy Level with Diffuser	N/A	N/A	10 mJ (measured over a 70 degree cone)
	Telescope expander	Navitar 1-50486 12x zoom lens [1]	145 mm in diameter with a focal length of 300 mm so the F number is $\frac{F}{2}$	N/A
LiDAR System	Timing delay	N/A	40 ns	13.3 ns
	Minimum operating distance (in air)	N/A	6 m	2 m
	External signal to start timing on SPAD camera	No	Yes	No

3.4.1 Timing Corrections and Minimum Operating Distance for System B

System B has a timing delay of 40 ns (as shown in Table 3.1) between the high-speed detector detecting a laser pulse and triggering the SPAD camera. This is because it uses a high-speed detector (Newport 818-BB-21) to trigger the SPAD camera to start timing. The high-speed detector is used to overcome the large jitter of the Bright Solutions laser.

This timing delay of 40 ns means System B's dToF measurements are 12 clock cycles less than the theoretical dToF value. In this thesis, the reported dToF measurements, expected dToF values and range gate minimum and maximum dToF values related to System B do not compensate for this timing delay. However, the equivalent distance values discussed in this thesis correct for this timing delay.

In addition, this timing delay means System B has a minimum operating distance of 6 m in air.

3.4.2 Timing Corrections and Minimum Operating Distance for System C

The FPGA controller for System C delays the SPAD window opening by 13.3 ns (as shown in Table 3.1). This timing delay means System C's dToF measurements are 4 clock cycles less than the theoretical dToF value. In this thesis, the reported dToF measurements, expected dToF values and range gate minimum and maximum dToF values related to System C discussed in this thesis do not compensate for this timing delay. However, the equivalent distance values discussed in this thesis correct for this timing delay.

In addition, this timing delay means System C has a minimum operating distance of 2 m in air.

3.5 Beer–Lambert Law

The Beer–Lambert Law is utilised in this thesis for calculating the beam attenuation coefficients to measure the level of obscurants in an environment using the number of ALs (to be presented in Section 3.6.1). For the experiments reported in this thesis, the attenuation of light is assumed to be uniform in all fog and water conditions. Therefore, the Beer–Lambert Law can be used to calculate the beam attenuation coefficient.

The beam attenuation coefficient is defined as the limit of the spectral attenuation to the distance of photon travel as that distance becomes vanishingly small [108]. This coefficient is used to understand the amount of laser power at different transmitted distances [45]. The literature often uses the terms beam attenuation coefficient and attenuation coefficient interchangeably.

There are currently different versions of the Beer–Lambert Law used in the literature. The version used in this thesis is commonly used in electro-optics. A different version is used in spectroscopy and it is not formulated in terms of the beam attenuation coefficient [150, 159].

The Beer–Lambert Law used in the electro-optics field is given in [45], and it is represented in this thesis as follows:

$$\mu = -\frac{1}{x} log(\frac{I}{I_0}), \tag{3.1}$$

where μ (given in m^{-1}) is the beam attenuation coefficient, x is the path length (in m), I is the power intensity of the transmitted laser beam (in W) and I_0 is the power intensity of the incident laser beam (in W) [131, 71, 47].

In the field of spectroscopy, the Beer–Lambert Law is usually presented in the form given in [150, 159] and is presented in this thesis as follows:

$$A = -log\frac{I}{I_0} = \sigma x C, \tag{3.2}$$

where A is absorbance, σ is the extinction coefficient, C is the substance's concentration in the sample and the remaining variables have the same meaning in Eq. (3.1). Eq. (3.1) will be the version that is used in this thesis.

From Eq. (3.1) and Eq. (3.2), we deduct that there is a linear relationship between the beam attenuation coefficient and the concentration of a substance [101]:

$$\mu = \sigma C \tag{3.3}$$

3.5.1 Path Length Value for Imaging Set-ups

When applying this law to calculate the beam attenuation coefficient for an obscured environment used in the imaging set-ups reported in this thesis, the path length x is altered to be only the distance travelled by the laser through the obscurant. This excludes any path length where there is no obscurant in that path's area. This alteration is applied in this thesis because the attenuation of the laser through air is assumed to be negligible compared to attenuation by obscurants.

When the obscurant is fog, the fog's average span is used as the path length x. The average span is calculated as the average of the maximum and minimum span of the fog observed by human eye. The fog's average span is used instead of its maximum span for the path length x because fog varies in distribution and the Beer–Lambert law assumes any obscurants are uniform in distribution in order to calculate the beam attenuation coefficient. In this thesis, I assume using the average span approximates the fog to be uniform over this distance.

3.6 Measuring the Level of Obscurant

The different metrics used in this thesis for quantifying the level of fog and water turbidity are discussed in this section. They are used in this thesis to quantitatively

describe the level of fog and water turbidity in the imaging set-ups reported in this thesis. The number of ALs is most commonly used metric for quantifying both fog and water turbidity levels. The SD depth is an alternative metric used for quantifying the turbidity level of water while visibility is an alternative metric for quantifying the fog level.

3.6.1 Number of Attenuation Lengths (ALs)

Number of attenuation lengths (ALs) is a common measure of the level of water turbidity and fog when discussing SPAD LiDAR imaging through fog [158, 157, 156, 60] and water [89]. The definition of one attenuation length is the distance at which the transmitted light power is reduced by a factor of $1/e$ of its initial value [60]. The attenuation length is thus calculated as:

$$N_{AL} = \mu d = -\frac{d}{x} \ln\left(\frac{P}{P_0}\right), \tag{3.4}$$

where μ is the beam attenuation coefficient and d is the distance between the SPAD detector and the target. x is the path length travelled by the laser through the obscurant (here is either fog or water) to reach the power meter. μ is calculated using the Beer–Lambert Law as discussed in Section 3.5 and its equation expansion follows Eq. (3.1) in Section 3.5. Following the discussion in Section 3.5.1, the average span of fog is used as the path length x in Eq. 3.4.

In the literature, various laser wavelengths are employed to measure the number of ALs. The choice of wavelength significantly influences the measured number of ALs due to different transmission characteristics through the obscurant. Therefore, care must be taken when comparing results for different SPAD LiDAR systems operating at different wavelengths. The value of the number of ALs cannot be calculated for an alternative wavelength based on the number of ALs for a given wavelength, as the calculation requires either measuring the input and output powers at the beginning and end of the obscurant, or knowing the beam attenuation coefficient of that wavelength in the specific imaging environment. In this thesis, all measurements related to the number of ALs are for a wavelength of 532 nm.

3.6.2 Secchi Disc (SD) Depth

The SD depth measurement (measured in m) is defined as the maximum depth at which the SD can be seen by the naked eye. A SD is a 30 cm-diameter circular disc with black and white quadrants. The function of an SD is to provide a simple visual index for visibility [125].

Even though the number of ALs is more commonly used in the SPAD LiDAR community for quantifying the water turbidity levels tested, it is difficult to measure the amount of transmitted power through natural waters when it comes to conducting field trials. The laser power at the target's location cannot be measured without the use of specialised equipment. Therefore, SD depth is a more pragmatic measure-

ment in this scenario. In Chapter 5, SD depth is used to describe the water turbidity level of outdoor natural water locations.

3.6.3 Visibility in Metres

Even though the number of ALs is a common measure for quantifying fog levels in the SPAD LiDAR community, it is not the standardised way of measuring fog levels. An additional measure called visibility is used in this thesis to attempt to provide a more comprehensive view of the level of fog used in the data collection.

Visibility describes how far the naked eye can see from the start of the fog. Visibility is calculated by measuring the transmittance of a continuous 532 nm laser beam. This is possible by using the formula for visibility:

$$V = \frac{1}{\mu} \ln \left(\frac{C_0}{C_{th}} \right), \tag{3.5}$$

where μ is the attenuation coefficient, C_0 is the target contrast ($C_0 = 1$ in this thesis), C_{th} is the threshold contrast and ln denotes the natural logarithm. In this thesis, C_{th} is set to 0.05 according to the CIE's (International Commission on Illumination) definition of the human eye's minimal perceptible contrast [157, 145]. The attenuation coefficient, μ, is calculated using the Beer–Lambert Law (Eq, 3.1) as discussed in Section 3.5.

Similar to the number of ALs, the visibility measure is also dependent on the laser's wavelength used to measure fog level. In this thesis, all measurements related to the visibility are for a 532 nm wavelength.

3.7 Target Classification Metrics

For evaluating the target classification algorithms in this thesis, the common performance metrics from multiclass classification algorithms are used. Multiclass classification is where each target can only be classified as one unique object class label and there are more than two available object class labels for classifying these targets [49]. More details of these metrics can be found in the reference associated with their description. There are many versions of macro F1 measures [119]; the one listed below is the one used in this thesis.

The definitions of the classification metrics are based on the following measures for multi-class classification [148]:

True positive (TP) : both classification and ground truth class equals the class of interest

False positive (FP) : classification class equals your class of interest, but ground truth is a different class

True negative (TN) : classification and ground truth classes do not equal the class of interest

False negative (FN) : classification class equals a different class, but ground truth is the class of interest

The definitions of the classification metrics are given below [148].

Overall accuracy This is the average accuracy percentage over all classes. The formula is:

$$\frac{\sum_{i=1}^{l} \frac{TP_i+TN_i}{TP_i+FN_i+FP_i+TN_i}}{l} \times 100 \tag{3.6}$$

where l is the total number of classes [148].

Precision This is calculated as:

$$\frac{TP}{TP+FP} \tag{3.7}$$

for each class. [148]

Macro precision Average precision over all the classes. [148]

Recall This is calculated as:

$$\frac{TP}{TP+FN} \tag{3.8}$$

for each class. [148]

Macro recall Average recall over all the classes. [148]

Error rate This is the average error rate over all classes. It is calculated by:

$$\frac{\sum_{i=1}^{l} \frac{FP_i+FN_i}{TP_i+FN_i+FP_i+TN_i}}{l} \times 100 \tag{3.9}$$

where l is the total number of classes [148]. 100 is used to make the value a percentage.

Macro F1 measure

$$\frac{2 \times \frac{\text{macro precision} \times \text{macro recall}}{\text{macro precision} + \text{macro recall}}}{100} \tag{3.10}$$

This value ranges between 0 to 1 [119]. The formula has a division by 100 because the macro recall and macro precision values are presented in terms of percentages in this thesis.

In Chapter 4, Section 4.6.3 employs all of these metrics to evaluate target classification performance. However, in Section 5.4.3.4 of Chapter 5 and Section 6.5.1.2 and 6.5.2.3 of Chapter 6, only overall accuracy is utilised. This discrepancy arises due to the unique nature of the target classification algorithms proposed in these latter chapters, which incorporate an additional classification class labelled "Undecided". This class serves to account for instances where targets cannot be classified using the rules in the classification algorithm. The algorithm in Section 4.6.3 of Chapter 4

does not introduce this additional class as it is constructed in such a way that it will always categorise the target into one of the existing classes. Therefore, an Undecided class is not necessary.

No instances of this Undecided class is used for testing the performance of these target classification algorithms. This means there are no TPs or FNs for this Undecided class. TPs and FNs being zero makes the Recall metric undefined for the 'Undecided' class, making the macro recall over all the classes not an appropriate metric. Because of this, it is decided that none of the additional metrics to overall accuracy should be used. This is because they should all be presented together to provide a cohesive understanding of the classification performance. Instead, the amount of FP classifications in this additional class is presented to provide an idea of how well the classification algorithm performs. In the thesis, these FP classifications for this Undecided class is simply referred to as Undecided classifications or by the phrase "classifying a target as the Undecided class".

3.8 Chapter Summary

Background explanations of various technical aspects related to this thesis are presented in this chapter. The next chapter will present a real-time target classification algorithm for live data collected by a SPAD array flash LiDAR system. The algorithm is also implemented on an embedded board.

Chapter 4

Real-Time Embedded Target Classification of Live Data

Real-time autonomous target classification becomes integral when applying the edge computing paradigm to a distributed network of mobile platforms that conducts surveillance. In contrast to transmitting a continuous stream of data to the central node for reconnaissance, mobile platforms can transmit only the information related to the target label. This reduction in transmitted data not only conserves transmission bandwidth but also minimises power consumption, enhancing reliability and security. However, the on-board portable battery limitations pose a significant challenge as they limit the operation time for mobile platforms, in particular Uncrewed Autonomous Vehicles (UAVs). While a larger battery provides more power, it also increases the mobile platform's weight, which reduces operation duration as well. Therefore, there is a need for onboard sensors and processors to have a minimal Size, Weight, and Power (SWaP) footprint in order to prolong operation time. To meet these hardware requirements, a viable solution is to perform real-time classification on an embedded processing board characterised by low power consumption and compact size.

To address these hardware constraints, this chapter presents a real-time target classification algorithm implemented on an embedded board to classify data of different single targets collected by a 32×32 Single Photon Avalanche Diode (SPAD) array flash Light Detection and Ranging (LiDAR) system. The SPAD array flash LiDAR system's low SWaP requirements means its implementation on the embedded board can be mounted onto a mobile platform. Moreover, the low data latency inherent in SPAD cameras enables rapid processing on an embedded board. The target classification algorithm is implemented with a live data acquisition module into an embedded software program to execute in real-time on a Nvidia Jetson TX2 module [8]. The Jetson TX2 is an embedded System-on-Module (SoM) board with low SWaP requirements, which means it can be mounted onto a mobile platform with the SPAD array flash LiDAR system for edge computing application on a distributed network.

This work marks a significant advancement as the first implementation of a real-time target classification algorithm for SPAD LiDAR data on an embedded board, as

confirmed by the discussions in Section 2.5 of Chapter 2. Leveraging the graphical processing unit (GPU) on-board the Jetson TX2 for parallel processing enhances the computational efficiency of the implementation, enabling real-time execution. The use of Nvidia's CUDA programming language further optimises the functions for execution on the Jetson TX2's GPU.

The target classification algorithm is called the random feature-detecting network. It operates on raw images in batches. In this chapter, a raw image is defined as an image formed with each pixel's direct Time-of-Flight (dToF) measurement over one frame. The network initially conducts image reconstruction by creating an average image from a batch of raw images and then conducts target detection as part of its tracking module. If a target is detected, the image then undergoes feature extraction and its features are stored in a vector. Linear classification determines the image target's class by multiplying the vector with a classification matrix, with the index of the highest value determining the target's class. The random feature-detecting network resembles a single-layer Convolutional Neural Network (CNN) because it uses convolutional filters to extract some of the image's features, which together act like a convolutional layer. In this chapter, the random feature-detecting network's performance is evaluated when it is implemented with different numbers of convolutional filters (4, 8, 16, 32, 64).

Two different implementations of the random feature-detecting network are presented in this chapter. One is an embedded software implementation which is used to execute the random feature-detecting network on the Jetson TX2. This implementation includes implementing the live data acquisition module with the algorithm, enabling real-time target classification of live SPAD LiDAR data. The second implementation is an offline software implemented which is used to execute training and preliminary testing of the random feature-detecting network. Offline training and preliminary testing are crucial for determining the algorithm's convolution filter values empirically.

To demonstrate the real-time execution and classification performances of the embedded software implementation of the random feature-detecting network, four different model airplanes serve as classification targets. While model airplanes are used for testing here, the classifier can be re-trained and tested for diverse targets. The choice of model airplanes stems from their common complex features, which provides a sufficiently difficult test for the random feature-detecting network to distinguish between classes. Offline training and preliminary testing as well as real-time testing use SPAD LiDAR data of airplanes at different orientations and positions. Real-time testing has the SPAD array flash LiDAR system connect to the Jetson TX2 and conduct live imaging, where its data is classified by the embedded software implementation of the random feature-detecting network executing on the Jetson TX2.

The chapter begins with a description of the random feature-detecting network in Section 4.1. Then the offline training and preliminary testing of the random feature-detecting network are explained in Section 4.2. Section 4.2.5 provides the methodology at which the convolutional filters are selected for the network. A more software-focussed discussion of the embedded software implementation on the Jetson TX2 is

provided in Section 4.3. This includes a background discussion of the usage of CUDA software Toolkit for parallel software implementation in Section 4.3.2. Section 4.4 details the differences between the offline and embedded software implementations of the random feature-detecting network, highlighting their negligible impact on target classification performance. The SPAD array flash LiDAR imaging method for collecting SPAD LiDAR data of different airplanes for offline training and preliminary testing is discussed in Section 4.5. The chapter concludes with Section 4.6 presenting the obtained results. This is followed by a comprehensive discussion of the classification and real-time execution performance of the embedded software implementation of the random feature-detecting network in Section 4.7.

The research presented in this chapter is based on collaborative work documented in the publications [102] and [178].

4.1 Random Feature-Detecting Network

The primary objective of the random feature-detecting network is to perform target classification. It begins by processing a batch of raw images into an average image, then extracts various features from the average image and store the features' information into one vector. Let X represent the singular vector containing all the features' information. Details of the feature information stored in vector X are presented in Section 4.1.2.1. One X is created, then target classification is performed through the application of linear classification. The equation for linear classification is expressed as follows:

$$XZ = Y \tag{4.1}$$

where Z denotes the classification matrix[1]. The classification matrix Z is trained from a collected dataset of images with known target labels. The values in Z are considered weights for the different feature vectors stored in vector X. Additional information regarding the computational steps involved in training can be found in Section 4.2. Each class label is associated with an index in the resulting vector Y. Target classification is performed by determining the class label that corresponds to the index with the highest value in vector Y.

The structure of the random feature-detecting network is visually presented in Figure 4.1. Within this structure, the linear classification module executes the operation in Eq. 4.1. Preceding the linear classification, the modules engage in preprocessing tasks, generating the vector X from a batch of 16 raw images. The denoising module performs image reconstruction by calculating an average image from the batch of raw images. This effectively reduces noise compared to an individual raw image. The tracking module performs target detection, and upon detection, the network advances to the data manipulation module. In the data manipulation module, feature vectors are created from the average image, and these vectors are concate-

[1]The classification matrix is written as W in [102]; here it is defined as Z instead.

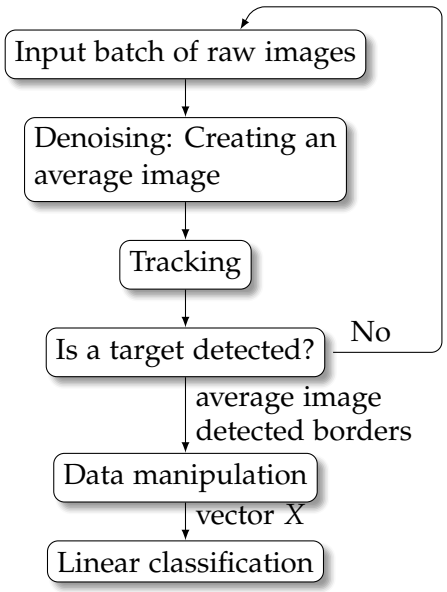


Figure 4.1: Function flow of the random feature-detecting network.

nated to form the vector X (from Eq. 4.1). Further details about the tracking and data manipulation modules can be found in Sections 4.1.1 and 4.1.2 respectively.

4.1.1 Tracking

The tracking module is responsible for target detection. It identifies the top, bottom, left and right borders that encapsulate the target within each average image. The random feature-detecting network assumes there is only one object present in each image, and that the object is the target. Therefore, the primary goal is to exclude the majority of the background for subsequent computation steps in the random feature-detecting network.

It is important to note that the removal of pixels outside of these borders does not occur until the cropping step of the data manipulation module, as described in Section 4.1.2. The role of the tracking module is solely to identify the values corresponding to the borders without performing any elimination of pixels beyond these boundaries.

Provisional border values are derived from the dToF values in the average image. These border values are computed separately for the top and bottom pairs, as well as the left and right pairs. For the top and bottom pair, pixel values in each row are summed into a single value, resulting in a vector of row sum values. For computational simplicity, these sum values are renormalised. Renormalisation involves subtracting each value of original row sum vector from the maximum value of the row sum vector. After renormalisation, a smoothing operation is applied to the renormalised row sums to mitigate the impact of sporadic noise pixels on border location. Once smoothing is completed, each value of the row sum vector is assessed if it is above a threshold value. Top and bottom border values are identified as indices

one index before and after of the indices of the values above the threshold. Similar calculations are applied to obtain left and right border values, with each column being summed into a value to create a column sum instead of a row sum.

A threshold is used to determine the border because rows containing the target exhibit a shorter dToF value sum compared with the background. This is because the target is assumed to be closer to the sensor than the background. This means that if an element in the row sum vector is above a threshold, the current and following rows are considered to contain the target. This same reasoning applies to the column sum vector. The threshold is determined empirically during algorithm development.

The subsequent computation of border values depends on the existence of a valid set of border values from the previous average image. In cases where no previous border values exist, the provisional border values are directly employed as the average image's border values.

When there is a set of border values from the previous average image, each border value of the current average image is determined as a weighted average of its provisional border value and the corresponding value from the previous average image. This calculation assumes that the border of the subsequent average image is likely to be in a similar position, considering the sequential order of images corresponds to the order of the raw images, where each raw image corresponds to one LiDAR cycle.

Following the computation of border values, the size of the block of pixels enclosed by these borders is scrutinised to ensure it falls within the specified size range. In the context of classifying airplanes, the block's side lengths are required to be within 10–30 pixels for successful detection. Failure to meet this criterion results in the tracking module deeming the target detection to be unsuccessful for the current average image.

4.1.2 Data Manipulation

After detecting a target in the average image, the data manipulation module is utilised to generate different feature vectors from this image, which are then concatenated to form vector X for linear classification. The function flow structure of this module is shown in Figure 4.2. As seen in the figure, the module can be divided into two distinct branches. Each branch starts with a different processing method on the image. After applying their various techniques, the images become feature images. Then both branches crop the feature images, compress them into a feature vector and then resample the feature vectors.

The first branch on the left begins with renormalising all pixel values of the average image. Renormalisation is computed by determining the maximum pixel value, and each renormalised pixel value is obtained by subtracting the original pixel value from the maximum pixel value.

The second branch on the right applies convolution, using 4×4 filters. The values for these filters determined empirically using a method detailed in Section 4.2.5. During performance testing, different numbers of convolutional filters (4, 8, 16, 32, and 64) are implemented for this convolutional step. This is to assess their impact on

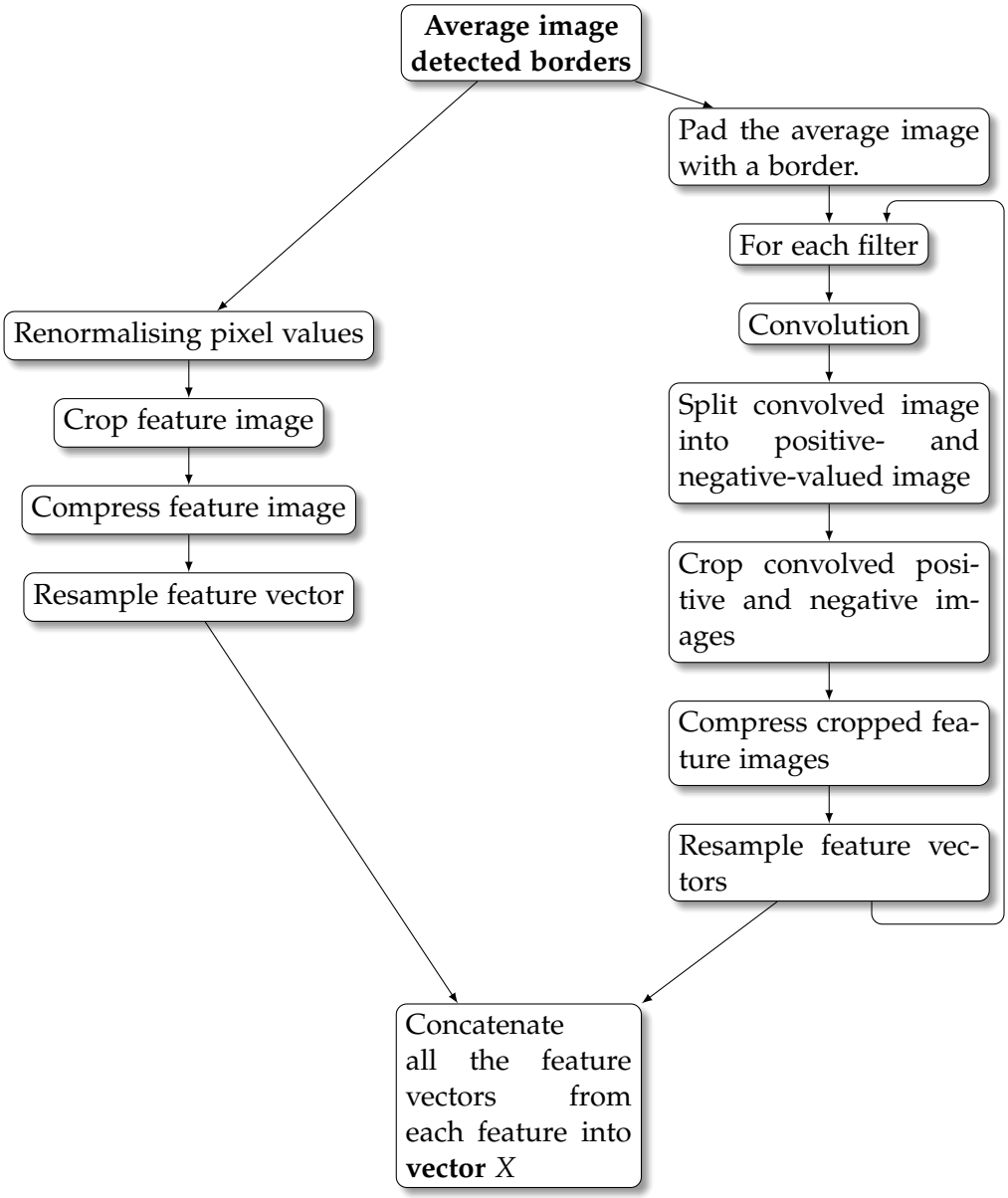


Figure 4.2: Function flow of the data manipulation module.

classification accuracy.

To preserve information on the edges during convolution, the average image is padded with a 3-pixel wide border before applying the filters. This border, one pixel less than the filter dimensions, ensures that edge pixels of the convolved image contain information from the average image's edge.

Subsequently, the convolved images are divided into positive- and negative-valued images. This process is found empirically to enhance classification accuracy compared with retaining positive and negative values together.

All feature images (the renormalised and convolved images) are then cropped.

For the feature images on the first branch, the indices are cropped to the border indices identified in the tracking module (Section 4.1.1). For the feature images on the second branch, the tracking module's border indices are offset by 3 pixels to account for the 3-pixel padding border introduced in the padding step.

Feature vectors are calculated as the sums of each row and column in these cropped feature images, and then concatenated to form vector X for linear classification (Eq. 4.1). The structure of vector X is shown in detail in Section 4.1.2.1.

Maintaining a consistent length for vector X is crucial for linear classification, as the classification matrix is established during training (described in Section 4.2) with unalterable dimensions. Given that resulting feature vectors possess different lengths due to cropping based on the tracking module's border's size (detailed in Section 4.1.1), they are resampled to a fixed length before concatenation. An efficient resampling algorithm is detailed in Section 4.1.2.2.

4.1.2.1 Structure of Vector X

After each of the feature vectors is resampled, they are concatenated together to create vector X, which is then used for linear classification (as shown in Eq. 4.1).

- Sum of each **row** of the cropped and **renormalised** image
- Sum of each **column** of the cropped and **renormalised** image
- Sum of each **column** of the cropped and **positive** values of the **convolved** images - there are a number of this type of vector and the total number depends on the number of filters used
- Sum of each **row** of the cropped and **positive** values of the **convolved** images - there are a number of this type of vector and the total number depends on the number of filters used
- Sum of each **column** of the cropped and **negative** values of the **convolved** images - there are a number of this type of vector and the total number depends on the number of filters used
- Sum of each **row** of the cropped and **negative** values of the **convolved** images - there are a number of this type of vector and the total number depends on the number of filters used

4.1.2.2 Resampling

When a vector requires resampling and its length is coprime with the new length, the upsample factor is determined as the product of the two lengths, while the downsample factor is set to the original vector length. However, managing these upsample and downsample factor values can introduce computational complexity.

The random feature-detecting network uses a resampling algorithm that artificially extends or shortens the original vector before applying the necessary upsampling and downsampling operations. Specifically, the original vector's length is modified, allowing for significantly lower upsample and downsample factors, thereby reducing the computational complexity of the resampling operation. For an increase in length, zeros are appended to both ends of the original vector. Conversely, for a reduction in length, elements are removed from both ends of the original vector.

The determination of the new length x for the original vector involves finding the minimum of the following function:

$$f(x) = \text{Penalty function}(x) * \text{LCM}(x, \text{output vector length}), \tag{4.2}$$

where

$$\begin{aligned}\text{Penalty function}(x) =& 1.7^{-\frac{\text{original vector length}^{0.63}}{10}} \times \text{remapping resistance factor} \\ & \times (x - \text{original vector length})^2 + 1\end{aligned} \tag{4.3}$$

The least common multiple (LCM) in Eq. 4.2 serves as an indicator of the relative computational cost associated with resampling. Meanwhile, the penalty function articulated in Eq. 4.3[2] quantifies the information loss resulting from the removal of elements or the introduction of additional zeros to the original vector before resampling. The remapping resistance factor plays a crucial role in determining the acceptable difference in length. A higher value implies lower tolerance. In the specific application of classifying airplanes using this network, the remapping resistance factor is configured at 1.25.

The original vectors consist of either the row or column sums of cropped feature images, as detailed in Section 4.1.2. Consequently, the act of adding or removing elements from either end corresponds to a modification of the edge information within the images. Given the inherently noisy nature of the raw images of SPAD LiDAR data, this algorithm operates under the assumption that modifying the values at the image edges has a negligible impact on the information contained within the image, and consequently, it has a negligible effect on the classification performance.

The number of elements to remove or add on the left and right is determined by the following formulas:

$$\begin{aligned}&\text{Number of elements to add or remove to the left} \\ &= \begin{cases} \frac{\text{new length} - \text{original length}}{2} & \text{if length difference is even} \\ \frac{\text{new length} - \text{original length} - 1}{2} & \text{if length difference is odd} \end{cases}\end{aligned} \tag{4.4}$$

[2]The penalty function provided in [102] is incorrect and does not contain the first factor reported in Eq. 4.3, which is the correct equation.

$$\text{Number of elements to add or remove to the right} = \begin{cases} \frac{\text{new length} - \text{original length}}{2} & \text{if length difference is even} \\ \frac{\text{new length} - \text{original length} + 1}{2} & \text{if length difference is odd} \end{cases} \tag{4.5}$$

To summarise the order of steps for this resampling algorithm, when the cropped feature vectors are created in the data manipulation module, the length of each of these vectors is scrutinised by the resampling algorithm. For each of these vectors, their lengths are considered as the original vector length by the resampling algorithm. The resampling algorithm determines a suitable new length for each cropped feature vector by determining what value x minimises Eq. 4.2. Subsequently, the resampling algorithm alters the length of each cropped feature vector using Eq. 4.4 and 4.5 to determine the number of zeros to append or the number of elements to remove. Finally, each of these altered vectors are resampled to the same output vector length.

The output vector length the same length as the number of rows in the classification matrix Z. This is to facilitate the computation of Eq. 4.1 for classification. In the specific context of classifying airplanes, the output vector length is consistently set at 36[3], for both feature vectors related to row sums and column sums.

For this resampling algorithm, the upsampling operation is executed using zero-order hold. For an upsampling factor of n, each element in the vector is replicated n times. Downsampling, on the other hand, involves keeping every $i + n$-th element for a downsampling factor of n, where $i = 1, 2, 3, 4, \ldots$.

To improve efficiency on an embedded board, the resampling algorithm utilises a precomputed lookup table. Given that the SPAD array size is consistently 32×32, the raw image resolution is always going to be 32×32 and the convolved image will always be 35×35. Therefore, the lengths of cropped feature vectors range from 1–35[4]. Therefore, the lookup table can pre-calculate the number of elements needed for removal or addition for all potential original vector lengths. This facilitates rapid computation for resampling.

4.2 Offline Training and Preliminary Testing

Offline training and preliminary testing procedures are executed for the random feature-detecting network to determine the values of the target classification matrix Z and assess the classification accuracy of the network using the calculated matrix Z. Both offline training and preliminary testing involve the use of collected data from the SPAD array flash LiDAR system, which is of the same targets that will be classified by the real-time embedded software implementation of the network.

Training and preliminary testing require a SPAD LiDAR dataset of objects from the different objects classes that the target can belong to. Each batch of raw images in this dataset is labelled with a one-hot counter corresponding to its class. The method

[3]Note: The output vector length is erroneously reported as 32 in [102] and [178].

[4]The reported vector length range in [102] and [178] is incorrect. The correct range of original vector lengths is 1–35, independent of cropping.

for allocating raw images to either training or testing depends on the number of raw images collected for each object class. In each training and testing iteration, the datasets for training and testing consist of different raw images to ensure the robustness of the classification.

As training involves the pre-processing modules of the random feature-detecting network, different convolution filters used in the data manipulation module also influence the outcome of the training, i.e., the value of the classification matrix Z. The effect of using different number of convolutional filters (which results in different numbers of feature vectors) on the classification accuracy is examined in the development of the random feature-detecting network. In particular, the number of filters that are of interest are 4, 8, 16, 32 and 64. For each number of filters, distinct sets of filter values are employed to train and test the random feature-detecting network, aiming to identify filter values that yield optimal classification performance. Further explanation of filter selection is provided in Section 4.2.5.

Both training and testing process the collected data into the vector X, which is the output of the pre-processing modules in the random feature-detecting network, as described in Section 4.1. These modules are before the linear classification module. Given that training and testing share the same pre-processing modules, they are implemented together in the offline software implementation to improve computational efficiency.

The order of execution of the offline software implementation starts with the pre-processing modules, then the data is organised into training and testing datasets. This is followed by training to determine the classification matrix Z. Then the implementation ends with testing, where the classification accuracy is determined for the random feature-detecting network with the classification matrix X calculated from training. The subsections below provide details of each of these steps.

4.2.1 Pre-processing Modules

In the initial step, each batch of raw images from the available dataset are processed by the pre-processing modules (those that are before the linear classification module, as detailed in Figure 4.1 in Section 4.1) to generate a collection of vector Xs. For each vector X, there is an associated ground truth one-hot vector that represents the object class label. The one-hot vector assigns a value of 1 at the index corresponding to the object class label.

4.2.2 Allocating Data into Training and Testing Datasets

The pairs of vector X and their corresponding ground truth class label vector are allocated to either the training or testing dataset. The distribution of these pairs is randomised during each iteration of training and testing. This randomisation is essential to ensure the robustness of the random feature-detecting network. Section 4.5.1 provides an explanation of how the airplane data set is divided for training and preliminary testing of the random feature-detecting network.

4.2.3 Training

Once the datasets are organised, training is initiated, during which the classification matrix Z is calculated using the training dataset. All the vector Xs from the training dataset are gathered into a matrix X_{train}. Subsequently, the classification matrix Z is calculated using Eq. 4.1 through the following formula:

$$Z = (X_{train}^T * Y_{\text{ground truth for training}})^T (X_{train}^T * X_{train} + b)^{-1}, \tag{4.6}$$

where b is the regularisation factor added to the diagonal of the matrix product. Equation 4.6 is derived by using the pseudoinverse of X_{train} and applying the Ridge regression (also called Tikhonov regularisation) [37]. When calculating the classification matrix for the four different airplanes that are imaged (as described in Section 4.5), the regularisation factor is set to 100. This value was found through trial and error to provide the best classification result.

$Y_{\text{ground truth for training}}$ is a matrix that consists of one-hot counter target labels. Each vector in $Y_{\text{ground truth for training}}$ corresponds to the target label for the corresponding vector X in X_{train}. These vectors in $Y_{\text{ground truth for training}}$ are placed in the same order as their corresponding vector Xs are placed in X_{train}.

4.2.4 Preliminary Testing

Preliminary testing is executed using the calculated classification matrix from training, and also the testing dataset. The classification output of each vector X is compared with its corresponding ground truth label vector. The classification accuracy is computed as the ratio of matching classification outputs to the total number of batches tested. To facilitate a fast comparison between the output vector Y from Eq. 4.1 and the one-hot counter label, the vector Y is converted into a binary vector. The entry with the maximum value in Y becomes 1, while the rest become 0. If all values in Y are 0, the vector becomes all 0. If all the values in Y are the same, then all of its entries become 1. The classification accuracy is used to select the appropriate convolutional filters, which is discussed in the next subsection (Section 4.2.5).

4.2.5 Convolutional Filter Selection

To demonstrate the real-time classification capability of the random feature-detecting network, four different types of airplanes are utilised as classification targets.

Various numbers of convolutional filters are employed in the network to assess the classification performance of the network for airplanes as targets. Specifically, 4, 8, 16, 32, and 64 filters are tested. To determine the best set of filter values for each number of filters, 10 sets of filter values are generated randomly offline for each number of filters. The set of filter values that yields the highest classification accuracy is selected for use in the real-time implementation of the network.

To determine the classification accuracy using each set of generated filter values, the random feature-detecting network undergoes offline training and preliminary

testing with each set of filter values. Raw images of airplanes are used to train and test the network. Groups of raw images are collected in recordings, where each recording only contains raw images of one type of airplane. Section 4.5 details the method for collecting raw images of different airplanes. The collected raw images are split into training and testing datasets based on their associated recording. Section 4.5.1 provides more detail on how the collected raw images are split for training and testing.

At each instance of preliminary testing, the raw images from 4 randomly selected airplane recordings are used. Training and testing are repeated 20 times to ensure that each set of filter values is tested on raw images of each class of airplane. This is because each recording only contains raw images of one type of airplane and it cannot be guaranteed that the 4 randomly selected airplane recordings correspond to the 4 different possible classes of airplanes in the classifier.

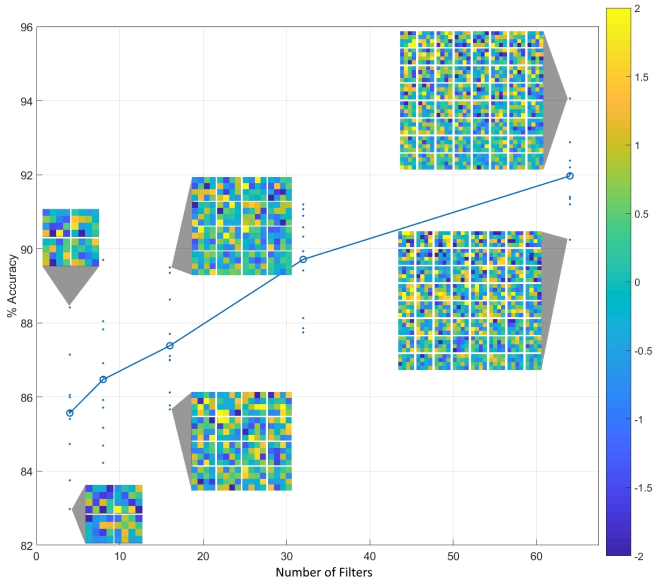


Figure 4.3: Mean classification accuracies of different number of filters and filter values tested over 20 different sets of data [102]

The classification accuracies for the generated sets of random filters are shown in Figure 4.3. The values used for some sets of filters are shown in this figure via a colour map, where there is a colour bar on the right. For example, for four sets of filters, there are four 4×4 colour maps shown. Each '+' data point in the figure represents the mean accuracy of one set of random filters after conducting training and testing over 20 random airplane data splits. Therefore, there are 10 '+' data points for each number of filters. Each 'o' data point represents the mean accuracy of all the sets of filters for that number of filters. For each number of filters, the set of filter values with the highest mean accuracy is selected for the real-time implementation.

4.3 Embedded Software Implementation

The random feature-detecting network is implemented onto the Jetson TX2 along with the live data acquisition module, where it obtains SPAD LiDAR data from the SPAD camera and converts the data into raw images. The random feature-detecting network uses the raw images from the live data acquisition module for target classification. Both the random feature-detecting network and live data acquisition module are executed on the Jetson TX2 to avoid any additional data transfer time that can be introduced by having the live data acquisition module run on a different computing board to the random feature-detecting network. This ensures the SPAD LiDAR data can be processed and classified in real-time.

The implementation of the random feature-detecting network follows the same structure detailed in Section 4.1.

In this section, the software design choices for the embedded software implementation of the random feature-detecting network are discussed in Section 4.3.1 and this is followed in Section 4.3.2 by an explanation of CUDA, which is a software programming model used by the embedded software implementation to optimise some of the functions in the random feature-detecting network. The live data acquisition module is not discussed is this thesis for legal reasons.

4.3.1 Software Design Choices

The embedded software implementation of the random feature-detecting network and the live data acquisition module adheres to three key software design choices: multithreading when applicable for performance improvement, leveraging L1 caches when possible, and offloading specific functions to the Jetson TX2's on-board GPU only when concurrent operations would be beneficial. The embedded software program utilises multithreading by separating data acquisition and the random feature-detecting network into two distinct threads. This choice of using multithreading is based on the fact that data acquisition is quicker than executing the random feature-detecting network, allowing the SPAD camera to acquire the next batch of data while the network processes the current batch. Consequently, the random feature-detecting network only idles during data transfer between threads.

The implementation leverages the 80 kB L1 cache on the Jetson TX2's ARM Cortex A57. The cache can accommodate each raw image's 1,024 numerical values, resulting in minimal execution time for data transfers between threads. Additionally, cache hits can be utilised for serial arithmetic functions, which is faster than transferring data between the central processing unit (CPU) and GPU for concurrent operations and then transferring the data back to the CPU.

The GPU in the Jetson TX2 is leveraged for embedded software implementation of the random feature-detecting network. CUDA is used to implement some of the operations for computation in the Jetson TX2's on-board GPU. CUDA is a programming model designed by Nvidia to efficiently utilise the GPU for computation. CUDA transfers the relevant data between the CPU to GPU such that the

GPU processed data is transferred back to CPU for further processing. Operations are parallelised using CUDA only when the reduction in time from concurrent GPU execution outweighs the additional time spent transferring data to the GPU.

The data transfer between the CPU and GPU extends the time it takes to execute an operation in the GPU. This makes it unattractive to parallelise certain operations in the random feature-detecting network. For example, renormalising each of the pixel values in the data manipulation module (Section 4.1.2) is not implemented in the GPU. This is because the image is only 32×32 in resolution so iterating the same operation 1,024 times in series on the L1 cache is faster than moving these values to the GPU for computation. A similar argument applies to other operations, such as splitting the convolved image into positive and negative values, and resampling the feature vector to a fixed length. These operations all require small number of iterations so their execution time is still shorter than transferring the data to the GPU for parallel computation. The main reason only small iterations are required is because the SPAD array size are 32×32 which means any feature images or vectors resulting from the SPAD camera's raw images are also small in size.

Convolution, summation of matrices to vectors, and matrix multiplication (used for computing linear classification Eq. 4.1) are the only operations from the random feature-detecting network that are implemented in CUDA for execution on the Jetson TX2's on-board GPU. In particular, the summation of matrix rows and columns (for compressing the cropped feature vectors as shown in Figure 4.2 in Section 4.1.2) is implemented using CUDA warp shuffle functions, which access the image at the register level for fast summation operations.

Two design steps are implemented to minimise the transfer time of data between CPU and GPU for operations implemented in CUDA. First, parallelised operations are called only once. For example, convolution in the implementation is only called once. This is because the values of its multiple filters are concatenated into a single array and then the multiple output images are concatenated into another single array. Second, the GPU's shared memory is used to store values.

For a detailed explanation of CUDA and its utilisation in the embedded software implementation of the random feature-detecting network for GPU execution, refer to Section 4.3.2.

4.3.2 CUDA Programming Model

The embedded software implementation on the Jetson TX2 utilises CUDA to parallelise certain operations within the random feature-detecting network. This optimises the execution of these operations. CUDA is a parallel computing platform and programming model introduced by Nvidia in 2006 [118]. Initially designed to enhance the processing capabilities for real-time, high-definition 3D graphics rendering, CUDA provides a software environment that supports combined CPU and GPU programming. It is specifically developed to optimise computing performance for Nvidia graphical processing units (graphical processing units (GPUs)) [118].

The fundamental idea behind CUDA is to enable collaborative work between the

CPU (referred to as "the Host") and the GPU (referred to as "the Device"). The CPU functions as the controller, managing data flow and initiating functions, while the GPU handles the computation of various functions and oversees memory management between them. In CUDA, kernels are created to address sub-problems. These kernels are functions executed in threads within thread blocks. A thread, as defined in CUDA, represents the execution of a process, and a block consists of 1,024 threads. These threads are organised into a grid structure [10].

A common structure of a GPU Program follows the process below [10]:

1. CPU allocates storage on GPU (cudaMalloc function)
2. CPU copies input data from CPU to GPU (cudaMemcpy function)
3. CPU launches kernel(s) on GPU to process the data (Kernel launch)
4. CPU copies results back to CPU from GPU

Kernels are executed on threads and blocks because of the structure of the GPU. The syntax to initialise kernels with multiple threads and blocks is:

```
«dim(bx,by,bz), dim(tx,ty,tz), shared memory» kernel (kernel
variables);
```

where `dim(bx,by,bz)` is the dimensions of the number of blocks that are used to execute the kernel. `dim(tx,ty,tz)` is the dimension for the number of threads. Shared memory is the amount of memory shared between threads; this memory is managed by the GPU. In the CUDA model, each thread accesses its own local memory, and shared memory with other threads in the same block. The GPU also has a global memory that is shared between blocks and communicates with the CPU memory [10].

Threads are organised into blocks because it is optimal to run threads for the same kernel in the same Streaming Multiprocessor (SM) of a GPU. The SM creates, manages, schedules, and executes threads in groups of 32 parallel threads called warps. It contains multiple simple processors and a memory, which is where the thread accesses its local and shared memory. The number of Streaming Multiprocessors (SMs) in a GPU depends on the particular type of board. One block of threads must be run in one SM only, but a SM can run multiple blocks if the blocks are small enough. The number of blocks depends on the memory and register usage of a kernel. The allocation of blocks to SMs are all done by the GPU and this is why it is important to declare blocks and threads in programs [10].

4.4 Difference between Offline and Embedded Software Implementations

The random feature-detecting network undergoes offline training and preliminary testing before being implemented as an embedded software program for real-time operation on the Jetson TX2 [8]. The offline training and preliminary testing follows

the procedures described in Section 4.2. Offline training and testing are essential for empirically determining convolution filter values, a process discussed in Section 4.2.5, and classification matrix Z values. Once filter values are selected, they are used in the embedded software program along with their corresponding classification matrix Z.

Section 4.4.1 details one difference between the offline and embedded software implementations. Section 4.4.2 details implementation differences for testing the embedded software implementation of the random feature-detecting network for real-time airplane target classification, compared to the offline software implementation used for conducing preliminary testing of the target classification performance of the random feature-detecting network. This difference arises due to the difference in imaging setup for testing the real-time performance of the embedded software implementation compared with the imaging setup used for collecting data to train and preliminary test the offline software implementation of the random feature-detecting network. Despite these variances, airplane classification performance is minimally impacted, as demonstrated by the classification results in Section 4.6.3.

4.4.1 Denoising of Real-Time Images

In the offline implementation, a moving average of 5 raw images is employed for denoising during offline training and preliminary testing. This number is identified as the minimum required to effectively create an average image and remove noise without compromising the information in each raw image. However, for the embedded software implementation, a block average of 16 raw images is utilised instead. This is because a batch of 16 raw images is transferred to the Jetson TX2 at each camera acquisition.

In the embedded software implementation, camera acquisition operates as a separate thread from the random feature-detecting network (as detailed in Section 4.3). The camera halts the acquisition of data while the network performs classification on a batch of raw images. The SPAD array flash LiDAR System A acquires data at a much faster rate than the classification thread. Therefore, to optimise the process, it is practical to process an average of 16 raw images for each execution of the random feature-detecting network. This approach avoids the camera waiting for the network to iterate its execution 12 times to classify a moving average of 5 raw images for each batch of 16 raw images. Processing all 16 raw images enables the network to classify the target with all available information while allowing the camera to capture the real-time scene more rapidly.

Given that camera acquisition and classification are implemented in separate threads in the embedded software program (as explained in Section 4.3.1), calculating an average of 16 raw images does not risk losing any real-time information and keeps the processing time brief.

4.4.2 Additional Check in Embedded Software Implementation

When testing the embedded software implementation of the random feature-detecting network for classifying airplane models, the imaging environment that is used is different compared with the original environment used for collecting data that is utilised for offline training. This variation leads to differences in appearance of the raw images, used for testing the embedded software implementation, compared with the raw images used for training. Consequently, an additional check is incorporated into the random feature-detecting network for the embedded software implementation. This check is designed to prevent background noise from being mistaken as a target for classification. The detected target from the tracking module (Section 4.1.1) goes through an additional check where it is only considered to be an airplane target only if its distance falls within the range where the airplane is positioned in this new imaging environment. This is determined by evaluating whether the 5th percentile value of the average image is below a threshold value of 940 clock cycles. This threshold value is computed from the 5th percentile value of an average image of an airplane, where the airplane is held at the furthest possible point in the new imaging environment.

4.5 Airplane Dataset for Training and Preliminary Testing

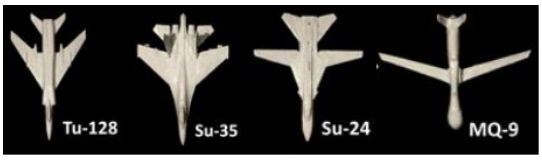


Figure 4.4: Model airplanes used for imaging [102].

Raw images of airplanes are employed for offline training and preliminary testing (as outlined in Section 4.2.5) of the random feature-detecting network. No raw images are used for validation. An explanation on the splitting the collected raw images of airplanes for training and preliminary testing is detailed in Section 4.5.1. The dataset comprises recordings of model airplanes, namely the Tu-128, Su-35, Su-24, and MQ-9, dropped in various orientations in front of the SPAD array flash LiDAR system. Figure 4.4 shows the images of the airplanes. The airplanes are painted white for standardisation. The dimensions of these model airplanes do not exceed 17 cm$\times$12 cm$\times$2 cm, ensuring they remain within the camera's field-of-view (FOV) when imaged at a closer distance.

Figure 4.5 shows the imaging set-up for collecting recordings of these airplanes. A dropbox is used to catch the airplanes after it is dropped. Each model airplane is dropped 100 times which produces 100 recordings. This results in a diverse dataset for training and preliminary testing. Separate recordings are made for each drop, along with horizontally flipped versions added to double the number of recordings in the dataset.

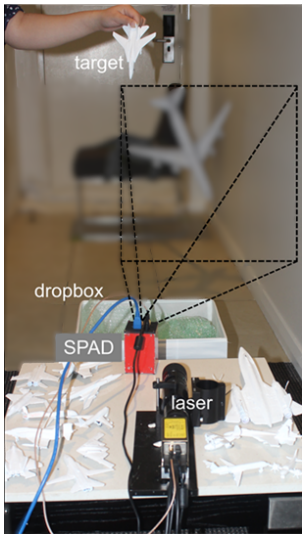


Figure 4.5: Imaging set-up—the black-dotted prism depicts the SPAD camera's FOV. [103]

In total, there are 800 recordings in the collected dataset, with 200 recordings for each airplane type. Each recording, on average, contains 427 raw images. Each recording is reduced in duration such that the airplane target appears in each of its raw images. Figure 4.6 shows a sample of the average images, where each image is from the denoising module processing a batch of 16 raw images.

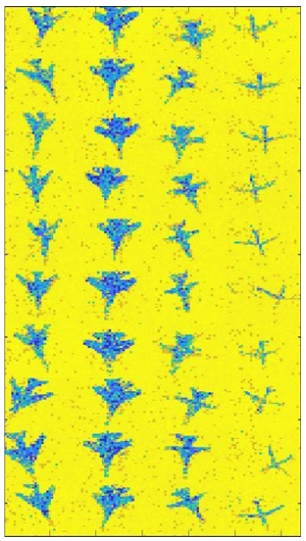


Figure 4.6: A selection of average images from the training dataset. The airplane types are (from left to right) Tu-128, Su-35, Su-24 and MQ-9. [102].

The airplanes are dropped in various orientations 40 cm from the front of the camera, simulating a large airplane flying across the camera's FOV. This approach ensures diversity in the dataset, with images of the airplane captured in various ori-

entations. The SPAD array flash LiDAR system's laser pulse width is 17 ns, and the SPAD camera has a timing resolution of 390 ps, resulting in pixel values corresponding to 44 different time bins for the model airplanes.

The specifications of the SPAD array flash LiDAR system used in this chapter, referred to as System A, are provided in Section 3.4 of Chapter 3. The target is consistently positioned 400 mm away from the camera, while the laser is located 473 mm away from the target, representing the calculated minimum distance for the laser.

4.5.1 Allocating Training and Testing Datasets

The allocation of raw images into either offline training or testing is determined based on the collected recordings. In other words, all the raw images from the same recording will be used to create images for either the training or testing dataset. Dividing the raw images based on their recordings is to prevent the training or testing dataset from containing too many images of an airplane at a certain position of its trajectory during its drop. Each recording contains raw images of the airplane model dropping in front of the SPAD array flash LiDAR system once. Therefore, each recording consists of raw images of the airplane appearing at different points of the FOV, ranging from top of the FOV, to bottom of the FOV.

As discussed in Section 4.2.5, 20 instances of training and testing are conducted for each set of filter values. At each instance, 99.5%[5] of all the collected recordings (equivalent to 796 recordings) are randomly selected for training. The remaining recordings (i.e. 4 recordings) are allocated for testing.

To understand the number of images used in training and testing, a few factors need to be considered. First of all, the average image reconstructed in the denoising module of the random feature-detecting network is used here to discuss the number of images used for training and testing the network. Even though the raw images are the inputs into the random feature-detecting network, their features are not directly used for classification. Instead, it is the average image's features that are used for classification. Therefore, the number of raw images in the recordings does not reflect the amount of images used for training and testing because the denoising module reconstructs an average image from a moving average of 5 raw images (as described in Section 4.4.1). The number of average images created from the collected recordings is considered as the number of images used for training and testing.

Secondly, the raw images from different recordings cannot be used together to create an average image. This is because each recording contains raw images of each separate airplane drop. Therefore, the total number of average images used in the training and testing dataset cannot be simply estimated from the total number of raw images from all recordings. Instead the number of average images must be calculated from the number of raw images in each separate recording first. Then the number of average images from each of the recordings for training are summed together to

[5] [102, Section 1.2] incorrectly reports the recordings are split in half for training and testing.

determine the total number of average images used in the training dataset. Likewise, the total number of average images for testing is calculated in the same manner.

Thirdly, the number of average images that can be created from each recording is different for each recording. Each recording is collected from a separate airplane drop hence will contain a different number of raw images (corresponding to the duration of the airplane drop). Therefore, the number of average images from each recording can only be estimated from the mean number of raw images in each recording. For the airplane recordings, the mean number of raw images per recording is 427. Therefore, each recording can produce 423 average images during training and testing, given an average image is created from 5 raw images (as described in Section 4.4.1).

By considering all these factors, the number of average images used in training can be estimated to be $796 \times 423 = 336,708$, given 796 recordings are used each time for training. And the total number of images used for testing is esimated to be $4 \times 423 = 1,692$, given 4 recordings are used each time for testing.

4.6 Results

The performance of the embedded software implementation of the random feature-detecting network on the Jetson TX2 is discussed in this section. A demonstration of the denoising and tracking modules is shown in Section 4.6.1 and 4.6.2, respectively. Section 4.6.3 provides the testing methodology and target classification performance of the embedded software implementation of the random feature-detecting network using different numbers of convolutional filters. The execution time of one iteration of the random feature-detecting network in the embedded software implementation for different numbers of filters is provided in Section 4.6.4. In Section 4.6.5, the target classification performance of the random feature-detecting network with 64 filters is provided when the network is used to classify a moving airplane.

4.6.1 Denoising

A comparison between raw images and the average image from the denoising module is shown in Figure 4.7. The figure shows that averaging a batch of raw images significantly increases the amount of details shown in the reconstructed image. Figure 4.7 consists of a 4×7 grid of images. Each column is for one type of airplane being classified. The first 6 rows of each airplane are the first 6 raw images taken from a batch of 16 raw images of each type of airplane. The bottom row is the average image of the same batch of 16 raw images for each type of airplane. This comparison shows the significant improvement in the amount of airplane details shown in the average image in comparison with the raw images.

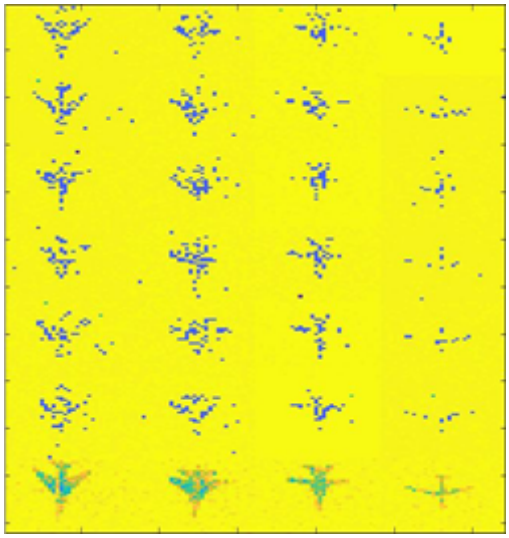


Figure 4.7: Effect of averaging 16 raw images. Here the first 6 rows contain the raw images and the last row contains the averaged images. Section 4.4.1 discusses why 16 raw images are averaged together in the embedded software implementation of the random feature-detecting network [102].

4.6.2 Tracking

The tracking module detects where the target is in each of the average images. Figure 4.8 shows some example average images where the model airplane is detected by the tracking module and a bounding box is drawn using the borders computed by the tracking module.

Figure 4.8: The bounding boxes determined by the tracking module for each airplane [102].

4.6.3 Real-Time Classification of a Static Image of a Target

In the embedded software implementation on the Jetson TX2, the random feature-detecting network undergoes real-time testing with different numbers of filters. The classification performance for each filter quantity is evaluated with each class of model airplane being held in three distinct orientations: vertically downwards, angled downwards, and dynamically. The dynamic orientation involves moving the airplane in a small circular motion to simulate the camera experiencing vibrations while on a mobile platform such as an UAV. For all three cases, the airplanes are positioned 40 cm away from the camera, and the dynamic case involves the airplanes moving within a radius of 1.5 cm from that position.

For each filter quantity and each orientation of holding the model airplane, the random feature-detecting network's classification performance with 32 batches of SPAD raw images is measured. The tracking module, as described in Section 4.1.1, and the additional detection check outlined in Section 4.4.2, ensure that all input batches to the random feature-detecting network contain an airplane in the raw image's FOV.

Figure 4.9 presents examples of average images of each airplane at different orientations used for classification testing. The average images in each column in this figure correspond to consecutive batches of input raw images acquired by the SPAD camera. The average images for the dynamic case resembles those for the vertical case because the dynamic case's circular trajectory is perpendicular to the SPAD's image plane. Therefore, the airplane appears to be moving slightly up and down relative to the SPAD camera. Despite the airplane model moving in this dynamic case, the average image of the airplane still appears static.

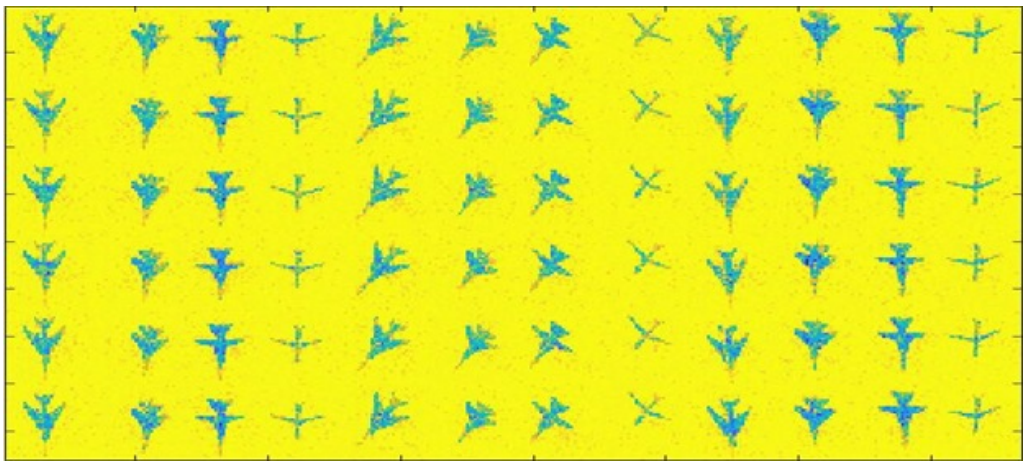


Figure 4.9: Average images saved from testing the classifier using a live raw image feed. There are 12 columns of average images. Every fourth column shows average images of each airplane being held in a different way (starting from the left): a) vertical, b) angled and c) dynamic (where the airplane is moving in a circle) [102].

Multiclass confusion matrices are used to analyse the performance of the ran-

dom feature-detecting network with different numbers of filters and orientations of the airplanes. An example of a multiclass confusion matrix is given in [78]. The predictions from the set of 32 classifications are recorded and used for calculating the performance measures in Table 4.1[6]. An explanation of these metrics is given in Section 3.7 of Chapter 3. The predictions are recorded as percentages for each airplane type. A discussion of these results and performance comparison with similar classification algorithms in the literature will be given in Section 4.7.

Table 4.1: Performance measures for different numbers of filters.

Measures	**Airplane Orientation**	**Number of Filters**				
		4	**8**	**16**	**32**	**64**
Overall Accuracy (%)	Vertical	95.31	98.44	98.83	99.22	100
	Angled	93.75	95.31	97.27	98.44	99.61
	Dynamic	91.41	94.53	96.88	97.66	98.44
	Average	**93.49**	**96.09**	**97.66**	**98.44**	**99.35**
Macro Recall (%)	Vertical	90.63	96.88	97.66	98.44	100
	Angled	87.50	90.63	94.53	96.88	99.22
	Dynamic	82.81	89.06	93.75	95.31	96.88
	Average	**86.98**	**92.19**	**95.31**	**96.88**	**98.70**
Macro Precision (%)	Vertical	92.37	97.22	97.77	98.48	100.00
	Angled	88.81	92.45	94.91	97.01	99.24
	Dynamic	85.99	91.05	94.64	95.86	97.06
	Average	**89.06**	**93.57**	**95.77**	**97.12**	**98.77**
Macro F1 measure	Vertical	0.91	0.97	0.98	0.98	1.00
	Angled	0.88	0.92	0.95	0.97	0.99
	Dynamic	0.84	0.90	0.94	0.96	0.97
	Average	**0.88**	**0.93**	**0.96**	**0.97**	**0.99**
Error Rate (%)	Vertical	4.69	1.56	1.17	0.78	0.00
	Angled	6.25	4.69	2.73	1.56	0.39
	Dynamic	8.59	5.47	3.13	2.34	1.56
	Average	**6.51**	**3.91**	**2.34**	**1.56**	**0.65**

This space is intentionally left blank.

[6]For Table 4.1, a similar table was presented in [102]. However, in the table in [102], an error was made where overall accuracy was the same as macro recall. The error is fixed and the re-calculated overall accuracy values are shown in this table. Also, average error rate in [102] is named as error rate in this table to match the terminology used in the literature.

4.6.4 Execution Time

The average execution time of the random feature-detecting network on the Jetson TX2 for conducting one target classification is presented in Table 4.2. These are the average execution times when the dynamic case is tested. The overall execution time is only the time it takes to execute the random feature-detecting network because data acquisition and the network are implemented in two separate threads, as discussed in Section 4.3.1.

Table 4.2: Average execution time of the random feature-detecting network in performing one target classification when it is implemented with different numbers of filters (ms). [102]

	Number of Filters				
	4	**8**	**16**	**32**	**64**
Tu-128	31.99	30.00	40.61	46.22	64.79
Su-35	28.63	32.39	39.76	48.43	66.45
Su-24	29.06	31.85	35.98	46.79	62.99
MQ-9	31.04	31.38	38.07	49.80	66.04
Average	30.18	31.40	38.61	47.81	65.07

The power consumption of the Jetson TX2 when executing the embedded classification program is found to be 5.1 W. Other embedded boards in the current Nvidia Jetson TX2 series have only improved in peripheral, as shown by the specifications in [8]. The GPU and CPU are the primary hardware that runs the embedded software program. From the specifications in [8], the GPU and CPU have remained the same since the time when the embedded software program is developed and implemented on the TX2 board.

4.6.5 Real-Time Classification of Moving Target

The random feature-detecting network is tested for real-time classification of a moving target. An airplane is simulated to fly past the camera by fixing the camera on a gimbal, and the gimbal moves the camera at a constant speed, scanning a static vertical model airplane from bottom to top. This simulates the airplane passing through the FOV of the SPAD camera.

The camera starts at a position where the airplane is outside its FOV. As the embedded software program for the random feature-detecting network is manually started, the gimbal begins moving the camera to scan the airplane and collect images. The gimbal stops moving, and the embedded software program is manually stopped when the airplane is outside the FOV. The number of detections during each scan is recorded, and the random feature-detecting network's confidence rate for each airplane type during the midpoint of the scan is shown in Table 4.3:

In each row of Table 4.3, the confidence rate is displayed for each type of plane. The confidence rate represents the percentage of classifications labelling the image as a certain type of airplane. Each group of four confidence rates in each row of Table

4.3 sums to 100. This is because classification is tested for each type of airplane one at a time. The percentage from the confidence rate is calculated over the number of classifications performed up to the midpoint of the scan. The number of classifications conducted by the midpoint of the scan is approximately half the total number of detections provided in Table 4.3.

Classification accuracy in this scenario is only reported for the random feature-detecting network with 64 convolution filters. This is because the previous static test results (as shown in Table 4.1) indicated that this number of filters had the best performance. A discussion of these results and a performance comparison with similar classification algorithms in the literature will be provided in Section 4.7. The right-most column presents the number of successful target detections.

Table 4.3: Target classification performance of the random feature-detecting network using 64 convolutional filters when the airplane is moving at a constant speed in FOV. The right-most column presents the number of successful target detections. [102]

Airplanes scanned	**Confidence rate of the mid-scan's raw image (%)**				**No. of detections**
	Tu-128	**Su-35**	**Su-24**	**MQ-9**	
Tu-128	**91.30**	0	8.70	0	44
Su-35	0	**86.96**	13.04	0	45
Su-24	0	0	**100**	0	43
MQ-9	0	0	4.76	**95.24**	40

4.7 Discussion

From Table 4.1 in Section 4.6.3, it is evident that the random feature-detecting network performs best with 64 filters. In this setting, its overall classification accuracy reaches 99.35%, and its error rate is at its lowest, 0.65%. Its F1 measure is 0.99, given its recall and precision scores are at 98.70% and 98.77%, respectively. Overall, the classification accuracy improves with an increasing number of filters. The average execution time for this classifier with 64 filters is around 65.07 ms. The power consumption for the classifier to execute in real time is 5.1 W. As there is no reported power consumption in other related work, the power consumption cannot be compared.

At the time of publication of the random feature-detecting network as a real-time classification algorithm [102], there was limited literature related to the classification of SPAD LiDAR data. The only known literature at the time was [136].

Since the publication of the random feature-detecting network in [102], additional literature for target classification have been published [117, 66, 114, 142, 134, 112, 113] but their target classification performance will not be compared with the random feature-detecting network. A review of the literature related to target classification of SPAD LiDAR data can be found in Section 2.5 of Chapter 4, which includes other levels of target discrimination, such as determining the orientation of a target, and performing target recognition and identification.

To provide a more comprehensive performance comparison with the literature at the time of publication of the random feature-detecting network [102], the random feature-detecting network is also compared with other target classification algorithms of 3D point clouds and RGB images. Classification of 3D point clouds is similar to SPAD LiDAR data, given both provide a 3D representation of a target. As for RGB images, the target classification algorithms that are selected for comparison are demonstrated to classify RGB images with pixel resolutions from 32×32 to 256×256, which is similar to the 32×32 pixel resolution of the SPAD LiDAR data that are classified by the random feature-detecting network.

The target classification algorithms selected for comparison utilise Convolutional Neural Networks (CNNs). This type of classification algorithm is chosen because the random feature-detecting network uses the convolution operation to extract features from the image, which is similar to how CNNs utilise the convolution operation. These classification algorithms were state-of-the-art CNNs at the time of publication of the random feature-detecting network [102]. Coincidentally, the only known literature related to classification of SPAD LiDAR data also utilises CNNs. Therefore, the rest of this section will be about comparing the random feature-detecting network with CNNs.

The classification performance of the random feature-detecting network is comparable to CNNs in the selected literature. The performance summary of these CNNs and the random feature-detecting network is presented in Table 4.4. The table incorporates a snapshot of the random feature-detecting network's performance when utilising 64 convolutional filters. In each row of performance metrics, the best value is highlighted in bold. The number of training and testing images used by each algorithm are provided in Table 4.5.

This space is intentionally left blank.

Table 4.4: Target classification performance overview of CNNs and the random feature-detecting network.

Measures	**2D network for SPAD data [136]**	**3D network for SPAD data [136]**	**VoxNet [98]**	**AlexNet [77]**	**GoogLeNet [151]**	**ResNet50 [62]**	**ShuffleNet [180]**	**Random feature-detecting network (64 filters)**
Overall accuracy (%)	N/A	N/A	N/A	40.09 [141]	68.15 [141]	68.96 [141]	N/A	**99.35**
Macro recall (%)	95.00	97.00	N/A	N/A	N/A	N/A	N/A	**98.70**
Macro precision (%)	95.00	97.00	N/A	N/A	N/A	N/A	N/A	**98.77**
Macro F1 measure	0.95	0.97	0.73	N/A	N/A	N/A	N/A	**0.99**
Error rate (%)	N/A	N/A	N/A	N/A	N/A	N/A	26.30	**0.65**
Average execution time	N/A	N/A	1ms to 0.5s on a Tesla K40 GPU	N/A	N/A	N/A	**15.2 to 108.8ms on a single thread on the Qualcomm Snapdragon 820 processor**	65.07 ms on the Jetson TX2
Power consumption	N/A	N/A	N/A	N/A	N/A	N/A	N/A	**5.1W**

Table 4.5: Overview of the number of training and testing images used by CNNs and the random feature-detecting network.

Measures	**2D network for SPAD data [136]**	**3D network for SPAD data [136]**	**VoxNet [98]**	**AlexNet [77]**	**GoogLeNet [151]**	**ResNet50 [62]**	**ShuffleNet [180]**	**Random feature-detecting network (64 filters)**
No. of training images	1923 SPAD images and 14 million 32×32 RGB images	1615 SPAD images	N/A	1.2 million 256×256 RGB images [77] and 100,000 32×32 RGB images [141]	1.2 million 224×224 RGB images [151] and 100,000 32×32 RGB images [141]	1.2 million 224×224 RGB images [62] and 100,000 32×32 RGB images [141]	N/A	336,708 SPAD images
No. of testing images	60 SPAD images	60 SPAD images	N/A	20,000 32×32 RGB images [141]			100,000 224×224 RGB images	1,692 SPAD images

In the rest of this section, a discussion will be provided for each algorithm's performance compared with the random feature-detecting network.

4.7.1 Comparison with Other SPAD array flash LiDAR Classification Algorithms

The authors in [136] developed a 2D network and a 3D network for target classification of data collected by a 64×64 SPAD array flash LiDAR system. The SPAD LiDAR data is converted to a 320×320 2D binary image for target classification by the 2D network. For the 3D network, the SPAD LiDAR data is converted to a 3D point cloud. The SPAD LiDAR data used to evaluate the 3D network has the same dToF measurement for all pixels related to the target. Therefore, the dToF measurement of each pixel is repeated over the z-plane to create a 3D point cloud for the 3D network [136]. In terms of algorithm processing, the use of point clouds has a different data structure to the raw images used for the random feature-detecting network. The point cloud proposed by the authors in [136] repeats a raw image over several z-planes while the random feature-detecting network only utilises one raw image.

When the random feature-detecting network utilises 16 or more filters, it results in higher precision, recall and F1 measure than the 2D network proposed by the authors in [136]. However, there are a few differences between the 2D network and the random feature-detecting network. First, the 2D network uses a VGG-16 model that is pre-trained on 14 million standard images and adds an extra four layers to fine-tune the network to classify SPAD LiDAR data [136]. This makes a total of 18 layers, which is much larger than the random feature-detecting network. Moreover, the 2D network here is able to use transfer learning to extend a pre-trained VGG-16 model. This is different from the random feature-detecting network, which is unable to store memory from previous learning.

The training data used by the 2D network is also different to the random feature-detecting network. There are 1,923 raw images used for training where the raw images are partitioned such that there are 641 raw images per class [136]. Another difference is that the raw images used to test the classification performance of the random feature-detecting network are 32×32 in size, while the original raw images used by the authors in [136] are 64×64, which are then scaled up to 224×224 to be compatible with the VGG-16 model's input layer [136]. Furthermore, the authors in [136] classified eight different objects into three classes (airplane, chair, Uncrewed Autonomous Vehicle (UAV)) [136], while the random feature-detecting network presented in this chapter is tested to classify four different airplanes into four different classes, where there is a one-to-one correspondence between each object and a class.

When compared with the 3D network proposed by [136] for SPAD LiDAR data, the random feature-detecting network has equal classification performance when using 32 filters and performs better at 64 filters. Despite the 3D network using a smaller number of training images, its structure is more complex than the random feature-detecting network. The 3D network is an 11-layered CNN [136], while the random feature-detecting network can be viewed as a single-layer CNN. The more

complicated structure implies that the 3D network's computation time and power may be more than those of the random feature-detecting network, making it less efficient to execute on embedded boards.

4.7.2 Comparison with VoxNet: a 3D Point Cloud Classification Algorithm

The random feature-detecting network performs better than VoxNet [98] for all the different numbers of filters that it has been developed to utilise. The macro F1 measure of the Multi-Resolution VoxNet is 0.73 [98], while the random feature-detecting network has a lower macro F1 measure of 0.88 at 4 filters. However, the random feature-detecting network has only been shown to classify 4 airplane types, while VoxNet has been demonstrated to classify 14 classes of objects [98]. VoxNet's training data size is unknown, so it is difficult to determine whether the random feature-detecting network is trained on a larger dataset. VoxNet also uses many more layers than the random feature-detecting network, which can be viewed as a single-layer CNN. VoxNet's CNN consists of two identical 5-layer networks working in parallel and then one final classification layer. The two parallel networks are used to process the data that is voxelised by two different-sized grids, 0.1 m and 0.2 m. This is to ensure the classifier is rotationally invariant.

VoxNet executes faster than the random feature-detecting network. The execution time of VoxNet varies between 1 ms to 0.5 s when executed on a Tesla K40 GPU [98]. The execution time increases with more pixels in the image to be classified. Given the SPAD LiDAR data has a low image resolution, a fair comparison with the random feature-detecting network's execution time will be using the lower bound of VoxNet's execution time range. This means VoxNet's 1 ms execution time is much faster than the random feature-detecting network's 65.07 ms execution time. However, a Tesla K40 GPU is used for VoxNet's implementation. The different computational hardware can enable the VoxNet to execute faster.

4.7.3 Comparison with RGB Image Classification Algorithms

The random feature-detecting network is compared with four different CNNs that classify RGB images. They are AlexNet [77], GoogLeNet [151], ResNet50 [62] and ShuffleNet [180].

The first three networks are extended by the authors in [141] to classify 32×32 RGB images. Therefore, their performances will be discussed together. Table 4.6 shows the average, minimum, and maximum accuracies of these three CNNs at classifying 20 different classes of objects from the datasets CIFAR-100 and CIFAR-10. Classes from CIFAR-100 and CIFAR-10 each make up half of the 20 testing classes.

The classification accuracy for each of the classes varies a lot and is much lower than the random feature-detecting network accuracy (with any number of filters). Even though the number of testing classes is greater than the number of airplane types used to test the random feature-detecting network, all three of these CNNs

Table 4.6: Target classification accuracy of different CNNs by the authors in [141].

	AlexNet [77]	**GoogLeNet [151]**	**ResNet50 [62]**
Average accuracies	40.0875	68.152	68.96
Maximum accuracy	95.9 (Truck from CIFAR-10)	97.1 (Truck from CIFAR-10)	99.2 (Motorcycle from CIFAR-10)
Minimum accuracy	0 (Bed from CIFAR-100)	0.84 (Streetcar from CIFAR-100)	33.4 (Table from CIFAR-100)

have far more layers than the random feature-detecting network. AlexNet has 8 layers [77], GoogLeNet has 22 layers [151], and ResNet50 has 50 layers [62]. Also, the authors take the pre-trained models of these CNNs and further fine-tune these models with three datasets: CIFAR-100 with 32×32 RGB images, ImageNet with varying sizes of RGB images, and CIFAR-10 with 32×32 RGB images. Further fine-tuning is possible because these models allow transfer learning. This is different from this chapter's random feature-detecting network as it is unable to store memory from previous learning.

Furthermore, the minimum accuracy of these three networks is always from one of CIFAR-100's object classes and the maximum accuracy always comes from one of CIFAR-10's object classes. One possibility for the difference between CIFAR-10 and CIFAR-100 is that CIFAR-10 only has 10 object classes while CIFAR-100 has 100 object classes with 20 superclasses. Superclasses are a broader category that encompasses more than one of the classes, so that each image has two labels, one representing its class and the other representing its superclass. Hence, each of the training images from CIFAR-100 contains two labels, which is different from the training dataset used to test the random feature-detecting network. Only one label is used for each image in the training dataset for the random feature-detecting network.

Each of these CNNs has been trained by approximately 1.2 million RGB images. Each training dataset for each CNN has different images, where they are datasets from different years of the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [77, 151, 62]. The authors of [141] further fine-tune the network with the CIFAR-100 dataset, which has 500 RGB images sized 32×32 per object class and 100 classes in total, and CIFAR-10 dataset, which has 5,000 RGB images sized 32×32 per object class and 10 object classes in total.

The random feature-detecting network consistently outperforms ShuffleNet [180] in terms of error rates across all numbers of filters. With its highest error rate at 6.51% when using 4 filters, the random feature-detecting network compares favourably with ShuffleNet, which exhibits a lowest error rate of 26.3% [180]. It is important to note that ShuffleNet is evaluated on the 224×224 RGB images from the ImageNet 2012 classification dataset [180]. This is different to the random feature-detecting network, which is assessed using live images of airplanes for real-time classification. Although ShuffleNet reports multiple error rates based on different filter configura-

tions, the exact method of error rate calculation is unclear. In contrast, the error rate in Table 4.1 for the random feature-detecting network is determined by averaging the percentages of false negatives and false positives for each model airplane.

In terms of execution times, the random feature-detecting network, operating with varying execution times between 30.18 and 65.07 ms, demonstrates competitive performance based on the number of filters employed. ShuffleNet, implemented on a single thread using a mobile processor (Qualcomm Snapdragon 820), shows similar execution times, ranging from 15.2 to 108.8 ms, with the exact duration depending on the configuration used [180]. While both networks are implemented on mobile processors, the random feature-detecting network leverages three threads and ShuffleNet uses a single thread. Since both networks have similar execution times, it is possible that ShuffleNet may have a shorter execution time if it can be implemented in multiple threads. Furthermore, ShuffleNet has been tested to process higher resolution images compared with the random feature-detecting network. This suggests that ShuffleNet may execute target classification faster than the random feature-detecting network, given its execution time similarity to the random feature-detecting network despite processing higher resolution images.

4.8 Chapter Summary

A target classification algorithm called the random feature-detecting network is implemented on the Jetson TX2 with a live data acquisition module to classify SPAD LiDAR data in real time. The algorithm's embedded software implementation processes a batch of 16 raw images into an average image, then extracts different feature from the average image and stores these features' information into a vector, which is then processed by linear classification to determine the target's class. Part of these features are extracted by convolution filters, whose values are randomly generated.

The classification performance of different versions of the algorithm are evaluated, where each version has a different number of filters. The different numbers of filters are 4, 8, 16, 32 and 64. For classifying 4 different types of model airplanes (Tu-128, Su-35, Su-24 and MQ-9), the classifier achieves an overall classification accuracy of 99.35% with an F1 score of 0.99 when using 64 filters, which is better than the classification performance of the two SPAD LiDAR data classification algorithms developed by the authors in [136], which was the only literature on target classification of SPAD LiDAR data at the time of publication of the random feature-detecting network in [102]. When compared with other target classification algorithms that process 3D point clouds and RGB images, the random feature-detecting network also offers better target classification performance.

The algorithm and live data acquisition module's total power consumption is 5.1 W. The algorithm's execution time in the embedded software implementation varies between 30.18 and 65.07 ms. The execution time depends the number of convolution filters that the algorithm uses. Despite the ShuffleNet target classification algorithm demonstrating a faster execution time, the random feature-detecting network was

the only target classification algorithm that had been demonstrated to perform real-time classification for SPAD array flash LiDAR images at its time of publication in [102]. Overall, this chapter has presented a real-time target classifier that executes on a Jetson TX2 and has a low SWaP requirement, which means it can be mounted on mobile platforms for remote surveys.

If the classifier is to be implemented on a different embedded board in the future, the hardware of the board must meet or exceed the computational performance capabilities of the Jetson TX2 to ensure that the classifier can maintain real-time execution.

In addition to imaging in air, remote surveys often encounter environments with obscurants, such as natural water bodies (e.g. rivers or oceans) and outdoor fog. Since the target classification algorithm discussed in this chapter has only been tested on targets imaged in air, it is important to consider these other environments. Obscured environments pose challenges in collecting accurate dToF measurements of targets, which can complicate target classification. Therefore, the next chapter will focus on image reconstruction of targets submerged in natural water environments, where accurately reconstructed images can provide better target classification performance.

Chapter 5

SPAD Image Reconstruction of Targets Submerged in Natural Waters

To increase the sensing versatility of mobile platforms such as Uncrewed Autonomous Vehicles (UAVs), this chapter focuses on above-water Single Photon Avalanche Diode (SPAD) Light Detection and Ranging (LiDAR) imaging of submerged targets in natural waters, which includes rivers and oceans. Natural water environments are common when performing remote surveillance. The ability to perform target detection and classification of underwater targets from the air is crucial for effective maritime operations for the military.

The turbid nature of natural water environments poses significant challenges for SPAD LiDAR imaging. Suspended particles, organic matter, and water itself induce light absorption [132] and scattering [109], which can degrade the Signal-to-Noise Ratio (SNR) of SPAD LiDAR images, complicating target discrimination tasks. This chapter focuses on algorithms that reconstructs a clear image of the submerged target for target detection and classification.

As an initial approach, two basic image-reconstruction algorithms for submerged target are tested in this chapter: the detection threshold and median algorithm, and the histogram averaging algorithm. Both are simple computational methods that use data collected from a small number of SPAD LiDAR frames (200–300). These image-reconstruction algorithms will be presented in detail later in the chapter, along with their computational demand analysis to demonstrate their real-time capability.

There are two main challenges in using SPAD array flash LiDAR system to collect above-water data of submerged targets for evaluating the effectiveness of these two algorithms. First, finding suitable sites for above-water imaging in natural waters is difficult. Such sites need structures to position the SPAD array flash LiDAR system above the water, and sufficient depth to separate the submerged target from the floor, ensuring the algorithm can distinguish between them based on depth information. Safety precautions further limit the number of viable locations, making outdoor experiments challenging and requiring extensive planning and approvals. If natural water sites are not used, simulating complex natural water conditions in a laboratory

tank becomes necessary, involving either transporting natural water or recreating it with realistic components including sediments and chlorophyll.

Second, creating a mechanical set-up that positions the SPAD array flash LiDAR system above water while keeping the underwater target within its field-of-view (FOV) is non-trivial. Fast water currents can displace the target, and turbid water can obscure its position.

These challenges highlight that effective reconstruction algorithms are not the only hurdle; reliable data collection of submerged targets using a SPAD array flash LiDAR system is equally crucial. Therefore, this chapter introduces a novel imaging set-up for above-water imaging of submerged targets.

In this chapter, three datasets are used to evaluate the performance of these two image-reconstruction algorithms. They are collected in two different laboratory environments and one outdoor natural water location, where different SPAD array flash LiDAR systems are used. The system specifications will be presented in detail later in this chapter.

The first dataset, from the Australian Institute of Marine Science (AIMS) indoor saltwater tank, is used to evaluate the algorithms' ability to reconstruct images of targets submerged in various turbidity conditions, created using sediments and chlorophyll. As an above-water imaging system would require significant engineering to create the imaging set-up, this first stage of imaging positions the SPAD array flash LiDAR system to horizontally image the target within the AIMS indoor saltwater tank. This dataset serves as an intermittent step where the feasibility of employing a SPAD array flash LiDAR system for imaging in natural waters is assessed. The reconstructed images from this dataset are used to gauge image quality in natural waters. In addition, target classification is evaluated to demonstrate image reconstruction quality.

To simulate a possible above-water imaging optical set-up, the SPAD array flash LiDAR system at the AIMS indoor saltwater tank is range-gated to only detect photons reflected within the water tank, which simulates the imaging of submerged targets without the interference of the water surface during above-water imaging. Some preliminary work related to this dataset has been published at a conference [101].

The collection of data for the second dataset establishes a methodology for imaging targets at various depths using an above-water SPAD array flash LiDAR system. The data collection location is at the DSTG indoor freshwater tank. A mechanical rig is constructed to position the LiDAR system above the water for imaging. The performance of the reconstruction algorithms in clear freshwater is evaluated, providing a baseline for the quality of reconstructed images in natural waters. Due to the laboratory environment mandating the use of multiple ceiling lights during experiments, the effect of the water surface on imaging is not investigated here because the indoor lighting condition is not reflective of outdoor lighting conditions at natural water locations.

The third dataset, from the Port Adelaide River, is used to assess the reconstruction algorithms for targets submerged in natural water environments. The same me-

chanical rig from the DSTG indoor freshwater tank is used for the imaging set-up, and the SPAD array flash LiDAR system is positioned such that it detects photons reflected from the water surface. The river, an estuary, consists of a mixture of saltwater and freshwater.

As discussed in Section 2.4 of Chapter 2, SPAD LiDAR image reconstruction of submerged targets is typically evaluated using freshwater tanks with Maalox acting as a turbidity agent. To the best of my knowledge, this is the first evaluation of image reconstruction techniques using real natural water environments and water tanks that simulate natural waters with real marine constituents.

Achieving successful image reconstruction in various natural water conditions is a result of iterative trial-and-error image processing. Evaluating the reconstructed images highlight the need for a comprehensive understanding of how water properties affect photon returns. Preliminary results on attenuation and scattering have been reported in [101] but are not reported in this thesis.

This chapter is structured in four parts. First, the methodology for collecting data of submerged targets by using a SPAD array flash LiDAR system is described. Section 5.1 explains the different SPAD array flash LiDAR systems used and their relevant features for image reconstruction. Section 5.2 details the imaging set-ups at the three different water locations.

The second part describes the image-reconstruction algorithms for the collected data, which are detailed in Section 5.3. This is followed by Section 5.3.1, which presents a computational demand analysis of these algorithms to demonstrate real-time capability.

The third part presents and discusses the reconstructed images from each water location. Section 5.4 covers the AIMS indoor saltwater tank results, followed by Section 5.5 with the DSTG indoor freshwater tank results and Section 5.6 with the Port Adelaide River results. In addition, target classification performance is presented in Section 5.4.3 for reconstructed images of data collected at the AIMS indoor saltwater tank.

The fourth part, Section 5.7, discusses the challenges of SPAD array flash LiDAR imaging of submerged targets in natural waters and proposes future research opportunities.

5.1 SPAD Array Flash LiDAR Systems

In this chapter, two different SPAD array flash LiDAR systems are employed for the imaging of underwater targets. These systems, identified as Systems B and C, are described in Table 3.1 in Section 3.4 of Chapter 3. System B is employed for imaging at the AIMS indoor saltwater tank, while System C is used for above-water imaging at the DSTG indoor freshwater tank and the Port Adelaide River.

Both systems share the same SPAD camera, but differ in their laser components. System B uses a Bright Solutions laser, which is larger and heavier, making it impractical for above-water imaging. In contrast, System C utilises a more compact Aréte

laser, which is better suited for above-water imaging tasks.

As mentioned in Section 3.2 of Chapter 3, direct Time-of-Flight (dToF) measurements are discussed in terms of clock cycles. As detailed in Section 3.4.1 and 3.4.2 of Chapter 3, the dToF measurements do not compensate for the timing delays of Systems B and C.

When no photons are registered by a SPAD detector during a LiDAR pulse cycle, the SPAD camera assigns its maximum possible dToF value to the pixel corresponding to the SPAD detector. For all the datasets discussed in this chapter, an upper-bound range gate is set during imaging. Therefore, the maximum possible value of a dToF measurement is the dToF value set for the upper-bound range gate. During post-processing, any dToF measurements with this maximum dToF value are discarded.

5.2 Data Collection Method

Three datasets are collected to evaluate the image-reconstruction algorithms, each from a different location: the AIMS indoor saltwater tank, the DSTG indoor freshwater tank, and the Port Adelaide River.

This section includes multiple subsections, beginning with Section 5.2.1, which describes the imaging targets used at each location. Following this, Section 5.2.2 provides details of the imaging set-up at the AIMS indoor saltwater tank.

Subsequent subsections detail the novel imaging set-up for performing above-water imaging of submerged targets. Section 5.2.3 describes the set-up used at the DSTG indoor freshwater tank for above-water imaging of submerged targets in clear freshwater. This includes details of the custom mechanical rig built for this novel imaging set-up. Section 5.2.4 covers the set-up used at the Port Adelaide River for above-water imaging of submerged targets in natural water. Finally, Section 5.2.5 explains the process of capturing dToF measurements of the target at various submerged depths at both the DSTG indoor freshwater tank and the Port Adelaide River, given their similar imaging set-ups.

5.2.1 Imaging Targets

Different imaging targets are used at each location. In the AIMS indoor saltwater tank, three thin plastic shapes, spray-painted in matte grey (as shown in Figure 5.1), serve as the targets. Each shape fits within a square with a side length of 40 mm.

For imaging in the DSTG indoor freshwater tank, a Secchi Disc (SD) with a diameter of 30 cm is used, as depicted in Figure 5.2. At the Port Adelaide River, a white disc of the same size as the SD is employed. The use of a white disc as a target in the river enhances the likelihood of detecting photons reflected from it. This is because the water is much more turbid compared with the clear fresh water in the DSTG indoor freshwater tank.

Table 5.1 below lists the different targets used in the different locations and the tables in this chapter that contain the reconstructed images of these targets.

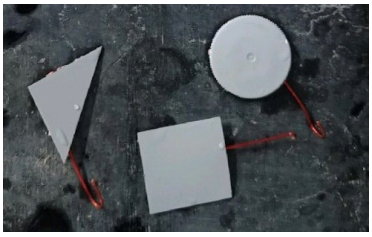


Figure 5.1: The thin plastic shapes spray painted in matte grey used for imaging. Note they have a wire hook attached so they can be attached to an optical mount for imaging in the AIMS indoor saltwater tank.

Figure 5.2: The black and white SD used for imaging.

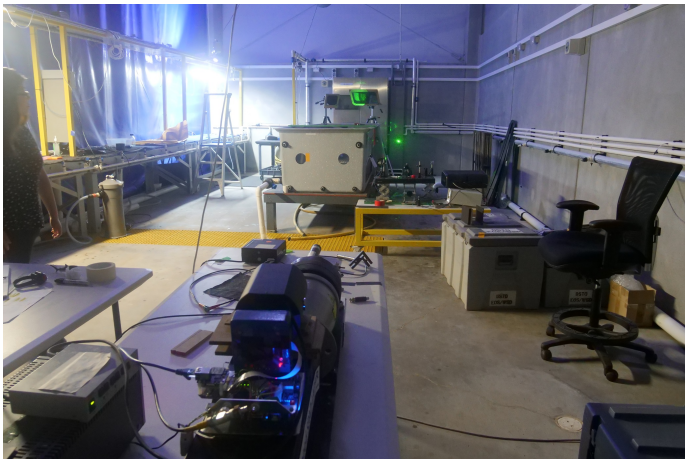
Table 5.1: Imaging targets used at each location.

Imaging Location	**Imaging Target**	**Table that contains reconstructed images**
AIMS indoor saltwater tank	Three grey geometric shapes	Table 5.2
DSTG indoor freshwater tank	30cm black and white SD	Table 5.3
Jetty at Port Adelaide River	A 30cm white disc, which is the same size as the SD used in DSTG indoor freshwater tank	Table 5.5

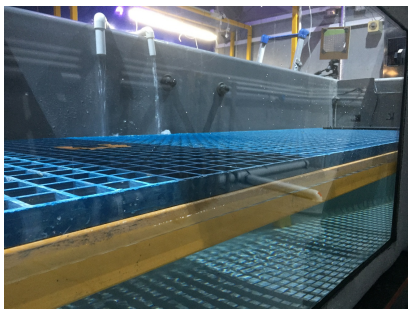
This space is intentionally left blank.

5.2.2 AIMS Indoor Saltwater Tank

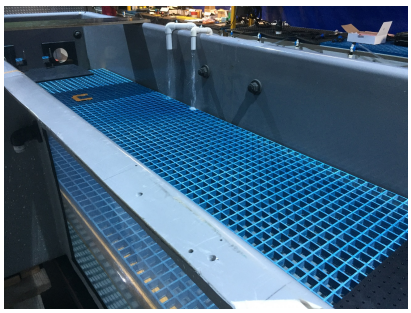
In the AIMS indoor saltwater tank, diverse ocean conditions are simulated by combining filtered saltwater with varying quantities of sediments and chlorophyll. SPAD LiDAR data are then collected for a submerged target in saltwater with different concentrations of sediments (1.30 to 8.62 mg/L), a singular concentration (2.46 µg/L) of chlorophyll, and a combined mixture of sediments (5.71 mg/L) and chlorophyll (2.44 µg/L). Only a single concentration of chlorophyll is utilised because of limited time. The selection of sediments and chlorophyll is based on their significance as two major optical constituents of the ocean [24]. According to [24], chlorophyll is prevalent in phytoplankton, while sediments consist of inorganic particles. These chosen constituents effectively emulate key aspects of oceanic optical properties, contributing to a more realistic simulation of diverse ocean conditions within the saltwater tank. The tables in Appendix B list the different amounts of these constituents that are used for imaging.



(a)



(b)



(c)

Figure 5.3: (a) AIMS indoor saltwater tank set-up in the laboratory. [11] (b) There are two large side windows on the right side of the tank used for the experiment. This is the side window that is near the front wall of the tank (with the portholes). (c) View of the other side window that is further from the front wall.

The dimensions of the saltwater tank are 4 m in length, 1 m in width, and 0.75 m in height. Figure 5.3(a) shows a photo of the laboratory where the saltwater tank is situated. The saltwater tank is in the middle of the laboratory. Two small circular acrylic portholes, are custom-cut into the tank and are visible in Figure 5.3(a). These portholes are strategically positioned to facilitate extended distances for SPAD array flash LiDAR imaging into the water tank. Designed to be 0.5 m apart and with a diameter of 0.12 m each, these portholes are integral to the experimental set-up. There is also an additional wall in the AIMS indoor saltwater tank, which is referred to as the back wall when evaluating reconstructed images. In addition, the tank is equipped with two large acrylic windows on one side, as illustrated in Figure 5.3(b) and 5.3(c) [101], but they are not used as part of the imaging set-up.

The next sub-section discusses how the saltwater is sourced. This is followed by Section 5.2.2.2 which details the sediments and chlorophyll added into the water tank for imaging. The remainder of this section discusses the imaging set-up and method.

5.2.2.1 Obtaining Saltwater from the Ocean

The AIMS indoor saltwater tank is situated within the AIMS SeaSim facility. It serves as a controlled environment for saltwater experiments, specifically designed for the study of coral reefs[1]. Filtered saltwater utilised in the laboratory experiments is sourced from the ocean. A detailed description of the SeaSim facility capabilities is provided in [6].

Before entering the saltwater tank, the saltwater passes through a fabric filter with a diameter of 1 µm to remove any unexpected debris. Once inside the tank, a pump is employed to ensure uniform mixing of the saltwater. Due to the absence of chlorine, the saltwater can only remain in the tank for a maximum of 12 hours after substances are added, as bacteria may begin to proliferate. Consequently, the tank is emptied and refilled within approximately 2 hours, including the time required for tank drainage, wall cleaning, and saltwater replenishment.

Thorough cleaning of the tank walls is imperative to prevent residuals that could adversely affect the saltwater quality during imaging processes. Throughout experiments, the saltwater temperature in the tank is maintained at 25 $^{\circ}C$ to ensure consistency and relevance to local environmental conditions.

5.2.2.2 Sediment and Chlorophyll Used in the AIMS Indoor Saltwater Tank

Sediments with diameters of 38 µm or less are used to increase the sediment concentration of the saltwater in a controlled manner. The sediments have been sourced directly from the ocean floor.

A certain sediment concentration is achieved in the AIMS indoor saltwater tank by adding finite amounts of sediments. The sediments are provided by the AIMS SeaSim staff. To ensure the sediments are evenly distributed throughout the water tank, the required amount of additional sediment is first mixed in a bottle with

[1]Visit this site more information: https://www.aims.gov.au/about/facilities/national-sea-simulator

filtered saltwater to create a solution. Examples of the bottles of pre-mixed sediment solution are shown in Figure 5.4. Then, this solution is evenly poured throughout the saltwater tank, followed by stirring of the tank's saltwater using the motor attached to the saltwater tank.

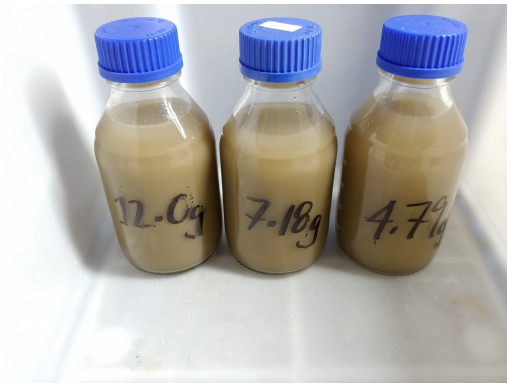


Figure 5.4: Sediment solutions used to increase sediment concentrations.

For chlorophyll, a mixture of equal parts (approx. 16.66%) of six different microalgae are used to increase the chlorophyll concentrations of the tank water. Microalgae are used since they contain chlorophyll [67]. The microalgae used are *Chaetoceros* sp. (CS-256), *Nanochloropsis oceania* (CS-702), *Isochrysis* sp. (CS-177), *Chaetoceros muelleri* (CS-176), *Dunaliella* sp. (CS-353) and *Rhodomonas salina* (CS-24/01). These microalgae are between 3–10 µm in diameter. These microalgae are lab-grown as coral food. The microalgae are maintained by the AIMS SeaSim staff.

For all the sediment and chlorophyll concentration values reported in this thesis, a baseline correction is applied to them. These corrected concentration values account for any existing sediments and chlorophyll in the saltwater which is sourced from the ocean.

5.2.2.3 Imaging Set-up

System B, with its specifications listed in Table 3.1 in Section 3.4 within Chapter 3, is utilised for imaging at the AIMS indoor saltwater tank. The SPAD array flash LiDAR system is oriented horizontally relative to the imaging target, as opposed to placing it vertically above the imaging target and water surface. Figure 5.5 shows a schematic of the imaging set-up. The acrylic portholes are 0.5 m apart and each has a diameter of 0.12 m.

The employed SPAD array flash LiDAR system has a minimum imaging distance of 6 m, so the system is positioned such that the system begins detecting photons reflected from inside the water tank and the system does not detect any photons reflected from the acrylic windows or the front wall of the AIMS indoor saltwater tank. This specific imaging set-up simulates photon returns collected from an above-water SPAD array flash LiDAR system range gated to exclude photon returns from

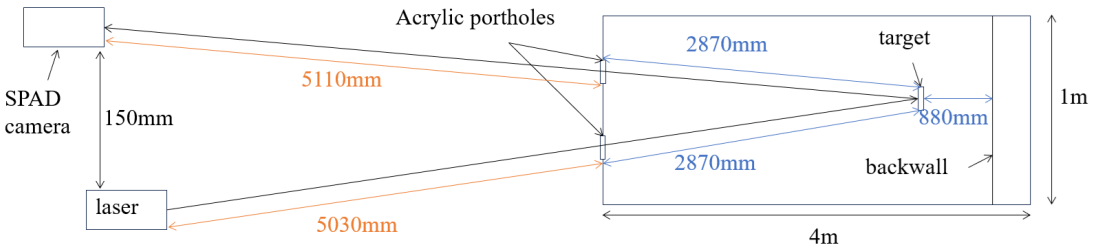


Figure 5.5: Schematic of AIMS indoor saltwater tank imaging set-up.

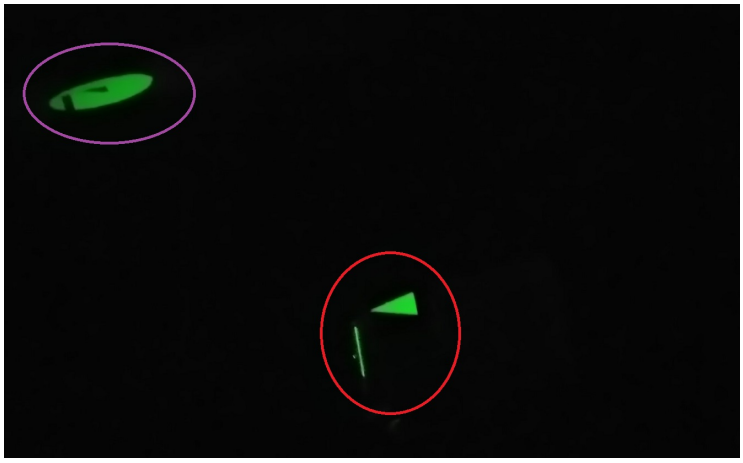


Figure 5.6: The grey right-angled triangle in the saltwater tank, held up by an optical mount and piece of wire, (highlighted by the red circle in the photo) is illuminated by a laser beam during calibration. Its shadow is on the back of the water tank, highlighted by the purple circle in the photo.

reflections and scattering at the water surface. Additionally, a maximum range gate of 100 clock cycles is set.

The imaging targets in the AIMS indoor saltwater tank, elaborated upon in Section 5.2.1, consist of three grey shapes. These shapes are affixed to the top of an optical mount with a height of 122 mm, positioned about 2.9 m away from the portholes. As seen in Figure 5.1 in Section 5.2.1, a small red wire hook is attached to each of the shapes. This wire is used to secure the shape onto the optical mount, and enable a quick change of shapes on the optical mount during imaging.

Figure 5.6 illustrates the laser illuminating a triangle submerged higher in the water, positioned by the optical mount on the left. The shadow of the triangle is observable on the back wall of the tank, positioned at the upper left of the photograph.

A dark environment is utilised to conduct the experiments in order to reduce the amount of ambient noise detected by the SPAD array flash LiDAR system.

5.2.2.4 Imaging Method

Imaging is conducted for different sediment concentrations first, with all imaging for each condition conducted within half an hour. This approach ensures that the sediment does not settle significantly, which could alter the saltwater conditions between shapes for the same sediment concentration. The first sediment condition has no added sediment, which corresponds to a corrected sediment concentration of 1.30 mg/L. Although no sediment is added, the filtered saltwater itself contains some residual sediment.

Subsequent sediment conditions involves increasing sediment concentrations by adding additional sediments to the saltwater tank for each new concentration. After collecting all data for these sediment concentrations, the tank is emptied, thoroughly cleaned, and refilled with new saltwater. Algae are then added to achieve a chlorophyll concentration of 2.46 µg/L. Following this, sediments are added to the tank to create a combined mixture with sediment concentration of 5.71 mg/L and chlorophyll concentration of 2.44 µg/L. The slight decrease in chlorophyll concentration from 2.46 µg/L to 2.44 µg/L is due to the fact that sediments are added the day after the chlorophyll is introduced, potentially causing some algae to die overnight and thus resulting in a lower chlorophyll measurement when the concentration is re-measured by the specialised fluorescence meter.

5.2.3 DSTG Indoor Freshwater Tank Imaging Set-up

A lab-controlled freshwater elliptical tank at DSTG is used to test the set-up of a SPAD array flash LiDAR system for above-water imaging. The tank, which measures 7 m in depth and 6 m in width, is continuously filtered and treated with chlorine to ensure a contamination-free environment. This ensures any of the observed effects in the data are not due to turbid conditions, which are already tested at the AIMS indoor saltwater tank described in Section 5.2.2.

The SPAD array flash LiDAR system is mounted on a bridge above the side of the water tank to perform above-water imaging of a submerged target. The system is secured using a custom mechanical rig designed to position it precisely over the water surface.

A schematic of the custom mechanical rig is shown in Figure 5.7. The rig is anchored to the handrails of the bridge over the water tank. The SPAD array flash LiDAR system is mounted onto a steel plate, with the laser positioned on the top side and the SPAD camera on the opposite side. This steel plate is then attached to an L-frame, which is robust enough to support the unit over the water. Additionally, the steel plate is connected to the L-frame with a hinge, allowing the imaging system to rotate and capture images at various angles. The red angle depicted in Figure 5.7 illustrates the camera angles reported in this chapter in relation to the position of the steel plate.

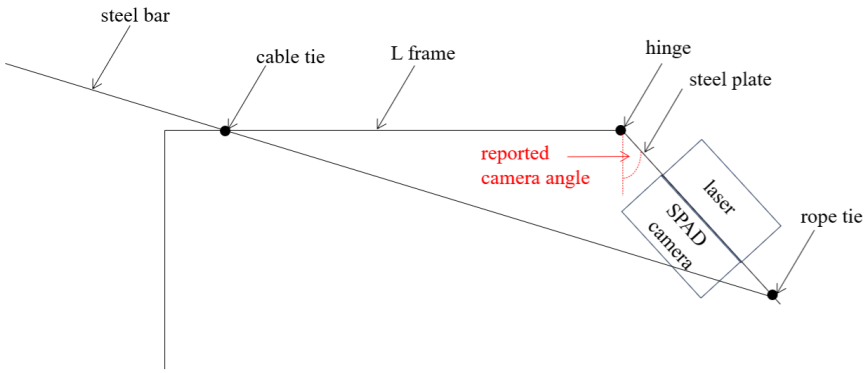


Figure 5.7: Schematic of custom rig from the side view.

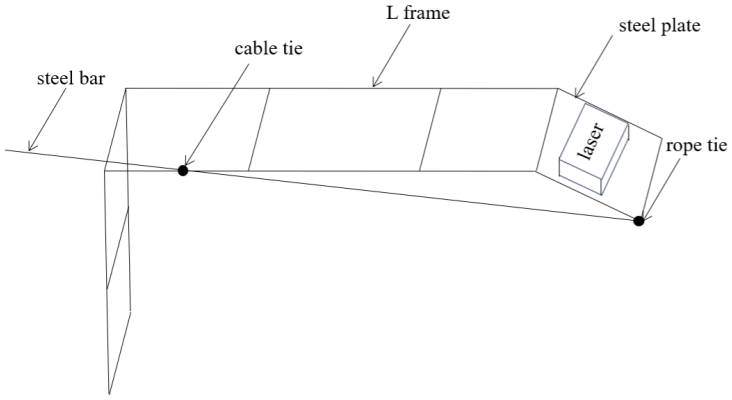


Figure 5.8: Schematic of custom rig from an angled view. Here it shows the steel bar is attached to only one side of the steel plate and L frame, so it is not blocking the FOV of the SPAD array flash LiDAR system. The SPAD camera is not shown here as it is underneath the steel plate in this angled view.

The schematic in Figure 5.8 illustrates the attachment of the steel bar to the rig. A narrow steel bar is affixed to the tip of the steel plate and the L-frame, enabling the steel plate to be pushed outward and the SPAD array flash LiDAR system to be angled. As shown in the figure, the steel bar is attached to only one side of the steel plate and L-frame, ensuring it does not obstruct the SPAD array flash LiDAR system's FOV.

The steel bar is secured to the steel plate by cutting a small hole at the corner tip of the plate and drilling holes in the bar. A rope is threaded through these holes, linking the tip of the plate to the tip of the steel bar. Additionally, the bar is fastened to the L-frame with a loose cable tie, allowing it to move the steel plate forward and backward without causing lateral movement.

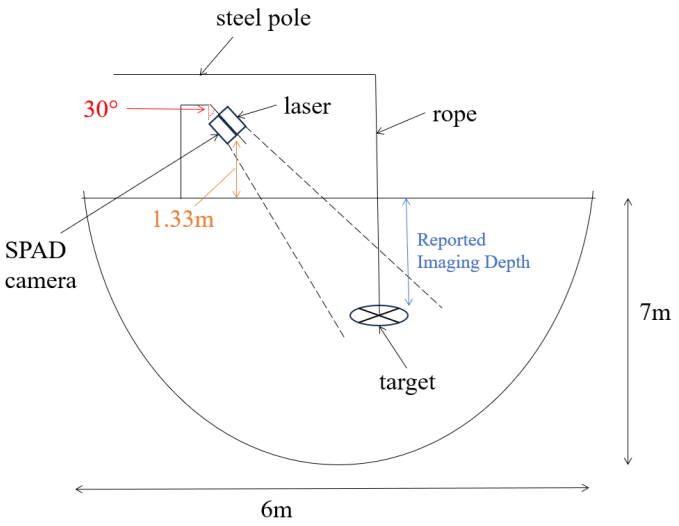


Figure 5.9: Schematic of DSTG indoor freshwater tank imaging set-up.

A schematic of the imaging set-up at the DSTG indoor freshwater tank is shown in Figure 5.9. The SPAD camera is positioned 1.33 m above the water surface and is angled at 30° from the vertical, focusing toward the centre of the tank. This angle, determined empirically, minimises the detection of laser reflections off the water surface and optimises image quality.

The imaging target, an SD, is suspended in the water using a rope and steel pole mechanism. The steel pole extends the target away from the imaging system, while the rope allows the target to be lowered into the water (as depicted by the vertical line attached to the target in Figure 5.9). The rope slides along the pole, enabling adjustment of the target's depth. A measuring tape attached to the rope tracks how much rope has slid along the pole, allowing precise measurement of the imaging depth.

As the multiple ceiling lights in the laboratory are not representative of outdoor lighting conditions at natural water locations, this set-up intentionally excludes the detection of photons reflected by the water surface. This is possible because the system's minimum operating distance in this configuration[2] is greater than its distance to the water surface.

Further details on the imaging method used at this site are provided in Section 5.2.5.

5.2.4 Port Adelaide River Imaging Set-up

Imaging is conducted at the jetty on the Port Adelaide River, an estuary where freshwater from the river merges with saltwater from the ocean. The water in this area is highly turbid, with the SD depth measured at approximately 3.4 m, at the time

[2]Minimum operating distance is 2 m, as discussed in Section 3.4 of Chapter 3.

Figure 5.10: The black and white SD is being lowered into the water at Port Adelaide River to measure the water turbidity level in terms of SD depth.

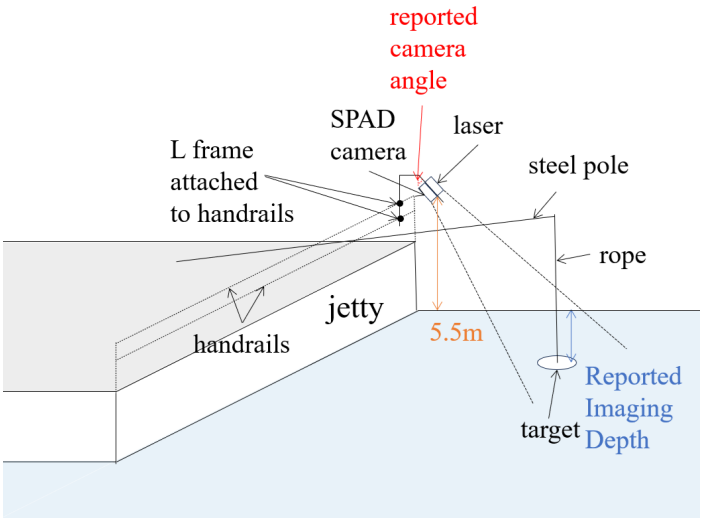


Figure 5.11: Schematic of Port Adelaide River imaging set-up.

of data collection. Details on measuring the SD depth and the rationale for using it instead of the number of attenuation lengths (ALs) to assess natural water quality are discussed in Section 3.6.2 of Chapter 3. Figure 5.10 shows a photo of the SD lowered into the water at the Port Adelaide River.

The weather condition is sunny at the time of data collection. However, the experimental set-up is located in a shaded area of the jetty, which provided shade for the imaging area in the river. Figure 5.11 shows a schematic of the imaging set-up on the jetty.

Similar to the set-up at the DSTG indoor freshwater tank, the same mechanical rig described in Section 5.2.3 for the SPAD LiDAR system is secured to the handrails on the side of the jetty. The imaging target, a white disc, is suspended in the water using the same rope and steel pole mechanism used for suspending the SD at the DSTG indoor freshwater tank, as described in Section 5.2.3. The main difference is that the steel pole for suspending the imaging target is positioned lower and to the right of the SPAD array flash LiDAR system, as shown in Figure 5.11.

To prevent the steel pole from appearing in the SPAD camera's FOV, as indicated by the two diagonal dotted lines in Figure 5.11, the pole is placed away from the SPAD array flash LiDAR system and angled diagonally towards the camera, ensuring that the target remains within the FOV for imaging. The steel pole is positioned in such a way that the rope appears from the bottom right corner of the SPAD camera's FOV.

When imaging at an initial target depth of 0.5 m, the mechanical rig positions the SPAD array flash LiDAR system at an angle of 56° from the vertical downward direction, labelled as the "Reported Camera Angle" in Figure 5.11. For target depths of 1 m and deeper, the angle is adjusted to 63° to keep the target within the SPAD camera's FOV.

Further details of the imaging method used at this site are provided in Section 5.2.5.

5.2.5 Imaging Method at DSTG Indoor Freshwater Tank and Port Adelaide River

Above-water dToF measurements of submerged targets are captured using a novel imaging set-up. The target is positioned directly below the SPAD array flash LiDAR system. However, direct vertical alignment can cause strong reflections from the water surface, which interfere with imaging even with range gating. To address this issue, the target is intentionally placed at a horizontal offset distance from the direct vertical line beneath the SPAD array flash LiDAR system, allowing for imaging at an angle that minimises surface reflections.

The target is submerged at various predetermined depths, pausing at each depth for imaging before being raised and taken out of the water. The target, tethered by a rope (as described in Section 5.2.3), is positioned at specific depths indicated by a measuring tape. During each descent, the SPAD array flash LiDAR system initiates imaging, and a DSTG-developed Graphic User Interface (GUI) is used to provide real-time feedback by displaying preliminary reconstructed images using basic arithmetic methods. These methods process each pixel's dToF measurements over a rolling batch of 32 frames to calculate the mean, median, and in addition, the dToF value of the histogram peak of the dToF measurements. If the target is visible in these preliminary reconstructed images, the GUI helps ensure it is centred within the FOV. If the target is not visible, a histogram of the dToF measurements for all pixels from the current batch of 32 frames is generated. From this histogram, an absence of a histogram peak around the target's expected dToF value indicates that the previously imaged depth is the lowest point at which the target can be captured.

When evaluating the performances of reconstruction algorithms, each reconstructed image is created by processing a data file that corresponds to a continuous collection of frames taken with the target at a specific depth. A new data file is created for each distinct depth. Multiple data files may be collected for a single depth. When referring to datasets in this chapter, a dataset consists of a set of data files collected during a single instance of lowering the target into the water, where the target

is not retrieved from water during the entire process. Data files obtained after retrieving and re-submerging the target are considered part of a different dataset. This method ensures consistency in water conditions within the same dataset, reducing the impact of spatial and temporal variations in water properties due to factors such as water currents.

5.2.5.1 Range Gating on the SPAD Array Flash LiDAR System

Different range gates are used for the SPAD array flash LiDAR system for imaging, depending on the imaging location. For the DSTG indoor freshwater tank, a maximum range gate of 44 clock cycles is set, while for the Port Adelaide River, the maximum range gate is set to 45 clock cycles.

These upper-bound range gates are set at a higher dToF value than the expected dToF value of the floor of each location. In the DSTG indoor freshwater tank, the lowest point of the tank floor has an expected dToF value of 17 clock cycles. In the Port Adelaide River, the floor is 10 m below the water surface, equating to an expected dToF value of 31 clock cycles.

For the Port Adelaide River dataset, an additional minimum range gate is set to 4 clock cycles. By adding 4 clock cycles to correct for timing delay (discussed in Section 3.4.2 of Chapter 3), this minimum range gate's value corresponds to a lower bound imaging distance of 4 m from the SPAD array flash LiDAR system, which is still above the water surface. This setting was empirically determined to improve image quality. This is likely because of a reduction of detected photons due to ambient noise, as the weather was sunny at the time of the data collection.

Conversely, for the DSTG indoor freshwater tank dataset, it was empirically determined at the time of data collection that no minimum range gate is necessary. This is because the SPAD LiDAR system's minimum operating distance of 2m, as discussed in Section 3.4 of Chapter 3, which excludes photon returns from the water surface.

5.3 Image-Reconstruction Algorithms

In this chapter, two basic image-reconstruction algorithms are considered: the detection threshold and median algorithm, and the histogram averaging algorithm. These algorithms are two of the most widely used image reconstruction algorithms in the field of SPAD LiDAR research. Both algorithms operate on a pixel-wise basis, processing a batch of frames each time to produce a reconstructed image. The number of frames used by the algorithm to reconstruct one image varies by dataset, and this information will be provided alongside the reconstructed images presented in this chapter.

The detection threshold and median algorithm first examines each pixel's total number of photon detections. If a pixel's total number of photon detections over the batch of frames is less than or equal to 0.1 (i.e. 10%) of the total number of frames in the batch, the pixel's reconstructed dToF value is considered undefined. If the photon

detections exceed this threshold, the pixel's reconstructed dToF value is determined as the median of the pixel's dToF measurements over the batch of frames.

The histogram averaging algorithm, proposed by the authors in [117], involves creating a histogram for each pixel's dToF measurements within the batch of frames. For each pixel, the algorithm identifies the histogram bin with the highest count, and if this bin's photon count value surpasses a predefined threshold, the dToF value corresponding to that bin is used as the pixel's reconstructed dToF value. Otherwise, the pixel's reconstructed dToF value is considered undefined. The threshold for this algorithm is consistent with the one used in the detection threshold and median algorithm, set at 0.1 (i.e. 10%) of the total number of frames in the batch.

In the reconstructed images presented in this chapter, the colour representing the smallest dToF value in the colour bar is also used to indicate an undefined value for a pixel. In all of the colour bars shown in this chapter, dark blue is used to represent both the image's smallest dToF value and the undefined values. This is an artefact of using Matlab's `imagesc` function to display the array of dToF values as an image. This colour-coding does not affect the image quality of the target in the reconstructed images. This is because the lowest value in all the colour bars corresponds to the closest distance from which a detected photon return is reflected, which is never where the target is positioned.

The next sub-section discusses the real-time capability of the two image-reconstruction algorithms presented here. A computational demand analysis is provided for each of these two image-reconstruction algorithms.

5.3.1 Real-Time Capability

The image-reconstruction algorithms investigated in this chapter are real-time capable because they both consists of a small number of steps and these steps employ simple computational operations. To illustrate this real-time capability, a summary of the detection threshold and median algorithm is provided in Figure 5.12 and a summary of the histogram averaging algorithm is provided in Figure 5.13.

Figure 5.12 and 5.13 demonstrate that both algorithms have a small amount of steps for processing one pixels' set of dToF measurements. From Figure 5.12, the most computationally intensive task for the detection threshold and median algorithm is calculating the median value, which uses a sorting operation. For the reconstructed images reported in this chapter, the algorithm is applied to process a maximum of 300 frames. Therefore, this means that the sorting operation will sort a maximum of 300 dToF measurements, which is a small amount and makes it possible for the median value to be computed in real time. As the most computationally intensive task is real-time capable, this implies the detection threshold and median algorithm is real-time capable for processing one pixel's set of dToF measurements.

This space is intentionally left blank.

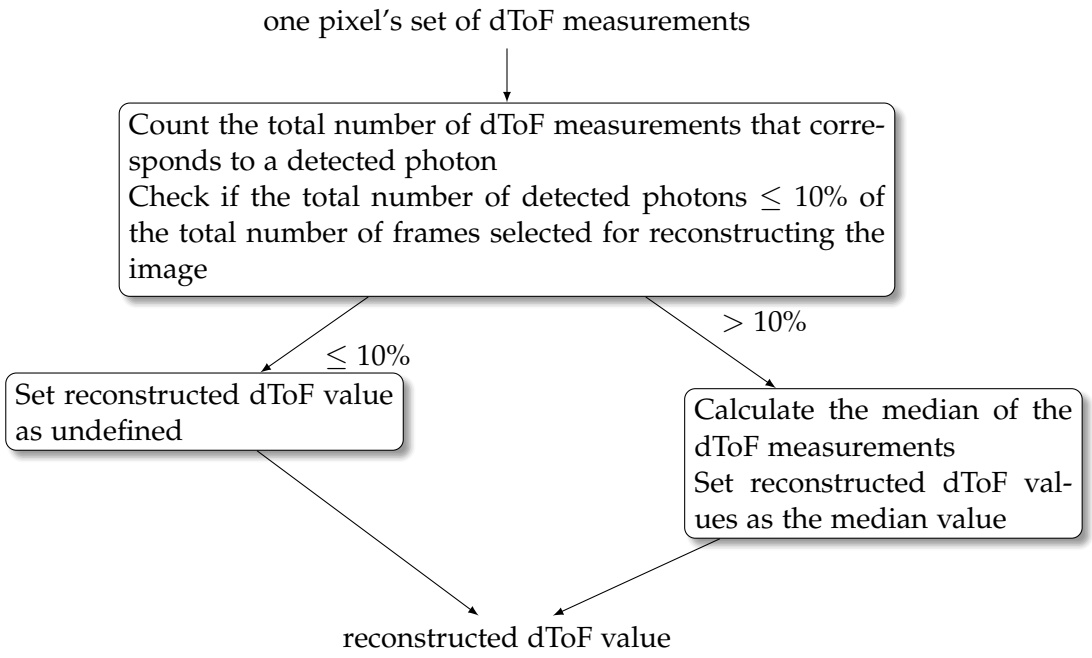


Figure 5.12: Summary of the detection threshold and median algorithm

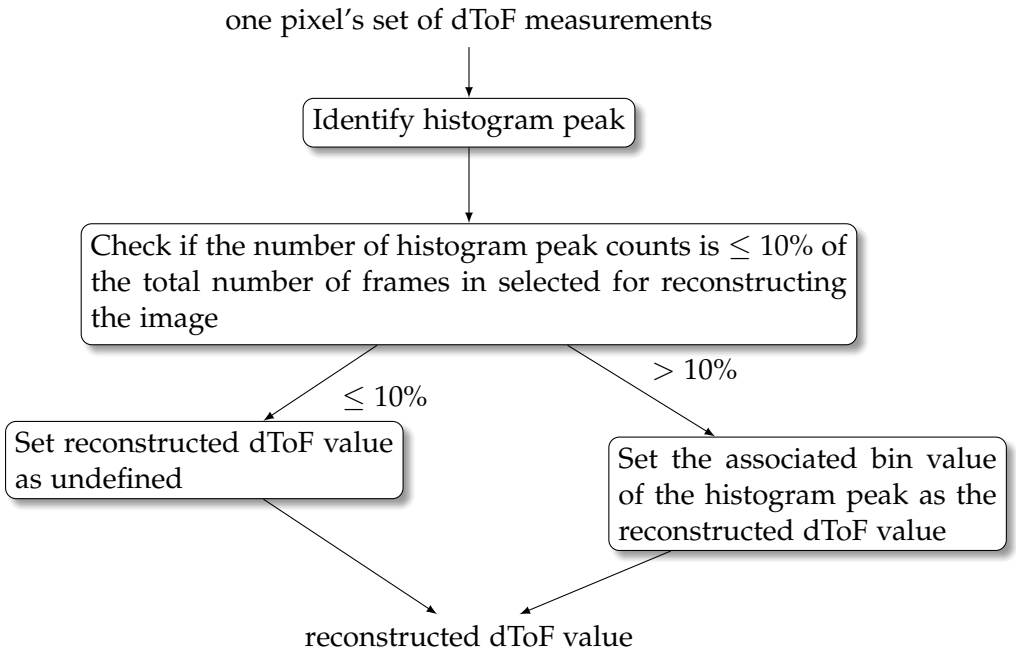


Figure 5.13: Summary of the histogram averaging algorithm

For the histogram averaging algorithm, the most computationally intensive task is identifying the histogram peak, while the other task is computationally trivial. The task of identifying the histogram peak is computationally expensive because it uses

sorting operation twice. Once for sorting the dToF measurements into a histogram and another time for sorting each of the histogram count values in order to determine the histogram peak. However, both instantiation of the sorting operation process a small amount of data. This is because the histogram averaging algorithm is evaluated over the SPAD LiDAR datasets collected at the DSTG indoor freshwater tank and Port Adelaide River. For the reconstructed images presented in this chapter, the algorithm only processes a batch of 200 or 300 frames at each execution, depending on the dataset.

For creating a histogram, the sorting operation only processes up to 300 dToF measurements. For identifying the histogram peak, the sorting operation processes the histogram count values over all possible bin values. From the range gate values presented in Section 5.2.5.1, it can be deduced that there will only be a maximum of 40 different possible dToF measurements in this chapter's reported SPAD LiDAR datasets. This means there is only a maximum 40 possible bin values. Therefore, despite the sorting operation being high intensity, the small number of possible bin values means the task of identifying the histogram peak for one pixel is real-time capable. Therefore, this demonstrates the histogram averaging algorithm is real-time capable for processing one pixel's set of dToF measurements.

Both the detection threshold and median algorithm and the histogram averaging algorithm repeat the same steps in Figure 5.12 and 5.13 over all pixels of the SPAD array. Instead of processing each pixels' dToF measurements in series, a graphical processing unit (GPU) can be utilised to speed up the computation. Each parallel thread can be used to process each pixels' set of dToF measurements. This is possible since it is common for a GPU to contain at least 1024 threads. Therefore, this approach shows that both of these algorithms are real-time capable in reconstructing an entire image.

The following sections present the reconstructed images for the datasets collected at the three locations described in Section 5.2. Section 5.4 reports the reconstructed images and target classification performance from data collected at the AIMS indoor saltwater tank. Section 5.5 provides reconstructed images from data collected at the DSTG indoor freshwater tank, while Section 5.6 presents the reconstructed images from data collected at the Port Adelaide River.

5.4 Image Reconstruction of Data Collected at AIMS Indoor Saltwater Tank

In images reconstructed from data collected in the AIMS indoor saltwater tank, the detection threshold and median algorithm explained in Section 5.3 is used. For this specific dataset, the algorithm employs a batch of 200 frames to reconstruct each image. Additionally, the dToF measurements are further range-gated before applying the detection threshold and median algorithm. This post-processing range gate is different to the range gate applied during SPAD array flash LiDAR imaging. It is

set between 1 and 10 clock cycles[3] because the back wall of the water tank has an expected dToF value of 9 clock cycles. Hence, no detected photons should have a dToF measurement exceeding 10 clock cycles.

The expected dToF values are derived from theoretical calculations of a photon's dToF from the SPAD LiDAR system to the target and back, considering the refractive index of water (1.33) and the speed of light in water ($2.26 \times 10^8 m/s$). The theoretical calculation assumes the SPAD LiDAR laser pulse propagates perpendicularly to the water tank's acrylic window, thus eliminating the need to factor in the angle of laser pulse propagation in air and water. Also, System B's timing delay of 40 ns is subtracted from the theoretical dToF value as part of calculating the expected dToF value. This is to ensure the expected dToF value is consistent with System B's dToF measurements which does not compensate for System B's timing delay, as discussed in Section 3.4.1 of Chapter 3. When converting the expected dToF value from seconds to clock cycles, the clock cycle value is rounded up to the nearest integer, ensuring consistency with the dToF measurements, which are always integers.

Table 5.2 below contains the reconstructed images for various concentrations of sediments and/or chlorophyll. Reconstructed images of a circle, triangle, and square are shown for each saltwater condition, except for the condition with 8.62 mg/L of sediment concentration. In this case, only the reconstructed images related to the circle and triangle are shown due to corruption of the data related to the square. The equivalent number of ALs for each saltwater condition is also provided in this table as a reference for water turbidity level.

Other saltwater conditions that also only have between 1.30 to 5.62 mg/L of sediments are not presented here because their image quality is similar to the presented reconstructed images in Table 5.2 within this sediment range. Appendix B contains these additional reconstructed images as well as the ones presented here[4]. All the reconstructed images from this dataset are depicted using the same colour bar for consistent comparison.

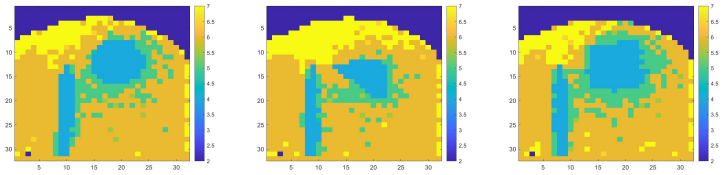
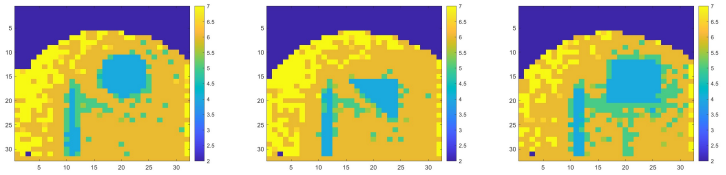
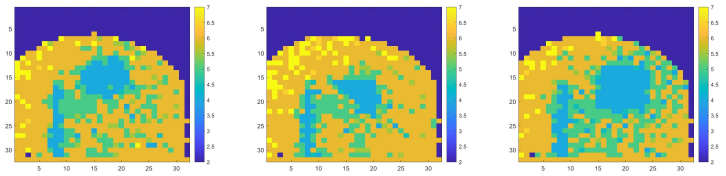
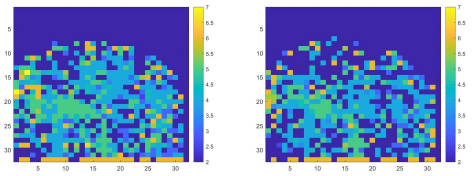
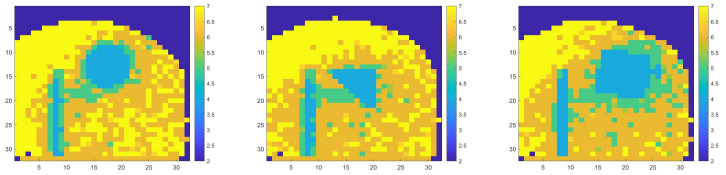
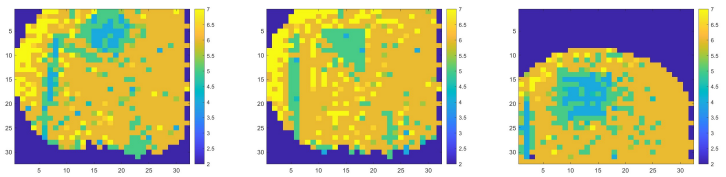
The reconstructed images are presented in a larger size in Appendix B for more detailed inspection. The reconstructed images presented here are made smaller to enable easier comparison of all images.

This space is intentionally left blank.

[3]This is inclusive of the dToF measurements with 1 and 10 clock cycles

[4]The sediment concentration values reported in [101] are not the corrected sediment concentration values. The values given in this thesis are the corrected values.

Table 5.2: Reconstructed images at different saltwater conditions by the detection threshold and median algorithm.

Amount of Substances Added	**No. of ALs**	**Reconstructed Images**
1.30 mg/L sediments	1.32	
3.70 mg/L sediments	2.35	
5.62 mg/L sediments	3.29	
8.62 mg/L sediments	4.28	
2.46 µg/L chlorophyll	1.91	
2.44 µg/L chlorophyll & 5.71 mg/L sediments	3.43	

In all the images, a block of dark blue pixels appears on the outer edges. These pixels form a curved boundary around the target, where the target is imaged from within the acrylic porthole. They represent the AIMS indoor saltwater tank's front wall around the acrylic porthole, where its location is shorter than the SPAD array flash LiDAR's minimum imaging distance (i.e. System B in Table 3.1 in Section 3.4 of Chapter 3). This causes no detection of photons reflected by the front wall and results in these pixels' reconstructed dToF values being undefined. As discussed in Section 5.3, the undefined values are shown as the same colour as the lowest reconstructed dToF value in the image's colour bar.

For all saltwater conditions except at 8.62 mg/L of sediments, the differently shaped targets can be identified visually. Section 5.4.1 discusses the visual quality of the reconstructed image in further detail. In addition, these images contain a straight thick vertical line, which appears next to the target. This represents the optical mount rod used to hold the target. Some reconstructed images also display a small thin horizontal line linking the target to the thick line, indicating the small wire piece used to attach the target to the optical mount. The background pixels in all the reconstructed images are from scattering, since the expected dToF value of the saltwater tank's back wall is 9 clock cycles.

In images relating to 8.62 mg/L of sediments, the background pixels are from detected photons reflected from water scattering, as their dToF measurements do not match the expected dToF value of the AIMS indoor saltwater tank's inside back wall.

The next subsection (Section 5.4.1) discusses the reconstruction performance of the detection threshold and median algorithm in more detail. It is followed by Section 5.4.2, which quantifies reconstruction performance in terms of different metrics of localisation accuracy. Then, Section 5.4.3 presents an evaluation of the accuracy in classifying different shapes in the reconstructed images under various saltwater conditions, focusing on conditions with only sediment concentrations.

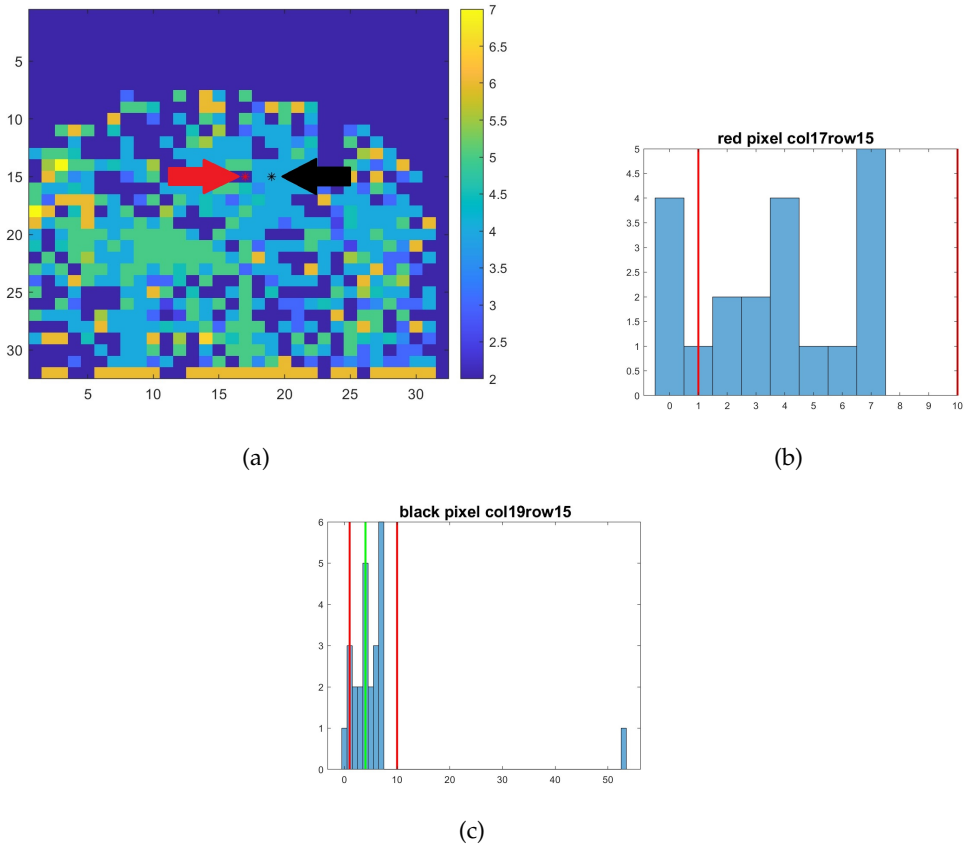
5.4.1 Reconstruction Performance via Visual Inspection

In all saltwater conditions, except where there is 8.62 mg/L of sediments, the reconstructed images of each shape are sufficiently discernible by visual inspection. This means target detection is possible by visually observing these reconstructed images. The reconstructed images at 5.62 mg/L of sediments have a higher noise level than the images at lower sediment concentration levels. This can be attributed to the increased amount of sediments causing an increased amount of scattering. Section 5.4.3 discusses in detail the classification of shapes from reconstructed images under saltwater conditions with increasing sediment concentrations.

In the case of 8.62 mg/L of sediments, the detection threshold and median algorithm fails to produce a discernible representation of the target in the image. This effect can be explained by examining the dToF measurements collected for target pixels, which are pixels at the expected spatial position of the target in the image. The

spatial position of the target in the image is inferred from the reconstructed images of the saltwater condition with 5.62 mg/L sediments. This is possible because at the time of the data collection, the imaging of the saltwater condition with 8.62 mg/L of sediments was conducted immediately after the imaging of the condition with 5.62 mg/L sediments. The same experimental set-up is used between imaging these two sets of data, except for the addition of sediments into the saltwater to increase its sediment concentration.

The dToF measurements of two different pixels at the target's position are exam-



(a) (b)

(c)

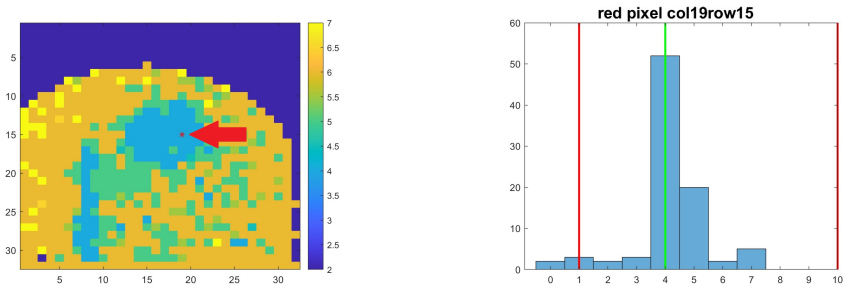
Figure 5.14: Histogram of two different target pixels (15,17) and (15,19), which are in the spatial position of the target in the reconstructed image of a circle at 8.62 mg/L of sediments. Figure (a) shows the reconstructed image with red (15,17) and black (15,19) asterisks marking which pixels are examined. The corresponding coloured arrows are also pointing at them. Figure (b) shows the histogram of dToF measurements for the red-marked pixel (15,17). Figure (c) shows the histogram of dToF measurements for the black-marked pixel (15,19). The title of each histogram states which colour-marked pixel is examined and the coordinates of the pixel. If there is a green line in the histogram, it represents the reconstructed dToF value for the pixel. The red vertical lines in the histogram depict the post-processing range gate used before reconstruction is applied.

ined via histograms and are shown in Figure 5.14. These two pixels are marked with red and black asterisks (indicated by arrows) in Figure 5.14(a). These two pixels are used to illustrate the general characteristics of the histograms of dToF measurements from the target pixels, which display similar characteristics to these two pixels.

The total number of photon detections within the post-processing range gate (indicated by the red vertical lines in the figures) for each of these two histograms is 16 for the red-marked pixel in Figure 5.14(b) and 23 for the black-marked pixel in Figure 5.14(c). These numbers are very close to 20, which represents 10% of the total number of frames used for reconstruction (200 frames). If the total number of detections is below 20, the detection threshold and median algorithm registers the reconstructed dToF value for that pixel as undefined. This is demonstrated by the red-marked pixel in Figure 5.14(a), where the total photon detection count is 16, resulting in an undefined reconstructed dToF value for that pixel. This indicates that using 10% of frames for the threshold value of is not suitable for determining whether a pixel corresponds to the position of the target.

Another characteristic of the data for the saltwater condition with 8.62 mg/L sediments is the distribution of dToF measurements for pixels related to the target, which exhibits a less consistent trendline. Specifically, these histograms do not display a clear peak. This contrasts with the distribution of dToF measurements collected in the saltwater condition with the next highest sediment concentration of 5.62 mg/L. An example of a histogram of dToF measurements from a target pixel in the reconstructed image for 5.62 mg/L is shown in Figure 5.15; the pixel is marked by a red asterisk in Figure 5.15(a).

In Figure 5.15(b), the histogram of dToF measurements for a target pixel in the reconstructed image shows a distinct histogram peak, indicating a significant number



(a) (b)

Figure 5.15: Histogram of the target pixel (15,19) in the reconstructed image of a circle at 5.62 mg/L of sediments. Figure (a) shows the pixel marked by a red asterisk and red arrow on the reconstructed image. Figure (b) shows the histogram of dToF measurements for this pixel. The title of the histogram states the coordinates of the pixel. The green line is the reconstructed dToF value of the pixel in the reconstructed image. The red vertical lines in the histogram depict the post-processing range gate used before reconstruction is applied.

of detected photons reflected from the target's surface. This clear histogram peak is absent in the histograms of dToF measurements relating to the target pixels in the 8.62 mg/L condition (see Figure 5.14). The lack of a clear peak results in inconsistent median values for each pixel, leading to a noisy reconstructed image.

Several other observations can be made from the reconstructed images in Table 5.2. For the reconstructed images of the target in saltwater containing only sediments and up to 5.62 mg/L of it, the images become noisier with increasing sediment concentration. The reconstructed image with 2.46 µg/L of chlorophyll shares similar image quality to images reconstructed in sediment-only conditions, except the background appears slightly different, with a different distribution of yellow and orange pixels. This difference can be attributed to chlorophyll having different scattering properties compared with sediments.

For the saltwater condition where there is a mixture of 2.44 µg/L of chlorophyll and 5.71 mg/L of sediments, the reconstructed images exhibit a higher noise level than all the other images except those with 8.62 mg/L of sediments. This is correlated with its water turbidity level having a higher number of ALs compared with the saltwater conditions of the other images (except those with 8.62 mg/L of sediments).

5.4.2 Reconstruction Performance via Different Localisation Accuracy Metrics

To provide a different perspective to the reconstruction performance of the detection threshold and median algorithm, its reconstructed images are evaluated in terms of pixel-wise absolute localisation accuracy and relative localisation accuracy. The pixel-wise absolute localisation accuracy measures the difference between each target pixel's reconstructed dToF values and its expected dToF value, while the relative localisation accuracy measures the range of the target pixels' reconstructed dToF values. As mentioned in Section 5.4.1, target pixels are pixels at the expected spatial position of the target in the image.

The imaged target at the AIMS indoor saltwater tank is a planar object. Therefore, the expected dToF values of the target pixels should be identical. For the imaging setup at the AIMS indoor saltwater tank, the expected dToF value of the target is calculated to be 6 clock cycles[5].

The pixel-wise absolute localisation accuracy is evaluated for the reconstructed images containing a discernible representation of the target (i.e. all images except the saltwater condition of 8.62 mg/L of sediments). It can be seen that the target pixels mostly have a reconstructed dToF value of 4 or 5 clock cycles, which is 1-2 clock cycles from the target's expected dToF value. The number of pixels with a reconstructed dToF value of 5 clock cycles only becomes the majority at the saltwater condition with a mixture of 2.44 µg/L of chlorophyll and 5.71 mg/L of sediments, where the water turbidity is the second highest tested level. Therefore, the pixel-wise absolute localisation accuracy of the reconstructed images is 1–2 clock cycles.

[5]The expected dToF value does not account for System B's timing delay of 40 ns, as discussed in Section 3.4.1.

The reconstructed dToF values of the target pixels are either 4 or 5 clock cycles, meaning the relative localisation accuracy is one clock cycle. This means the reconstructed dToF values of the target pixels are mostly consistent, even though their reconstructed dToF values do not always match the expected dToF value of the target.

5.4.3 Target Classification of Reconstructed Images

The target classification performance of reconstructed images of targets submerged in saltwater at different sediment concentrations is assessed by determining whether the target classification's overall accuracy degrades when the image is captured at increasing sediment concentrations. All reconstructed images up to a sediment concentration of 5.62 mg/L are utilised for the evaluation of target classification performance. This does not only include the reconstructed images for sediment concentrations shown in Table 5.2, but also for the other sediment concentrations (that are up to 5.62 mg/L) listed in Table B.1 of Section B.1 in Appendix B. The reconstructed images at 8.62 mg/L are not evaluated for target classification because the target is not discernible in them, as shown in Table 5.2.

For target classification, a set of rules based on the target's area and target's object ratio are used for differentiating the possible shapes of the target. The target's object ratio is defined as the ratio of the target's area to its bounding box's area. The target's area is calculated as the number non-zero pixels in the target's bounding box (which contains only the target). These rules follow those developed by [33], but use a different threshold value because the reconstructed images have fewer sharp edges compared with those in [33]. New threshold values are used for the target classification algorithm here, which are determined empirically.

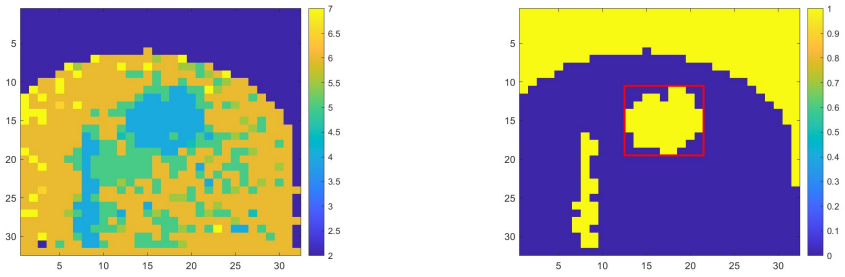
As discussed in Section 3.7 of Chapter 3, the classifier includes an additional prediction class called "Undecided" for cases where the target does not fit any of the classification rules for the three classes of shapes. However, no instances of this Undecided class are used to test the classifier, because this class is only meant to account for instances in which the classifier cannot determine a class for the target based on its rules.

The classification rules require a bounding box to be drawn around the target. Here, a target detection algorithm is used. The output of the target detection algorithm provides the area of the target and properties of the target's bounding box. The dimensions of the bounding box are provided, which can be used for calculating its area.

Since target classification performance is the focus of this section's investigation, a standard target detection algorithm augmented in an ad hoc way is applied here and is detailed in Section 5.4.3.2. If the target classification algorithm is to be applied to a dataset collected via a different imaging set-up, a different detection algorithm may be needed, depending on the background pixels of the reconstructed image.

To enable more accurate target detection, the reconstructed image is converted to a binary image and then denoised, creating a simpler representation of the recon-

structed image. Figure 5.16(a) shows the reconstructed image, while Figure 5.16(b) shows the corresponding denoised binary image, with the red box representing the bounding box determined by the target detection algorithm. The red box indicates that the target detection algorithm is able to create a boundary box that contains only the target.



(a) (b)

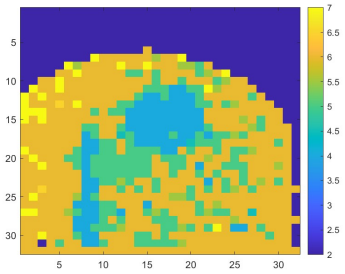
Figure 5.16: Comparison between the reconstructed image (Figure 5.16(a)) of the circle target in the AIMS indoor saltwater tank with 5.62 mg/L of sediment and its denoised binary image (Figure 5.16(b)) with bounding box showing the segmented image contains only the target.

Section 5.4.3.1 explains the steps applied to convert the reconstructed image to a denoised binary image, then Section 5.4.3.2 presents details of the target detection algorithm. The real-time capability of this target classification algorithm is then presented in Section 5.4.3.3. This is followed by a discussion of the target classification results in Section 5.4.3.4.

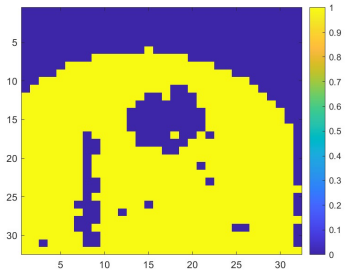
5.4.3.1 Converting the Reconstructed Image to a Denoised Binary Image

Several steps are applied to the reconstructed image to produce the denoised binary image. Figure 5.17 shows the image at every step of this process. First, the 5th percentile of the reconstructed image is used as the threshold value for converting the image to a binary image. Any value below or equal to the threshold value is converted to 0. The 5th percentile is chosen because the target's distance is within this percentile. This results in the image shown in Figure 5.17(b), where the target pixels have the value 0.

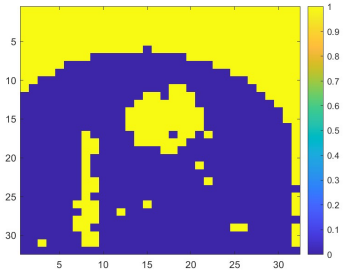
Next, scattered pixels are removed and holes are filled to produce an image suitable for detection, as demonstrated by the steps shown in Figure 5.17(c) to 5.17(f). The image is inverted in Figure 5.17(e) and scattered pixels are removed from it (in Figure 5.17(f)) to effectively fill holes in the original binary image. Finally, the image is inverted one last time, as depicted in Figure 5.17(g), to ensure that target pixels contain the value 1, thereby enabling the target detection algorithm to segment the target from the image.



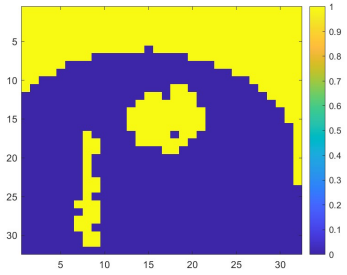
(a) Reconstructed image



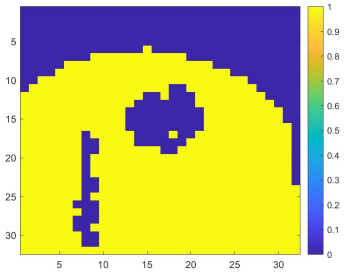
(b) Thresholded to binary image



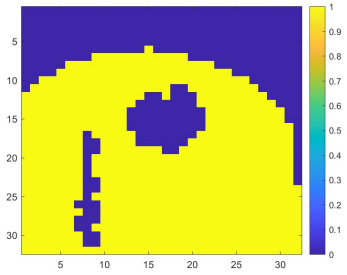
(c) Flipped binary image



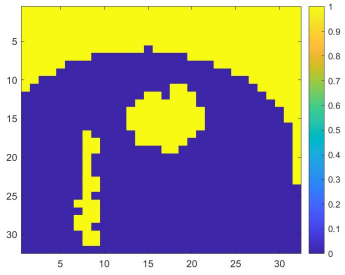
(d) Removed scattered pixels



(e) Flipped binary image



(f) Removed scattered pixels (effectively filling holes)



(g) Denoised binary image

Figure 5.17: Images depicting each processing step applied to a reconstructed image before it is used for target detection. This is of a square submerged in saltwater condition with 5.62 mg/L of sediments.

5.4.3.2 Target Detection

The target detection algorithm can be described in two parts. The first part performs detection of objects in the denoised binary image (from the previous section). Here, an object is defined to be a group of connected pixels of the value 1 in the binary image. Pixels are considered connected if they are adjacent in any of the eight possible directions: north, south, east, west, northeast, southeast, southwest, or northwest. An example implementation of this is Matlab's `regionprops` function.

Given that each denoised binary image contains many different geometric artefacts in addition to the target, many objects are detected in each image. Therefore, the second part of the target detection algorithm applies a small number of empirically determined rules to select which object is the target. These rules are related to the object's bounding box's area, position and dimensions, and are evaluated sequentially. An object is discarded immediately if it fails to meet any one of the criteria. The object that satisfies these criteria is selected as the target. The detection algorithm outputs the area and bounding box of this target. They are used for classification of the target's shape, as described in Section 5.4.3.

Figure 5.16(b) in Section 5.4.3 shows the bounding box determined by the target detection algorithm, and it shows the bounding box to only include the target. Images where no target is detected are discarded and not tested for target classification.

This target detection algorithm successfully detects a target for all the denoised binary images except for 125 out of 900 images of the square target in a sediment concentration of 1.78 mg/L. Therefore, when testing the overall accuracy of the target classification algorithm at each sediment concentration, 900 images are used for each possible shape of the target, except for the square at 1.78 mg/L of sediments, where only 775 images are evaluated.

The denoised binary images in Table B.2 in Appendix B show the target detection result (i.e. the bounding box) for an image in each sediment concentration and each possible shape of the target.

5.4.3.3 Real-Time Capability of Target Classification Algorithm

The execution time of the target classification algorithm is not measured because the implementation is not optimised. An analysis of the operations used in this algorithm can demonstrate that the algorithm is real-time capable. The target classification algorithm can be summarised by Figure 5.18. It includes the detection threshold and median algorithm because the target classification algorithm requires the detection threshold and median algorithm's reconstructed images. Therefore, the detection threshold and median algorithm should be included in the consideration of whether this target classification algorithm is real-time capable.

All of the steps shown in Figure 5.18 are real-time capable. The detection threshold and median algorithm in the first step is already shown to be real-time capable in Section 5.3.1. The use of threshold in steps 2, 5 and 6 is computationally simple.

For step 3, the task of removing scattered pixels and filling holes requires the search operation to identify holes and noise pixels in the reconstructed image. How-

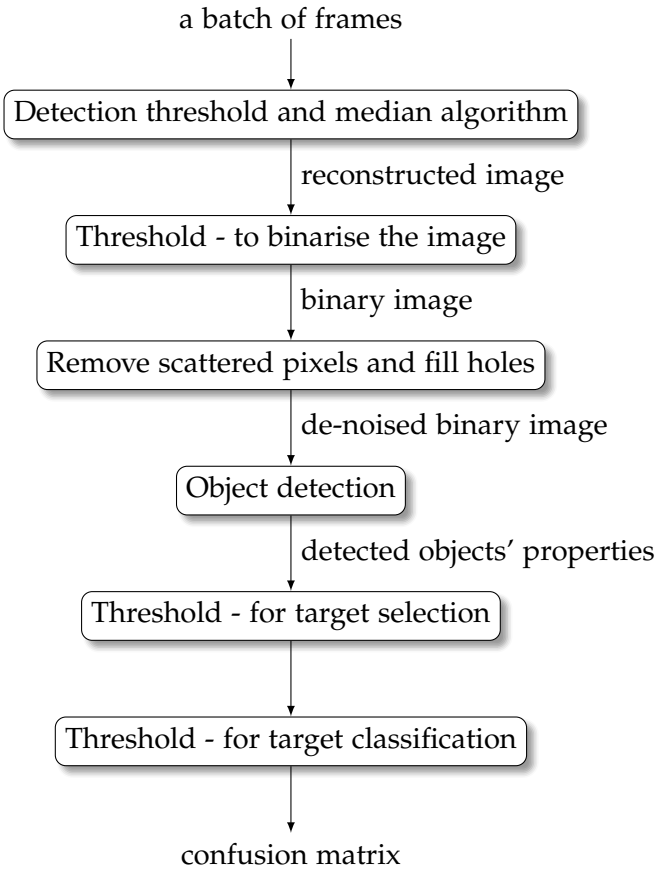


Figure 5.18: Target classification algorithm of reconstructed images from the detection threshold and median algorithm

ever, the reconstructed image only contains $32 \times 32 = 1024$ pixels. Therefore, the search operation only needs to be executed on 1024 pixels. This small number of pixels means this step is real-time capable.

For step 4, object detection is a standard operation in real-time computer vision for images with higher pixel resolution than the 32×32 images evaluated in this algorithm. Therefore, it's clear that this step is real-time capable.

From this discussion, the target classification algorithm is real-time capable.

5.4.3.4 Target Classification Results

The total number of denoised binary images tested for target classification for each sediment concentration is 2,700, with 900 images for each shape. The only exception is at 1.78 mg/L, where the total number of images is 2,575, due to 125 fewer images of the square being used for classification. This is because the target detection algorithm is unable to segment an image containing the target in those 125 images, as discussed in Section 5.4.3.2. Only the classification of the three shapes is tested for sediment

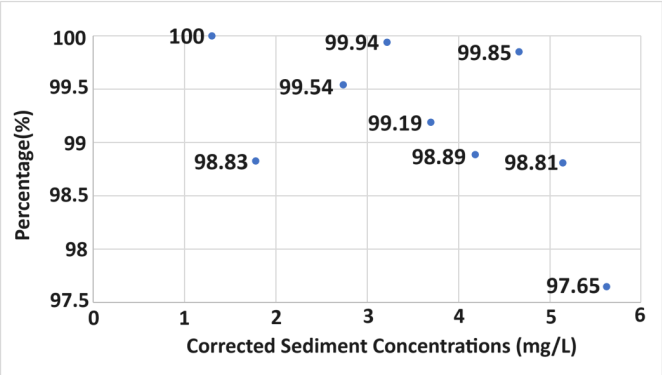


Figure 5.19: Overall accuracy in classifying binary denoised images (i.e. the denoised binary version of the reconstructed images) of a circle, a square or a triangle in varying sediment concentrations.

concentrations up to 5.62 mg/L[6] because the data for the square target at 8.62 mg/L is corrupted.

The overall accuracy of target classification of the binary denoised images (i.e. the denoised binary version of the reconstructed images) from data collected under different sediment concentrations is presented in Figure 5.19[7]. The definitions of overall accuracy are given in Section 3.7 of Chapter 3. It can be seen that the overall accuracy is high for all sediment concentrations[8]. The equivalent number of ALs for the saltwater condition with this range of sediment concentrations is 1.32 to 3.29. Table B.1 of Section B.1 in Appendix B provides a detailed list of the equivalent number of ALs for each sediment concentration. The table also contains the reconstructed images and their corresponding denoised binary images for data collected at saltwater conditions with each sediment concentration.

For sediment concentrations up to 3.70 mg/L (equivalent to 2.35 ALs), the algorithm's misclassifications are primarily due to the algorithm classifying shapes as Undecided. The Undecided class is used to account for instances where the classifier cannot definitively determine a shape based on the predefined rules, as explained in Section 3.7 of Chapter 3.

As sediment concentrations increase beyond 3.70 mg/L, the misclassifications include cases where shapes are incorrectly classified as a different class, in addition to being categorised as Undecided. Figure 5.20 shows the number of Undecided classifications as the total percentage of classified denoised binary images for each sediment concentration. This figure demonstrates that the percentage of Undecided classifications remains below 2.5% for sediment concentrations up to 5.62 mg/L,

[6]The sediment concentration values reported in [101, Fig. 7] are not the corrected sediment concentration values. The values given in this thesis are the corrected values.

[7]The overall accuracy values in [101, Fig. 7] is incorrect; the values have been updated in this figure.

[8]The values for F1-measure, macro precision, and macro recall given in [101] are based on an overly simplistic performance analysis. They should not be used in this classification scenario because the classifier is not tested on any instances of the Undecided class, as discussed in Section 3.7 of Chapter 3.

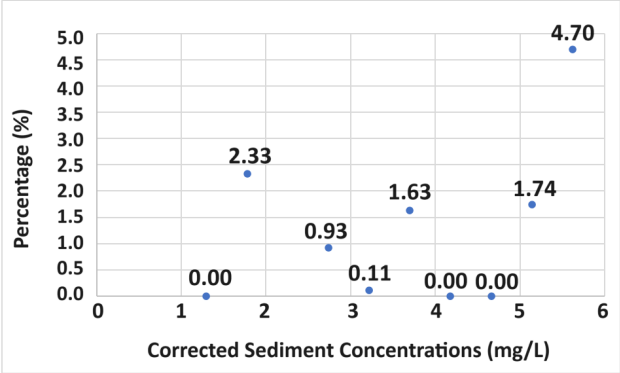


Figure 5.20: Percentage of classification categorising an image as Undecided over all the images of all three shapes at varying sediment concentrations.

indicating a relatively low rate of indecision in the classifier's performance for these conditions.

From both Figures 5.19 and 5.20, no clear trend in overall accuracy with increasing sediment concentration is evident. The only noticeable trend is a dip in classification accuracy at 5.62 mg/L (equivalent to 3.29 ALs). This drop in overall accuracy may be due to the higher sediment concentration, which introduces additional noise in the reconstructed images, as shown by the images for 5.62 mg/L sediments in Table 5.2. This dip in overall accuracy indicates a need to enhance either the target classification algorithm or the image-reconstruction algorithm to ensure that overall accuracy remains above 98% and the percentage of Undecided classifications stay below 2.5%. Addressing these aspects could help maintain consistent target classification performance across varying sediment concentrations.

5.5 Image Reconstruction of Data Collected at DSTG Indoor Freshwater Tank

The histogram averaging algorithm proves to be more effective than the detection threshold and median algorithm for image reconstruction of data collected at the DSTG indoor freshwater tank, as shown in Table 5.3. For this dataset, a batch of 300 frames is used for reconstructing each image by either algorithms.

Expected dToF values are used to validate the dToF values in these reconstructed images. They are calculated from the theoretical dToF value, which is the amount of time it takes for a photon to travel from the SPAD array flash LiDAR system to the target and back. This calculation considers the refractive index of water (1.33), adjusting for the speed of light in water and using Snell's law to account for changes in propagation angle due to refraction.

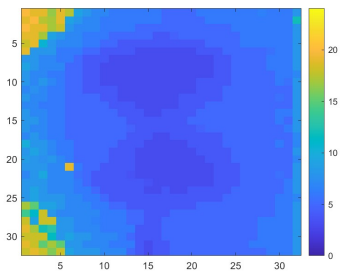
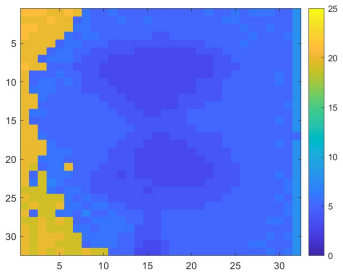
Once the theoretical dToF value is calculated, System C's timing delay of 13.3 ns is subtracted from the theoretical dToF value as part of calculating the expected dToF value. This is to ensure the expected dToF value is consistent with System B's dToF

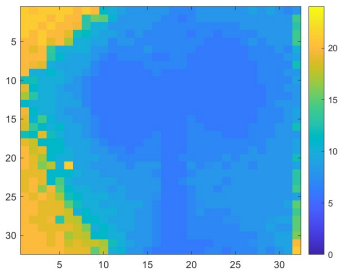
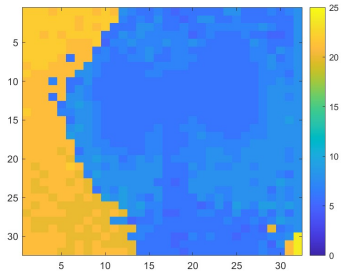
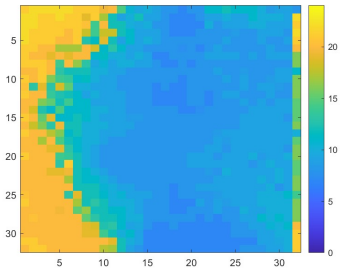
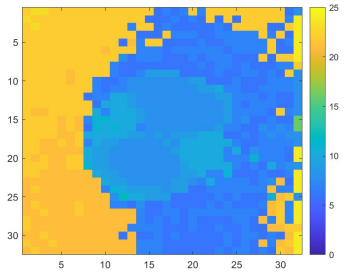
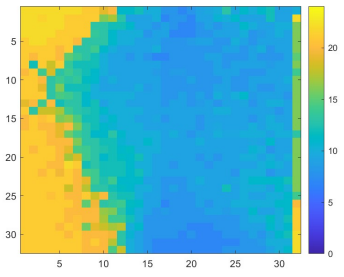
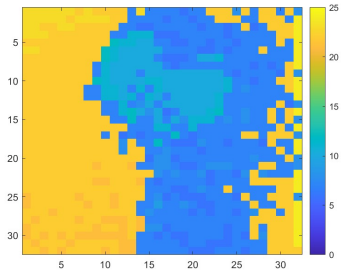
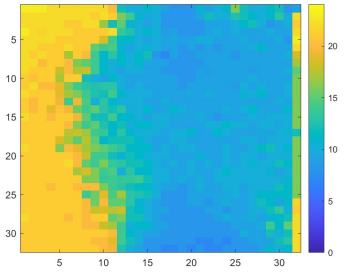
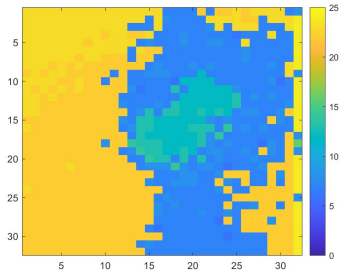
measurements which does not compensate for System C's timing delay, as discussed in Section 3.4.2 of Chapter 3.

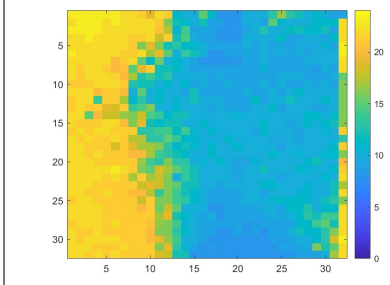
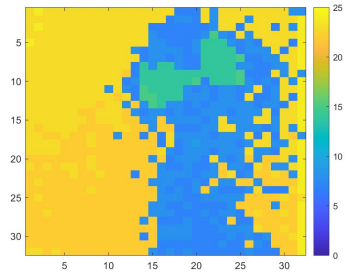
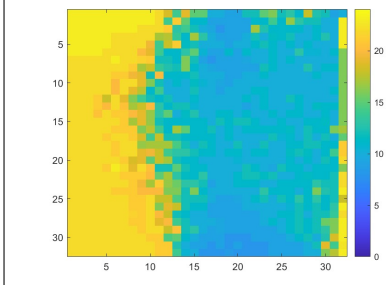
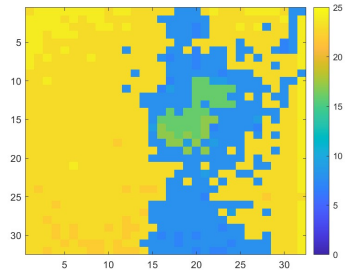
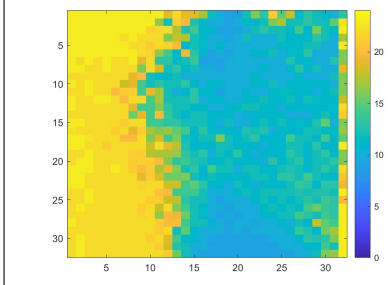
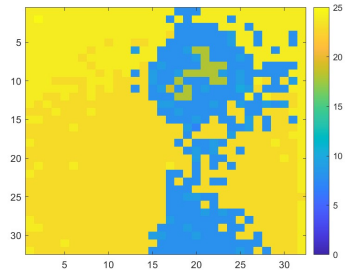
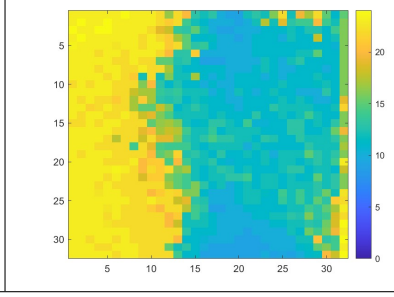
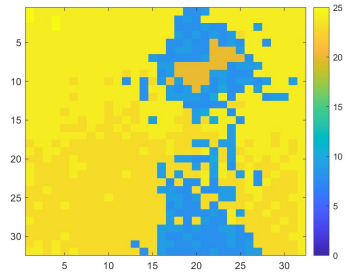
A post-processing range gate is utilised, before applying either of the reconstruction algorithms. This is different to the range gate applied during SPAD array flash LiDAR imaging. This is to further filter out unwanted photons and enhance image quality. The post-processing range gate values used are determined empirically to provide the best reconstructed image quality. The upper bound post-processing range gate value is uniformly set at 25 clock cycles for all depths. This value exceeds the expected dToF value of the tank's floor. The lower bound post-processing range gate value is set to 2 clock cycles for target depths from 0.5 m to 4.5 m, and 5 clock cycles for a target depth of 5 m.

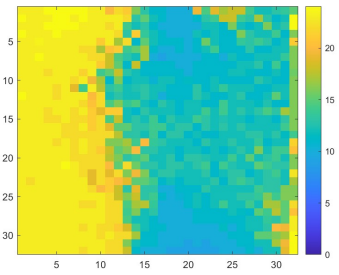
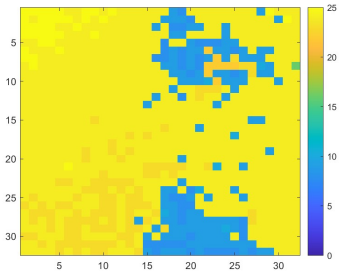
Table 5.3 presents a comparison between the two image-reconstruction algorithms, showcasing reconstructed images of the target at various depths below the water surface. Each image in the table uses the same colour bar for consistent comparison. The expected dToF value for the target at each depth is provided in the table.

Table 5.3: Different reconstructed images at various target depths below the water surface of the DSTG indoor freshwater tank.

Depth (m)	**Expected dToF value of the target(clock cycles)**	**Detection Threshold and Median image**	**Histogram Averaging image**
0.5	1		

1	2		
1.5	4		
2	5		
2.5	7		

3	8		
3.5	10		
4	11		
4.5	13		

5	14		

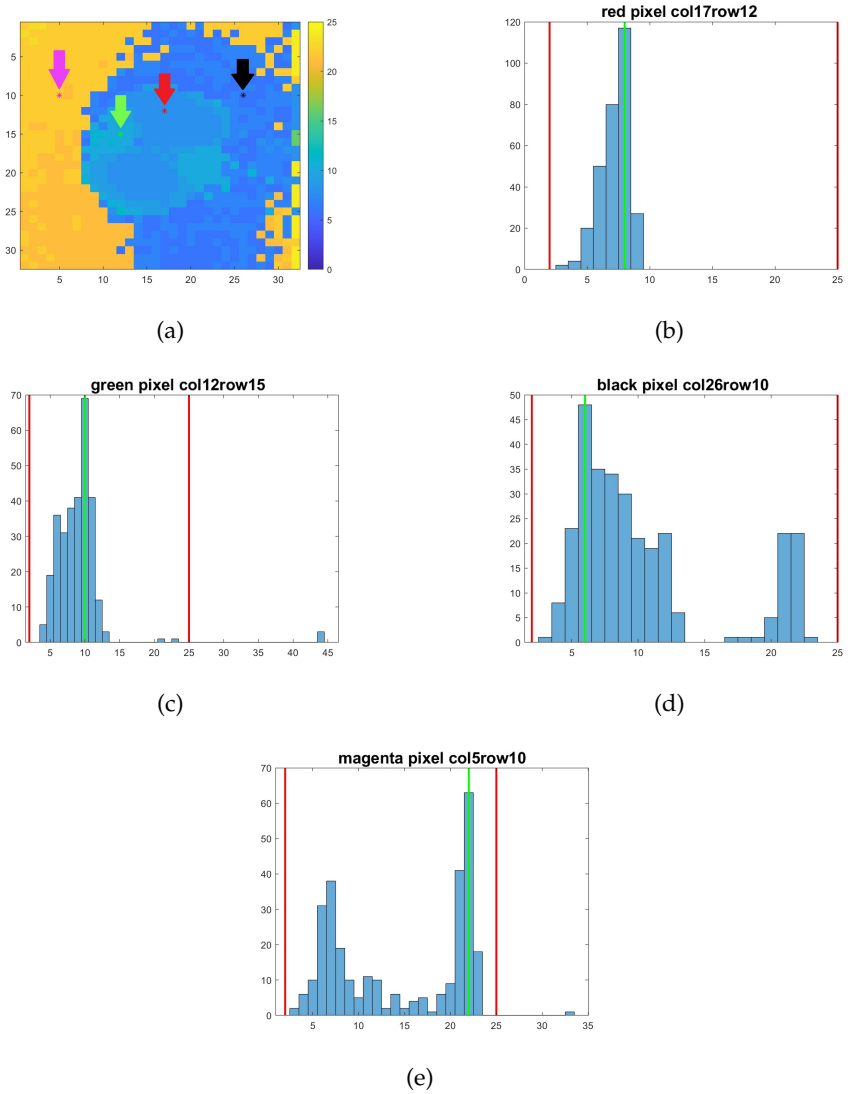
In all the reconstructed images in Table 5.3, only two of the quadrants of the SD are discernible in most images. Although we lack an RGB image to confirm the precise orientation of the SD, we assume these are the white quadrants of the SD.

Even though the reconstructed images represent depth information, the black quadrants of the SD target are not clearly visible for most of the reconstructed images, except the histogram averaging algorithm reconstructed image at 1.5 m depth. This phenomenon can be explained by variations in light reflection from the target compared to laser scattering in water. A detailed analysis of the pixel-wise histograms of dToF measurements provides an explanation for these effects as well as other observed effects in these images. Figure 5.21 provides histograms related to pixels in the histogram averaging algorithm's reconstructed image at 1.5 m depth.

Each histogram in Figure 5.21 is an accumulation of dToF measurements of each pixel marked with a coloured asterisk (indicated by an arrow) in Figure 5.21(a). The red lines in these histograms indicate the post-processing range gate values, while the green vertical line shows the reconstructed image value.

Although we lack an RGB image to confirm the precise orientation of the SD in Figure 5.21(a), we assume, based on experience, that the red-marked pixel in the reconstructed image is located on the white quadrant of the SD. Conversely, the green-marked pixel is presumed to be on the black quadrant of the SD. The red-marked pixel (Figure 5.21(b)) shows a higher peak in photon counts compared with the histogram for the green-marked pixel (Figure 5.21(c)). Theoretically, more photon counts should be observed in the white quadrant because the white colour reflects significantly more photons. In contrast, the less reflective black quadrant, represented by the green-marked pixel, shows a histogram peak attributed more to water scattering than to the target itself. Consequently, the histogram bin value for the histogram peak differs between the black and white quadrants, resulting in varying pixel values in the reconstructed image by the histogram averaging algorithm.

For the black-marked and magenta-marked pixels in Figure 5.21(a), these represent different types of background pixels. The black-marked pixel is in an area illuminated by the laser, while the magenta-marked pixel is in a region without laser illumination but still shows photon returns from ambient light and the bottom of the water tank. This is confirmed by their respective histograms: Figure 5.21(d) shows



(a) (b) (c) (d) (e)

Figure 5.21: A closer inspection into a histogram averaging algorithm image of an SD at 1.5 m depth in the DSTG indoor freshwater tank. Figure (a) shows the histogram averaging algorithm image with red, green, black and magenta asterisks and arrows marking pixels with different values. The histograms of dToF measurements used to reconstruct the distance value are shown for the red (b), green (c), black (d) and magenta (e) pixels. The title of each histogram states which colour-marked pixel is examined and the coordinates of the pixel. The green line is the reconstructed dToF value of the pixel in the reconstructed image. The red vertical lines in the histogram depict the post-processing range gate used before reconstruction is applied.

more detected photons reflected near the water surface, which could be caused by water scattering, and Figure 5.21(e) shows more photon counts from the freshwater tank's bottom.

The histogram characteristics discussed here for the red-, black- and magenta-marked pixels of the histogram averaging algorithm reconstructed image at 1.5 m depth are also applicable to other histogram averaging algorithm images in Table 5.3, where the white quadrants of the SD and effects from water scattering are also visible.

The difference in histogram characteristics provides a possible explanation for why the black quadrant is not visible in these other reconstructed images. It is possible that the small number of photons returns from the less reflective black quadrant were not detected by the SPAD camera when imaging due to the relatively larger number of photon returns from water scattering.

The next subsection (Section 5.5.1) discusses the reconstructed images in more detail and assesses their quality. This is followed by Section 5.5.2, which discusses a phenomenon observed in the histograms of the collected dToF measurements, specifically the existence of two histogram peaks for data from targets at depths of 2.5 m–4.5 m. Understanding these data characteristics enables a more thorough evaluation of the histogram averaging algorithm's global target localisation performance, which is discussed in Section 5.5.3, which addresses a systematic discrepancy between the reconstructed dToF value for pixels at the target's expected spatial position in the image, and the expected dToF value.

5.5.1 Reconstruction Performance via Visual Inspection

As only the white quadrants of the SD are visible in most reconstructed images in Table 5.3, the reconstruction performance is evaluated on whether the white quadrants are discernible in the images.

The reconstructed images in Table 5.3 demonstrate that the histogram averaging algorithm provides a more discernible image of the white quadrants compared with the detection threshold and median algorithm. The white quadrants become indiscernible in the detection threshold and median algorithm's reconstructed images for depths beyond 1.5 m. In contrast, the reconstructed images by the histogram averaging algorithm retains discernibility of the white quadrants for depths to 4.5 m.

The discernibility of the white quadrants in these reconstructed images means the white quadrants can be used for target detection of the SD. Target classification using only features of the white quadrants is possible, on the condition that other target classes do not consist of only white quadrants as part of their image features.

While the white quadrants are not discernible in the histogram averaging algorithm reconstructed image at a 5 m depth, the image still provides sufficient information for target detection. It contains a small number of pixels representing the white quadrants of the SD, making it possible to determine the presence of a target within the FOV. Unlike the histogram averaging algorithm's reconstructed images of targets at shallower depths, this reconstructed image cannot be extended for target classification applications in the future because the SD's features are not distinguishable in the image.

5.5.2 Two Peaks in the Collected Data

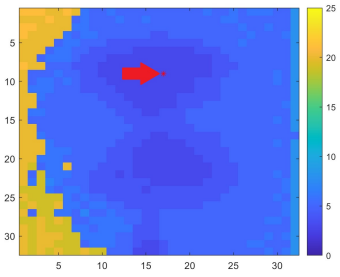
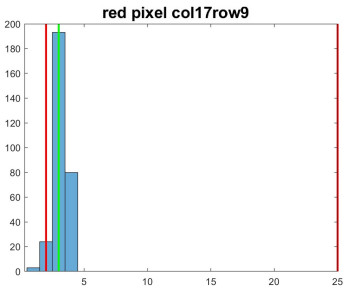
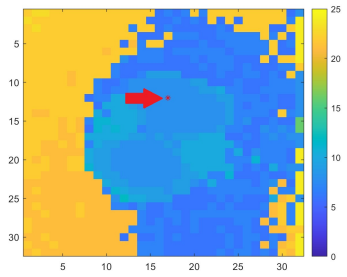
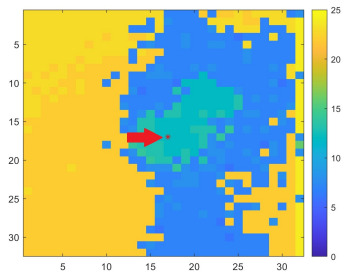
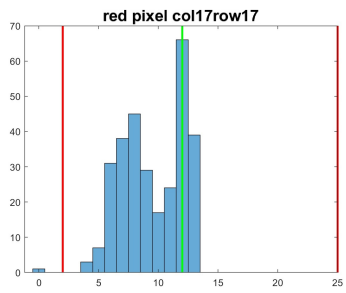
A two-peak phenomenon emerges in the pixel-wise histogram of dToF measurements when the target is positioned 2.5 m to 4.5 m below the water surface. This effect is particularly evident for pixels corresponding to the white quadrant of the SD. Table 5.4 presents a series of histograms of dToF measurements for a pixel in reconstructed images of the white quadrant at various submerged depths. For each depth, the examined pixel is located on the white quadrant. A red asterisk (indicated by red arrow) is used to mark the pixel in each of the reconstructed images. The reconstructed image's dToF value (indicated by the green line in the histograms) consistently aligns with the second peak, which corresponds logically to the target being at greater depths and, therefore, farther away from the SPAD array flash LiDAR system.

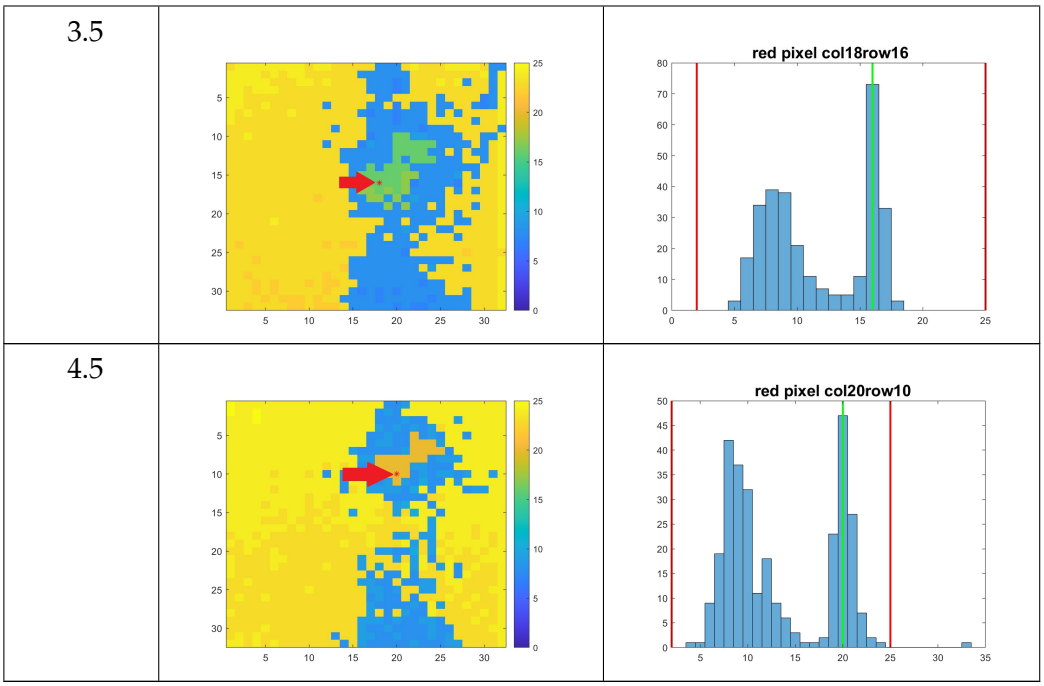
The first peak in these histograms is attributable to scattering in the water. As discussed in Section 5.2.3, any photon returns from the water surface are automatically filtered out by the SPAD array flash LiDAR's internal timing delay. Thus, the photon detections counted by the first histogram peak result solely from scattering rather than direct reflections from the water surface. This is confirmed by the first peak spanning approximately 10 clock cycles (around 6 to 15 clock cycles) as illustrated by the histogram for a target depth of 4.5 m in Table 5.4. The peak appears to have a shorter span in the other histograms because the target is at a shallower depth, causing the tail end of the peak to be overlapped by the target's histogram peak.

In contrast, the histograms of the red-marked pixel at shallower depths (0.5 m and 1.5 m) exhibit a single peak, likely due to the target's proximity to the water surface. In these cases, the photon counts from the target coincide with those from scattering, resulting in a single peak in the histograms.

This space is intentionally left blank.

Table 5.4: The histogram averaging algorithm image of targets at different depths and the histogram of dToF measurements at the pixel marked with red asterisk and arrow in the image. The pixel is selected to be on the SD's white quadrant and its dToF measurements is representative of the dToF measurements collected by other pixels on the white quadrants. The green line is the reconstructed dToF value of the pixel in the reconstructed image. The red vertical lines in the histogram depict the post-processing range gate used before reconstruction is applied.

Depth (m)	**Histogram averaging algorithm image with red-marked pixel**	**Histogram of dToF measurements of red-marked pixel**
0.5		
1.5		
2.5		



The distribution of dToF measurements might explain why the detection threshold and median algorithm is less effective for reconstructing images from this dataset. The majority of dToF measurements are at shorter distances than the target, causing the median value to not align with the histogram's second peak which corresponds to the target's depth. This misalignment results in less accurate reconstructions, as the median value does not represent the actual distance to the target effectively. Consequently, the detection threshold and median algorithm struggles to produce clear images, particularly when the target is positioned at greater depths.

5.5.3 Systematic Error in Global Target Locations

As the histogram averaging algorithm provides a better reconstructed image than the detection threshold and median algorithm (as discussed in Section 5.5.1), the former algorithm is evaluated for its localisation accuracy. However, it is difficult to evaluate the localisation accuracy in pixel-wise resolution for these reconstructed images. This is because it is difficult to accurately distinguish between the pixels that corresponds to the black quadrant from background pixels that corresponds to water scattering. Therefore, a more reliable localisation accuracy measure would be to evaluate the global target location that can be deduced from these reconstructed images.

From the discussions in this section, it is evident that the higher number of reflections from the white quadrant provides more reliable data for reconstruction. Consequently, the reconstructed global location of the target is defined as the reconstructed dToF value of one of the pixels corresponding to the white quadrant of the

SD. The reconstructed dToF value of the majority of pixels associated with the white quadrant is consistent, as evidenced by the pixels sharing the same colour within each reconstructed image. Therefore, the value of any one of these pixels can be used as the global target location.

In Section 5.5.2, Table 5.4 presents the histogram of dToF measurements of a pixel on the white quadrant of the SD for each imaging depth. The reconstructed dToF values, indicated by the vertical green line in the histograms in Table 5.4, are used as the global target location by this algorithm.

Comparing these global target location values with the target's expected dToF values shown in Table 5.3, it becomes apparent that they do not match consistently. The discrepancy in their values increases with target depth, ranging from 2 clock cycles to 7 clock cycles.

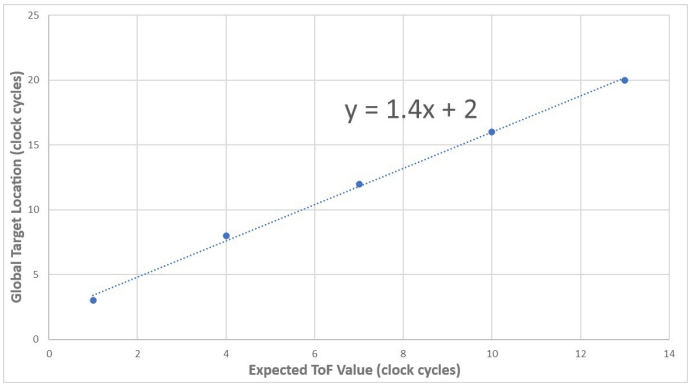


Figure 5.22: Global target locations (y-axis) compared with the expected dToF values.

Despite this mismatch, a linear relationship between these values is observed. This relationship is illustrated in Figure 5.22, where the global target locations are plotted against the corresponding expected dToF values from Table 5.3. This linear relationship suggests a systematic error in the imaging of the target using the SPAD array flash LiDAR system. Therefore, it can be deduced that the histogram averaging algorithm can reconstruct accurately, on the premise that further investigation is required on the cause of this systematic error.

5.6 Image Reconstruction of Data Collected at Port Adelaide River

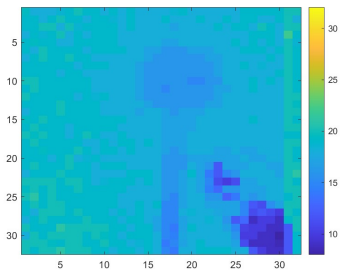
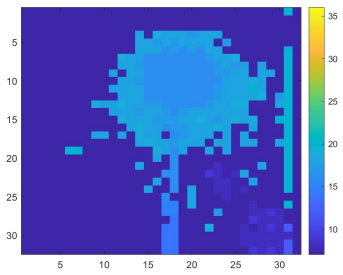
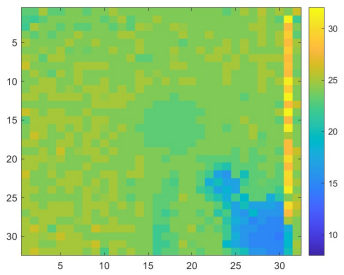
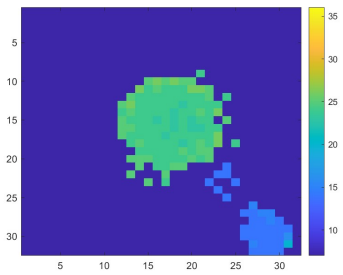
Similar to the results from the DSTG indoor freshwater tank (see Table 5.3), the reconstructed images using the histogram averaging algorithm provide a more discernible representation of the target compared with those reconstructed using the detection threshold and median algorithm.

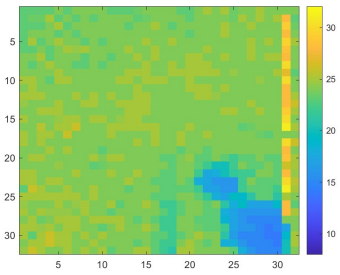
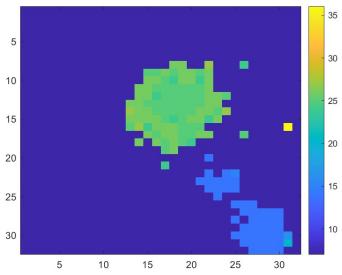
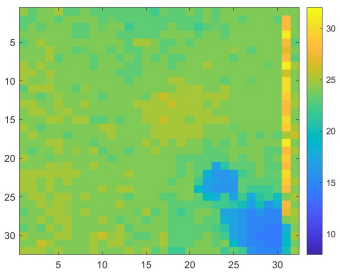
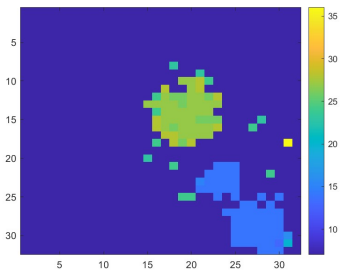
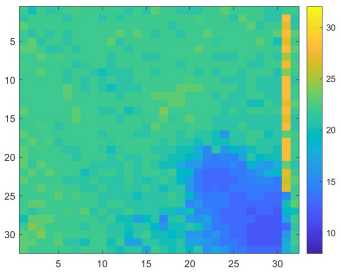
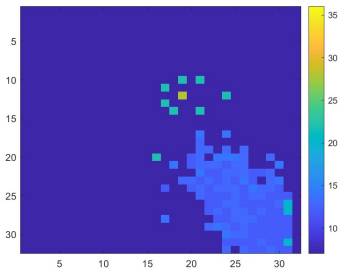
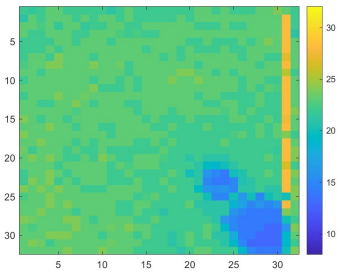
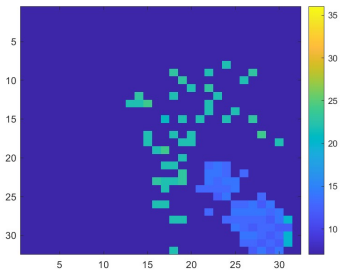
Table 5.5 below shows a comparison of reconstructed images between the detection threshold and median algorithm and the histogram averaging algorithm. The images are from data collected of the target at various depths below the water surface

and different camera angles, each created from a batch of 300 frames. All images in this table use the same colour bar to allow for direct comparison. No post-processing range gate was applied to this dataset before image reconstruction.

The reconstructed dToF value of the target pixels are validated against the target's expected dToF value in Table 5.5. Target pixels are pixels at the expected spatial position of the target in the image. The calculation of the expected dToF values follows the same method described in Section 5.5.

Table 5.5: Different reconstructed images at various target depths below the water surface of Port Adelaide River,

Depth (m) and camera angle (deg)	**Expected dToF value of the target (clock cycles)**	**Detection Threshold and Median image**	**Histogram Averaging image**
0.5, 56	18		
1.0, 63	24		

1.5, 63	26		
2.0, 63	28		
2.5, 63	30		
3.0, 63	32		

In all the histogram averaging algorithm images, the background pixels are all the pixels with undefined values. The background pixels' histogram peak's photon count is below 10% of the total number of frames used to reconstruct an image. This indicates that the peak represents noise. Meanwhile, detection threshold and median algorithm performs a check on whether the total number of detected photons

is above 10% of total number of frames, instead of only the histogram peak. Therefore, its reconstructed dToF value is not filtered out by this check and is used in the reconstructed image. This highlights the effectiveness of employing histogram peak to filter out noise.

A group of pixels (with defined reconstructed dToF values), forming a diagonal pattern at the bottom right corner, is consistently observed in all reconstructed images by both algorithms. These pixels appear light blue in reconstructed images at depths of 1.0 to 3.0 m, and dark blue in the image of the target at 0.5 m. The discrepancy in reconstructed dToF values in the 0.5 m deep target image arises because the camera angle differs from that in other images (as shown in Table 5.5), placing the camera closer to the rope.

These pixels represent the rope suspending the target in the water, positioned on the right side of the SPAD camera (see Figure 5.11 in Section 5.2.4). This is confirmed by their reconstructed dToF values, which are shorter than the expected dToF value of the water surface. Specifically, the expected dToF value for the water surface is 21 clock cycles for images with targets at depths of 1.0 to 3.0 m, and 16 clock cycles for the image with the 0.5 m deep target.

The consistent reconstructed dToF values of these pixels across images of different target depths (except for the 0.5 m depth due to the different camera angle) further confirm that these pixels represent the rope. This consistency is due to the rope's proximity to the camera being unaffected by the target's depth, as described in Section 5.2.4.

The pixels related to the rope do not appear as a distinct line because the image is reconstructed with data collected from 300 frames, which is acquired over 12 seconds. The acquisition time is long because System C's Aréte laser operates at 25 Hz. This long acquisition time means the SPAD array flash LiDAR system captures the swaying rope in various positions. When reconstructing the image with data of the rope at various positions, this results in a blurred appearance in the reconstructed image.

The next subsection (Section 5.6.1) discusses why the histogram averaging algorithm produces better reconstructed images than the detection threshold and median algorithm. Section 5.6.2 quantifies the reconstruction performance of the histogram averaging algorithm via relative localisation accuracy. Then Section 5.6.3 discusses the challenges associated with using the histogram averaging algorithm to reconstruct targets at depths greater than 2.0 m. Subsequent sections explore the distribution of dToF measurements in Section 5.6.4 and discuss the global target absolute localisation accuracy of the histogram averaging algorithm in Section 5.6.5.

5.6.1 Reconstruction Performance by Visual Inspection

Table 5.5 demonstrates that the histogram averaging algorithm yields a more discernible reconstructed image of the target compared with the detection threshold and median algorithm, even though the histogram averaging algorithm cannot provide a discernible reconstruction of the target at depths beyond 2.0 m. For a detailed

explanation of the challenges faced by the histogram averaging algorithm at greater depths than 2.0 m , see Section 5.6.3.

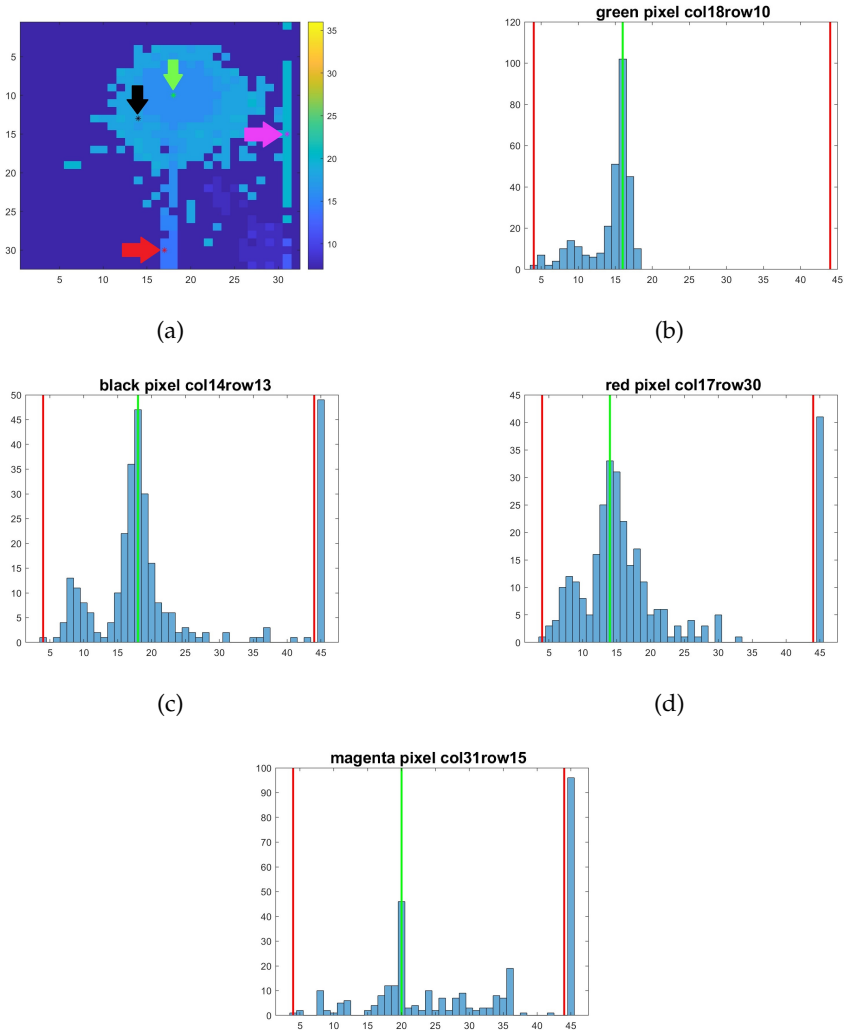
For data collected at target depths from 1.0 m to 2.0 m, the histogram averaging algorithm delivers the clearest representation of the target, which appears as a circular shape in the reconstructed images. This demonstrates that these reconstructed images can be used for visual target detection. In contrast, the target is less discernible in the images reconstructed using the detection threshold and median algorithm.

For the histogram averaging algorithm, its reconstructed image for the target at 0.5 m differs from those at 1.0 m, 1.5 m and 2.0 m due to the different camera angles. This results in more reflections from the water surface being detected by the SPAD array flash LiDAR system. The next subsection provides a detailed explanation of the artefacts seen in the histogram averaging algorithm's reconstructed image for the target at 0.5 m.

5.6.1.1 Histogram Averaging Algorithm Reconstructed Image at 0.5 m Depth

The artefacts seen in the histogram averaging algorithm's reconstructed image at 0.5 m can be explained by examining the pixels' histogram of dToF measurements. Figure 5.23 presents histograms of dToF measurements for pixels in different regions of the reconstructed images, each marked with a coloured asterisk (indicated by the same coloured arrow). These histograms help explain the attributes of each pixel's region.

This space is intentionally left blank.



(a) (b) (c) (d) (e)

Figure 5.23: A closer inspection into a histogram averaging algorithm image of a white disc submerged at 0.5 m depth at the Port Adelaide River. Figure (a) shows the histogram averaging algorithm image with green, black, red and magenta asterisks and arrows marking pixels with different values. The histograms of dToF measurements used to reconstruct the distance value are shown for the green (b), black (c), red (d) and magenta (e) pixels. The title of each histogram states which colour-marked pixel is examined and the coordinates of the pixel. The green line is the reconstructed dToF value of the pixel in the reconstructed image. The red vertical lines in the histogram depict the post-processing range gate used before reconstruction is applied.

Figure 5.23(b) illustrates that the circular region containing the green-marked pixel most likely represents the target. The histogram for this pixel shows a reconstructed dToF value of 16 clock cycles, which is 2 clock cycles away from the expected dToF value for the target of 18 clock cycles (as detailed in Table 5.5). The small discrepancy of this value to the expected dToF value, combined with the circular shape of the region, suggests that it accurately represents the target. The discrepancy between the target's reconstructed and expected dToF values does not mean the target is localised incorrectly. This is because the reconstruction algorithm produces different reconstructed dToF values for all the pixels related to the target so some pixels have the same reconstructed dToF value as the expected dToF value. The localisation accuracy of this algorithm is further elaborated in Section 5.6.5.

The second region of interest in the reconstructed image of the 0.5 m depth target is the "halo" surrounding the target. Figure 5.23(c) shows the histogram for the black-marked pixel in this region, shown in Figure 5.23(a). The histogram indicates that this halo corresponds to scattering from the water surface. Its histogram peak is at 18 clock cycles, close to the water surface's expected dToF value of 16 clock cycles. While the reconstructed dToF value is the target's expected dToF value, it is physically not plausible for this region to represent the target as well as the inner circular region, as it would be physically impossible for the same surface to have two distinct depth values.

The narrow region associated with the red-marked pixel (at the bottom of Figure 5.23(a)) is hypothesised to represent reflections from the water surface. The histogram for this pixel, presented in Figure 5.23(d), shows a peak at 14 clock cycles, which is within 2 clock cycles of the water surface's expected dToF value of 16 clock cycles. The shorter distance recorded is likely due to the SPAD camera's imaging angle of 56° from vertical, causing photon returns closer to the image's bottom edge to be detected sooner than those from the top half of the camera.

Another narrow region near the right edge of the reconstructed image of the 0.5 m depth target is suspected to be an artefact of an electronic error in the SPAD camera. This is supported by the histogram of the magenta-marked pixel, a pixel in this region, in Figure 5.23(e), which shows a peak at 20 clock cycles with a distribution that differs from the other histograms presented in Figure 5.23.

All histograms in Figure 5.23 show a non-trivial number of detected photons with dToF measurements preceding the expected dToF value for the water surface (16 clock cycles for this reconstructed image's target depth). These detected photons are attributed to ambient noise in the environment. A more detailed discussion on this noise is provided in Section 5.6.4.

5.6.2 Reconstruction Performance via Relative Localisation Accuracy

To provide a more quantitative analysis of the reconstruction performance of the histogram averaging algorithm, the relative localisation accuracies of the target pixels are assessed for the reconstructed images that contain a discernible representation of the target. They are the images of the target at depths 0.5 m to 2 m. Similar

to Section 5.4.2, the relative localisation accuracy measures the range of the target pixels' reconstructed dToF values.

The range of the target pixels' reconstructed dToF values is calculated as the difference between their minimum and maximum values. Since the target is clearly visible in the reconstructed images, identifying which pixels correspond to the target is straightforward. Table 5.6 presents the result.

Table 5.6: Range of reconstructed dToF values.

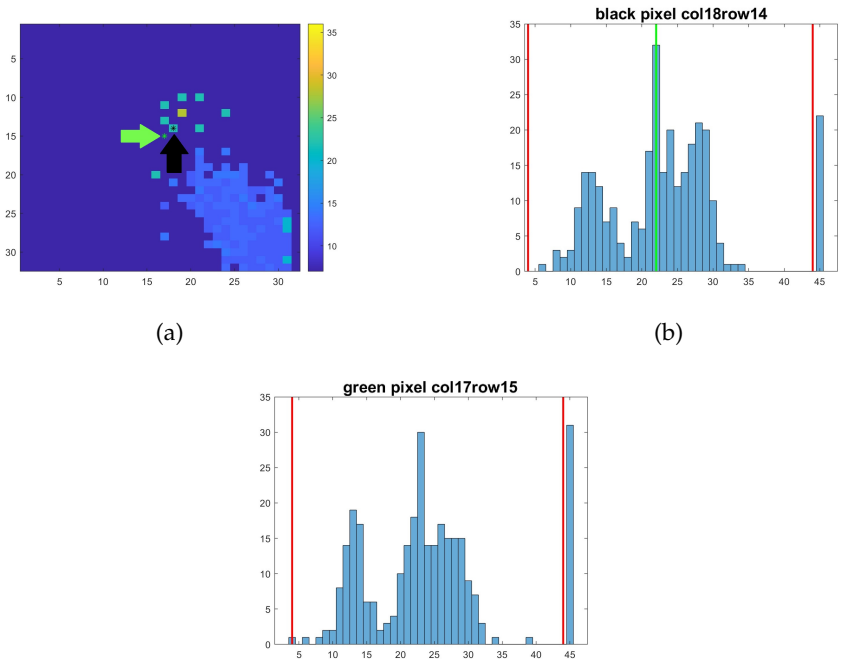
Depth (m)	**Range of reconstructed dToF values (clock cycles)**
0.5	16-19
1	23-24
1.5	24-27
2	24-28

The range of the reconstructed dToF values in Table 5.6 indicates that the reconstructed dToF values for target pixels can vary by up to 5 clock cycles. This variation may be attributed to the extended acquisition time of the batch of frames used for reconstruction. The SPAD array flash LiDAR system employed for this dataset operates at 25 Hz (as noted for System C in Table 3.1 in Section 3.4 of Chapter 3). Thus, a batch of 300 frames requires 12 seconds of acquisition time. In a dynamic environment such as a river with constantly moving water, the target might have shifted during this period, leading to variations in the reconstructed dToF values observed in the reconstructed images.

5.6.3 Reconstruction Difficulty with the Histogram Averaging Algorithm

Despite the histogram averaging algorithm demonstrating better performance over the detection threshold and median algorithm, it still faces significant challenges when reconstructing images for targets positioned at depths greater than 2.0 m. At these greater depths, identifying a clear histogram peak around the target's depth becomes increasingly difficult. For example, Figure 5.24 illustrates this issue for a target submerged at 2.5 m. The figure shows histograms of dToF measurements for two different colour-marked pixels within the reconstructed image. These histograms highlight the difficulty in identifying clear peaks corresponding to the target's expected dToF value of 30 clock cycles, underscoring the limitations of the histogram averaging algorithm at greater depths.

From Figure 5.24, both the black (Figure 5.24(b)) and green (Figure 5.24(c)) pixels' histograms lack a distinct peak around the 30-clock cycle mark, which hampers the effectiveness of the histogram averaging algorithm for reconstructing the target



(a) (b)

(c)

Figure 5.24: A closer inspection into a histogram averaging algorithm image of a white disc submerged at 2.5 m depth at the Port Adelaide River. Figure (a) shows the image with black and green asterisks and arrows marking pixels with different values. The histograms of dToF values are shown for black (b) and green (c). The title of each histogram states which colour-marked pixel is examined and the coordinates of the pixel. If there is a green line in the histogram, it represents the reconstructed dToF value for the pixel. The red vertical lines in the histogram depict the post-processing range gate used before reconstruction is applied.

image. The most prominent peaks in these histograms are likely due to scattering from the water surface, appearing 1 to 3 clock cycles away from the water surface's expected dToF value of 21 clock cycles. This trend is also observed in data collected at a depth of 3.0 m, which exhibits similar histogram characteristics.

The histograms in Figure 5.24 reveal a notable amount of detected photons with dToF measurements preceding the expected 21 clock cycles for the water surface, indicating the presence of ambient noise. Further analysis of this noise is provided in Section 5.6.4.

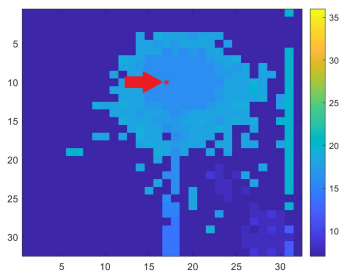
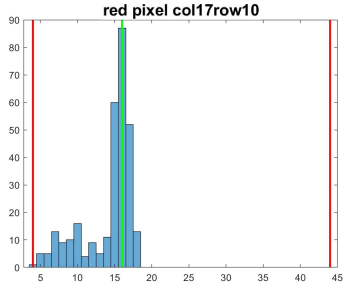
Additionally, the histograms for targets at depths of 2.5 m and 3.0 m from the Port Adelaide River do not exhibit the two-peak phenomenon observed in Section 5.5.2 for data collected at the DSTG indoor freshwater tank. At the DSTG tank, this two-peak phenomenon is evident for targets at depths ranging from 2.5 to 4.5 m. Although Figure 5.24 shows multiple peaks for data collected of a target at 2.5 m depth, these additional peaks correspond to reflections from the air and water

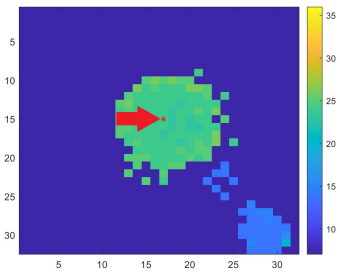
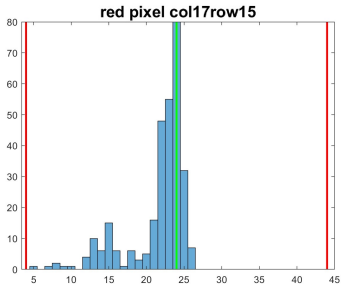
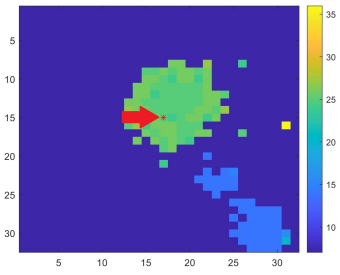
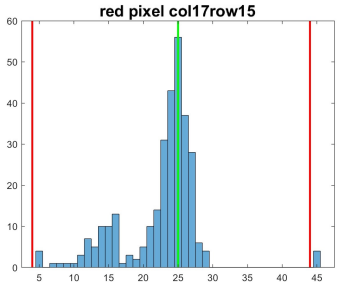
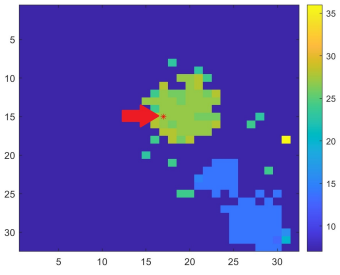
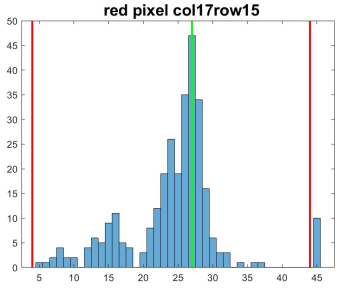
surface. In contrast, the histograms from the DSTG indoor freshwater tank only contain detected photons with dToF measurements from below the water surface. For dToF measurements beyond the water surface (which has an expected dToF value of 21 clock cycles) in Figure 5.24, the histogram's expected second peak cannot be clearly distinguished. The absence of this second peak indicates that natural water environments significantly affect the SPAD array flash LiDAR signals, complicating accurate target reconstruction at deeper depths.

5.6.4 In-Air Photon Noise

In all the histograms of dToF measurements discussed in this section, there is consistently a non-trivial number of detected photons with dToF measurements preceding the expected dToF value of the water surface. This is illustrated further in Table 5.7, where the histogram of dToF measurements of a target pixel (marked by a red asterisk in the reconstructed image shown in the first column) is presented. Despite the data being collected in shade, the outdoor weather was sunny at the time of data collection, resulting in significant ambient noise from daylight. This persistent noise contributes to the SPAD array flash LiDAR system detecting photons with dToF measurements shorter than the water surface's expected dToF value, complicating the accurate reconstruction and interpretation of the target's depth for future image-reconstruction algorithms.

Table 5.7: The histogram averaging algorithm image of targets at different depths and the histogram of dToF measurements at the pixel marked with red asterisk and arrow in the image. The green line is the reconstructed dToF value of the pixel in the reconstructed image. The red vertical lines in the histogram depict the post-processing range gate used before reconstruction is applied.

Depth (m)	**Histogram Averaging image with red-marked pixel**	**Histogram of dToF measurements of red-marked pixel**
0.5		

1.0		
1.5		
2.0		

In Table 5.7, it is evident that all histograms exhibit one or multiple small peaks at dToF measurements shorter than the expected dToF value for the water surface. While the distribution of dToF measurements for ambient light is theoretically expected to be uniform, these peaks display a cyclic pattern in relation to dToF measurements. Specifically, the photon counts drop to fewer than 2 counts every 5–7 clock cycles.

This cyclic behaviour is identified as an artefact of an internal electronics error in the SPAD camera. Such errors can introduce periodic fluctuations in the photon count data, affecting the accuracy of depth estimation and further complicating the analysis of dToF measurements for image reconstruction. Addressing these artefacts is crucial in the future for improving the reliability of the reconstruction algorithm and ensuring accurate target localisation in future imaging scenarios.

5.6.5 Global Target Absolute Localisation Accuracy

The global target absolute localisation accuracy of the histogram averaging algorithm is assessed using reconstructed images that contain a discernible representation of the target, which are the reconstructed images from Table 5.5 in Section 5.6 of targets submerged to a depth 2.0 m. To assess the global target absolute localisation accuracy, the global target location deduced from the reconstructed image is compared with the target's expected dToF value. For targets that are non-planar objects, the expected dToF value will be different for each target pixel. However, as the imaged target at the Port Adelaide River is a planar object, the expected dToF values of the target pixels are all the same.

As discussed in Section 5.6.2, the reconstructed dToF values of the target pixels are inconsistent. Therefore, to evaluate the absolute localisation accuracy of the global target, the median of the target pixels' reconstructed dToF values are analysed for each target depth's reconstructed image. Since the target is clearly visible in the reconstructed images, identifying which pixels correspond to the target is straightforward. Table 5.8 provides the result below.

Table 5.8: Comparing the median and expected dToF value of the target pixels

Depth (m)	**Expected dToF values (clock cycles)**	**Median of the reconstructed dToF value (clock cycles)**
0.5	18	18
1	24	24
1.5	26	25
2	28	27

All median values in Table 5.8 are within one clock cycle of the expected dToF value. Deviations in the median dToF values are noted for the reconstructed images of targets at the two greatest depths. This discrepancy may be due to the use of a longer rope for suspending the target at lower depths, which causes more swaying in the water. This issue introduced by the swaying rope will be discussed in Section 5.7.1.

It can be concluded that absolute localisation accuracy with the histogram averaging algorithm is high, with the target pixels' reconstructed dToF values within one clock cycle of expected dToF values.

Obtaining reliable data is another critical aspect to reconstructing a discernible image of a submerged target. Ensuring that a submerged target is within the SPAD

array flash LiDAR's FOV can be complex. The associated challenges and potential improvements to the experimental set-up are discussed in the next section.

5.7 Imaging Set-up Difficulties at Outdoor Natural Water Locations

As discussed in the introduction of this chapter, developing an effective reconstruction algorithm is not the only challenge in producing a discernible image of a submerged target. Collecting reliable data of a submerged target using a SPAD array flash LiDAR system presents its own set of difficulties. Specifically, several insights and lessons about imaging in outdoor natural water locations are learnt from the experience of collecting data using the SPAD array flash LiDAR system at the Port Adelaide River.

Three distinct challenges were encountered when conducting SPAD array flash LiDAR imaging at Port Adelaide River for data collection, each of which will be addressed in the following subsections.

Firstly, the water current caused the target to move during imaging, complicating the task of capturing a large set of frames with the target stationary at a specific position. This issue is discussed in Section 5.7.1. To mitigate this problem, data used for the reconstructed images in Table 5.5 of Section 5.6 were collected from the side of the jetty where the water is calmer.

Secondly, reflections from sunlight on the water surface posed difficulties in imaging under natural light conditions, as detailed in Section 5.7.2. To address this, data collection at Port Adelaide River was conducted in the shaded area of the jetty to prevent sunlight from degrading the quality of the reconstructed images.

Finally, locating the target by eye became increasingly challenging once it was lowered past the SD depth. The SD depth, as discussed in Section 3.6.2 of Chapter 3, represents the maximum distance at which a human eye can still observe a target in natural water. For the data collected in this chapter, the river water at the imaging location had an SD depth of 3.4 m at the time of data collection. Consequently, it was only possible to collect data of the target only submerged to 3 m during data collection, as locating the target at 3.5 m and beyond proved difficult. Section 5.7.3 explores the challenges of locating targets past the SD depth, the associated problems, and potential solutions.

The section concludes with Section 5.7.4, which discusses the use of a self-propelled moving target as a potential solution for future outdoor imaging in natural waters. This approach explains how this choice somewhat counter-intuitively reduces the challenges described in the previous subsections.

5.7.1 Water Current

The water current in the river causes the target to move, making it challenging to capture stable images of the target in a fixed position. At Port Adelaide River, this

issue is partially mitigated by selecting an area with calmer waters. For future imaging efforts, suspending a weight directly below the target on the rope could help stabilise the target in the water, allowing for imaging in areas with stronger currents. However, this solution is effective only for targets at shallow depths. As the imaging depth increases, so does the length of the rope needed to suspend the target, which can exacerbate movement due to the greater surface area of the rope affected by the current.

Several alternative solutions can address this issue, with varying effectiveness depending on the natural water environment. One option is to anchor the target to the riverbed. While this method could be effective, it may require significant logistical planning to transport and deploy a heavy anchor, and it may be impractical if the riverbed is too deep. Another approach is to replace the rope with a rigid structure to suspend the target. This can reduce movement caused by currents, but introduces its own challenges, such as adding artefacts to the reconstructed images. These artefacts would necessitate additional processing to mask or filter them out before applying image-reconstruction algorithms. Additionally, depending on its design, a rigid structure might complicate handling and increase the complexity of the imaging set-up.

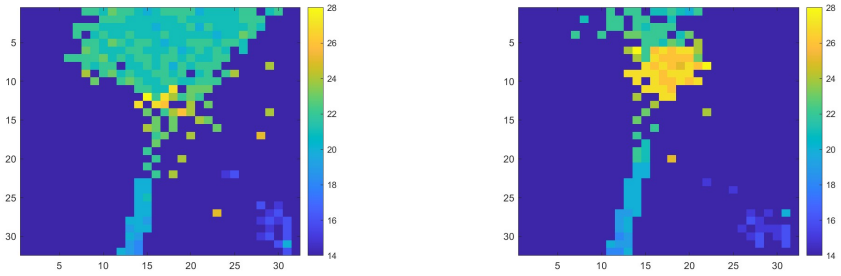
Finally, using a self-propelled moving target is a potential solution for greater depths; this will be discussed in detail in Section 5.7.4. This approach could mitigate the challenges associated with current-induced target movement and offer a practical alternative for imaging in natural waters.

5.7.2 Sunlight Reflection on Water Surface

One major challenge when imaging outdoors into the river is the reflection of sunlight on the water surface. This issue is apparent from the images reconstructed from data collected on the sunny side of the jetty at Port Adelaide River, which are shown in Figure 5.25. The imaging set-up used to collect this data is the same as the set-up used for collecting data at the Port Adelaide River presented in this chapter (as described in Section 5.2.4). These reconstructed images in Figure 5.25 illustrate this problem with sunlight reflection.

The reconstructed images in Figure 5.25 are created from a batch of 300 frames using the histogram averaging algorithm, which is the same algorithm identified in Section 5.6 to provide better reconstructed images for the Port Adelaide dataset, which was collected in the shade (see the image in the right-most column in Table 5.5 in Section 5.6). These images show similar features to the reconstructed images from data collected in the shade, with the only notable difference being the presence of a patch of light green pixels at the top of the images.

The light green patch of pixels at the top of the two reconstructed images in Figure 5.25 results from detected photons originating from sunlight reflecting off the water surface, which is close to the water surface's expected dToF value of 21 clock cycles. In contrast, the target submerged at 1.5 m appears as a yellow circle in these images, corresponding to its expected dToF value of 26 clock cycles.



(a) (b)

Figure 5.25: Histogram averaging images of the target submerged at 1.5 m depth (equivalent to an expected dToF value of 26 clock cycles from the SPAD array flash LiDAR system) in the Port Adelaide River under bright sunlight. The camera angle is at 63° in (a) and a slightly larger angle than 63° is used in (b).

The difference between these two images is that Figure 5.25(b) is collected with the SPAD array flash LiDAR system adjusted to a larger camera angle than the 63° camera angle used to collect the data that is used to reconstruct the image in Figure 5.25(a). This adjustment in camera angle is made empirically, by using the preliminary reconstructed image of the target that is available in real-time on the DSTG-developed GUI during data collection, as outlined in Section 5.2.5. The camera angle adjustment is minor, as confirmed by the two reconstructed images in Figure 5.25 showing the same reconstructed dToF values for the target and background pixels in the new image.

The change in angle for Figure 5.25(b) minimised the impact of sunlight reflection on the image reconstruction of the target. This is a significant improvement when compared with Figure 5.25(b), where the photon returns from the sunlight prevented the SPAD array flash LiDAR system from detecting photons reflected from the target. The sunlight prevents detections of photons reflected from the target because sunlight reflection off the water surface is closer to the SPAD camera than the submerged target, and the SPAD camera detects only the first photon return within a LiDAR cycle.

5.7.3 Imaging at Depths Greater than the Natural Water's SD Depth

Reconstructing a discernible image of a target submerged at greater depths in natural waters presents significant challenges, as discussed in Section 5.6.3. It is equally challenging to collect reliable data using a SPAD array flash LiDAR system at these depths. Specifically, data collection becomes notably difficult when the target is lowered past the SD depth of natural waters. As detailed in Section 3.6.2 of Chapter 3, the SD depth represents the maximum distance at which a human eye can still perceive a submerged object.

Once the target is beyond the SD depth, locating it visually becomes extremely

challenging. Even with a weight to enhance the target's stability (as discussed in Section 5.7.1), maintaining the target's precise position remains difficult. These factors collectively contribute to the difficulty in confirming the target's exact location.

Uncertainty about the target's location leads to several issues. First, it becomes challenging to ensure that the target is within the SPAD camera's FOV. Consequently, the data collected cannot be reliably used to assess the effectiveness of the reconstruction algorithm. This uncertainty complicates the diagnosis of reconstruction failures, which could stem from various factors, such as the target being outside the FOV, the SPAD camera failing to detect photon returns from the target, deficiencies in the reconstruction algorithm itself, or other environmental influences.

Another issue arising from not knowing the target's location is the difficulty in determining the expected dToF value of the target, making it impossible to validate the reconstructed dToF value.

Potential improvements for locating the target include attaching a laser to the target to project a non-destructive beam visible as a spot in a raw image, or as a spot in the preliminary reconstructed images displayed on the DSTG-developed GUI discussed in Section 5.2.5. Additionally, using a rigid structure to replace the rope for suspending the target, as suggested in Section 5.7.1, may help counteract movement caused by water currents. Alternatively, employing a self-propelled moving target, as discussed in Section 5.7.4, offers another potential solution.

5.7.4 Use of a Moving Target

Another approach to address the challenge of keeping the target stationary in the water is to use a self-propelled moving target for imaging. This target can follow a predetermined path, which simplifies tracking and validation. A Remotely Operated Vehicle (ROV) is a suitable candidate for this purpose, as it is equipped with an internal system capable of countering water currents and maintaining a planned trajectory. This method also ensures that the target's location is known at each time point, facilitating validation of its presence in the reconstructed images.

This approach effectively addresses the issues highlighted in Section 5.7.1, where maintaining a stationary target at deeper depths proves difficult. It also resolves the problem of locating the target beyond the SD depth, as discussed in Section 5.7.3.

5.8 Chapter Summary

This chapter examined image reconstruction of data collected by an above-water SPAD array flash LiDAR system of submerged single targets in natural waters. The performance of two basic image-reconstruction algorithms are presented: the detection threshold and median algorithm, and the histogram averaging algorithm. Their computational demand analysis demonstrates they are real-time-capable, where the analysis highlights they consists of small number of steps and even at computationally intensive steps, they are processing a small amount of data. While the basic computational methods enable these algorithms to be real-time capable, they also

inhibit the image reconstruction performance of these algorithms. The image reconstruction performance of these algorithms are evaluated against three datasets collected at three different water locations, where the algorithms' image reconstruction performance worsen for deeper depths or higher number of ALs in each dataset.

The first dataset was collected at the AIMS indoor saltwater tank, which simulated the imaging of submerged targets at different turbidity conditions without the interference of the water surface. The detection threshold and median algorithm proved effective for reconstructing visually discernible images of the imaged target for water turbidity conditions of up to 3.43 ALs, which is created by a mixture of 2.44 µg/L chlorophyll and 5.71 mg/L sediments. This demonstrates that these images can be used to conduct visual target detection. However, it failed at higher sediment levels (8.62 mg/L, equivalent to 4.28 ALs).

The reconstruction performance of the detection threshold and median algorithm is quantified in term of two different metrics. The pixel-wise absolute localisation accuracy is 1–2 clock cycles. Despite the lower absolute localisation accuracy, the relative localisation accuracy of the reconstruction algorithm is found to be one clock cycle.

The evaluation of target classification using the reconstructed images of different shapes submerged in sediment-only conditions showed no trend in overall classification accuracy with increasing sediment concentration, except for a significant dip in accuracy at the highest concentration (5.62 mg/L, 3.29 ALs), although overall accuracy remained above 95%. A computational demand analysis of the target classification algorithm demonstrates that it is real-time capable.

The second dataset was collected at the DSTG indoor freshwater tank. The imaging set-up for collecting this dataset used a novel imaging set-up for above-water SPAD array flash LiDAR imaging of submerged targets. A custom mechanical rig built for positioning the SPAD array flash LiDAR system above the water surface is used in this set-up.

For reconstruction performance of this second dataset, both reconstruction algorithms can only reconstruct the white quadrant of the SD, which is the imaging target. This is attributed to the difference in the amount of reflected photon between the black and white quadrants.

The detection threshold and median algorithm cannot reconstruct a discernible image of the target for a clear freshwater depth of more than 1 m. This is surprising given this algorithm is able to reconstruct up to a turbidity level of 3.43 ALs at the AIMS indoor saltwater tank, where the portion of the imaging distance between the target and imaging system that is in the water environment is approximately 2.87 m. This inconsistency highlights a need for further study into the effect of water quality on SPAD array flash LiDAR systems.

The histogram averaging algorithm outperformed the detection threshold and median algorithm in reconstructing discernible images of targets submerged to 4.5 m depth. At a submerged depth of 5 m, the target is detectable in the histogram averaging algorithm's reconstructed images. However, the SD features are not discernible in the reconstructed image, making it difficult to extend the application of this image

for target classification in the future. A comparison between the reconstructed and expected dToF values demonstrates that the histogram averaging algorithm captures the trend of the target imaged at increasing depths, but the reconstructed dToF values contain a systematic error.

The third dataset uses the same mechanical rig and similar imaging set-up in the DSTG indoor freshwater tank to image targets submerged in the Port Adelaide River, where turbidity levels were at 3.4 m SD depth, at the time of data collection. The target in the reconstructed images by the histogram averaging algorithm is discernible, for data collected with the target submerged down to 2 m depth in natural water. This demonstrates that target detection is possible.

For the reconstructed images from the third dataset, an analysis of the reconstruction performance in terms of relative localisation accuracy is 5 clock cycles. Further investigations are required to determine why the range of reconstructed dToF values for the targets pixels is so large despite the imaging target being planar, where the expected dToF values should be the same for all target pixels. Global target absolute localisation accuracy with the histogram averaging algorithm is one clock cycle.

In data collected from the target at 2.5 m and 3.0 m, the target cannot be seen in the images reconstructed by the histogram averaging algorithm. A closer inspection of the collected dToF measurements shows no photon returns from the target are detected when the target is submerged at these depths. This is indicated by the absence of a secondary histogram peak, which can be observed from histograms of dToF measurements collected of the target submerged at similar depths at the DSTG indoor freshwater tank. The insufficient number of detected photons reflected from the target may be attributed to the complex nature of natural waters. This indicates that a more advanced image-reconstruction algorithm is needed to reconstruct a discernible image of the target at these greater depths in natural water location. In order to develop a real-time capable algorithm that can reconstruct images at greater depths or higher number of ALs, it will require a further study of the complex and variable nature of natural waters.

Studying the relationship between natural water properties and their Inherent Optical Properties (IOPs) is one approach to understanding the complex and variable nature of natural waters. Understanding these complex phenomena and their effects on SPAD LiDAR data will enable more sophisticated computational methods to be developed, where these methods may include lightweight neural or FPGA-based methods. For example, by modelling the effect of these properties on dToF measurement distributions, it may be possible to utilise lightweight computational methods to filter out noises in the SPAD LiDAR data caused by the effects of natural water; this may result unveiling the histogram peak that corresponds to the target's location. Preliminary research on the impact of natural water properties on IOPs is available in [101], although it is not included in this thesis.

Obtaining reliable data of submerged targets with a SPAD array flash LiDAR system is another critical aspect to developing better image reconstruction algorithms. Challenges in data collection and imaging in natural waters are identified from the experience of conducting imaging at the Port Adelaide River for creating the third

dataset. Challenges include target movement due to currents and sunlight interference. Future improvements for the imaging set-up could include the use of an ROV to move in a pre-determined path for more accurate imaging and target localisation below the SD depth.

Other than natural water environments, another common imaging environment during remote surveys is outdoor fog. The next chapter investigates image reconstruction for this type of imaging condition, where the histogram of dToF measurements present a similar two-peak characteristic as seen at the DSTG indoor freshwater tank.

Chapter 6

SPAD Image Reconstruction of Targets Obscured by Fog

Fog is another obscurant commonly encountered during remote surveillance by Uncrewed Autonomous Vehicles (UAVs). It creates a challenging imaging environment for Single Photon Avalanche Diode (SPAD) array flash Light Detection and Ranging (LiDAR) systems. Fog's light-scattering properties result in the majority of detected photons being noise rather than reflected from the target [183]. This makes it difficult to accurately reconstruct an image of the target, leading to poor target-discrimination performance. This may be detrimental for military operations if the target-discrimination information is incorrect.

This chapter focuses on developing image-reconstruction algorithms for SPAD array flash LiDAR data collected of a single-target imaged in fog. By evaluating the quality of these reconstructed images, it can be determined whether they can be used for target detection. The first two algorithms presented in this chapter also have their reconstructed images evaluated for target classification. These algorithms employ simple computational operations on restricted sets of data (500–1000 for these algorithms), suggesting they have the potential for real-time capability, even though their execution times are not measured. In this chapter, a computational demand analysis accompanies the description of each algorithm to demonstrate real-time capability.

The first approach is called the range gate and process algorithm. For this approach, a range gate is set on the SPAD array flash LiDAR system when imaging targets through fog. This enables a lot of the backscattered photons to not be detected. The image is reconstructed by using the direct Time-of-Flight (dToF) value associated with the peak of the histogram of dToF measurements as the reconstructed dToF value. The image quality of the reconstructed image is denoised by applying area correlation and threshold methods.

To demonstrate that the reconstructed images from the range gate and process algorithm can be classified, dToF measurements of two different wooden frigate silhouettes are collected using a SPAD array flash LiDAR system. The target is classified by determining which frigate class's mean area it is closest to. This area is defined as the total number of non-zero pixels in the reconstructed image. A portion of the collected images is used as training data to determine each class's mean area.

The imaging environment for the collected data of these frigate silhouettes has the fog concentrated at a short distance span between the SPAD array flash LiDAR system and the target, rather than having the fog distributed throughout the entire span between the SPAD array flash LiDAR system and the target. Combined with using a range gate around the known target location for data collection, this approach is not robust enough for real fog environments where there is a wider fog span. In real fog, range-gated data may require additional processing to create an accurately reconstructed image of an obscured target.

With robustness in mind, the second approach investigates modelling dToF measurements that are collected without a range gate. It employs a finite mixture model, which is a probability distribution that consists of a finite mixture of different probability distribution models, for depth estimation in each pixel of the reconstructed image. This approach is referred to as the finite mixture model algorithm in this thesis. It aims to model the entire range of possible dToF measurements of the SPAD array flash LiDAR system, taking into account the maximum possible distance of the imaging scene.

The finite mixture model algorithm is inspired by [139], which demonstrates that dToF measurements of detected photons reflected from fog scattering can be modelled by a statistical distribution such as a gamma distribution. The authors in [139] successfully reconstructed images by determining each of the pixel's reconstructed dToF value from the residual probability distribution between the probability distribution modelling all of the pixel's dToF measurements and the fitted gamma distribution to the same set of dToF measurements. Inferring the dToF value of each pixel in a reconstructed image from the residual probability distribution is also commonly employed in other literature [86, 181, 183, 182, 137]. Section 2.3 of Chapter 2 discusses this and the paper [139] in more detail.

Compared with the proposed algorithm in [139], the finite mixture model algorithm presented in this chapter differs in several ways. Most importantly, it does not determine each pixel's reconstructed dToF value from the residual probability distribution. Instead, the approach determines each pixel's reconstructed dToF value from a finite mixture model fitted to a set of dToF measurements. This means it models the dToF measurements of detected photons reflected from the fog and target concurrently, removing the need to compute a residual probability distribution. The reconstructed dToF value of each pixel can be inferred after one single probability distribution fit of the dToF measurements.

The finite mixture model consists of lognormal and Gaussian probability distributions. The model assumes the dToF measurements of detected photons reflected from fog can be modelled by one or more lognormal distributions while the dToF measurements of detected photons reflected from the target can be modelled by a Gaussian distribution. The target's localised distance is defined to be the mean of the Gaussian component, as in [139]. However, the finite mixture model algorithm reported here uses a lognormal distribution instead of a gamma distribution (which is used in [139]) to model the dToF measurements of detected photons reflected from fog. This is because the finite mixture model algorithm approaches the fog obscurant

problem from a phenomenological approach instead of physics modelling or convolution approaches. The lognormal is chosen because its trend shape is observed to be a sufficient match to the distribution of the dToF measurements in a histogram. This makes it a phenomenological approach, while a gamma distribution has been demonstrated to be derived from a physics model of multiple scattering events. The authors in [139] derive the gamma distribution model from assuming that the distance between the consecutive scattering events of a photon follows an exponential distribution.

The Expectation-Maximization (EM) algorithm [41] is employed in the finite mixture model algorithm to fit the probability distributions to the measured dToF measurements of each pixel over a batch of frames. It is an iterative algorithm that can approximate parameters of the probability distribution. It is chosen over more common methods such as Maximum Likelihood Estimation (MLE) because it is often difficult to find an analytical solution for these parameters [23, 35]. In addition, the EM algorithm is known to be short in computation time. In the SPAD array flash LiDAR community, only the authors in [137] have employed the EM algorithm in their image-reconstruction algorithm. As discussed in Section 2.3 of Chapter 2, the authors in [137] use EM algorithm to estimate parameters of the fog model's probability distribution, which models the dToF measurements reflected from fog using a Poisson distribution with its mean parameter dependent on a unit step function.

To demonstrate the image reconstruction performance of this finite mixture model algorithm, it is tested on a SPAD LiDAR dataset of three planar shapes (circle, triangle, and square) obscured by fog, collected by a SPAD array flash LiDAR system. The histogram of dToF measurements in this collected dataset shares some common attributes with dToF measurements collected in the DSTG indoor freshwater tank, as seen in Section 5.5.2 of Chapter 5. In particular, all the histograms of dToF measurements have two peaks, where the first one represents detected photons reflected from obscurant scattering and the second is from the target. This means that the finite mixture model algorithm can be potentially extended to reconstruct images of single targets submerged in water.

Target classification is demonstrated with the reconstructed images from this finite mixture model algorithm. The processing steps for target classification converts the reconstructed image into a binary image using a threshold method, then apply a few image processing techniques to improve the quality of the reconstructed image for classification. Afterwards, the total number of non-zero pixels are counted to determine the class of shape.

The reconstructed images of the shapes are only discernible and classifiable for collected data with a fog level of a maximum of 50.8 m visibility/1.14 attenuation lengths (ALs) (in terms of 532 nm). With the fog level increasing, the amount of detected photons reflected from the target becomes significantly less than from the fog. This is where the finite mixture model algorithm is weak; it is unable to fit a Gaussian distribution accurately to the dToF measurements related to the target once this occurs. Therefore, an alternative algorithm needs to be explored. This leads to the Fourier descriptor algorithm, which utilises spatial correlation between pixels at

each possible dToF measurement to perform global target localisation, which leads to image reconstruction. The algorithm is able to perform global target localisation even when there is a small number of detected photons reflected from the target.

The Fourier descriptor algorithm differs conceptually from the previous two approaches discussed in this chapter, offering a new perspective on image reconstruction. Instead of reconstructing an image on a pixel-by-pixel basis, this algorithm reconstructs all pixels simultaneously. Rather than processing the dToF measurements at each pixel over a batch of frames, it processes the collected data by creating a binary image for each possible value in the range of dToF measurements from a batch of frames. An image related to photon counts of one of the possible values in the range of dToF measurements is referred to as a slice when discussing this algorithm. When constructing a binary image for each possible dToF value's slice, each pixel's value is assigned a value of 1 if one or more photons are detected at this specific dToF value within the current batch of frames. Otherwise, it has a value of 0. The resulting set of slices is referred to as a binary histogram stack in this algorithm.

The Fourier descriptor algorithm reconstructs all pixels of an image at once by determining which slice of the binary histogram stack contains the closest representation of the target's silhouette. This is conducted by utilising Fourier descriptors, which are able to provide a unique signature for each different object's silhouette. Fourier descriptors can also be extended to be size- and rotation-invariant. Fourier descriptors are calculated for each slice in order to determine which slice contains the most similar silhouette to the target.

To ensure accurate Fourier descriptors are computed for each slice, noise pixels are removed and holes are filled for each slice of the binary histogram stack. An additional check in apparent target size is also employed in this algorithm to filter out noise pixels before Fourier descriptors are calculated for each slice.

Given that the imaging target used to evaluate this algorithm is planar, its expected dToF value should be the same for all pixels at the target's spatial position in the image. This means the target should only locate in one slice of the binary histogram stack. Therefore, for this chapter's implementation of the Fourier descriptor algorithm, the target's location is selected as the dToF value corresponding to the slice with the descriptor most similar to the target. Moreover, image reconstruction can be performed by simply assigning the chosen slice's dToF value to pixels where the slice's binary image has the value 1, and assign the remaining pixels' values as undefined.

The SPAD LiDAR dataset used to evaluate the finite mixture model algorithm is also used to test the Fourier descriptor algorithm. In this instance, only the data related to the planar triangle target is used for evaluating the Fourier descriptor algorithm. Promising preliminary results show the algorithm can perform absolute localisation of the triangle target at a fog level of 2.28 ALs and visibility of 25.4 m.

The motivation for using Fourier descriptors comes from the observation that the target remains consistently discernible in the finite mixture model algorithm's reconstructed images, even when some images exhibit poor reconstruction performance in terms of relative localisation accuracy. The target is discernible in the image because

its silhouette is clearly visible and resembles that of the actual target. Therefore, the Fourier descriptor aims to model this silhouette in order to enhance overall reconstruction performance.

From the discussion in Section 2.3 of Chapter 2, it is clear that none of the existing literature has approached reconstruction of fog-obscured targets using image features such as Fourier descriptors. The closest approach to image features that has been implemented is the use of spatial correlation techniques to reconstruct the image. For example, the authors in [60] encoded spatial correlation information between pixels in the regularisation terms used for image reconstruction. Another similar approach to image features is to perform denoising on the reconstructed image, such as the authors in [127] using inpainting.

As the Fourier descriptor algorithm relies on a clear silhouette of the target to perform localisation, this algorithm is unable to localise a target that is occluded by other objects. This is because the target's whole silhouette will not be fully visible in the image. Therefore, the Fourier descriptor algorithm's localisation performance will be poor for targets in dense clutter. To use the Fourier descriptor algorithm to localise an occluded target, an additional inference method needs to be applied beforehand to extrapolate from an image of the target's occluded silhouette to create an image with the target's complete silhouette. This will then allow the Fourier descriptor algorithm to be applied for localisation.

Target classification has not been demonstrated for the reconstructed images by the Fourier descriptor algorithm. Nonetheless, the algorithm can be extended for classification in the future. As Fourier descriptors are unique for different shapes, Fourier descriptors of different object classes can be calculated beforehand. During classification, the target can be classified by determining which object class has the most similar Fourier descriptor to the target's Fourier descriptor. For real-time implementation, a lookup table of different targets' Fourier descriptors can be implemented for fast real-time classification on-board mobile platforms. Once the algorithm is extended for classification, it can also be extended to localise multiple different targets with different shapes at different depths. The algorithm can be utilised to localise different Fourier descriptors that correspond to different targets.

In the future, the Fourier descriptor algorithm can be extended to image reconstruct and classify targets that are non-planar objects. From the perspective of the binary histogram stack, a non-planar object will appear over a consecutive series of slices. Therefore, a non-planar object will contain a sequence of Fourier descriptor instead of a planar object having a single Fourier descriptor. For target localisation, the Fourier descriptor algorithm needs to be extended to search for a series of consecutive slices that contains a particular sequence of Fourier descriptors. Similarly, classification of targets that are non-planar objects will involve determining which object class's sequence of Fourier descriptors matches best with the target's sequence of Fourier descriptors. Image reconstruction of a target that's a non-planar object will not be just from a slice of the binary histogram stack. Instead, image reconstruction will involve combining the binary images of all the histogram slices that are associated with the target, where a pixel's reconstructed dToF value is the dToF value of

the first slice in the identified sequence of slices that contain the value 1 at the same pixel position. Otherwise, the pixel stays undefined.

The remainder of the chapter is organised as follows. The imaging process using the SPAD array flash LiDAR to collect two distinct datasets, namely the Frigates Fog Dataset FFD and Shapes Fog Dataset (SFD), is detailed in Section 6.1. This section provides details about the imaging conditions and imaging targets captured in the datasets.

Moving on to the presented methodologies, Section 6.2 outlines the range gate and process algorithm. It encompasses image reconstruction techniques, denoising processes, computational demand analysis and the classification approach employed. Additionally, the robustness of using range-gated data is explored in Section 6.2.3.

Following this, Section 6.3 presents the finite mixture model algorithm. It begins with a description of the finite mixture model in Section 6.3.1, followed by an in-depth mathematical explanation of the application of the EM algorithm in Section 6.3.1.1, with a presentation of the EM algorithm's convergence condition in Section 6.3.1.2, and then the finite mixture model algorithm's initialisation process in Section 6.3.1.3. The target classification algorithm for the reconstructed images from this algorithm is discussed in Section 6.3.2, along with the efficacy of employing a pixel-wise algorithm in Section 6.3.4. The real-time capability of the finite mixture model algorithm and the classification algorithm are then discussed in Section 6.3.3, where their computational demand analyses are presented.

Finally, Section 6.4 presents the Fourier descriptor algorithm, which utilises the Fourier descriptor, an image feature, for target localisation. The computational demand analysis for this algorithm is presented in Section 6.4.2.2, which discusses the algorithm's real-time capability.

Transitioning to the results and discussion in Section 6.5, the performance of each algorithm is evaluated. The range gate and process algorithm's different performance outcomes are presented in Section 6.5.1, which includes image reconstruction results in Section 6.5.1.1, with subsequent target classification results detailed in Section 6.5.1.2. The performance of the finite mixture model algorithm is then discussed in Section 6.5.2, which is in terms of different performance metrics in its subsections. Finally, a comparison of the global target locations extracted by the Fourier descriptor algorithm and the finite mixture model algorithm is provided in Section 6.5.3, which demonstrates the Fourier descriptor algorithm's improvement in absolute localisation accuracy for the global target location.

The work presented in this chapter has been published in [100] (for the range gate and process algorithm), [99] (for the finite mixture model algorithm) and [104] (for the Fourier descriptor algorithm).

6.1 SPAD Data Collection Method

The algorithms developed in this chapter are evaluated using SPAD array flash LiDAR data collected in a long indoor dark tunnel. The technical specifications of

the SPAD array flash LiDAR system used are detailed under System C in Table 3.1 in Section 3.4 of Chapter 3.

Two separate datasets are collected and used to evaluate different algorithms. One dataset imaged two different white wooden frigate silhouettes through fog, while the other captured cardboard cutouts of three simple geometric shapes (circle, triangle, and square) through fog. In this thesis, the frigate silhouette dataset will be referred to as the Frigates Fog Dataset (FFD), and the geometric shapes dataset will be referred to as Shapes Fog Dataset (SFD). The FFD is used for evaluating the performance of the range gate and process algorithm, while the SFD is used to evaluate the finite mixture model algorithm, with only the triangle data of the SFD being used for testing the Fourier descriptor algorithm.

For FFD, a range gate is set on the SPAD array flash LiDAR system for 69 to 110 clock cycles. Unlike FFD, a range gate is not employed to collect data for the SFD dataset. The SPAD array flash LiDAR system is set to have a minimum delay of 10 clock cycles (equivalent to a target distance of 5 m) to avoid scattering directly from the laser while still capturing as much of the fog in the tunnel as possible. For both datasets, if a pixel registers no photon return, its dToF measurement is recorded as the maximum clock cycle value, which is 110 clock cycles for the SPAD array flash LiDAR system used in the imaging set-up of both datasets. Therefore, any pixels with a dToF measurement of 110 clock cycles is processed as no photon detected.

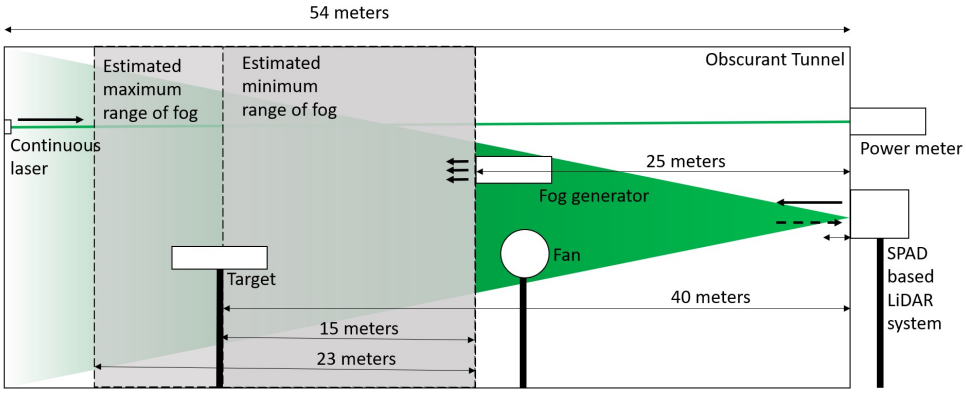


Figure 6.1: FFD's imaging set-up of tunnel for measurements through fog.

Other than the targets, the same equipment is used in the same locations for the imaging set-ups for data collection of the FFD and SFD. Figures 6.1 and 6.2 illustrate the equipment layout for the two datasets, respectively. The same 54 m-long tunnel is used for conducting imaging to collect data for both datasets. The Rave AF1214 Fog Machine and the Rave Heavy Fog water-based liquid (consisting of 42% Propylene glycol, <5% glycerine, and distilled water [9]) are used to generate fog to obscure the targets. The fog machine heated the water-based liquid and forced it into the atmosphere under pressure. The fog machine is positioned 25 m away from the LiDAR system. The SPAD array flash LiDAR system and power meter are situated in an approximately 3 m by 5.5 m room at the tunnel's entrance. A 532

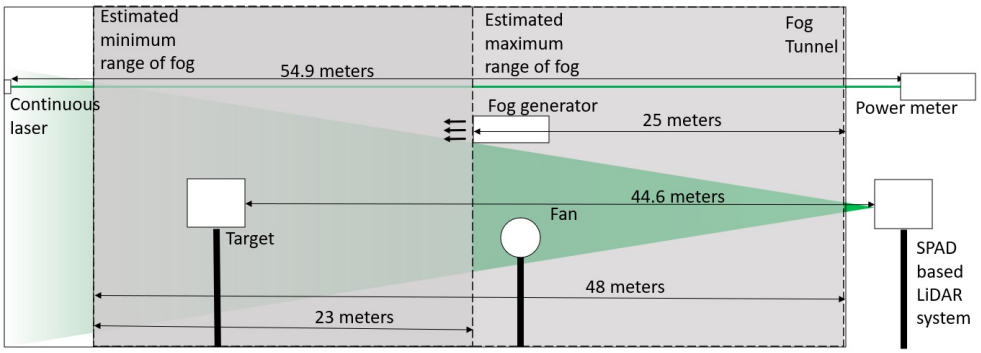


Figure 6.2: SFD's imaging set-up of tunnel for measurements through fog.

nm wavelength, 14 mW continuous wave laser is used and placed at the back of the tunnel, facing the power meter. The fog levels reported in this chapter are based on the received power measurements. The fog levels are reported in terms of visibility (cf. Section 3.6.3 of Chapter 3) and the number of ALs (cf. Section 3.6.1 of Chapter 3). Imaging is conducted in the dark to reduce environmental noise.

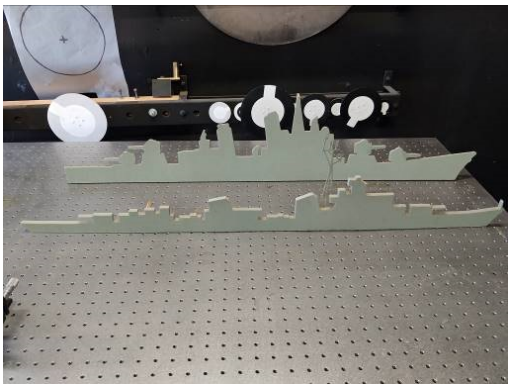


Figure 6.3: Cut-out silhouettes of two different frigates.

As mentioned earlier, the FFD and SFD contain data that are collected from imaging different targets. For FFD, Figure 6.3 displays the two different frigate silhouettes before they were painted white to use for imaging in data collection. One silhouette measures 120.5 cm in length, while the other measures 133 cm in length[1]. In contrast, for the SFD, white cardboard cut-outs of 3 different shapes (circle, triangle, and square) are used as targets, as depicted in Figure 6.4. These shapes are sized such that a 610 mm square could be circumscribed around them. The shapes are attached to 3 planks of long timber, and a pulley system is employed to raise these shapes into the field-of-view (FOV) of the SPAD array flash LiDAR system. This set-up enables the imaging of individual shapes with minimal interruption to the flow of fog in the tunnel, allowing the fog condition to remain as consistent as possible for all 3 shapes.

[1]An incorrect unit was noted in [100]. The correct length is in cm instead of m.

(a) Circle (b) Triangle (c) Square

Figure 6.4: Different shapes cut out from cardboard and painted white for imaging. These shapes are connected to a pulley mechanism that enable each shape to be lifted up from the floor for imaging. Figure 6.4(c) shows the design of this mechanism causes the triangle to be placed in front of the square when the square is imaged. This does not cause an issue as the triangle is directly in front of the square, so its tiny depth difference to the square is not detected by the SPAD array flash LiDAR system.

Figure 6.4(c) illustrates the design of this mechanism, which causes the triangle to be positioned in front of the square when the square is imaged. This arrangement does not pose an issue, as the triangle is situated right in front of the square, and its slight depth difference is not detected by the SPAD array flash LiDAR system used in this experimental set-up. At the time of data collection, additional fog was released only after all three shapes were imaged separately.

Another difference between the set-ups of the two datasets is the direction in which the fan is oriented, which influences the extent of fog dispersion. For FFD, the fan is directed perpendicular to the line of sight to the back of the tunnel, which helps keep the fog circulating in the narrower area of the tunnel. This can be seen in Figure 6.5 where the entrance to the tunnel does not have fog. At the other end of the tunnel, the fog is estimated to dissipate at around 15–23 m from the generator (as illustrated in Figure 6.1). For SFD, the fan is directed towards the rear of the tunnel. The fog is observed to dissipate at an approximate distance of 6 m from the back of the tunnel and spread up to the tunnel's entrance, as illustrated in Figure 6.2.

For FFD, data collection occurred at fog levels with visibilities ranging from 76.9 to 14.1 m, corresponding to 0.68 to 3.69 ALs. In contrast, for SFD, the fog levels spanned from 185 to 12 m in visibility, corresponding to 0.31 to 4.85 ALs.

As discussed in Chapter 3, visibility and the number of ALs are employed to quantify the fog levels that obscured the targets during imaging. Calculating these terms requires determining the beam attenuation coefficient, which is computed using Eq. 3.1 from Section 3.5 of Chapter 3. To obtain this coefficient value, the measured power of a continuous 532 nm laser beam across each fog level is used as the

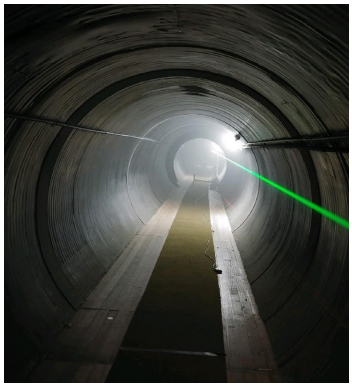


Figure 6.5: A photo of the long tunnel with lights on, with the testing target barely visible. Experiments are conducted in the dark. The fog level here is 25.8 m visibility and 2.25 ALs, which are calculated from laser power measurements of the continuous laser beam on the right.

term I in Eq. 3.1. This power is measured by the power meter at the tunnel entrance. The power of the continuous laser is used as the term I_0 in Eq. 3.1.

As discussed in Section 3.5.1, the term x in Eq. 3.1 is assigned the average distance of the fog span. For FFD, the fog is estimated to span between 15–23 m (as depicted in Figure 6.1). Thus, an average value of 19 m is used for x in Eq. 3.1. Conversely, for SFD, the fog span is estimated to span between 23–48 m, as shown in Figure 6.2. Consequently, an average value of 35.5 m is used for x in Eq. 3.1.

6.2 Reconstructing Range-Gated Data

The range gate and process algorithm is presented in the first subsection for reconstructing images, then it is followed by an explanation of the target classification algorithm used to evaluate the quality of the reconstructed images. Finally, a discussion of the robustness of this algorithm is provided in the final subsection.

The results from evaluating the reconstruction and classification performances of this algorithm are reported in Section 6.5.1.

6.2.1 Range Gate and Process Algorithm

The range gate and process algorithm comprises of three components. The first component is the collection of dToF measurements utilising a range gate. For the evaluation of this algorithm in reconstructing images of frigate silhouettes (i.e. FFD dataset), the dToF measurements are gathered with the range gate set to 69 to 110 clock cycles, as discussed in Section 6.1. Furthermore, as discussed in Section 6.1, in instances where a pixel records no photon return, its value is registered as the maximum clock cycle value, which is 110. Therefore, all pixels with a value of 110 clock cycles are excluded from the raw data before undergoing processing via the range gate and process algorithm.

The second component involves reconstructing the image. A pixel-wise method is employed, wherein each pixel's dToF measurements over a batch of frames are organised into a histogram, and the dToF value corresponding to the histogram peak is utilised as the reconstructed dToF value[2]. A batch of 500 frames is utilised for each image reconstruction.

The third component is denoising, which consists of two techniques that are applied to the reconstructed image to enhance its quality. The first technique is called area correlation, which involves comparing each pixel's value with its surrounding pixels. If a pixel is within 3 clock cycles of less than a certain number of its surrounding 3×3 pixels, it is discarded. This step aims to eliminate noisy background regions characterised by random pixel values. For the fog condition used to evaluate this algorithm (14.1 m visibility and 3.69 ALs), if there are less than 2 pixels that are within 3 clock cycles of a pixel, then the pixel is discarded.

After area correlation, a threshold value is employed as a second denoising technique to further filter out noise, resulting in a clearer image. This threshold is determined empirically from the reconstructed image at this stage, considering the difference between the expected dToF values of the target and the background.

6.2.1.1 Real-Time Capability

The execution time of the range gate and process algorithm is not measured because the implementation is not optimised. To demonstrate the real-time capability of this algorithm, an analysis of its operation is conducted. Figure 6.6 shows a summary of this algorithm.

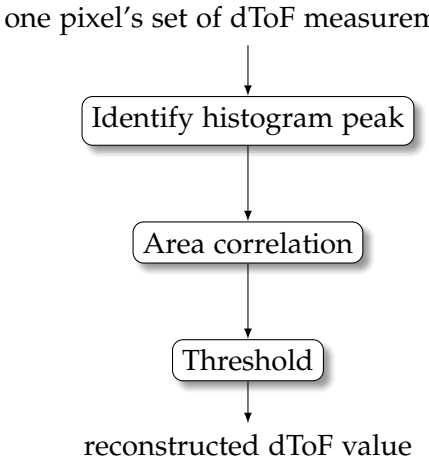


Figure 6.6: Summary of range gate and process algorithm

From Figure 6.6, it is clear the range gate and process algorithm have a small amount of steps when processing each pixel. The most computationally intensive

[2]To avoid misunderstanding, the histogram algorithm referred to in [100, Section IV] identifies the bin of the histogram peak, not including the subsequent application of a threshold, as in [100, Section III].

task is identifying the histogram peak for each of the pixels, which has been shown to be real-time capable for a small amount of data in Section 5.3.1 in Chapter 5. Here, the algorithm is only processing 500 frames and the number of bins in its histogram is 40, given the dToF measurements vary between 69 to 109 clock cycles (as discussed in Section 6.2.1).

The remaining tasks are simple in computation, and can be parallelised if needed to enable real-time execution. For speed up processing for all pixels, each pixel's processing can be implemented on a parallel thread, as argued for the algorithms in Section 5.3.1 in Chapter 5 which also processes each pixel's dToF measurements independently. Therefore, this demonstrates that the range gate and process algorithm is real-time capable for reconstructing an image.

6.2.2 Evaluating Reconstructed Images for Target Classification

To illustrate that the reconstructed image from this range gate and process algorithm can be classified, dToF measurements of two different white wooden frigate silhouettes are collected using a SPAD array flash LiDAR system (as described for FFD in Section 6.1). The target is classified by determining which frigate class's mean area it is closest to. The area of the silhouette is defined as the total number of pixels with non-zero values in the reconstructed image. All the pixels in the reconstructed image can be counted because the de-noising techniques have removed most of the noisy pixels, making the majority of non-zero values to be part of the silhouette shape in the image. Counting the number of non-zero pixels in a 32×32 is clearly real-time capable.

An additional class, called "Undecided", is created to account for scenarios where the area is equidistant to both frigates' mean areas. As outlined in Section 3.7 of Chapter 3, instances of the Undecided class are excluded from testing, serving solely to address scenarios where the classifier cannot assign an object class to the target based on predefined rules.

To determine the mean area for each frigate class for classification, training is conducted by calculating the mean area of each type of frigate silhouette from some of the frames in the FFD. For each fog condition, 100 reconstructed images of each frigate are generated for training, and the mean of each frigate's area is calculated and used for target classification. These 100 reconstructed images are generated from a sliding window approach, where each image is reconstructed from a batch of frames shifted by an index of one from the previous batch. To test the target classification performance, a different set of 100 reconstructed images of each frigate are utilised. Similarly, these images are also generated using the same sliding window approach for generating reconstructed images for training. However, the frames used for testing is different to the frames used for training.

6.2.3 Robustness of using Range Gated Data

This approach demonstrates the potential of utilising range gating for imaging targets obscured by fog, employing simple arithmetic techniques to reconstruct target images. However, when imaging unknown targets at unknown distances in unfamiliar environments, there is not sufficient time to query all possible distances via range gating to collect real-time target data. While this lack of speed may not pose a problem for stationary targets, it could prove problematic for fast-moving targets. A rapidly moving target may appear blurry if the SPAD array flash LiDAR system fails to query all sets of range gates quickly enough. To address this challenge, the upcoming section presents the finite mixture model algorithm, which focuses on reconstructing images of targets collected from SPAD array flash LiDAR systems that do not utilise a range gate.

6.3 Modelling dToF Measurements Collected without the Use of Range Gating

This section presents the finite mixture model algorithm and the target classification algorithm used to evaluate its reconstructed images. The goal of the finite mixture model algorithm is to reconstruct an image by determining the dToF value of each pixel independently. For each pixel, the finite mixture model algorithm utilises the EM algorithm[3] to fit a finite mixture model to the dToF measurements from a batch of frames.

The SFD is used to evaluate the quality and target classification performance of reconstructed images generated by the finite mixture model algorithm. For this dataset, the finite mixture model algorithm requires a batch of 500 frames to reconstruct one image.

Before fitting a finite mixture model to the dToF measurements from the SFD, any values outside the range of 0 to 100 clock cycles are discarded. This step aims to remove detected photons reflected from the back wall of the tunnel, as described in Section 6.1. The dToF measurements of detected photons reflected from the back wall appear as a spike when their dToF measurements are included in plotting the histogram of dToF measurements of all detected photons. Including this spike in the evaluation of the finite mixture model algorithm would necessitate more sophisticated modelling. To avoid this unnecessary complication, the dToF measurements of the detected photons reflected from the wall are excluded. The focus of this algorithm is on modelling the dToF measurements from both the fog and the target.

The finite mixture model is described in Section 6.3.1, followed by a mathematical explanation of applying the EM algorithm to fit the finite mixture model to the dToF measurements in Section 6.3.1.1. Finally, Section 6.3.2 details the processes used to

[3]There is an error in the reporting in [99] where Kernel Density Estimator (KDE) is not used as part of the finite mixture model algorithm, the EM algorithm fits the finite mixture model directly to the dToF measurements of each pixel.

conduct target classification on the reconstructed images by the finite mixture model algorithm.

The results from evaluating the reconstruction, classification and localisation performances of this algorithm are reported in Section 6.5.2.

6.3.1 Finite Mixture Model Algorithm

The finite mixture model is a probability distribution that consists of one or two lognormal distributions (depending on the fog level) and one Gaussian distribution. The lognormal distributions model the dToF measurements of detected photons reflected from the fog, while the Gaussian distribution models the dToF measurements of detected photons reflected from the target. The distance to the target is determined as the mean of the Gaussian distribution. For pixels that do not contain the target, the Gaussian distribution erroneously fit some of the dToF measurements of detected photons reflected from the fog, with the lognormal distribution fitting the remaining dToF measurements. Consequently, the reconstructed dToF values from the background pixels are less consistent compared to those from the target pixels, which are pixels at the expected spatial position of the target in the image.

In reconstructing images from the dToF measurements collected in the SFD, the Gaussian distribution is assumed to follow the lognormal distributions. This assumption is incorporated into formulating the EM algorithm to fit the finite mixture model to these dToF measurements. It is based on the experiment set-up, where the target is positioned at the furthest point. An illustration of this arrangement is provided in Figure 6.7, where it shows the Gaussian distribution is at a later clock cycle value than the lognormal distribution.

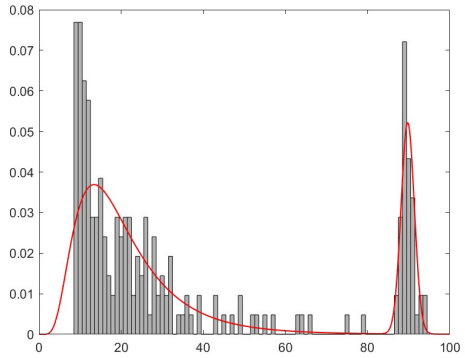


Figure 6.7: A histogram of the dToF measurements of detected photons at pixel (17,17) for a circle target obscured by a fog level with visibility of 94.5 m and 0.61 ALs compared with a finite mixture model with 1 lognormal and 1 Gaussian distributions that is fitted to the same set of dToF measurements.

6.3.1.1 Expectation-Maximization (EM) Algorithm

The EM algorithm is utilised to iteratively estimate the parameters of a finite mixture model. In the context of the fog model for a single pixel, this model represents the probability of measuring a certain dToF value. While MLE is a more common method for estimating parameters of finite mixture models from datasets [35], it is not applicable in this context due the difficulty in computing them analytically for mixture models [23, 35]. In fact, the EM algorithm is specifically applicable for computing MLE from incomplete data [41]. Although the existence and uniqueness of solutions cannot be guaranteed with the EM algorithm [35, 111], in the context of employing the fitted finite mixture model for reconstructing dToF values of a target, an approximation is deemed sufficient.

To create a finite mixture model that represents the probability of measuring a certain dToF value at a pixel, the EM algorithm processes the dToF measurements at the pixel over a batch of frames. Let x be the variable that represents a dToF measurement. Each possible dToF measurement is denoted as $x = x_i$. The dataset $D = \{x_1, ..., x_N\}$ represents the set of dToF measurements, where N is the maximum number of possible dToF measurements in the current pixel for the current batch of frames. The EM algorithm assumes that x is independent and identically distributed (IID), following a finite mixture model with K mixture components.

Here, the term mixture component refers to one of the probability distributions in the finite mixture model. When discussing the finite mixture model, the Gaussian distribution in the finite mixture model can be referred to as a Gaussian component. Likewise, the lognormal distribution can be referred to as a lognormal component.

The finite mixture model $p(x|\Theta)$ represents a probability distribution of dToF measurements of the detected photons, defined as follows:

$$p(x|\Theta) = \sum_{k=1}^{K} \alpha_k p_k(x|\underline{\theta_k}) = \sum_{k=1}^{K} \alpha_k p(x|\underline{z_k}, \Theta), \tag{6.1}$$

where $k = 1, ..., K$ is the index of the k-th mixture component, $\Theta = (\underline{\theta_1}, ..., \underline{\theta_K})$ represents a K-tuple and $\underline{\theta_k}$ contains the parameters for the probability distribution of the k-th mixture component. $\underline{z_k}$ serves as a one-hot counter to indicate the k-th mixture component. It is a vector with length K containing the value of 1 only at its k-th entry, with its remaining entries are zero. $\alpha_k = p(\underline{z_k}|\Theta)$ denotes the weight of the k-th mixture component in the finite mixture model. This weight represents the probability that a randomly selected x_i value belong to a k-th mixture component, with the constraint $\sum_{k=1}^{K} \alpha_k = 1$. Both $p_k(x|\underline{\theta_k})$ and $p(x|\underline{z_k}, \Theta)$ are mathematically equivalent representations of the probability distribution of the k-th mixture component.

The EM algorithm calculates the values of θ_k and α_k (for all $k = 1, ..., K$) in the finite mixture model equation. In the context of modelling the dToF measurements of detected photons reflected from fog, the finite mixture model has one Gaussian component and up to two lognormal components. Both types of probability distribution is defined by the same type of parameters so $\underline{\theta_k} = [\mu_k, \Sigma_k]$ for each of the components. However, μ_k and Σ_k have different statistical meaning between Gaussian and lognor-

mal distributions. In the case of the Gaussian distribution, μ_k represents the mean and Σ_k represents the variance of the dToF measurements x_i in dataset D. Whilst for the lognormal distribution, μ_k and Σ_k are not the mean and variance of dataset D. Instead, they are defined as parameters for the lognormal distribution. Another interpretation of μ_k and Σ_k for the lognormal distribution is they are the mean and variance of the equivalent Gaussian distribution of the lognormal distribution. The equivalent Gaussian distribution can be fitted to a transformed set of dToF measurements in the form of $ln(x_i)$, where the dToF measurements x_i from dataset D follows the lognormal distribution.

Initialisation of the parameters $\underline{\theta_k} = [\mu_k, \Sigma_k]$ and α_k are conducted before they are computed iteratively by the EM algorithm. The initial values are denoted as $\theta_k^0 = [\mu_k^0, \Sigma_k^0]$ and α_k^0. The initialisation employs a k-means++ inspired algorithm. The initialisation algorithm is discussed in detail in Section 6.3.1.3.

Each iteration of the EM algorithm aims to compute the parameters α_k, μ_k and Σ_k. Known formulas exist for calculating these parameters when the finite mixture model contains only Gaussian components (i.e. Gaussian Mixture Models (GMMs)) [154]. Therefore, the parameters of the Gaussian components are determined using these formulas. For the lognormal component's parameters, the Gaussian component's formulas are adapted accordingly. The formulas for this computation will be presented later in this section.

Each iteration can be divided into an E-step and an M-step. These two steps will be explained separately and the relevant formulas will be presented for the lognormal and Gaussian components.

E-step: The E-step computes values known as membership weights, aiming to determine the probability of each value x_i, belonging to each mixture component. This assumes that each possible value x_i of the variable x is generated by only one of the k-th mixture components. The membership weight is calculated as follows:

$$w_{ik} = p(z_k|x_i, \Theta) = \frac{p_k(x_i|\underline{\theta_k}) \cdot \alpha_k}{\sum_{m=1}^{K} p_m(x_i|\underline{\theta_m}) \cdot \alpha_m}, 1 \leq k \leq K, 1 \leq i \leq N. \tag{6.2}$$

This is calculated for all dToF measurements x_i, $1 \leq i \leq N$ and all mixture components $1 \leq k \leq K$. Therefore, this results in a $K \times N$ matrix of membership weights. Furthermore, the membership weights are defined such that $\sum_{k=1}^{K} w_{ik} = 1$, meaning each column of w_{ik} sums to 1.

This equation is valid irrespective of the type of distribution, as it directly applies Bayes' rule [154]. During calculation, the probability distributions ($p_k(x_i|\underline{\theta_k})$ and $p_m(x_i|\underline{\theta_m})$) in Eq. 6.2 follow either the Gaussian distribution or the lognormal distribution, depending on whether the mixture component is defined to be Gaussian or lognormal distribution, respectively. The formulas for the Gaussian and lognormal distributions are provided below.

For a Gaussian component, $p(x|\theta_k)$ takes the form of a Gaussian probability density function which is the following:

$$p_k^G(x|\underline{\theta_k}) = \frac{1}{(2\pi)^{1/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}\frac{(x-\mu_k)^2}{\Sigma_k}}, \tag{6.3}$$

where μ_k is the mean and Σ_k is the variance of the Gaussian component k.

For a lognormal component, $p(x|\theta_k)$ takes the form of a lognormal probability density function, which is the following:

$$p_k^L(x|\underline{\theta_k}) = \frac{1}{(2\pi)^{1/2}|\Sigma_k|^{1/2}x} e^{-\frac{1}{2}\frac{(\ln(x)-\mu_k)^2}{\Sigma_k}}, \tag{6.4}$$

where μ_k and Σ_k are defined as the parameters of the lognormal component k.

For the first iteration of the E-step, the initial values of the parameters $\underline{\theta_k^0} = [\mu_k^0, \Sigma_k^0]$ and α_k^0 are used in these Eq. 6.2, 6.3 and 6.4 for $\underline{\theta_k} = [\mu_k, \Sigma_k]$ and α_k.

M-step: The M-step utilises the membership weights w_{ik} and the dataset D to calculate the new values of Θ and α_k. First, the sum of the membership weights N_k for the k-th mixture component is calculated to determine the number of dToF measurements x_i that can be modelled by mixture component k. This is given by:

$$N_k = \sum_{i=1}^{N} w_{ik} \tag{6.5}$$

Then the mixture weights α_k for the finite mixture model (as shown in Eq. 6.1) are calculated by:

$$\alpha_k^{new} = \frac{N_k}{N}, 1 \leq k \leq K, \tag{6.6}$$

The calculations of the sum of membership weights (N_k) and mixture weights (α_k) are the same regardless of the type of probability distribution defined for the mixture component. However, the other parameters vary depending on the type of probability distribution of each component.

The mean μ_k and variance Σ_k of the Gaussian component are updated using the following formulas:

$$\mu_k^{new} = (\frac{1}{N_k}) \sum_{i=1}^{N} w_{ik} \cdot x_i, 1 \leq k \leq K \tag{6.7}$$

$$\Sigma_k^{new} = (\frac{1}{N_k}) \sum_{i=1}^{N} w_{ik} \cdot (x_i - \mu_k^{new})^2, 1 \leq k \leq K \tag{6.8}$$

For a lognormal component, the calculation of its parameters μ_k and Σ_k adopts a different formula to the Gaussian component. The formulas in Eq. 6.7 and Eq. 6.8 are employed to calculate μ_k and Σ_k for a Gaussian component only. However, for a variable x that follows a lognormal distribution, the transformed variable $ln(x)$ follows a Gaussian distribution. Therefore, the parameters μ_k and Σ_k of a lognormal distribution fitted to x will have the same value as the mean μ_k and variance Σ_k Gaussian distribution fitted to the transformed data $ln(x)$. Therefore, the lognormal

component's parameters μ_k and Σ_k can be calculated by applying Eqs. 6.7 and 6.8 to the data in the form of $\ln(x)$ instead of x. This leads to the formulas for updating the lognormal component's parameters to be:

$$\mu_k^{new} = (\frac{1}{N_k}) \sum_{i=1}^{N} w_{ik} \cdot \ln(x_i), 1 \leq k \leq K \tag{6.9}$$

$$\Sigma_k^{new} = (\frac{1}{N_k}) \sum_{i=1}^{N} w_{ik} \cdot (\ln(x_i) - \mu_k^{new})^2, 1 \leq k \leq K \tag{6.10}$$

As the EM algorithm is iterative, a convergence condition is used to determine when the algorithm stops further calculations. The convergence condition is determined by a log-likelihood measure derived from [154]. This is discussed in Section 6.3.1.2. If the convergence measure is not met within 50 iterations, the algorithm halts and outputs the parameter values ($\underline{\theta_k} = [\mu_k, \Sigma_k]$ and α_k) from the current iteration.

6.3.1.2 Convergence Condition for EM Algorithm

The EM algorithm is an iterative process aimed at determining the values Θ and α_k in Eq. 6.1. In this finite mixture model algorithm, a solution is defined to be converged when the absolute difference in log-likelihood between the current iteration's parameters Θ and α_k and those of the previous iteration is equal to or less than 0.01. The implementation halts after 50 iterations if this convergence criterion is not met.

The log-likelihood under the IID assumption of variables is the following for a Gaussian distribution:

$$\ln l(\Theta) = \sum_{i=1}^{N} \ln p(x|\Theta) = \sum_{i=1}^{N} \left(\ln \sum_{k=1}^{K} \alpha_k p_k^G(x_i|\underline{\theta_k}) \right) \tag{6.11}$$

For the lognormal distribution case, the log-likelihood changes to the following:

$$\ln l(\Theta) = \sum_{i=1}^{N} \ln p(x|\Theta) = \sum_{i=1}^{N} \left(\ln \sum_{k=1}^{K} \alpha_k x_i p_k^L(x_i|\underline{\theta_k}) \right) \tag{6.12}$$

The expression in the summation has an extra factor of x_i because $x_i p_k^L(x_i|\underline{\theta_k})$ is equivalent to $p_k^G(\ln(x_i)|\underline{\theta_k})$ by the following:

$$\begin{aligned} x_i p_k^L(x_i|\underline{\theta_k}) &= x_i \times \frac{1}{(2\pi)^{1/2}|\Sigma_k|^{1/2} x_i} e^{-\frac{1}{2}\frac{(\ln(x_i)-\mu_k)^2}{\Sigma_k}} \\ &= \frac{1}{(2\pi)^{1/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}\frac{(\ln(x_i)-\mu_k)^2}{\Sigma_k}} \\ &= p_k^G(\ln(x_i)|\underline{\theta_k}) \end{aligned} \tag{6.13}$$

This means Eq. 6.12 calculates the log-likelihood of the equivalent Gaussian distribution with mean μ_k and variance Σ_k. This formula is used because the update of the parameters μ_k and Σ_k (from Eq. 6.9 and 6.10) is calculated as the mean and

variance of the Gaussian distribution with data in the form of $\ln(x_i)$. Therefore, it is mathematically consistent to continue calculating the log-likelihood in terms of the same Gaussian distribution.

As the log-likelihood value is a summation of the log-likelihood values for all the lognormal and Gaussian components in the finite mixture model, the log-likelihood for the mixture of lognormal and Gaussian distributions can be expressed as follows:

$$\begin{aligned} \ln l(\Theta) &= \sum_{i=1}^{N} \ln p(x|\Theta) = \sum_{i=1}^{N} \left(\ln \sum_{k=1}^{K} L(x_i, \alpha_k, \underline{\theta_k}) \right) \\ L(x_i, \alpha_k, \underline{\theta_k}) &= \begin{cases} \alpha_k p_k^G(x_i|\underline{\theta_k}) & \text{last k-th component} \\ x_i \alpha_k p_k^L(x_i|\underline{\theta_k}) & \text{otherwise} \end{cases} \end{aligned} \tag{6.14}$$

6.3.1.3 Parameter Initialisation

As the EM algorithm is an iterative process, the parameters $\underline{\theta_k} = [\mu_k, \Sigma_k]$ and α_k of the finite mixture model in Eq. 6.1 need to be initialised first. The initial values of these parameters are denoted as $\underline{\theta_k^0} = [\mu_k^0, \Sigma_k^0]$ and α_k^0.

The initialisation process employed for the finite mixture model algorithm draws inspiration from the k-means++ algorithm[4] outlined in [18]. The process forms clusters from the raw data points, which are the dToF measurements $x = x_i$. The number of clusters to be formed are dictated by the pre-defined number of mixture components (K) in the finite mixture model. In the application of this initialisation process to the finite mixture model algorithm, the number of mixture components is up to 3, depending on the number of mixture components employed. Each of these clusters C_k contain data that can be fitted with one of the mixture components.

The initial values $\underline{\theta_k^0} = [\mu_k^0, \Sigma_k^0]$ and α_k^0 of each mixture component can be calculated from these clusters. Each cluster is used to calculate the parameters of each mixture component. The proportion of points of each cluster compared to the data set D is used to set the initial value of each component's mixture weight α_k^0. The cluster with the highest mean value is used to set the parameters of the K-th component (and labelled as the K-cluster). This is because the finite mixture model assumes the K-th mixture component follows a Gaussian distribution. The K-th cluster's mean μ_{C_k} and variance Σ_{C_K} are directly used as the K-th mixture component's Gaussian distribution's mean and variance values, denoted as μ_K^0 and Σ_K^0.

As the finite mixture model is defined to have only one Gaussian component, the remaining clusters are used for determining the parameters of the other mixture components, which follow the lognormal distribution. Each remaining cluster's points are assumed to follow a lognormal distribution. However, each of these remaining clusters' mean and variance values cannot be directly used as the parameters of the lognormal components. This is because the parameters of the lognormal distribution

[4]There is a minor error in the reporting in [99] where k-means++ algorithm is not used to initialise the parameters for the finite mixture model algorithm, instead the initialisation process is inspired by the k-means++ algorithm.

are not its mean and variance. Therefore, the following formula is the calculation of the initial parameters μ_k^0 and Σ_k^0 of the lognormal components given its mean and variance are μ_{C_k} and Σ_{C_k} respectively. These formulas follow directly from the standard equations of using the parameters of a lognormal distribution to calculate its mean and variance. This is the part of the parameter initialisation process where it differs from the k-means++ algorithm.

$$\mu_k^0 = \ln(\frac{\mu_{C_k}^2}{\sqrt{\mu_{C_k}^2 + \Sigma_{C_k}}}) \tag{6.15}$$

$$\Sigma_k^0 = \ln(1 + \frac{\Sigma_{C_k}}{\mu_{C_k}^2}) \tag{6.16}$$

This parameter initialisation process has several other differences from the k-means++ algorithm. First of all, the initialisation here is not an iterative algorithm because the dToF measurements x_i span a small range of between 0 to 100 clock cycles. This means one iteration is sufficient to approximate an adequate set of initial values of the parameters, which the EM algorithm can then use for initialisation to accurately estimate these parameters throughout its iterative process. Secondly, the Mahalanobis distance is used instead of the square of the Euclidean distance when calculating distance between the data points and the centre of each cluster. Thirdly, additional checks are employed to ensure the value of any of the cluster's Σ_{C_k} is not zero, which ensures none of the Σ_k^0 values is zero. This is because the EM algorithm breaks down if any of the Σ_k^0 values are zero. The values Σ_{C_k} are used as the value of Σ_k^0 or are used to calculate Σ_k^0. The values set for Σ_{C_k} are still sensible as it is still within the range of value in the dataset D. The EM algorithm can further compute the values of μ_k, Σ_k and α_k iteratively.

The details of the initialisation process are shown by the pseudocode presented in **Algorithm 1**.

This space is intentionally left blank.

Algorithm 1 Parameters initialisation for the finite mixture model algorithm.

1: For each $k \in \{1, ..., K\}$, initialise the variance Σ_{C_k} of each k-th mixture cluster to be the variance of the entire data set D.
2: Take one centre c_1, chosen uniformly at random from the set of dToF measurements $x \in D$.
3: Remove all dToF measurements x_i in the data set D that is the same value as c_1.
4: Take a new centre c_l, choosing from the remaining dToF measurements $x \in D$ with probability $\frac{M(x,c_j,\Sigma_{C_j})}{\sum_{x \in D} M(x,c_j,\Sigma_{C_j})}$, where $M(x, c_j, \Sigma_{C_j}) = \frac{|x-c_j|}{\Sigma_{C_j}}$ is the Mahalanobis distance of the point $x \in D$ to its closest centre c_j where $1 \leq j < l$.
5: Repeat the previous step until there are K centres altogether.
6: For each $k \in \{1, ..., K\}$, set the cluster C_k to be the set of points in D that have the closest Mahalanobis distance to c_k than they are to c_m for all $m \neq k$.
7: Set the centre with the largest value to be the initial value μ_K^0 of the K-th mixture component, which is the mixture component that follows a Gaussian distribution.
8: Calculate the centre values μ_{C_k} of the remaining clusters where $k < K$.
9: Set the initial mixture weight α_k^0 to be the portion of points in the cluster C_k with respect to all the points in the data set D.
10: Calculate the sample variance Σ_{C_k} of each of the clusters C_k.
11: If all Σ_{C_k} value are zero (i.e. the number of unique points are less than K), then make all Σ_{C_k} values equal to the variance of the data set D.
12: If less than K number of Σ_{C_k} values are zero, for the Σ_{C_l} values that are zero, set them as the mean of all the Σ_{C_m} values that are non-zero, where $m \neq l$ and $1 \leq l, m \leq K$.
13: Set the initial variance of the K-th mixture component Σ_K^0 to be the variance Σ_{C_K} of cluster C_K, this mixture component follows the Gaussian distribution.
14: Calculate the initial parameters μ_k^0 and Σ_k^0 of the remaining mixture components (which follows the lognormal distribution) using Eq. 6.15 and 6.16.

6.3.2 Target Classification of Reconstructed Images

The reconstructed images from the finite mixture model algorithm undergo evaluation for target classification. As part of this process, the reconstructed images are first converted into binary images and then denoised, as illustrated in Figure 6.8. Section 6.3.2.1 below provides details on the steps for binary conversion and denoising.

Converting the reconstructed image to a binary format ensures accurate target classification through a straightforward target classification algorithm. This simplification is essential to enable real-time autonomous target classification, as it streamlines the computational process without compromising accuracy.

The total number of non-zero pixels in the binary image is utilised as a metric to classify the target in the reconstructed image. As depicted in Figure 6.8, it is important to note that the non-zero pixels in the binary image represent the background[5]. By classifying the amount of background pixels, the different shapes can be distinguished, as the amount of background pixels is related to the type of shape in the image. This is because the binary image is de-noised to the extent that the majority of pixels are either from the background or the shape. The threshold values for the number of background pixels are determined empirically to provide the best possible classification performance.

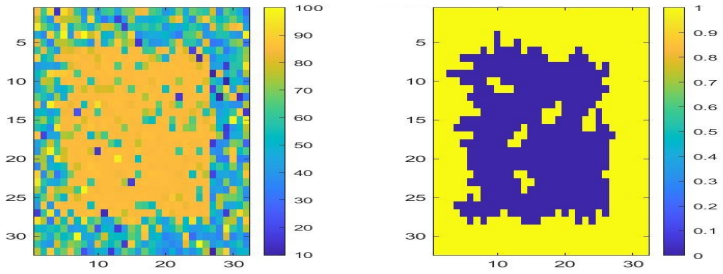


Figure 6.8: Reconstructed image (left) and its corresponding denoised binary image (right) employed for classification. These images represent a square obscured by a fog level of 50.8m visibility and 1.14 ALs

For evaluating target classification with the SFD, each class of target is a shape, and the classifier has a specific range of pixel counts to distinguish between the shapes. As explained in Section 3.7 of Chapter 3, the target classification algorithm incorporates an additional class termed 'Undecided.' This class is introduced to accommodate instances where the target in the reconstructed image does not conform to any of the classification rules for the various shapes. During classifier testing, no instances of the Undecided class are employed, as its purpose is solely to handle cases where the classifier cannot determine an object class for the target based on predefined rules.

6.3.2.1 Converting the Reconstructed Image to a Denoised Binary Image

The process employed to convert the reconstructed image into a denoised binary image is similar to the process described in Section 5.4.3.1 of Chapter 5; the difference here is the initial converted binary image's target pixels have the value 1 instead of 0. This results in the final binary image having its target pixels as 0 instead of 1. This difference enables the background pixels to be used here for classification instead of the target pixels.

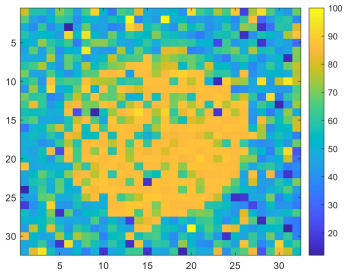
[5]The paper [99] provides incorrect details regarding the classification process, implying that the pixels from the shape are classified. However, in reality, it is the background pixels that are utilised for classification.

The initial converted binary image's target pixels have a value of 1 instead of 0 because the threshold value used here is different to the process presented in Section 5.4.3.1 of Chapter 5. The threshold value varies for each image and is determined as the 60th percentile pixel value of the reconstructed image[6]. The choice of the 60th percentile is based on the observation that the reconstructed dToF values of the target pixels in the reconstructed image constitute the top 40% of the reconstructed dToF values. As the target pixels have a higher value than the threshold, this results in its binary value being 1 instead of 0.

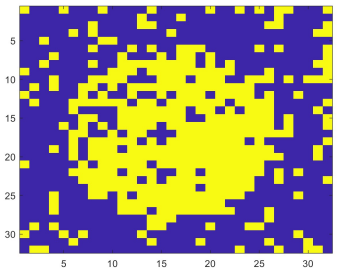
After the reconstructed image is converted to a binary image, subsequent steps involve filling holes and removing scattered pixels. This processing is necessary because as the fog thickens, there is a decrease in the number of detected photons reflected from the target, resulting in fewer pixels representing the target. Consequently, this may lead to an increase in holes and noise within the binary image. Figure 6.9 illustrates these processing steps. The binary image undergoes multiple inversions, during which scattered pixels are removed. The process of removing scatter pixels from an inverted image effectively fills in holes in the original binary image. Finally, the image is inverted one last time (as depicted in Figure 6.9(g)) to ensure that background pixels contain the value 1, thereby enabling the count of non-zero pixels to be equivalent to counting the total number of background pixels, where this count value is used for target classification.

This space is intentionally left blank.

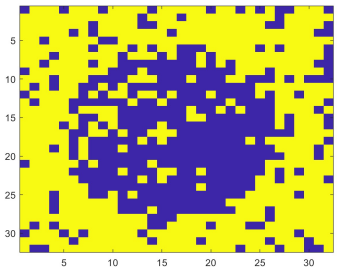
[6]The definition of this threshold value in [99] is incorrect; it is directly calculated from the reconstructed image, not the median image.



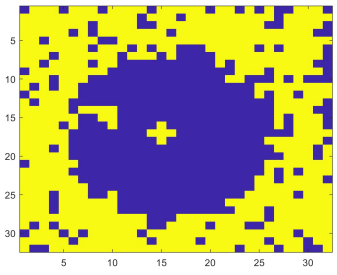
(a) Reconstructed image



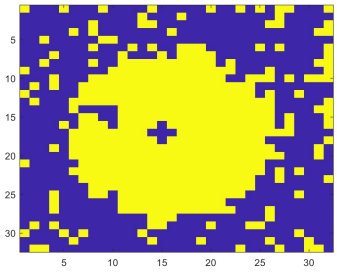
(b) Thresholded to binary image



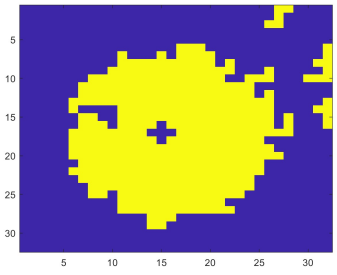
(c) Flipped binary image



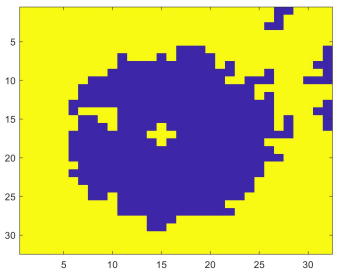
(d) Removed scattered pixels



(e) Flipped binary image



(f) Removed scattered pixels



(g) Denoised binary image

Figure 6.9: Images depicting each processing step applied to a reconstructed image for target classification. The data used here is of a circle obscured by a fog level of 50.8m visibility and 1.14 ALs.

6.3.3 Real-Time Capability

The execution times of the finite mixture model algorithm and the target classification algorithm of the reconstructed images are not measured here because their implementations are not optimised. To demonstrate their real-time capability, an analysis of their operations is conducted. Section 6.3.3.1 provides an analysis of the finite mixture model algorithm and Section 6.3.3.2 provides an analysis of the target classification algorithm.

6.3.3.1 Finite Mixture Model Algorithm

Figure 6.10 shows a summary of processing the dToF measurements of one pixel using the finite mixture model algorithm described in Section 6.3.1. For the parameter initialisation step in the finite mixture model algorithm, this figure refers to Algorithm 1 in Section 6.3.1.3 which presents a detailed pseudocode to this step. When considering this figure in conjunction with Algorithm 1, it is clear that most of the finite mixture model algorithm consists of mathematical operations.

The most computationally expensive task out of the mathematical operations in the finite mixture model algorithm is multiplication. This algorithm may use up to approximately 1.8 million multiplication operations to process dToF measurements from on pixel. This assumes the algorithms uses up to 50 iterations and processes up to 1000 frames, which is the maximum number of frames that are used during evaluation of the algorithm in Section 6.5. With appropriate implementation, state-of-the-art GPUs is able to execute this number of multiplication operations in real-time. For comparison, the Jetson TX2 utilised in Chapter 4 is an embedded board with an on-board GPU, and it can perform 1.33 TFlops in its default configuration [8]. This is commonly demonstrated in high speed signal processing and machine learning algorithms. Therefore, this discussion demonstrates the mathematical operations used in this algorithm are real-time capable for processing one pixel's dToF measurements.

After multiplication, the next most computationally expensive step in this algorithm is parameter initialisation, which is the first step in Figure 6.10. The pseudocode for parameter initialisation is presented in detail in Algorithm 1 in Section 6.3.1.3. By examining this pseudocode, it can be seen that Step 4 is the most computationally expensive because the sorting operation is used to determine which dToF measurement $x \in D$ has the highest probability $\frac{M(x,c_j,\Sigma_{C_j})}{\sum_{x\in D} M(x,c_j,\Sigma_{C_j})}$. Nonetheless, the finite mixture model algorithm is evaluated against the SFD and only processes a batch of a maximum of 1000 frames for each instantiation. Therefore, there is only a maximum of 500 dToF values in the dataset D, which means the sorting operation will be real-time capable given it is sorting a small amount of values. In addition, line 4 in Algorithm 1 may repeat up to two times, which depends on the total number of mixture components that is used[7]. When line 4 needs to be repeated twice, two sorting operations conducted in series is still real-time capable given the sorting operation is still only sorting 1000 values each time.

[7] number of mixture components is discussed in Section 6.3.1

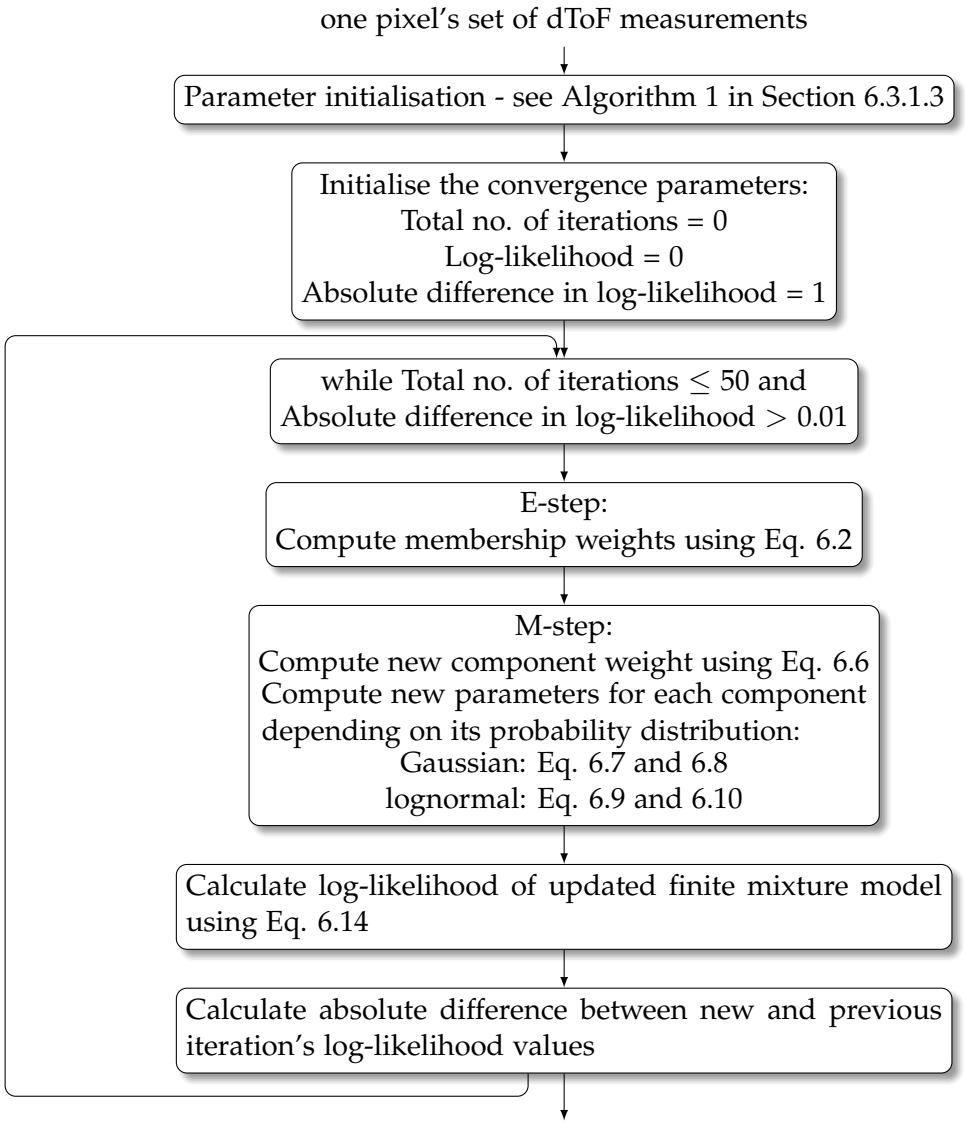


Figure 6.10: Summary of finite mixture model algorithm

This discussion demonstrates that the finite mixture model algorithm is real-time capable for processing one pixel's set of dToF measurements. To enable real-time processing of dToF measurements from all SPAD pixels, a parallel implementation can be employed to process each pixel in each parallel thread. This implementation follows the same discussion in Section 5.3.1 of Chapter 5 for processing each pixel in each parallel thread.

6.3.3.2 Target Classification Algorithm

Figure 6.11 presents a summary of the algorithm for classifying reconstructed images from the finite mixture model algorithm. It includes the finite mixture model algorithm in the summary because the target classification algorithm processes the finite mixture model algorithm's reconstructed images of the fog-obscured targets. Therefore, the finite mixture model algorithm should be included in the consideration of whether this target classification algorithm is real-time capable.

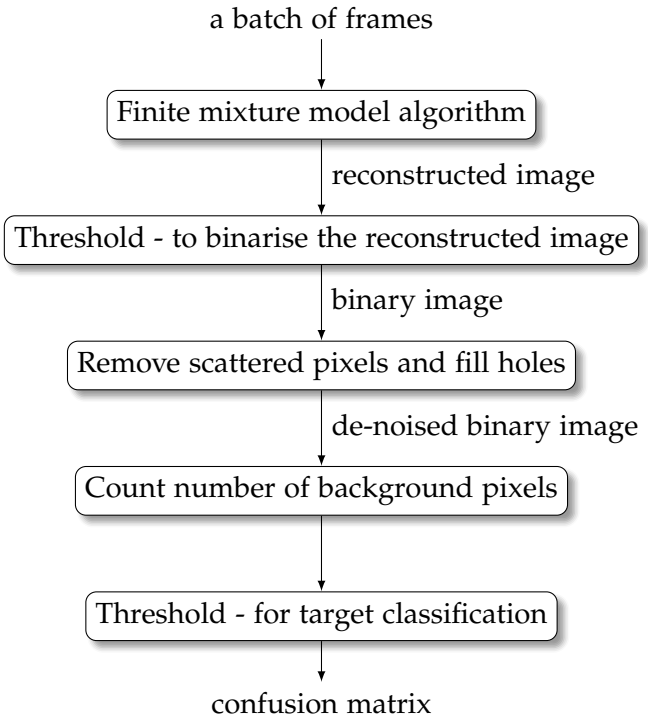


Figure 6.11: Target classification using reconstructed images from the finite mixture model algorithm

The most computationally expensive task in this algorithm is remove scattered pixels and fill holes, which is step 3 in Figure 6.11. As discussed in Section 5.4.3.3 of Chapter 5 which also examines this task in a different target classification algorithm, this task is real-time capable given the small pixel resolution of the image.

The Finite Mixture Algorithm has already been shown to be real-time capable, as discussed in Section 6.3.3.1. The remaining tasks are all computationally simple,

therefore making this classification algorithm real-time capable.

6.3.4 The Weaknesses of Pixelwise Approaches

The finite mixture model algorithm suffers from poor localisation accuracy, as evidenced in Section 6.5.2.4. This drawback can be attributed to two primary factors. Firstly, the pixel-wise approach yields varying reconstructed dToF value for each target pixel, leading to inconsistencies. Secondly, the algorithm models each pixel independently, lacking a mechanism to determine the target's spatial position within the 32×32 array image. Consequently, it struggles to identify which pixel is a target pixel and therefore have these target pixels' reconstructed dToF values dictate the target's location. A more suitable approach for reconstructing fog-obscured images would involve leveraging spatial information alongside the finite mixture model algorithm's processing across the entire imaging range of the SPAD array flash LiDAR system. The next section introduces the Fourier descriptor algorithm, which endeavours to address this challenge by employing an image feature-based approach to determine the target's spatial position within the 32×32 array and subsequently determine the target's location.

6.4 Fourier Descriptor Algorithm

The Fourier descriptor algorithm adopts an image feature-based approach for image reconstruction. This algorithm differs conceptually from the previous two image-reconstruction algorithms discussed in this chapter, in that it reconstructs all pixels simultaneously instead of a pixel-by-pixel basis. The algorithm analyses the data from all of the pixels, creating a binary image for each possible value in the range of dToF measurements. A binary image related to photon counts of only one of the possible values in the range of dToF measurements is referred to as a slice when discussing this algorithm. Each binary image pixel for each slice is 1 if the total photon count over a batch of frames is non-zero for the slice's corresponding dToF value, otherwise the pixel value is 0. This data structure is different to what is applied in the previous two image-reconstruction algorithms, which independently analyse each pixel's dToF measurements. The data structure employed in this algorithm is described in Section 6.4.1.

Given the Fourier descriptor algorithm is evaluated on data from the SFD, which consists of planar imaging targets, this means the expected dToF value for all target pixels should be the same. Target pixels are pixels at the expected spatial position of the target in the image. Therefore, the target's entire silhouette can be represented in one of the slices. This allows the Fourier descriptor algorithm to perform global target localisation by determining which slice contains the closest silhouette to the target, which is solved by computing Fourier descriptor for each slice. The chosen slice's corresponding dToF value provides the global target location.

From the global target location, image reconstruction is a simple conversion from the chosen slice's binary image. The reconstructed image has the slice's associated

dToF value at the same pixels where the slice's binary image has a non-zero value, and the rest of the reconstructed image's pixels are set as undefined.

A detailed explanation of the entire algorithm is presented in Section 6.4.2, with a detailed explanation of how to calculate a Fourier descriptor provided in Section 6.4.2.1.

The results from evaluating this algorithm's global target localisation performance in terms of absolute localisation accuracy is reported in Section 6.5.3.1.

6.4.1 Data structure

For each possible value in the range of dToF measurements, the Fourier descriptor algorithm constructs a binary image in which each pixel represents the number of photon counts at the corresponding dToF value. Each image related to each possible dToF value is referred to as a slice when discussing this algorithm.

Each slice's binary image is created in two steps. For each slice's dToF value, an image is created where each pixel's value is the number of photon counts with the corresponding dToF measurement over a batch of frames. A set of these images is called a histogram stack. Then, each slice's binary image is created by binarising each slice's image in the histogram stack. The binary image's pixel value is 1 if the corresponding pixel value in the histogram stack's image is non-zero. Otherwise, the pixel value is 0. Binary images are employed to facilitate easier and faster processing of spatial information. A set of these binary images is called a binary histogram stack.

As an example, Figure 6.12 depicts the transformation of four frames of 3×3 dToF measurements into a binary histogram stack. The measurements are demonstrated to convert into a histogram stack, then into a binary histogram stack. The bin values refer to each possible value in the range of dToF measurements.

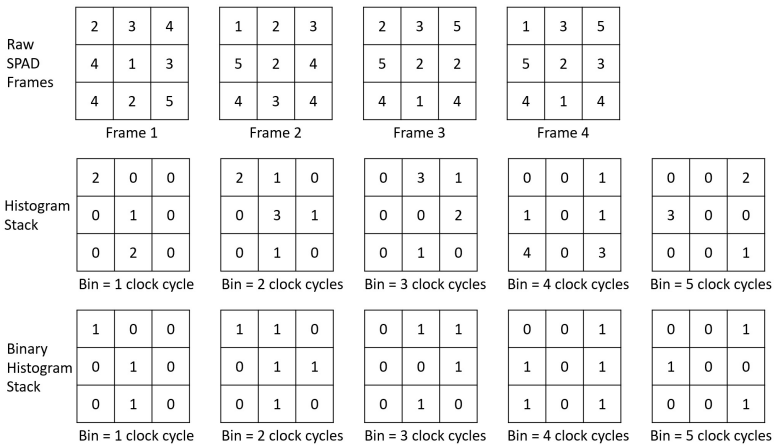


Figure 6.12: Example of turning dToF measurements collected over four frames into histogram stack and binary histogram stack.

In context of the SFD, the maximum dToF measurement is 100 clock cycles, approximating to a distance of 52 m from the SPAD array flash LiDAR system to the back of the fog (as depicted in Figure 6.2). This equates to the binary histogram stack having a maximum of 100 slices because there can only be a maximum of 100 different values in the dToF measurements.

6.4.2 Algorithm

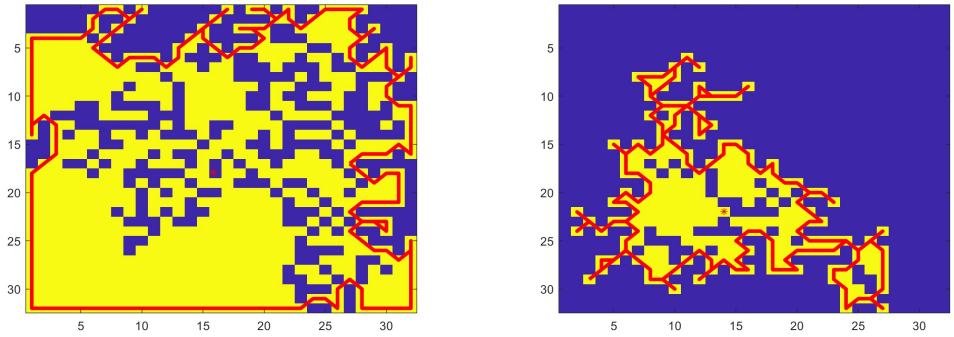
The Fourier descriptor algorithm aims to identify the slice within the binary histogram stack that best matches the image features of the target. By identifying the slice, the target's location is assigned as the dToF value associated with the slice. The algorithm accomplishes this by comparing the Fourier descriptor of each slice to that of the target. The slice with the closest resemblance in Fourier descriptor is deemed the winner, and its associated dToF value is designated as the location of the target.

Given the Fourier descriptor algorithm is evaluated on the SFD which consists of targets that are planar objects, the algorithm has an assumption that only a maximum of one single target can locate at each of the distance intervals corresponding to each of the dToF measurements by the SPAD array flash LiDAR system. Therefore, the algorithm computes at most one Fourier descriptor for each slice of the binary histogram stack. The process of computing Fourier descriptors is detailed in Section 6.4.2.1.

Before a Fourier descriptor is calculated for each slice, the algorithm first identifies and removes noise regions, followed by filling in any holes within a given slice. This process follows the same approaches described in Section 6.3.2.1 in this chapter and Section 5.4.3.1 in Chapter 5 for removing scattered pixels and holes. Here, scattered pixels are considered as noise regions. A region containing pixels with a value of 1 is flagged as noise if the total number of such pixels falls below a threshold, which is a fraction of the apparent target area at that location. Similarly, a region with pixels of value 0 is recognised as a hole if its total count of 0-valued pixels is below the same threshold. The binary image is inverted and scattered pixels are removed from it to effectively fill holes in the original binary image. The calculation of the apparent target area is detailed in Eq. 6.17 in Section 6.4.2.1.

6.4.2.1 Fourier Descriptors

Fourier descriptors provide a frequency domain description of the contour of an object [48]. In the Fourier descriptor algorithm, the calculation of Fourier descriptors follows a method described in [48] for shape recognition. The calculation begins with a boundary being traced clockwise around a region of 8-connected pixels with a value of 1. In the proof-of-concept implementation used in this approach, MATLAB's `bwtraceboundary` function is employed for this purpose. The resulting trace depends on the initial point and search direction provided as inputs to the function. Figure 6.13 illustrates two examples of boundary traces on slices generated from data in the SFD.



(a) (b)

Figure 6.13: Example of boundary traces with the red asterisk representing the mean of the boundary trace coordinates. The slices' associated dToF values are 50 (a) and 85 (b) clock cycles. The fog level of this data has visibility of 25.4 m/2.28 ALs.

Since the exact spatial position of the target within each slice's binary image is unknown, it is not feasible to specify a particular pixel and direction as the initial point and search direction for boundary tracing. Therefore, a set of 4000 unique and randomly generated combinations of initial pixels and search directions are used. The initial pixels are selected from all 32×32 pixels of the SPAD array, and the initial search directions include all 8 possible directions: North, North-East, East, South-East, South, South-West, West, and North-West.

Not every initial boundary pixel and search direction will successfully trace an object's boundary, but 4000 different combinations cover roughly half of the possible combinations of pixels and search directions. Compared to using all possible combinations, this approach provides approximately the same likelihood that an object is traced in each slice's binary image, as opposite directions at the same pixel yield the same trace. For each single binary histogram stack generated from a batch of frames, the same 4000 combinations of initial pixels and search directions are used. A new set of combinations is generated only when a new batch of frames is processed.

All generated boundary traces for each slice are further filtered by performing an area check to determine whether the object could be the target. A boundary trace is retained only if the object's area falls within a tolerance of 50% of the apparent target size at the slice's dToF value. This tolerance level has been empirically determined to work well with the SFD used to evaluate this algorithm.

The apparent target area can be estimated using ground truth data, which is a set of frames of the same target where its location is known. This set of frames is also from the SFD but is separate to the frames used for evaluating the Fourier descriptor algorithm. A binary histogram stack is generated for this ground truth data, and after applying the denoising steps described, the target's area can be determined from the slice associated to the target's expected dToF value. The area is defined as the number of pixels occupied by the target.

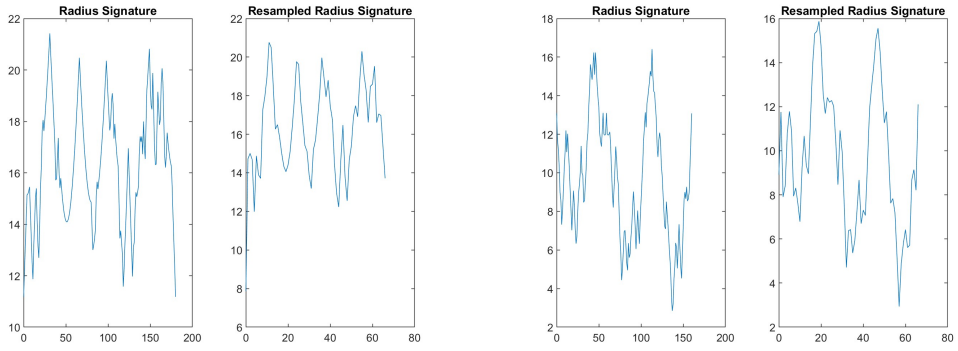
The apparent target area can then be calculated using the following formula:

$$\text{apparent area of target in slice associated with a dToF value of k} = \frac{\text{target's area in slice associated with a dToF value of p}}{(\frac{k}{p})^2}, \tag{6.17}$$

where p is the expected dToF value of the target.

To reduce computation time, the area of a bounding box that circumscribes the boundary trace is used instead of the exact area encompassed by the boundary trace. This approach is mathematically sound because the area of a region is proportional to the area of the bounding box that circumscribes it. This area check is a preliminary filter and does not account for shape differences between a detected object and the target. However, the algorithm corrects for this in subsequent steps by using Fourier descriptors.

The coordinates of each remaining boundary trace are converted to a radius signature, which records the distances of consecutive boundary trace points from the mean point of the same trace. Examples of radius signatures are shown in Figure 6.14. The radius signature is then resampled to the same length as the target's expected radius signature. The target's expected radius signature is calculated from applying the described steps on data of the same target collected in the same imaging set-up as SFD but without any obscurants. Resampled radius signatures are also shown in Figure 6.14.



(a) (b)

Figure 6.14: Example of boundary traces turned into radius signatures and then resampled radius signatures. The bin numbers are 50 (a) and 85 (b) clock cycles. The fog level of this data has visibility of 25.4 m/2.28 ALs.

Once the resampled radius signature is calculated, a Fast Fourier Transform (FFT) is performed to create the Fourier descriptor. This radius signature resampled to produce equivalent FFT components for correct comparison with the target's expected Fourier descriptor, which is calculated from the target's expected radius signature.

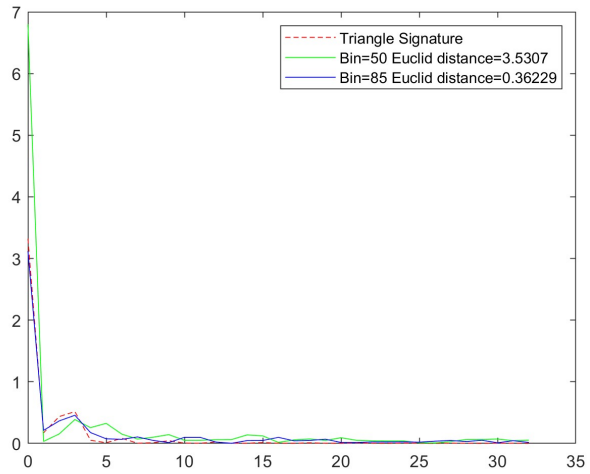


Figure 6.15: Example of Fourier descriptors of the radius signatures shown in Figure 6.14. The fog level of this data has visibility of 25.4 m/2.28 ALs.

Since the phase of the Fourier descriptor depends on the starting point of the boundary trace, only the magnitude of the Fourier descriptor is used for comparison to determine which slice contains the target. All frequency components are used and normalised to create a size-invariant descriptor, allowing comparison of detected shapes at different depths (and therefore different apparent sizes) to a fixed-size pre-computed target descriptor. Figure 6.15 shows two objects' Fourier descriptors compared with the target's Fourier descriptor.

The Euclidean distance between each remaining boundary trace's Fourier descriptor and the target's expected Fourier descriptor is calculated. The slice containing the boundary trace with the Fourier descriptor that has the closest Euclidean distance to the target's expected Fourier descriptor is then selected as the one containing the target. The global target location is assigned as the dToF value of this chosen slice.

6.4.2.2 Real-Time Capability

The execution time of the Fourier descriptor algorithm is not measured here because the implementation has not been optimised. Instead, an analysis of its operations steps is made to determine its real-time capability. An overview of the algorithm's steps is summarised in Figure 6.16. The Fourier descriptor algorithm is designed to operate on batches of frames so it can be implemented for real-time processing on an embedded board. When evaluated for its performance against the data in SFD, the algorithm uses a batch of 500 frames when processing data collected at a fog level of 43.8 m visibility and 1.33 ALs, while it uses 1,000 frames for a visibility of 25.4 m and 2.28 ALs.

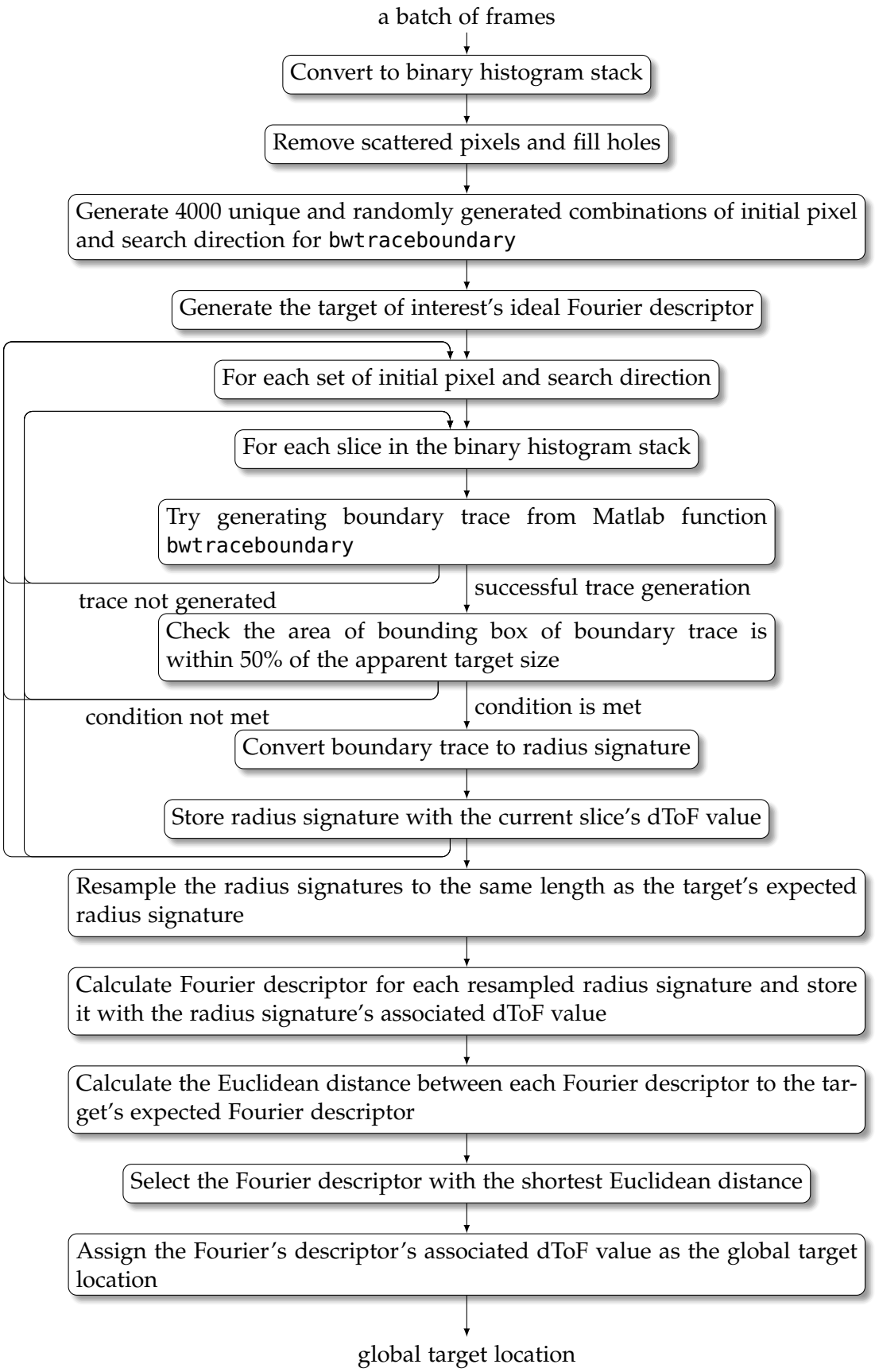


Figure 6.16: Summary of Fourier descriptor algorithm

Figure 6.16 shows that the most computationally-expensive task in the Fourier descriptor algorithm is converting a batch of frames into a binary histogram stack (1st step in the figure). This task is computationally expensive because it uses the sorting operation to create a histogram for each pixel's set of dToF measurements. However, there are only 32×32 pixels. Moreover, only a maximum of 1000 frames are used for each instantiation when evaluating this algorithm against the SFD. This means only $32 \times 32 = 1024$ histograms need to be created and for each histogram, and only up to 1000 dToF measurements are being sorted. Even though the sorting operation is expensive, applying it a small number of times means creating a histogram for each pixel is still real-time capable.

If creating 1024 histograms causes the algorithm's execution time to be longer than the data acquisition time, it can be parallelised where each parallel thread is used for sorting each pixel's dToF measurements in to a histogram. It is common for a graphical processing unit (GPU) to contain at least 1024 threads. Therefore, this discussion shows the task of converting the batch of frames into a binary histogram stack is real-time capable.

The second most computationally-expensive task is selecting the Fourier descriptor with the shortest Euclidean distance to the target's expected Fourier descriptor (second last step in Figure 6.16). There are only 4000 Euclidean distance values. Therefore, following the same discussion as sorting a small number of values for a histogram, this task is real-time capable.

The third most computationally-expensive task is generating the boundary trace for 4000 combinations of initial pixel and search direction (7th step in Figure 6.16). This is because the task of generating a boundary trace (from one set of initial pixel and search direction) requires multiple search operations. The boundary is traced pixel by pixel where at each pixel, the search operation is executed at each pixel to search in all directions for the next pixel that has the boundary. Given the image used for evaluating this algorithm is 32×32 in pixel resolution, the boundary cannot have more than $32 \times 32 = 1024$ pixels long. Therefore, the search operation is only conducted for a maximum of 1024 times to create one boundary trace, where the small number of instantiations of the search operation makes generating a boundary trace real-time capable. To improve the speed of generating 4000 boundary traces, parallelisation can be used where each parallel thread can be used to generate a boundary trace for each set of initial pixel and search direction. It is common for GPUs to have at least 4000 threads available. Therefore, generating 4000 different boundary traces is real-time capable.

The fourth most computationally-expensive task is to remove scattered pixels and fill holes, which is step 2 in Figure 6.16. This task requires the search operation to identify holes and noise pixels in each slice's binary image. As discussed in Section 5.4.3.3 in Chapter 5, each search operation is real-time capable over one binary image. Even though this task requires the search operation to be executed for 100 times over 100 binary images[8], this task can be parallelised where each parallel thread executes

[8]number of slices discussed in Section 6.4.1

one search operation. Therefore, this demonstrates this task is real-time capable.

The fifth most computationally-expensive task is generating 4000 unique and random combinations of initial pixel and search direction (3rd step in Figure 6.16). This is because this task uses the operation of random number generation. There are many existing hardware random number generators that can be employed for real-time execution. Therefore, this means this task is real-time capable. This device will need to be mounted onto the mobile platform if this algorithm is to execute on it. Another solution to enabling this task to be real-time capable is to pre-generate these random values. This will not affect the performance of the algorithm because generating 4000 sets of initial pixel and search direction covers roughly half of the possible combinations (as discussed in Section 6.4.2.1), which is enough for one of these combinations to generate a boundary trace.

The remaining tasks in Figure 6.16 are real-time capable because they are standard real-time operations in digital signal processing. From this discussion, it is clear that the Fourier descriptor algorithm is real-time capable.

In the following section, the evaluation of the quality and target classification performance of the reconstructed images are discussed for the range gate and process algorithm, and the finite mixture model algorithm. The performance of the Fourier descriptor algorithm is discussed in terms of global target localisation.

6.5 Results and Discussion

In this section, the performance of the image-reconstruction algorithms presented in this chapter is evaluated using different metrics. The reconstruction performance metrics employed are relative localisation accuracy and pixel-wise absolute localisation accuracy. Other than quantitative measures, visual inspection is also used to determine the reconstruction performance of the algorithms, determining whether the images can be used for visual target detection. The reconstructed image's quality is based on whether a naked eye can identify the silhouette of the imaging target in the image.

In this chapter, relative localisation accuracy is evaluated as the percentage of pixels in the reconstructed image that fall within one clock cycle of the most frequently occurring reconstructed dToF value in the image. Allowing for a one-clock-cycle leeway accounts for any timing errors attributable to the SPAD array flash LiDAR system. Identifying the most frequently occurring reconstructed dToF value in the image involves organising all reconstructed dToF values into a histogram and identifying the dToF value associated with the histogram's peak.

Pixel-wise absolute localisation accuracy is evaluated as the percentage of pixels that have its reconstructed dToF value within a predefined threshold of its expected dToF value. This is a technique utilised in [158, 183].

Target classification of the reconstructed images is also evaluated in terms of overall accuracy; the definition of overall accuracy is provided in Section 3.7 of Chapter 3.

In addition to reconstruction and classification performance, global target localisation performance is evaluated by using absolute localisation accuracy; it is calculated as the absolute difference between the target's expected dToF value and the target's estimated distance, as determined from the reconstructed image.

Each algorithm's performance is presented in terms of one or more of these metrics. As discussed in Section 2.3 of Chapter 2, there is no standardised metric for evaluating the quality of reconstructed images. Different metrics are used to report the performance of different algorithms, where different metrics highlight different characteristics of their reconstructed images by different algorithms.

Table 6.1 lists the dataset (FFD and SFD) and various fog levels employed to evaluate each of the image-reconstruction algorithms presented in this chapter. The table also lists the number of frames used to reconstruct one image. Background explanation of the use of visibility as metric for fog level is provided in Section 3.6.3 of Chapter 3. An explanation of the use of ALs is given in Section 3.6.1 of Chapter 3.

This space is intentionally left blank.

Table 6.1: Fog levels tested for each approach.

Approach	**Dataset**	**Visibility (m)**	**Number of ALs (at 532 nm)**	**Number of frames required to produce one reconstructed image**
Range gate and process algorithm	FFD	14.1	3.69	500
Finite mixture model algorithm (1 lognormal and 1 Gaussian)	SFD - all three shapes	185, 104.7, 101.7, 94.5 and 72.9	0.31, 0.55, 0.57, 0.61 and 0.80	500
Finite mixture model algorithm (2 lognormals and 1 Gaussian)	SFD - all three shapes	57.2 and 50.8	1.01 and 1.14	500
Fourier descriptor algorithm	SFD–only the right-angled triangle	43.8	1.33	500
Fourier descriptor algorithm	SFD–only the right-angled triangle	25.4	2.28	1000

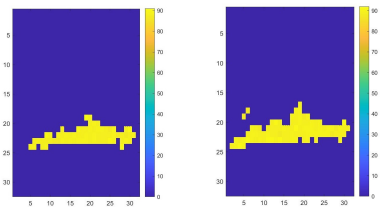
The performance results of the range gate and process algorithm are presented in Section 6.5.1, and then the performance results of the finite mixture model algorithm are presented in Section 6.5.2. Finally, the performance result of the Fourier descriptor algorithm is presented in Section 6.5.3.

6.5.1 Range Gate and Process Algorithm Performance

The metrics used to evaluate the range gate and process algorithm are reconstruction performance evaluated via visual inspection and the overall accuracy of target classification of reconstructed images. The former is discussed in Section 6.5.1.1 and the latter is presented in Section 6.5.1.2.

6.5.1.1 Reconstruction Performance Evaluated via Visual Inspection

The reconstruction performance of the range gate and process algorithm is evaluated using the data with the maximum fog level from FFD, which has a fog level of 3.69 ALs and 14.1 m visibility. Details of FFD are given in Section 6.1. Reconstructed images of each of the two frigates collected at the maximum fog level are shown in Figure 6.17.



(a) (b)

Figure 6.17: Reconstructed images of the two different frigate silhouettes located at 40 m from the camera, obscured by a fog level of 14.1 m visibility/3.69 ALs (with respect to 532 nm).

By comparison with the imaged wooden frigate silhouettes shown in Figure 6.3 in Section 6.1, it is evident by visual inspection that the reconstructed images of the frigates are of discernible quality, making these images suitable for target detection.

6.5.1.2 Target Classification of Reconstructed Images

Using the approach described in Section 6.2.2, the overall accuracy in classifying the two frigate silhouettes is 93%[9] for data collected at a fog level with visibility of 14.1 m and 3.69 ALs, which is the maximum fog level collected for FFD. The formula for overall accuracy is given in Section 3.7 of Chapter 3.

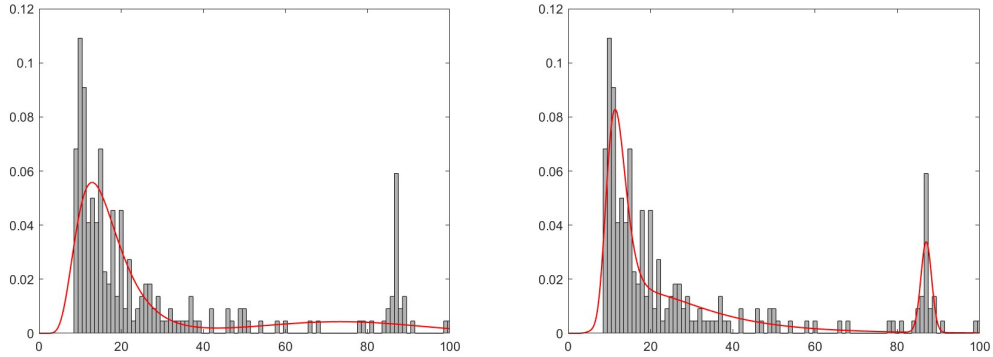
[9]The overall accuracy presented in [100] did not include the Undecided class in its calculation, hence has a different value to this.

There are zero instances where the algorithm classifies an image of these frigates as belonging to the Undecided class. As discussed in Section 6.2.2, this Undecided class is used for classifications where the target's area is equidistant to the mean area values of the two frigates. Therefore, the reason there are zero instances of Undecided may be because the margin between the area values of the two frigates is under 10 pixels, where the area is measured in number of pixels for classification.

6.5.2 Finite Mixture Model Algorithm Performance

To reconstruct each image for evaluation, the finite mixture model algorithm fits a finite mixture model containing up to two lognormal and one Gaussian distributions to each pixel's dToF measurements over a batch of frames. To ensure consistency, data over a moving subset of frames is used for producing each new reconstructed image. As mentioned in Section 6.3.1.1, the Gaussian distribution is assumed to be the final mixture component in the finite mixture model when fitting the data from the SFD dataset.

For fog levels between 0.31 ALs (185 m visibility) and 0.80 ALs (72.9 m visibility), the algorithm reconstructs each image from a batch of 500 frames, using 1 lognormal and 1 Gaussian distributions. For fog levels at 1.01 ALs (57.2 m visibility) and 1.14 ALs (50.8 m visibility), the algorithm reconstructs using a batch of 1,000 frames and utilises 2 lognormal and 1 Gaussian distributions. This adjustment in parameters helps the finite mixture model to better account for varying levels of fog and provides more accurate depth estimation.



(a) 1 lognormal probability distribution (b) 2 lognormal probability distributions

Figure 6.18: Both plots contain a histogram of the dToF measurements of detected photons at pixel (17,17) for a circle target obscured by a fog level with visibility of 57.2 m and 1.01 ALs. They are compared finite mixture models with either (6.18(a)) 1 lognormal and 1 Gaussian distributions or (6.18(b)) 2 lognormal and 1 Gaussian distributions fitted to the same set of dToF measurements.

The algorithm requires more frames at higher fog levels because increased fog density leads to greater photon scattering, resulting in fewer photons returning from

the target. This is evident in Figure 6.18, where it compares the fit of the finite mixture model using different numbers of lognormal distributions. The red line represents the finite mixture model and the grey bars depict the histogram of dToF measurements. The contrast of the two plots in this figure show two lognormal distributions provide a better fit than one. Furthermore, Figure 6.19 illustrates that the image reconstructed from two lognormal distributions has the target exhibit a more discernible quality, which means a better reconstruction performance when evaluated via visual inspection.

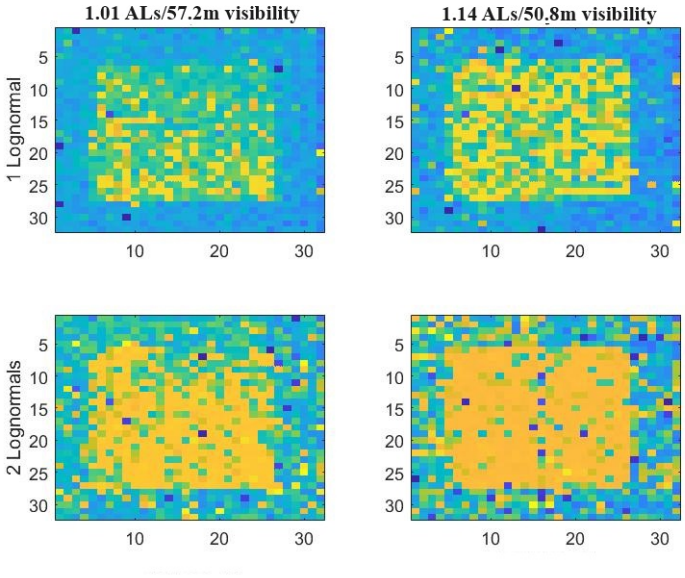


Figure 6.19: Reconstructed images using 1 lognormal and 1 Gaussian (top row) versus 2 lognormals and 1 Gaussian (bottom row) for fog levels at 1.01 ALs/57.2 m visibility (left column) and 1.14 ALs/50.8 m visibility (right column)

The finite mixture model algorithm's reconstruction performance is first evaluated in terms of relative localisation accuracy and is presented in Section 6.5.2.1. This is followed by a brief discussion of the reconstruction performance in terms of visual inspection in Section 6.5.2.2. Then the overall accuracy is presented in Section 6.5.2.3 for target classification of the reconstructed images. Finally, the algorithm's reconstruction performance is evaluated via pixel-wise absolute localisation accuracy, and the results are presented in Section 6.5.2.4. This additional reconstruction performance metric is intended to assess the effectiveness of the finite mixture model in accurately modelling each pixel's dToF measurements to determine the reconstructed dToF value for each pixel. This perspective is not possible with relative localisation accuracy since its reference dToF value is the highest occurring reconstructed dToF value in the reconstructed image, which does not necessarily correspond to the target's expected dToF value. In addition, Section 6.5.2.5 discusses the fit of the finite mixture model to the dToF measurements in more detail.

Ground truth images of each imaged target are used for calculating the pixel-

wise absolute localisation accuracy. Each pixel's dToF value in the ground truth image is used as the expected dToF value for comparing with the corresponding pixel's reconstructed dToF value in the reconstructed image. In addition, ground truth images are used to provide performance comparison for the relative localisation accuracies calculated for the reconstructed images.

To create ground truth images, additional data is collected with the imaged targets in the same imaging set-up as the SFD (see Section 6.1) but in an environment without fog. Each pixel's value in the ground truth image is an average of its dToF measurements over 500 frames, where only dToF measurements that are between 0 to 100 clock cycles are used[10]. This alignment is crucial as the dToF measurements processed by the finite mixture model algorithm are also constrained to 0 to 100 clock cycles before reconstruction.

6.5.2.1 Relative Localisation Accuracy

Relative localisation accuracy is used to evaluate image reconstruction performance because the reconstructed dToF measurements of the target pixels should exhibit consistent relative distances from each other, ideally within the same clock cycle range. The imaged targets in the SFD are cardboard cut-outs and are oriented perpendicular to the optical axis, which means the target pixels should all have the same expected dToF value. The beginning of Section 6.5 provide the definition of relative localisation accuracy.

As the relative localisation accuracy is not evaluating for the absolute localisation accuracy, the target's expected dToF value is not used as the reference value for evaluating how closely the reconstructed dToF values align to. Instead, the highest occurring dToF value in the reconstructed image is used as this reference value. This is because the target typically occupies the majority of the reconstructed image (as seen in Figure 6.20), which results in the highest occurring reconstructed dToF value in the reconstructed image corresponding to the estimated dToF value of the target itself.

This space is intentionally left blank.

[10]The ground truth images used in [99] were not calculated with restricted dToF values. Therefore, the values in Figure 6.20 and Table 6.2 are updated accordingly compared with [99, Section 6.1] and [99, Section 6.3]. Differences only arise for the values related to the last row (i.e. reconstructed images of the square). However, these differences are minimal, with only a maximum of a few percentage points for Figure 6.20 and decimal differences for Table 6.2.

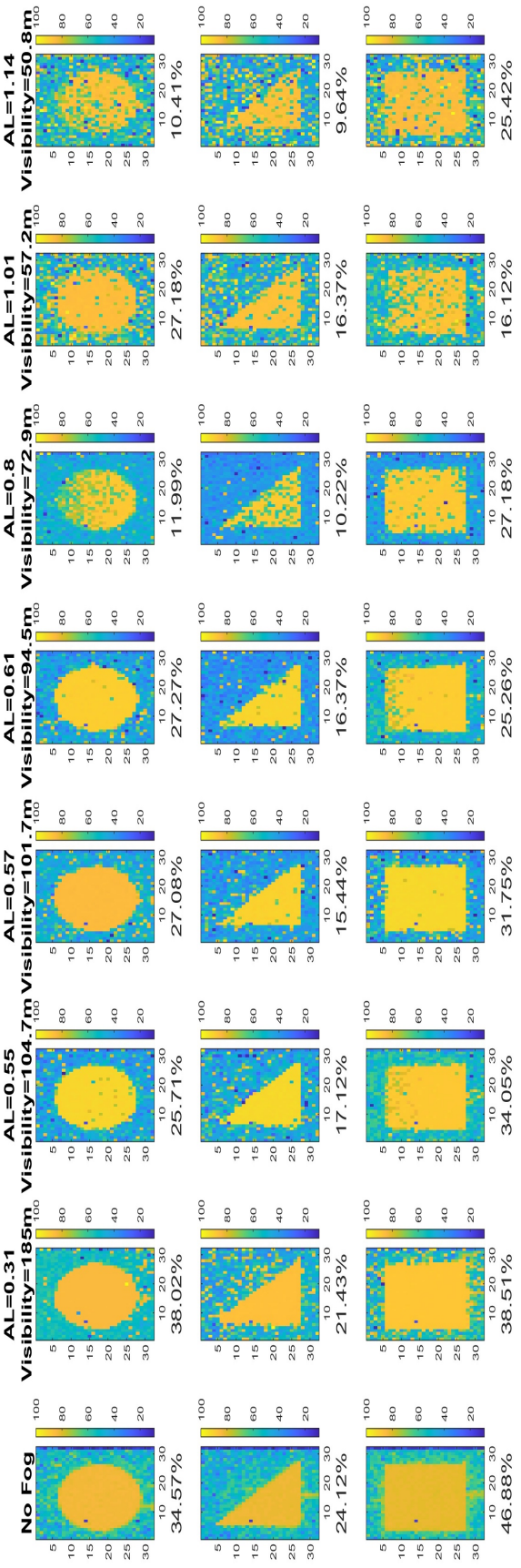


Figure 6.20: Relative localisation accuracy percentages for different shapes and fog conditions.

The relative localisation accuracy percentages for different shapes in different fog conditions are shown in Figure 6.20. Fog conditions vary across columns, increasing from left to right. Each percentage represents an average derived from 10 distinct processed images. While only one image is shown for visualisation purposes, the reconstruction remains consistent across all 10 reconstructed images. The first column presents the relative localisation accuracy percentages for different shapes in the ground truth image, where the image is created from data collected without fog. The percentages presented in this column is only evaluated over one ground truth image. Details of the creation of the ground truth image is provided in Section 6.5.2.

Comparing the relative localisation accuracy percentage value for each shape and fog condition with the relative localisation accuracy percentage value of the corresponding shape's ground truth image provides a relative assessment of the reconstruction quality. The relative localisation accuracy values calculated for the ground truth images serve as the benchmark, representing the optimal relative localisation accuracy percentage achievable by the finite mixture model algorithm. Generally, there is a downward trend in percentages with decreasing fog visibility, aligning with expectations due to heightened scattering in denser fog conditions. However, occasional discrepancies may arise where percentages deviate from this trend. These anomalies could be attributed to imperfect fits of the probability distribution model or the presence of significant noise pixels at the same distance, as observed in the case of the square at 57.2 m visibility and 1.01 ALs. As discrepancies are related to the fit of the finite mixture model, this will be further discussed in Section 6.5.2.4 which discusses the reconstruction performance in terms of the pixel-wise absolute localisation accuracy.

6.5.2.2 Visual Inspection

Despite the decreasing relative localisation accuracy presented in Section 6.5.2.1, the targets are discernible in the reconstructed images, making them suitable for target detection tasks. An example of a reconstructed image for each fog condition and image target is presented in Figure 6.20, and it can be observed that each target shape's silhouette is clearly identifiable in all these images.

6.5.2.3 Target Classification of Reconstructed Images

Target classification performance is reported for its overall accuracy at various experimental fog level (between 185–50.8 m visibility and 0.31–1.14 ALs). The fog levels tested are the same level listed in the two rows of Table 6.1 that are related to the finite mixture model algorithm. At each fog level, 10 different reconstructed images for each shape are used to test the target classification algorithm. The details of the target classification algorithm are provided in Section 6.3.2.

The overall accuracy for classifying different shapes are tested for each fog level, and is 100% for all fog levels. For all fog levels, there are zero instances where the al-

gorithm classifies a shape to be the additional Undecided class[11], which is discussed in Section 3.7 of Chapter 3. This high overall accuracy is attributed to the target's proximity to the SPAD array flash LiDAR system, ensuring its visibility within all experimental fog levels. Despite these promising results, classification performance at lower visibility levels warrants further investigation, particularly where visibility falls below the target's distance. Future work will focus on assessing overall accuracy under such challenging conditions.

6.5.2.4 Pixel-Wise Absolute Localisation Accuracy

The pixel-wise absolute localisation accuracy of a reconstructed image is calculated by determining the percentage of pixels that have their reconstructed dToF values within a predefined threshold of their expected dToF values. For evaluating the finite mixture model algorithm, the pixel's expected dToF value is defined as its corresponding pixel's value in the ground truth image, as discussed in Section 6.5.2. Also, the predefined threshold amount is assigned as 3 clock cycles for evaluating the finite mixture model algorithm.

Table 6.2 presents the pixel-wise absolute localisation accuracy for the reconstructed images of all shapes at various tested fog levels from the SFD dataset. Each percentage is an average calculated from the same set of 10 reconstructed images used in Section 6.5.2.1.

Table 6.2: Pixel-wise absolute localisation accuracy for reconstructed images for each fog level

Number of ALs	0.31	0.55	0.57	0.61	0.80	1.01	1.14
Visibility (m)	185	104.7	101.7	94.5	72.9	57.2	50.8
Circle	47.70%	7.36%	42.02%	7.64%	12.42%	34.26%	19.05%
Triangle	14.22%	10.10%	9.36%	8.13%	7.71%	10.65%	11.90%
Square	7.21%	12.79%	4.85%	10.71%	4.89%	9.49%	13.58%

As observed in Table 6.2, there is no consistency in the pixel-wise absolute localisation accuracy values, and all the percentages fall below 50%. This suggests that the algorithm is not suitable for accurately reconstructing dToF value to localise the target. The discrepancy arises from the imperfect fit of the estimated finite mixture model to the dToF measurements of detected photons. To illustrate this, the fit of the finite mixture model to the dToF measurements is presented in the next subsection.

[11]The classification accuracy in [99] did not include the Undecided class in its calculation. However, because of the 100% accuracy, the inclusion of this extra class does not affect the value.

6.5.2.5 Data Fit of the Finite Mixture Model

Comparing the trends of the histogram of dToF measurements to the finite mixture model provides a means to evaluate the fit of the finite mixture model's probability distribution to the data.

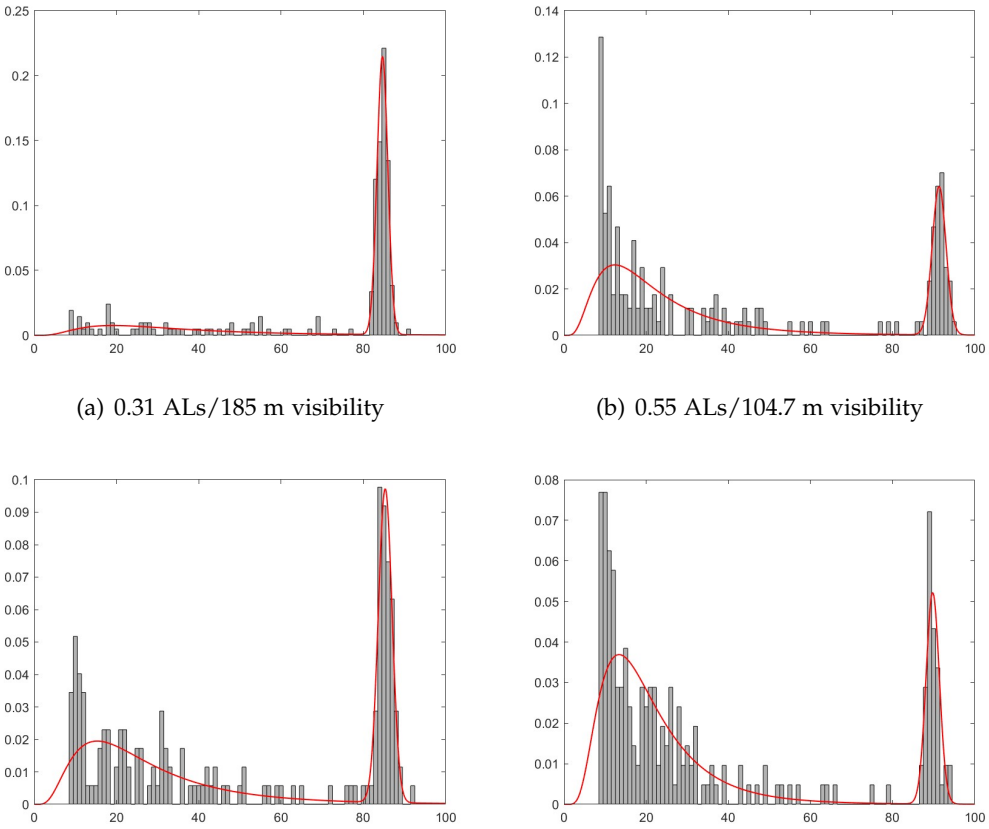
In [99], the probability distribution resulting from the finite mixture model algorithm is compared with the probability distribution fitted by the Kernel Density Estimator (KDE) algorithm. Since KDE is another estimation algorithm, it only provides an approximation to the distribution of the dToF measurements. Therefore, it is more appropriate to directly employ the histogram of dToF measurements, which is an exact representation of the distribution of dToF measurements.

Figure 6.21 comprises a series of plots[12] comparing the fitted finite mixture model's probability distribution with the histogram of dToF measurements, with each plot based on dToF measurements collected by pixel (17,17) in a different fog level. At each fog level, the histograms and probability distributions of the dToF measurements at pixel (17,17) are representative of the histograms and probability distributions of the dToF measurement at other target pixels. The counts on each histogram are normalised against their maximum value, converting the counts into probability values for comparability with the finite mixture model.

In each plot in Figure 6.21, the histogram peak at the beginning corresponds to dToF measurements of detected photons reflected from fog, while the histogram peak at the end relates to dToF measurements of detected photons reflected from the target. This comparison provides insight into the accuracy of the model's representation of the underlying value distribution of the dToF measurements.

This space is intentionally left blank.

[12]The plots are displayed over two pages.



(a) 0.31 ALs/185 m visibility

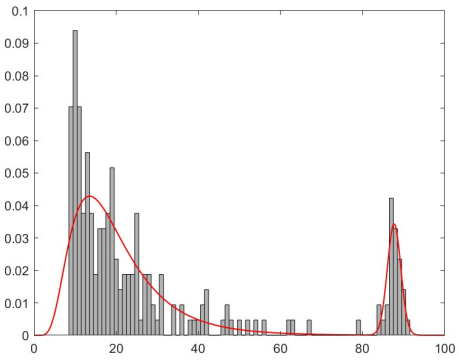
(b) 0.55 ALs/104.7 m visibility

(c) 0.57 ALs/101.7 m visibility

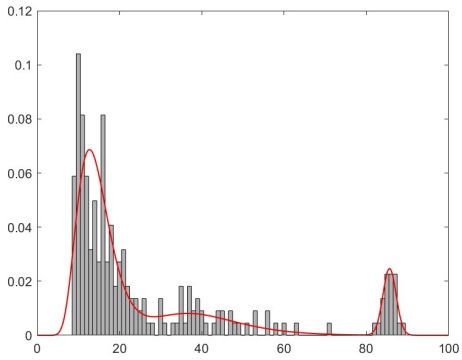
(d) 0.61 ALs/94.5 m visibility

Figure 6.21: The finite mixture model algorithm is fitted to the dToF measurements of detected photons at pixel (17,17) for the circle target obscured by different fog levels. Pixel (17,17) is located on the target in all the reconstructed images for all the fog levels. The fitted finite mixture model is the red line in each of the plots and they are compared to the set of grey bars, which is the histogram of dToF measurements with its count values normalised. The x-axis is number of clock cycles and y-axis is probability of photon detection. Figure (f) and (g) are fitted using a batch of 1,000 frames and a mixture of two lognormal and one Gaussian distributions, while the other figures are fitted using a batch of 500 frames and a mixture one lognormal and one Gaussian distributions.

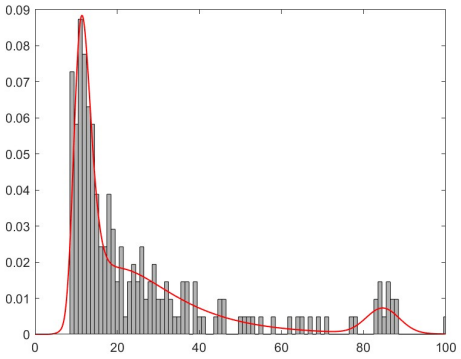
This space is intentionally left blank.



(e) 0.80 ALs/72.9 m visibility



(f) 1.01 ALs/57.2 m visibility



(g) 1.14 ALs/50.8 m visibility

Figure 6.21: The finite mixture model algorithm is fitted to the dToF measurements of detected photons at pixel (17,17) for the circle target obscured by different fog levels. Pixel (17,17) is located on the target in all the reconstructed images for all the fog levels. The fitted finite mixture model is the red line in each of the plots and they are compared to the set of grey bars, which is the histogram of dToF measurements with its count values normalised. The x-axis is number of clock cycles and y-axis is probability of photon detection. Figure (f) and (g) are fitted using a batch of 1,000 frames and a mixture of two lognormal and one Gaussian distributions, while the other figures are fitted using a batch of 500 frames and a mixture one lognormal and one Gaussian distributions.

With increasing fog levels, the histogram peak related to detected photons reflected from fog increases in height in all the plots in Figure 6.21. Conversely, the histogram peak related to detected photons reflected from the target decreases in height. This decrease causes the finite mixture model algorithm to "miss" the target's histogram peak. Mathematically, this indicates that the finite mixture model fails to accurately model the dToF measurements reflected from the target, as there are insufficient amount of dToF measurements to identify a distinct cluster. This issue is illustrated in Figure 6.22.

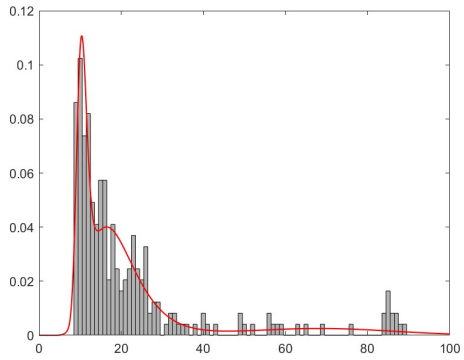


Figure 6.22: Comparing the finite mixture model (red line) with the histogram of dToF measurements. This is for dToF measurements collected by 1,000 frames of a square obscured by a fog level of 1.14 ALs/50.8 m visibility. The finite mixture model is not able to model the peak at around 83 to 93 clock cycles, which is the accumulation of dToF measurements of detected photons reflected from the square (which is the target).

The Gaussian distribution used in the finite mixture model algorithm fails to adequately capture the target's histogram peak, leading to inaccuracies in estimating the mean of the Gaussian distribution, which represents the reconstructed dToF value. This phenomenon occurs randomly due to the iterative nature of the EM algorithm used for parameter estimation. However, these occurrences become more frequent with increasing fog levels.

Since this algorithm reconstructs dToF values for each pixel, the increased occurrence of inaccuracies results in more pixels having inaccurately reconstructed dToF values. Consequently, the reconstructed images at higher fog levels exhibit less clarity of the target, as demonstrated by the images in Figure 6.20.

6.5.3 Fourier Descriptor Algorithm

As the Fourier descriptor algorithm estimates the dToF value of the global target for image reconstruction, it is evaluated for global target localisation using absolute localisation accuracy.

Reconstructed images of the target are not presented here because the reconstructed image would simply be a conversion from the slice's binary image containing the target. Since the algorithm achieves global target localisation by exploiting the target's silhouette, a high absolute localisation accuracy would indicate that the target's silhouette will also be clearly visible in the reconstructed image. Additionally, the image reconstruction performance with respect to relative localisation accuracy is inherently 100%, as the reconstructed image is converted from a slice of the binary histogram stack. As mentioned before, all the non-zero pixels in the binary image are associated with the target and are converted to the same reconstructed dToF value for the reconstructed image.

The data used to evaluate the absolute localisation accuracy of the Fourier descriptor algorithm are from the SFD at two different fog conditions: 43.8 m visibility/1.33 ALs and 25.4 m visibility/2.28 ALs[13]. In particular, the data associated with the right-angled triangle is utilised. The three vertices of this triangle are at coordinates (6,5), (6,27), and (28,27). These vertices are determined from inspecting a reconstructed image of the same right-angled triangle from the SFD imaged without fog, but under the same imaging set-up. In this reconstructed image, each pixel's reconstructed dToF value is the average of the dToF measurements collected over 1,000 frames.

For removing noise and filling holes in the slices of the binary histogram stack, empirical testing has determined that a fraction of 0.02 is effective for a fog level with 43.8 m visibility and 1.33 ALs, while a fraction of 0.035 works well for a fog level of 25.4 m visibility and 2.28 ALs.

The target is positioned 44.6 m away, as depicted in Figure 6.2, corresponding to an expected dToF value of 85 clock cycles (refer to Section 3.4.2 of Chapter 3 for the timing delay correction used to convert clock cycle values to distance values).

When using Eq. 6.17 from Section 6.4.2 to calculate the apparent area of this triangle at different dToF slices, the target's area at its expected dToF value is determined from the same batch of 1,000 frames of non-obscured data. It calculates the area of the target at the slice associated with the expected dToF value of 85 clock cycles. Hence p in Eq. 6.17 is defined to be 85 clock cycles.

6.5.3.1 Comparing Global Target Absolute Localisation Accuracies

The Fourier descriptor algorithm's absolute localisation accuracy of the global target location is evaluated on 100 global target location values determined by the algorithm processing 100 batches of frames. This is conducted for each of the two fog levels described in the previous section and in Table 6.1. Batches of frames are chosen to be evenly separated across the data collected over all the frames in the SFD for the two chosen fog levels (as shown in Table 6.1).

For the higher fog condition (43.8 m visibility and 1.33 ALs), batches of 1,000 frames are used by the Fourier descriptor algorithm, and a fraction of 0.035 is utilised for calculating the threshold for removing noise and holes (see Section 6.4.2 for explanation of the utilisation of this threshold value). Conversely, for the lower fog condition (25.4 m visibility and 2.28 ALs), 500 frames are used for each batch, and a fraction of 0.025 is employed since there is less noise in the data.

Figure 6.23(a) and 6.23(c) present the histogram of the 100 global target location values computed by the Fourier descriptor algorithm for the lower and higher fog condition, respectively.

To demonstrate performance improvement by the Fourier descriptor algorithm, its global target locations are compared with the global target locations of the finite mixture model algorithm. Here, the finite mixture model algorithm employs a

[13]These number of ALs values are updated from the values reported in [104].

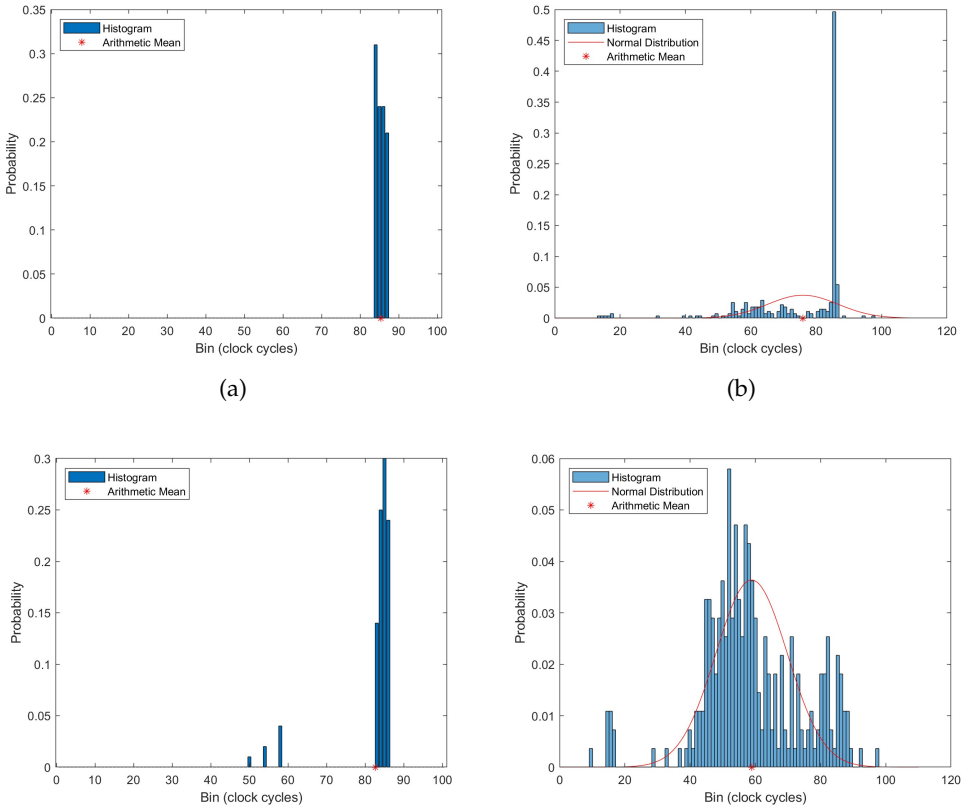


Figure 6.23: Comparison of the histogram of global target location values from the Fourier descriptor algorithm ((a) & (c)) and the finite mixture model algorithm presented in Section 6.3 ((b) & (d)). In the top row, the visibility is 43.8 m and the fog level is 1.33 ALs. For the bottom row, the visibility is 25.4 m and the fog level is 2.28 ALs. The red asterisk on the x-axis represents the arithmetic mean of each algorithm. Their values are 85.35 (a), 75.93 (b), 82.66 (c) and 58.86 (d). The Gaussian distribution is plotted in (b) and (d), where it is associated with the arithmetic mean of the finite mixture model algorithm's global target location values. Their variances are 115.42 and 120.27 respectively.

mixture of two lognormal and one Gaussian distributions for fitting the dToF measurements. As discussed in Section 6.3.1, the Gaussian distribution's mean for each pixel corresponds to the target's localised distance at that pixel. As the imaged target is planar, these locations from each pixel can be treated as global target locations for comparison with the Fourier descriptor algorithm's global target locations. The histograms in Figure 6.23(b) and 6.23(d) are of the global target location values from the finite mixture model algorithm, as presented in Section 6.3.

From the discussion of the finite mixture model algorithm's reconstruction performance in Section 6.5.2, it is known that the algorithm requires more frames at

higher fog levels because of the increased fog scattering. The fog levels reported here are higher than the levels reported for the finite mixture model algorithm in Section 6.5.2. Therefore, all of the collected data for each of the fog levels are used by the finite mixture model algorithm to reconstruct the dToF value of the target for each pixel. The total number of frames in the collected data is 1,500 for the higher fog level and 1,900 for the lower fog level.

The finite mixture model algorithm cannot determine which pixel in the raw frame is a target pixel, unlike the Fourier descriptor algorithm. To provide a fair comparison, the finite mixture model algorithm is applied to reconstruct dToF measurements of the target only for pixels within the right-angled triangle with vertices at coordinates (6,5), (6,27), and (28,27). As discussed in Section 6.5.3, this right-angled triangle and its vertices are identified from inspecting a reconstructed image of the right-angled triangle from SFD imaged under the same imaging set-up but without fog. Details of this reconstructed image is provided in Section 6.5.3. From this reconstructed image, it can be deduced that there are 276 target pixels. This results in the finite mixture model algorithm having 276 global target location values for each fog level.

Figure 6.23(b) and 6.23(d) present the histogram of the 276 global target location values computed by the finite mixture model algorithm. The red asterisks represent the values' arithmetic mean.

As each global target location value from the finite mixture model algorithm is from a mean of a Gaussian distribution, the arithmetic mean of these values also follows a Gaussian distribution. The variance of this arithmetic mean is calculated from the algorithm's 276 global location values. The resulting Gaussian distribution of these global target values is plotted as a red line over the histograms in Figure 6.23(b) and 6.23(d) to illustrate that the finite mixture model algorithm provides not only a point estimate but an entire distribution of the global target location.

To provide a comparison to the finite mixture model algorithm's arithmetic mean, the arithmetic mean of the Fourier descriptor algorithm's 100 global target location values is also computed and plotted as a red asterisk over the histograms in Figure 6.23(a) and 6.23(c).

The results, presented in Figure 6.23, highlight a significant enhancement in absolute localisation accuracy for the higher fog level (25.4 m visibility/2.28 ALs) and an improved accuracy for the lower fog level (43.8 m visibility/1.33ALs) by the Fourier descriptor algorithm. The Fourier descriptor algorithm provides a better global target localisation performance than the finite mixture model algorithm despite employing data collected over fewer frames in its image-reconstruction process.

6.6 Chapter Summary

Three different algorithms are presented for image reconstruction of single targets that are obscured by fog when imaged by a SPAD array flash LiDAR system. By visual inspection, the range gate and process algorithm is demonstrated to reconstruct

discernible images of wooden frigate silhouettes at a fog level of 14.1 m visibility and 3.69 ALs using data collected over a batch of 500 frames. This demonstrates target detection is possible by visual inspection. The classification of the reconstructed images achieves an overall accuracy of 93%. Classification is done by comparing the sum of non-zero pixels in the reconstructed image with each frigate's mean area, which is calculated with a small set of training data. In addition, computational demand analysis highlights that the algorithm uses a small amount of steps and processes a small amount of data, making it real-time capable.

The finite mixture model algorithm has decreasing reconstruction performance when evaluated in terms of relative localisation accuracy. However, when it is evaluated by visual inspection, all its reconstructed images are discernible for fog levels up to 50.8 m visibility/1.14 ALs, making these images applicable for target detection. The algorithm processes data collected over a batch of 500 frames per reconstructed image. The overall accuracy of classifying the imaged targets from these reconstructed images is 100% for all tested fog levels. Target classification converts the reconstructed image to a binary image, then, after applying some additional denoising steps, uses sum of non-zero pixels to classify the shape.

The pixel-wise absolute localisation accuracy is poor for the finite mixture model algorithm. This may be due to the pixel-wise approach of this algorithm leading to inconsistencies. Through computational demand analysis, it is shown that a state-of-the-art GPU such as the Nvidia Jetson TX2 module will be able to execute this algorithm in real time. In particular, it will be able to execute the approximately 1.8 million multiplication operations required per pixel for this algorithm. The algorithm's remaining operations are demonstrated to require far less time in comparison, ensuring that this algorithm is real-time capable.

The Fourier descriptor algorithm utilises the dToF measurements registered by all pixels collectively to perform global target localisation. By utilising a GPU onboard an embedded board, the parallelisation of this algorithm's computationally expensive tasks (such as histogramming and generating boundary trace) and using the algorithm to only process a small amount of frames will ensure this algorithm is real-time capable.

The Fourier descriptor algorithm is demonstrated to localise a triangle imaged by a SPAD array flash LiDAR system for a fog level of up to 25.4 m visibility/2.28 ALs by processing data collected over a batch of 1,000 frames. The algorithm's output global target locations are compared with those determined using the finite mixture model algorithm on data of a planar target in a fog level of 25.4 m visibility/2.28 ALs. This is at a higher fog level than previously evaluated by the finite mixture model algorithm. The comparison demonstrates that the Fourier descriptor algorithm has a higher global target absolute localisation accuracy than the finite mixture model algorithm. This implies that the Fourier descriptor algorithm will be able to provide a better image reconstruction performance.

While target classification performance is not tested, the use of Fourier descriptors in this algorithm suggests potential for extension to target classification, given that Fourier descriptors are unique for different geometric shapes, and can be extended

to be size- and rotation-invariant.

While reconstructed images are not demonstrated for the Fourier descriptor algorithm, the images will be a simple conversion of the generated slice's binary image. As this algorithm requires the binary image to contain a discernible representation of the target in order to calculate an accurate Fourier descriptor for global target localisation, a high global target localisation accuracy implies that the target is discernible in the binary images. This demonstrates that images reconstructed using the binary images from the Fourier descriptor algorithm will be suitable for target detection and classification.

Chapter 7

Conclusion

Real-time-capable target-discrimination algorithms have been presented in this thesis for processing data collected by a Single Photon Avalanche Diode (SPAD) array flash Light Detection and Ranging (LiDAR) system of single targets situated in three different types of environments: air, natural waters and fog. A real-time target classification algorithm, called the random feature-detecting network, is presented in Chapter 4 for classifying live data of four different model airplanes (Tu-128, Su-35, Su-24, and MQ-9), collected with a SPAD array flash LiDAR system in clear air conditions in a laboratory setting. The initial stages of this algorithm successfully perform image reconstruction and target detection. The target classification performance of the random feature-detecting network has an overall accuracy of up to 99.35%, with an F1-score reaching 0.99.

The random feature-detecting network is implemented with a live data acquisition module into an embedded software program to execute in real time on the Jetson TX2. The embedded software program utilises multithreading to implement the target classification algorithm in a separate thread to the live data acquisition module, which enables faster execution. The power consumption of the embedded software program is 5.1 W. The random feature-detecting network's execution time on the embedded software implementation ranges between 30.18 and 65.07 ms. These performance metrics demonstrates the feasibility of deploying the embedded software implementation on a Jetson TX2, with its low Size, Weight and Power (SWaP) profile rendering it suitable for edge computing applications on compact mobile platforms within a distributed network.

The remainder of this thesis presents image-reconstruction algorithms where the reconstructed images can be used for target detection.

Two basic image-reconstruction algorithms, the detection threshold and median algorithm and the histogram averaging algorithm, are presented in Chapter 5 for collected SPAD LiDAR data of submerged targets in natural waters. Both are real-time-capable algorithms, as demonstrated by the computational demand analysis which indicates these algorithms use a small number of computational steps and processes a small amount of data (i.e.200-300 frames). The reconstruction performance of these two algorithms highlights a trade-off of increasing real-time capability in lieu of effective image reconstruction performance at higher number of ALs or deeper target depths. Datasets collected from three different locations are used to evaluate the al-

gorithms' image reconstruction performances: Australian Institute of Marine Science (AIMS) indoor saltwater tank, Defence Science and Technology Group (DSTG) indoor freshwater tank, and the Port Adelaide River. A novel imaging set-up is presented to conduct above-water imaging at the latter two locations.

The detection threshold and median algorithm's reconstructed images are suitable for detection of single targets imaged from a horizontal imaging set-up in the AIMS indoor saltwater tank. Its reconstructed images are discernible for a water turbidity level of up to 3.43 attenuation lengths (ALs), where the saltwater contains 2.44 µg/L chlorophyll and 5.71 mg/L sediments. In addition, the overall target classification accuracy using this algorithm's reconstructed images for sediment-only saltwater conditions is above 95% . The tested sediment-only saltwater conditions have up to 5.62 mg/L of sediments (equivalent to water turbidity level of 3.29 ALs).

The reconstructed images by the histogram averaging algorithm provides a better representation of the targets than the detection threshold and median algorithm for datasets collected from an above-water imaging set-up at the DSTG indoor freshwater tank and the Port Adelaide River. This makes the histogram averaging algorithm more suitable for reconstructing images for above-water detection of submerged targets in natural water locations.

Nonetheless, the histogram averaging algorithm is unable to reconstruct discernible images of a target submerged beyond a depth of 2 m in the Port Adelaide River. To enable discernible images of greater depth targets to be reconstructed, two streams of work are required. First, it is necessary to develop a more advanced image-reconstruction algorithm, which will require a better understanding of the complex and variable nature of natural waters. Second, a better SPAD array flash LiDAR imaging set-up must be created to ensure reliable data collection at natural water locations, where the target location can be reliably validated.

Three different image-reconstruction algorithms are presented in Chapter 6 for targets obscured by fog and imaged by a SPAD array flash LiDAR system. The range gate and process algorithm successfully reconstructs images of two different wooden frigate silhouettes obscured by a fog level of up to 14.1 m visibility and 3.69 ALs. This demonstrates that these reconstructed images can be utilised for target detection. The overall accuracy of target classification on all reconstructed images from this algorithm is 89.5%. However, the effectiveness of this algorithm relies on fog having a limited span, which is then exploited by setting a range gate in the SPAD array flash LiDAR system during data collection to avoid detecting photons reflected from the fog, thereby providing a clearer image of the target. Consequently, a more robust algorithm is necessary for scenarios where fog persists over longer distances, obscuring the target's location.

The finite mixture model algorithm successfully reconstructs images of three different shapes from SPAD LiDAR data captured through fog using a SPAD array flash LiDAR system without a range gate set, in a fog level of up to 50.8 m visibility and 1.14 ALs. The reconstructed images provides discernible images of the targets, making them suitable for target detection tasks. Overall shape classification accuracy of the finite mixture model algorithm's reconstructed images is 100% for different fog

levels up to 50.8 m visibility and 1.14 ALs.

The computational efficiency of the Expectation-Maximization (EM) algorithm is leveraged by the finite mixture model algorithm to reduce the computational time of fitting a finite mixture model to a set of direct Time-of-Flight (dToF) measurements. Moreover, a computational demand analysis is conducted and demonstrates that the finite mixture model algorithm is real-time-capable if implemented on an embedded board with a GPU, such as the Jetson TX2.

The histogram of dToF measurements collected in fog shares the two-peaks characteristics with dToF measurements obtained through water. In particular, the first peak originates from dToF measurements of the obscurant and the second peak originates from dToF measurements of the target. This implies that it may be possible to extend the finite mixture model algorithm to reconstruct images of single targets in natural water environments.

To enhance the fog level at which a target can be detected through effective image reconstruction of data collected by a SPAD array flash LiDAR system, the Fourier descriptor algorithm is the last algorithm presented in Chapter 6. Unlike the previous pixelwise approaches, this algorithm analyses the dToF measurements registered by all pixels collectively. Its focus is in global target localisation, where it utilises an image feature called Fourier descriptor to determine the most likely location of the target. Image reconstruction is possible by employing a simple conversion of the binary images is associated with the Fourier descriptor most similar to the target's expected Fourier descriptor. The computational demand analysis highlights that the Fourier descriptor algorithm is real-time-capable if it leverages the parallel computation capability of a GPU. Most of its computational expensive tasks can either be parallelised or the task is designed to only process a small amount of data.

The global target localisation performance of this algorithm is more accurate than the global target location values inferred from the finite mixture model algorithm, for data collected in fog levels up to 25.4 m visibility and 2.28 ALs, which is at a higher than the maximum fog level at which the finite mixture model algorithm can reconstruct a discernible image of the target.

Target classification was not demonstrated for the reconstructed images of the Fourier descriptor algorithm because it has only been evaluated for reconstructing images of one shape. Nonetheless, the algorithm holds promise for extension to shape classification. This is because Fourier descriptors exhibit unique characteristics for different geometric shapes.

7.1 Future Work

There are several future directions for the studies presented in this thesis. Firstly, Section 7.1.1 discusses potential enhancements for the Fourier descriptor algorithm presented in Section 6.4 of Chapter 6. It includes a proposal for a hybrid approach combining the finite mixture model algorithm (presented in Section 6.3 in Chapter 6) and the Fourier descriptor algorithm. Secondly, there is scope for extending the

imaging of underwater targets to greater depths. This is elaborated on in Section 7.1.2. Finally, a broader research question comparing the two different approaches (physics vs phenomenological) for the modelling of dToF measurement is discussed in Section 7.1.3.

7.1.1 Better Target Classification in Fog

The Fourier descriptor algorithm developed for fog (in Section 6.4 of Chapter 6) can undergo several improvements. Firstly, employing a more robust boundary-tracing algorithm would enable a more accurate capture of the shape's silhouette, thus extending the algorithm's applicability to dToF measurements collected at fog levels exceeding the algorithm's current maximum reconstruction fog level of 25.4 m visibility or 2.28 ALs. Additionally, an enhanced boundary-tracing mechanism would yield more precise Fourier shape descriptors, facilitating improved comparison with the target's expected Fourier descriptor.

Secondly, this algorithm could be expanded into a classifier for different shape targets by utilising distinct Fourier shape descriptors for target localisation.

Thirdly, its localisation performance could be further enhanced by integrating the finite mixture model algorithm described in Section 6.3 of Chapter 6. The finite mixture model contains information regarding the fog and target locations for each pixel. Leveraging this information could eliminate slices associated with certain dToF values from undergoing Fourier shape descriptor calculation, thereby speeding the algorithm's computation.

7.1.2 Imaging into Natural Waters

To extend SPAD array flash LiDAR imaging to deeper waters (beyond 10 m), several challenges must be addressed. Firstly, a deeper understanding of Inherent Optical Properties (IOPs) for various natural water constituents is essential. This can be achieved through empirical measurements or by extrapolating theoretical relationships from existing literature. Preliminary research on the impact of natural water properties on IOPs is available in [101], although it is not included in this thesis.

Secondly, acquiring a more powerful laser is necessary. Natural water induces significant laser scattering, so the use of a more potent laser would ensure that at least a few photons reach the target and are detected by the SPAD camera.

Thirdly, the development of a more sophisticated algorithm is crucial for extracting target information from dToF measurements made by a SPAD array flash LiDAR system. This new algorithm should focus on modelling photon travel through water utilising IOPs measurements. By doing so, it would create a better model for the dToF measurements and provide better image reconstruction of the submerged target.

7.1.3 Physics vs Phenomenological

While developing algorithms for dToF measurements of obscured targets collected by a SPAD array flash LiDAR system, it becomes evident that existing image-reconstruction algorithms typically adopt one of two approaches for depth estimation of the target. They either employ a physics model of photon detection and propagation or investigate the phenomenology of the detected photons' dToF measurements. This prompts a future research question: in which medium does the accuracy gain in reconstructed dToF values from modelling the physics of photon propagation and detection outweigh the gain in processing speed from processing the dToF measurements via the phenomenological perspective?

I hypothesise that mediums such as air, fog, and shallow clear natural water benefit more from the phenomenological approach. Based on the characteristics dToF measurements presented in Section 5.5.2 in Chapter 5 and Section 6.5.2.5 in Chapter 6, the dToF measurements of detected photons reflected by the target can be easily distinguished from the dToF measurements of the background for data collected in these mediums. Therefore, a phenomenological approach is possible because an algorithm can be applied to autonomously distinguish between target and background's dToF measurements based on the distribution of dToF measurements. In contrast, mediums with more obscurant (e.g. turbid natural water) benefit more from a physics model. This is because these mediums have far more light scattering properties than mediums such as air, fog and shallow clear natural water. Therefore, the distribution of dToF measurements of detected photons in the latter mediums may exhibit less structure. Consequently, employing the physics model may become critical to identify detected photons as noise and filter them out to ensure effective image reconstruction for accurate target detection and classification.

Appendix A

Additional Details for the SPAD array Flash LiDAR Systems

This chapter presents additional details of the different Single Photon Avalanche Diode (SPAD) array flash Light Detection and Ranging (LiDAR) systems presented in Table 3.1 in Section 3.4 of Chapter 3.

A.1 System B laser remarks

Telescope expander for Bright Solutions Laser A microscope objective is selected such that the output beam from the laser will almost fill the 145 mm output lens.

A.2 System C laser remarks

Diffuser for laser A diffuser is placed in front of the laser to make the system eye safe and this results in a laser beam divergence of approximately 115°. The diffuser is a circle of tracing paper placed within a simple beam expander consisting of two negative lenses spaced 40 mm apart. This gives a good even illumination at an energy level of 10 mJ (measured over a 70 degree hemisphere).

Appendix B

Images of Targets Submerged in the AIMS Indoor Saltwater Tank

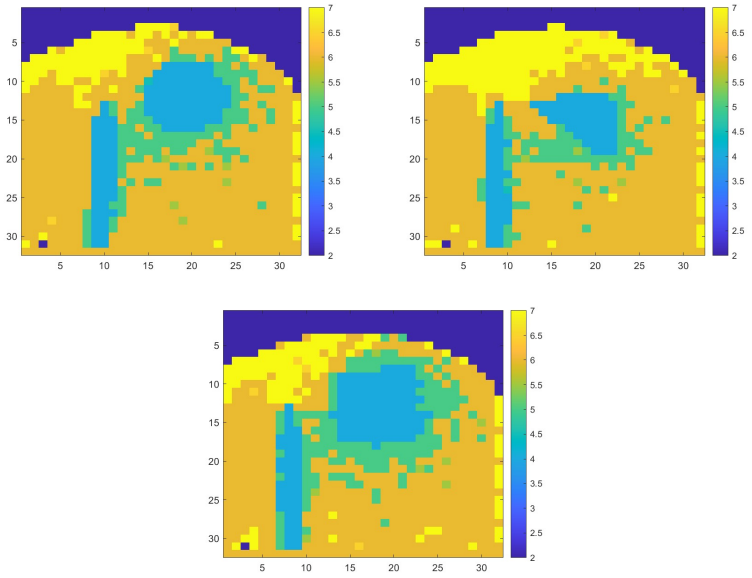
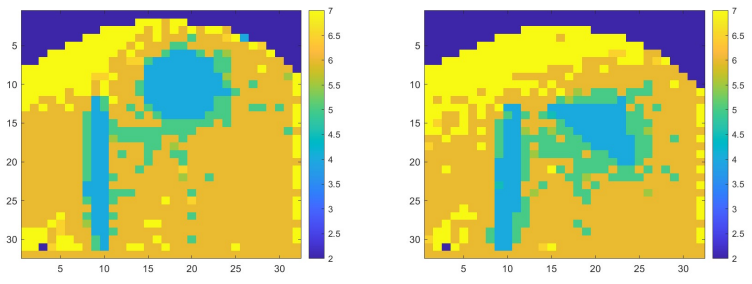
This chapter shows an extended list of the reconstructed images and denoised binary images discussed in Section 5.4 of Chapter 5. Each image shown in this Appendix is created from a batch of 200 frames. Each section shows images processed from data collected in saltwater conditions containing different types of marine constituents: sediments, chlorophyll, and mixture of sediments and chlorophyll.

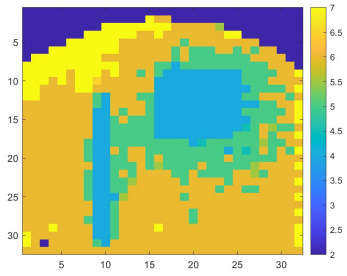
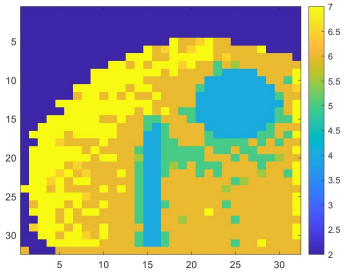
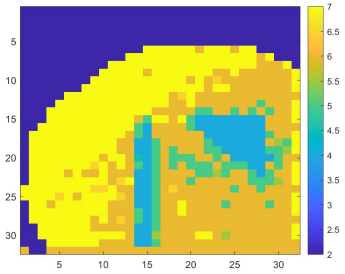
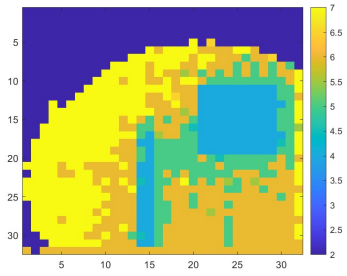
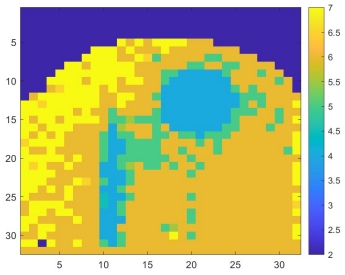
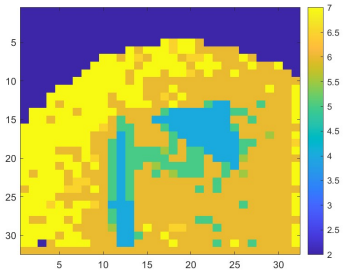
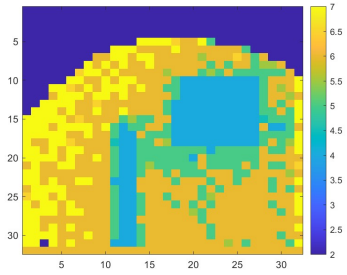
This space is intentionally left blank.

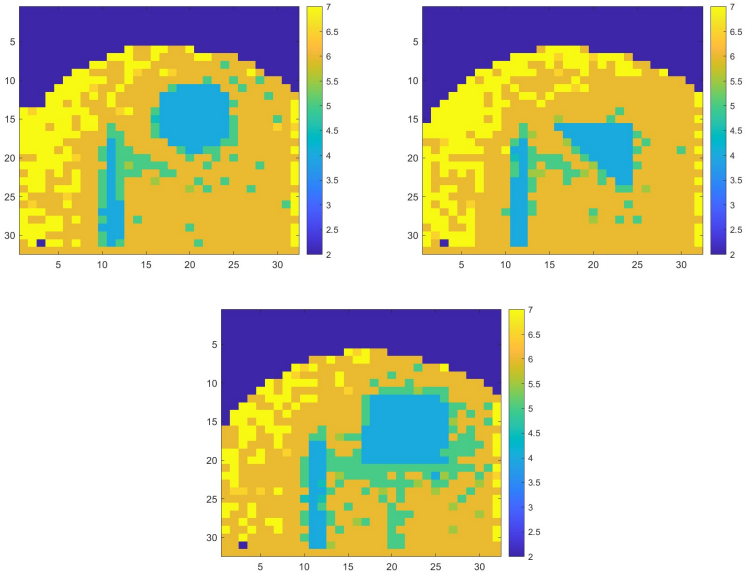
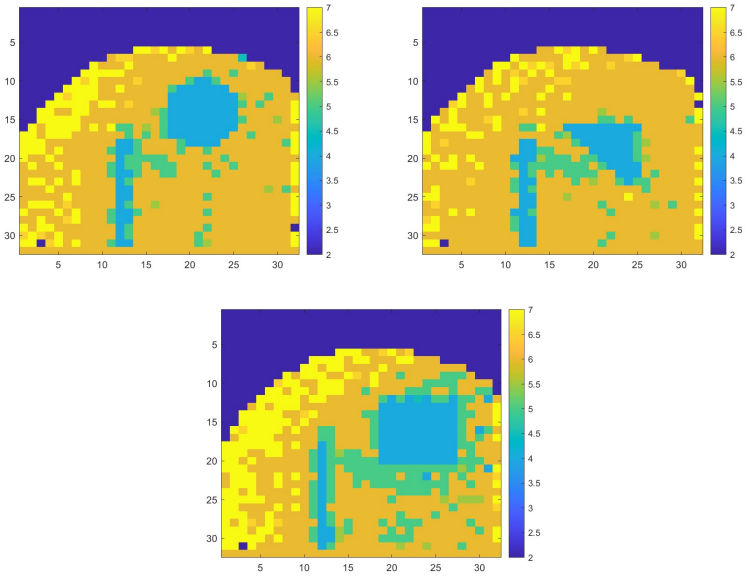
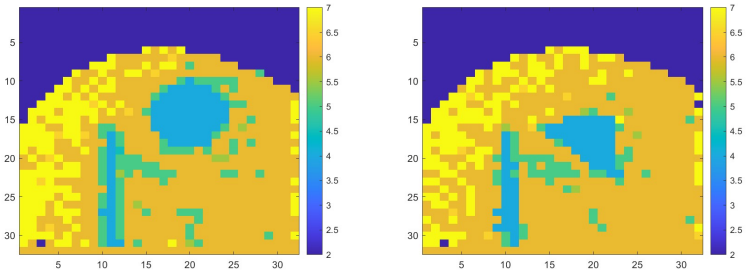
B.1 Sediments

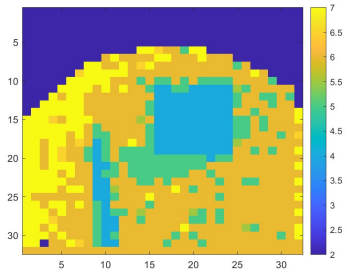
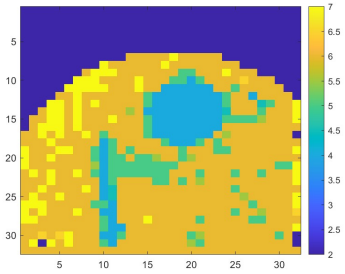
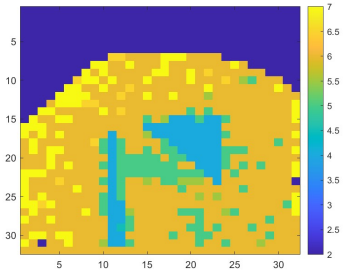
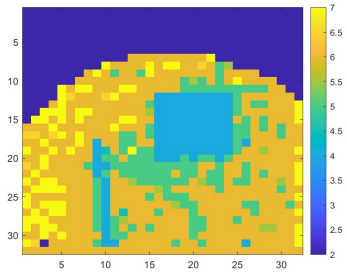
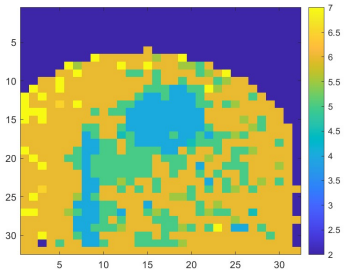
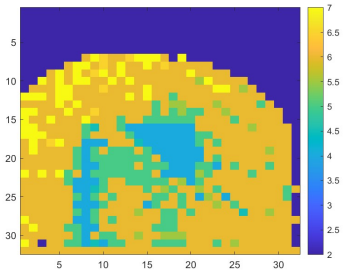
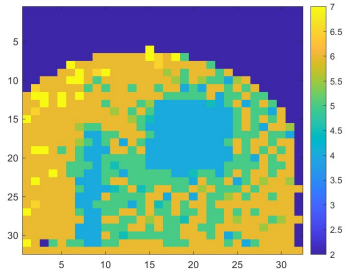
Table B.1 shows the reconstructed images at various sediment conditions. Afterwards, Table B.2 shows the denoised binary images at various sediment conditions. The red box in each of the denoised binary images show the segmented image determined by the target detection algorithm described in Section 5.4.3.2 of Chapter 5.

Table B.1: Reconstructed images of various shapes at different sediment concentrations.

Sed. Conc. (mg/L)	**No. of ALs (w.r.t. 532nm)**	**Reconstructed Image**
1.30	1.32	
1.78	1.40	

		
2.74	1.84	  
3.22	2.03	  

3.70	2.35	
4.18	2.41	
4.66	2.57	

		
5.14	2.83	  
5.62	3.29	  

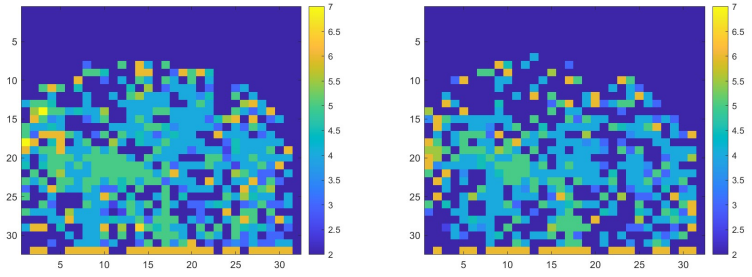
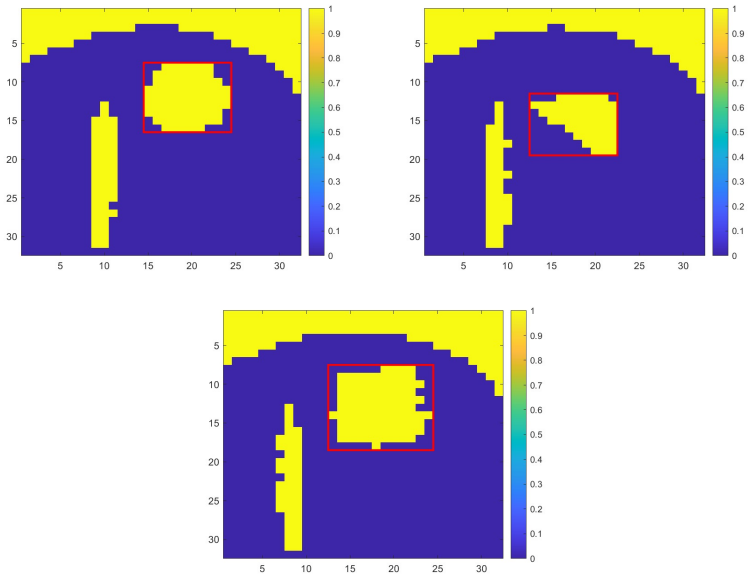
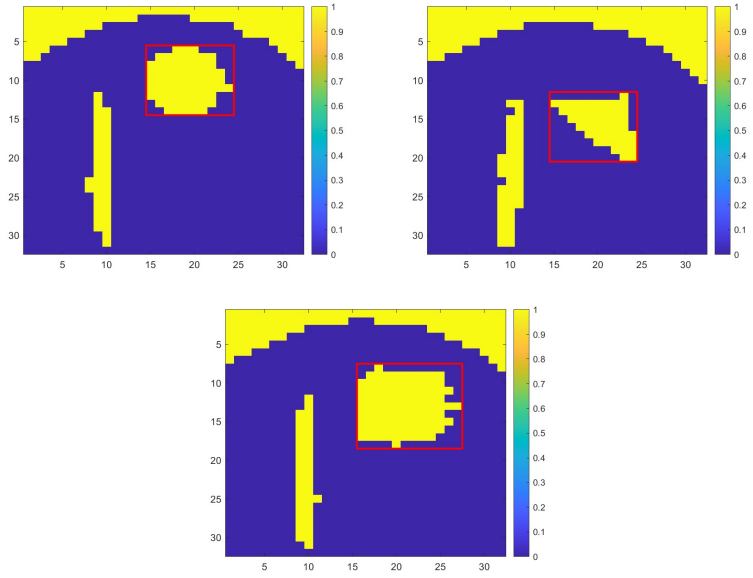
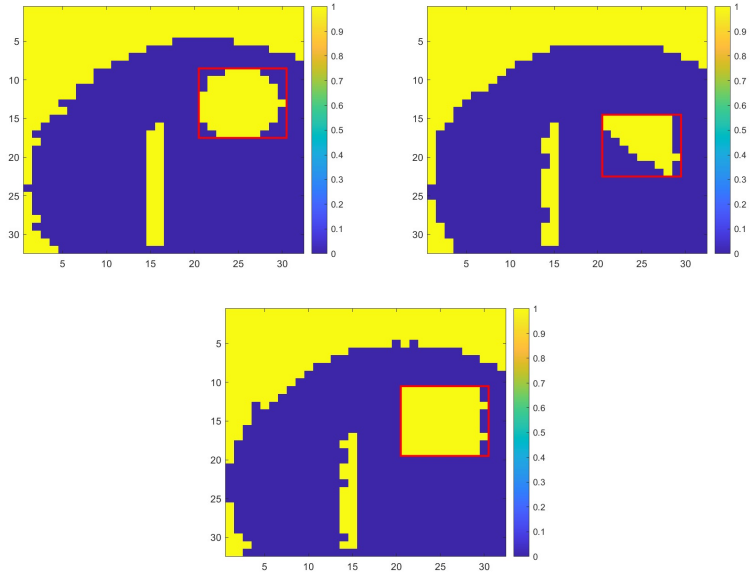
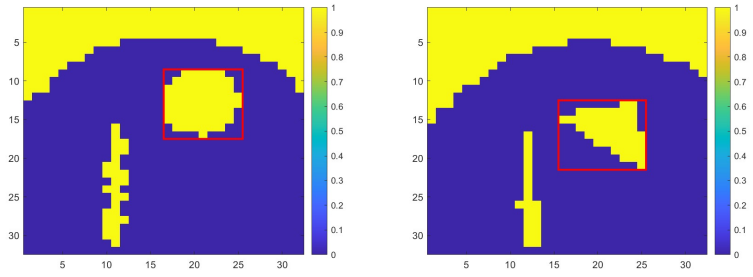
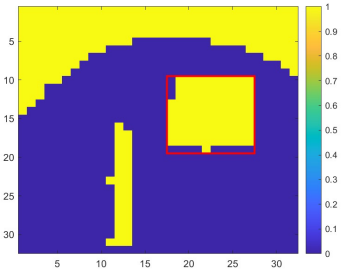
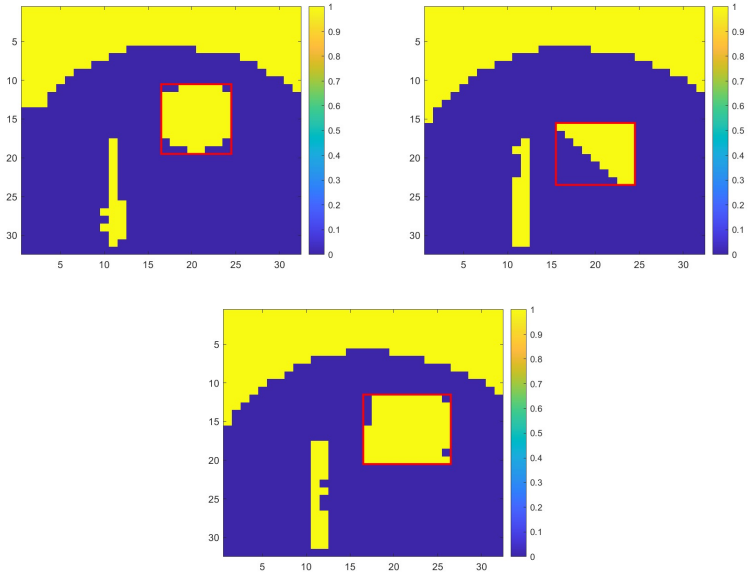
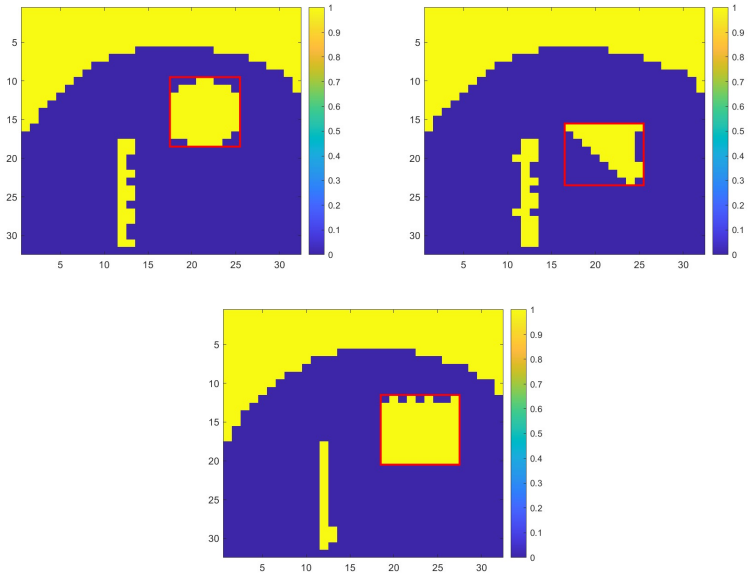
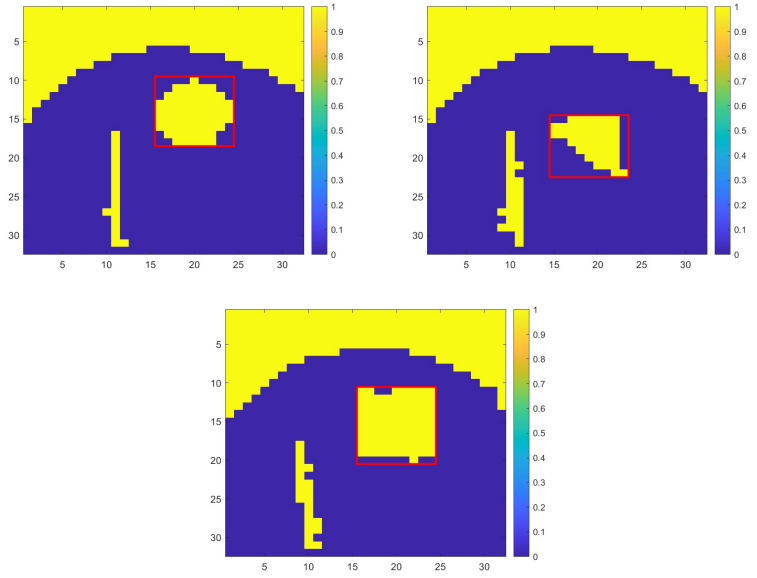
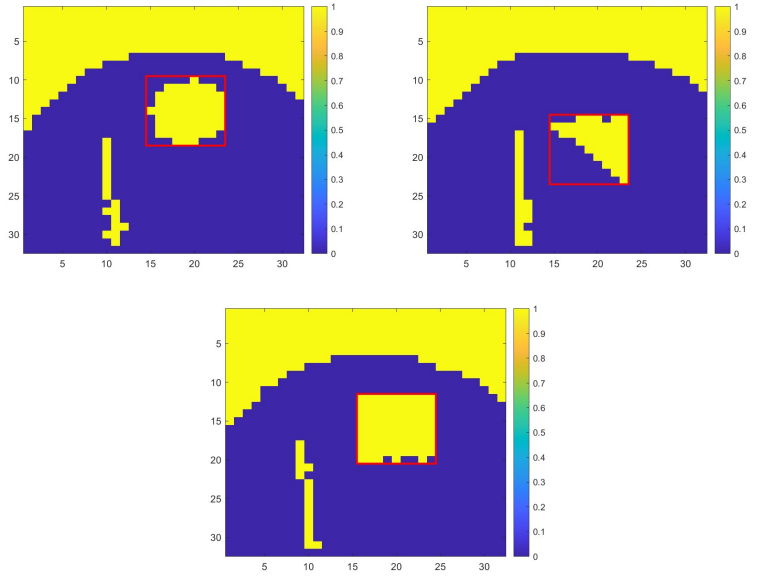
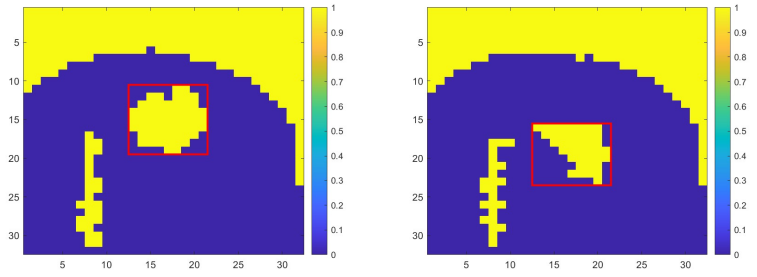
8.62	4.28	

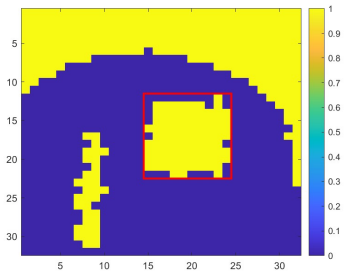
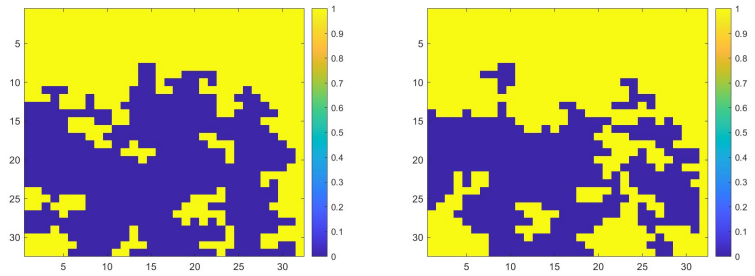
Table B.2: Denoised binary images of various shapes at different sediment concentrations with bounding box of the selected shape showing on each image.

Sed. Conc. (mg/L)	**No. of ALs (w.r.t. 532nm)**	**Denoised Binary Image**
1.30	1.32	

1.78	1.40	
2.74	1.84	
3.22	2.03	

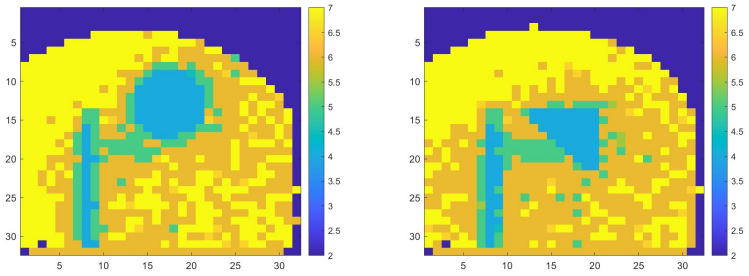
		
3.70	2.35	
4.18	2.41	

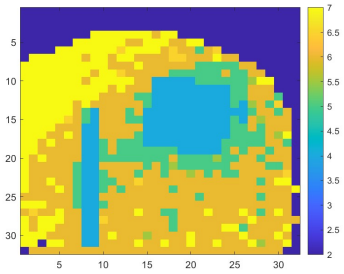
4.66	2.57	
5.14	2.83	
5.62	3.29	

		
8.62	4.28	

B.2 Chlorophyll

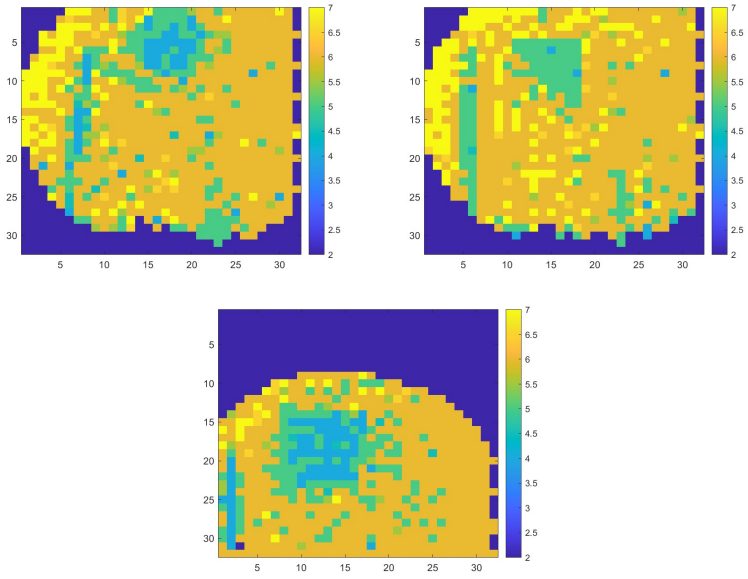
Table B.3: Reconstructed image of various shapes at 2.46 µg/L of chlorophyll. (Measured the power ratio of a 532 nm which gives 1.91 ALs with respect to 532 nm)

Reconstructed Image




B.3 Mixture of Sediments and Chlorophyll

Table B.4: Reconstructed image of various shapes at 2.44 µg/L and 5.71 mg/L. (Measured the power ratio of a 532 nm which gives 3.43 ALs with respect to 532 nm)

Reconstructed Image


References

1. “12x Zoom High Mag Zoom Lenses,” Inspection Equipment, Accessed: March 18, 2019. [Online]. Available: http://www.inspection.ie/navitar-1-50486-navitar-12x-zoom-lens-with-2mm-fine-focus-description-navitar-12x-zoom-lens-with-2mm-fine-focus.html (cited on page 42)

2. “CUBE Lasers,” Coherent, Accessed: March 12, 2019. [Online]. Available: https://www.coherent.com/resources/datasheet/lasers/COHR_CUBE_DS_0121_1.pdf (cited on pages 41 and 42)

3. “Edge Computing,” Accenture Australia, Accessed: January 17, 2024. [Online]. Available: https://www.accenture.com/au-en/insights/cloud/edge-computing-index (cited on page 1)

4. “FB600-10 - Ø1" Bandpass Filter, CWL = 600 ± 2 nm, FWHM = 10 ± 2 nm,” Thorlabs, Accessed: January 18, 2024. [Online]. Available: https://www.thorlabs.com/thorproduct.cfm?partnumber=FB600-10 (cited on page 41)

5. “Microchannel Plates and MCP Detectors and Imaging Systems,” Del Mar Photonics, Accessed: October 13, 2019. [Online]. Available: http://www.dmphotonics.com/MCP_MCPImageIntensifiers/mcp_references.htm (cited on page 36)

6. “National Sea Simulator Capabilities,” Australian Institute of Marine Science, Accessed: October 25, 2024. [Online]. Available: https://www.aims.gov.au/national-sea-simulator/capabilities (cited on page 89)

7. “Navitar Machine Vision - 1/2" Format Lenses,” Navitar, Accessed: January 18, 2024. [Online]. Available: https://navitar.com/products/imaging-optics/low-magnification-video/navitar-machine-vision/navitar-machine-vision-12-format/ (cited on page 41)

8. “Nvidia jetson TX2,” Nvidia, Accessed: February 14, 2023. [Online]. Available: https://www.nvidia.com/en-au/autonomous-machines/embedded-systems/jetson-tx2/ (cited on pages 49, 63, 72, and 167)

9. “Safety Data Sheet,” MSDS.COM.AU Pty Ltd, Accessed: November 2, 2024. [Online]. Available: https://www.jaycar.com.au/water-based-fog-machine-liquid-1l/p/AF1212?srsltid=AfmBOoqfORYjoNUYaiWdpVROfmkEwq84NrlQEEoDh8aaV-yPBZSOXmjZ (cited on page 149)

10. "CUDA C++ Programming Guide," Nvidia, 2016, Accessed: October 23, 2016. [Online]. Available: https://docs.nvidia.com/cuda/cuda-c-programming-guide/ (cited on page 63)

11. AIMS SeaSim, "Lasers in SeaSim tanks," Twitter, Sep. 2018, Accessed: December 12, 2019. [Online]. Available: https://twitter.com/SeaSim_AIMS/status/1044060263656243201 (cited on pages xviii and 88)

12. M. A. Albota and others, "Three-dimensional imaging laser radars with Geiger-mode avalanche photodiode arrays," *Lincoln Lab. J.*, vol. 13, no. 2, pp. 351–370, 2002, Accessed: January 22, 2024. [Online]. Available: http://ridl.cis.rit.edu/products/publications/lincoln%20lab/13_2laserradars3d.pdf (cited on page 37)

13. Y. Altmann, S. McLaughlin, and M. E. Davies, "Fast online 3D reconstruction of dynamic scenes from individual single-photon detection events," *IEEE Trans. on Image Process.*, vol. 29, pp. 2666–2675, Nov. 2020, Accessed: March 20, 2023, doi: 10.1109/TIP.2019.2952008. [Online]. Available: https://ieeexplore.ieee.org/document/8897087/ (cited on page 32)

14. Y. Altmann, X. Ren, A. McCarthy, G. S. Buller, and S. McLaughlin, "Robust Bayesian target detection algorithm for depth imaging from sparse single-photon data," *IEEE Trans. on Comput. Imag.*, vol. 2, no. 4, pp. 456–467, Dec. 2016, Accessed: March 7, 2023, doi: 10.1109/TCI.2016.2618323. [Online]. Available: http://ieeexplore.ieee.org/document/7592397/ (cited on pages 12 and 32)

15. ——, "Lidar waveform-based analysis of depth images constructed using sparse single-photon data," *IEEE Trans. on Image Process.*, vol. 25, no. 5, pp. 1935–1946, May 2016, Accessed: March 14, 2022, doi: 10.1109/TIP.2016.2526784. [Online]. Available: https://ieeexplore.ieee.org/document/7401051/ (cited on pages 12, 20, and 32)

16. ——, "Target detection for depth imaging using sparse single-photon data," in *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Shanghai, China, Mar. 2016, pp. 3256–3260, doi: 10.1109/ICASSP.2016.7472279 (cited on pages 12 and 32)

17. S. J. Anderson, "Target classification, recognition and identification with HF radar," in *Proc. RTO SET Symp. (RTO-MP-SET-080)*, Oslo, Norway, Oct. 2004, p. 25, Accessed: November 7, 2024. [Online]. Available: https://www.sto.nato.int/publications/STO%20Meeting%20Proceedings/RTO-MP-SET-080/MP-SET-080-25.pdf (cited on page 3)

18. D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proc. of the Annu. ACM-SIAM Symp. on Discrete Algorithms*, ser. SODA '07, vol. 8. New Orleans, Louisiana, U.S: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035, doi: 10.1145/1283383.1283494 (cited on page 161)

19. B. F. Aull *et al.*, "Geiger-mode avalanche photodiodes for three-dimensional imaging," *Lincoln Lab. J.*, vol. 13, no. 2, pp. 335–349, 2002, Accessed: January 22, 2024. [Online]. Available: http://ridl.cfd.rit.edu/products/publications/lincoln%20lab/13_2geigermode3d.pdf (cited on page 37)

20. W. Becker, "Time-Correlated Single Photon Counting (TCSPC)," in *Advanced Time-Correlated Single Photon Counting Techn.*, ser. Springer Series in Chemical Physics, A. W. Castleman, J. Toennies, and W. Zinth, Eds. Berlin, Germany: Springer, 2005, vol. 81, ch. 2, sec. 2.4, p. 20, Accessed: March 8, 2023, doi: 10.1007/3-540-28882-1. [Online]. Available: http://link.springer.com/10.1007/3-540-28882-1 (cited on page 40)

21. ——, "Preface," in *The bh TCSPC Handbook*, 9th ed. Berlin, Germany: Becker & Hickl GmbH, Sep. 2021. [Online]. Available: www.becker-hickl.com (cited on page 40)

22. L. Bian *et al.*, "High-resolution single-photon imaging with physics-informed deep learning," *Nature Commun.*, vol. 14, p. 5902, Sep. 2023, Accessed: August 1, 2024, doi: 10.1038/s41467-023-41597-9. [Online]. Available: https://www.nature.com/articles/s41467-023-41597-9 (cited on pages 14 and 32)

23. M. Blume, "Expectation maximization: A gentle introduction," Institut fur Informatik, Technische Universitat Munchen, Accessed: February 15, 2024. [Online]. Available: http://www.cse.cuhk.edu.hk/~cslui/CSCI3320/Expectation%20Maximization%20-%20A%20Gentle%20Introduction.pdf (cited on pages 145 and 157)

24. E. Boss, C. Roseler, and C. Mobley, "Introduction to optical constituents of the ocean," in *Ocean Optics Web Book*. Creative Commons Attribution, Oct. 2021, Accessed: March 13, 2024. [Online]. Available: https://www.oceanopticsbook.info/view/optical-constituents-of-the-ocean/introduction-to-optical-constituents-of-the-ocean (cited on page 88)

25. D. Bronzi, F. Villa, S. Tisa, A. Tosi, and F. Zappa, "SPAD figures of merit for photon-counting, photon-timing, and imaging applications: A review," *IEEE Sensors J.*, vol. 16, no. 1, pp. 3–12, Jan. 2016, Accessed: February 2, 2024, doi: 10.1109/JSEN.2015.2483565. [Online]. Available: https://ieeexplore.ieee.org/document/7283534 (cited on pages 14, 37, and 40)

26. G. Buller and A. Wallace, "Ranging and three-dimensional imaging using time-correlated single-photon counting and point-by-point acquisition," *IEEE J. of Sel. Topics in Quantum Electron.*, vol. 13, no. 4, pp. 1006–1015, Jul./Aug. 2007, Accessed: April 30, 2022, doi: 10.1109/JSTQE.2007.902850. [Online]. Available: http://ieeexplore.ieee.org/document/4301404/ (cited on page 36)

27. G. S. Buller *et al.*, "Depth imaging at kilometer range using time-correlated single-photon counting at wavelengths of 850 nm and 1560 nm," in *Proc. SPIE*,

H. Mohseni, M. H. Agahi, and M. Razeghi, Eds., vol. 8460, San Diego, California, U.S, Oct. 2012, p. 84601I, doi: 10.1117/12.965890 (cited on page 32)

28. S. Chan *et al.*, "Long-range depth imaging using a single-photon detector array and non-local data fusion," *Scientific Reports*, vol. 9, no. 8075, May 2019, Accessed: April 30, 2022, doi: 10.1038/s41598-019-44316-x. [Online]. Available: http://www.nature.com/articles/s41598-019-44316-x (cited on page 32)

29. J. Chang *et al.*, "Confidence interval guided single photon LiDAR depth imaging," in *Proc. SPIE*, Q. Dai, T. Shimura, and Z. Zheng, Eds., vol. 12767, Beijing, China, Nov. 2023, p. 127670G, doi: 10.1117/12.2688934 (cited on pages 13 and 32)

30. S. Chen *et al.*, "Learning non-local spatial correlations to restore sparse 3D single-photon data," *IEEE Trans. on Image Process.*, vol. 29, pp. 3119–3131, Dec. 2020, Accessed: March 14, 2022, doi: 10.1109/TIP.2019.2957918. [Online]. Available: https://ieeexplore.ieee.org/document/8931246/ (cited on pages 12 and 32)

31. ——, "Non-local restoration of sparse 3d single-photon data," in *Proc. 27th European Signal Process. Conf. (EUSIPCO)*, A Coruna, Spain, doi: 10.23919/EUSIPCO.2019.8902525 (cited on pages 12 and 32)

32. S. Chen, X. Su, Z. Zhang, W. Xu, J. Wang, and W. Hao, "Fast and robust restoration of single photon 3D data using parameterized kernel," *IEEE J. of Sel. Topics in Quantum Electron.*, vol. 30, no. 1: Single-Photon Technologies and Applications, pp. 1–8, Jan./Feb. 2024, Accessed: August 1, 2024, doi: 10.1109/JSTQE.2023.3269747. [Online]. Available: https://ieeexplore.ieee.org/document/10107749/ (cited on pages 13 and 32)

33. S. V. Chhaya, S. Khera, and P. Kumar S, "Basic geometric shape and primary colour detection using image processing on matlab," *Int. J. Res. in Engin. and Technol.*, vol. 4, no. 5, pp. 505–509, May 2015, Accessed: February 23, 2020, doi: 10.15623/ijret.2015.0405094. [Online]. Available: https://ijret.org/volumes/2015v04/i05/IJRET20150405094.pdf (cited on page 107)

34. I. Cusini *et al.*, "Historical perspectives, state of art and research trends of SPAD arrays and their applications (Part II: SPAD arrays)," *Frontiers in Phys.*, vol. 10, p. 906671, Jul. 2022, Accessed: March 24, 2023, doi: 10.3389/fphy.2022.906671. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fphy.2022.906671/full (cited on pages 14 and 37)

35. V. R. da Silva and A. Yongacoglu, "EM algorithm on the approximation of arbitrary PDFs by Gaussian, gamma and lognormal mixture distributions," in *Proc. 7th IEEE Latin-American Conf. on Commun. (LATINCOM)*, Arequipa, Peru, Nov. 2015, pp. 1–6, doi: 10.1109/LATINCOM.2015.7430127 (cited on pages 145 and 157)

36. C. Dai, W.-L. Ye, C. Yu, X. Huang, and Z.-P. Li, "Long-range photon-efficient 3D imaging without range ambiguity," *Opt. Lett.*, vol. 48, no. 6, pp. 1542–1545, Mar. 2023, Accessed: November 3, 2024, doi: 10.1364/OL.485127. [Online]. Available: https://opg.optica.org/ol/abstract.cfm?uri=ol-48-6-1542 (cited on pages 12, 13, and 32)

37. P. De Chazal, J. Tapson, and A. Van Schaik, "Learning ELM network weights using linear discriminant analysis," in *Proceedings of ELM-2014 Volume 1*, J. Cao, K. Mao, E. Cambria, Z. Man, and K.-A. Toh, Eds. Cham, Switzerland: Springer Int. Publishing, 2015, vol. 3, pp. 183–191, series Title: Proceedings in Adaptation, Learning and Optimization, Accessed: February 12, 2024, doi: 10.1007/978-3-319-14063-6_16. [Online]. Available: https://link.springer.com/10.1007/978-3-319-14063-6_16 (cited on page 59)

38. D. Delic and S. Afshar, "Background to single-photon lidar systems," in *More-than-Moore Devices and Integration for Semiconductors*, F. Iacopi and F. Balestra, Eds. Cham, Switzerland: Springer Nature Switzerland AG, 2023, ch. 6, sec. 1.1, pp. 195–200, Accessed: March 8, 2023, doi: 10.1007/978-3-031-21610-7. [Online]. Available: https://link.springer.com/10.1007/978-3-031-21610-7 (cited on pages 35, 36, 37, and 40)

39. ——, "Flash 3d imaging laser radar and applications," in *More-than-Moore Devices and Integration for Semiconductors*, F. Iacopi and F. Balestra, Eds. Cham, Switzerland: Springer Nature Switzerland AG, 2023, ch. 6, sec. 1.2, pp. 200–205, Accessed: March 8, 2023, doi: 10.1007/978-3-031-21610-7. [Online]. Available: https://link.springer.com/10.1007/978-3-031-21610-7 (cited on pages 36 and 37)

40. ——, "Neuromorphic computing for compact LiDAR systems - introduction," in *More-than-Moore Devices and Integration for Semiconductors*, F. Iacopi and F. Balestra, Eds. Cham, Switzerland: Springer Nature Switzerland AG, 2023, ch. 6, sec. 1, pp. 191–195, Accessed: March 8, 2023, doi: 10.1007/978-3-031-216 10-7. [Online]. Available: https://link.springer.com/10.1007/978-3-031-21610-7 (cited on page 1)

41. A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the *EM* algorithm," *J. of the Roy. Statistical Soc.: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, Sep. 1977, Accessed: December 28, 2022, doi: 10.1111/j.2517-6161.1977.tb01600.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1111/j.2517-6161.1977.tb01600.x (cited on pages 145 and 157)

42. V. Devrelis, J. Mau, G. Day, and D. Delic, "Estimation of underwater horizontal visibility for divers and ROVs," in *Proc. Global Oceans 2020: Singapore – U.S. Gulf Coast*, Biloxi, MS, U.S, 2020, pp. 1–6, doi: 10.1109/IEEECONF38699.2020.93894 02 (cited on page 10)

43. V. H. Dhulla, S. S. Mukherjee, A. O. Lee, N. Dissanayake, B. Ryu, and C. Myers, "256 x 256 dual-mode CMOS SPAD image sensor," in *Proc. SPIE*, M. A. Itzler,

K. A. McIntosh, and J. C. Bienfang, Eds., vol. 10978, Baltimore, U.S, May 2019, p. 109780Q, doi: 10.1117/12.2523084 (cited on page 6)

44. K. Drummond, S. McLaughlin, Y. Altmann, A. Pawlikowska, and R. Lamb, "Joint surface detection and depth estimation from single-photon Lidar data using ensemble estimators," in *Proc. IEEE Sensor Signal Process. for Defence Conf. (SSPD)*, Edinburgh, United Kingdom, Sep. 2021, pp. 1–5, doi: 10.1109/SSPD51364.2021.9541522 (cited on page 26)

45. D. D. Duncan, "Visibility range," in *The Infrared & Electro-Optical Systems Handbook - vol.2: Atmospheric Propagation of Radiation*, F. G. Smith, J. S. Accetta, and D. L. Shumaker, Eds. Michigan and Washington, U.S: Infrared Inf. Analysis Center, ERIM and SPIE Opt. Eng. Press, 1993, ch. 1, sec. 1.4.5.1, p. 123. (cited on pages 43 and 44)

46. C. Fu *et al.*, "Three-dimensional imaging via time-correlated single-photon counting," *Appl. Sci.*, vol. 10, no. 6, p. 1930, Mar. 2020, Accessed: March 8, 2023, doi: 10.3390/app10061930. [Online]. Available: https://www.mdpi.com/2076-3417/10/6/1930 (cited on pages 13 and 32)

47. W. Gomaa, A. F. El-Sherif, and Y. H. El-Sharkawy, "Underwater laser detection system," in *Proc. SPIE*, S. R. K. Clarkson, W. Andrew, Ed., vol. 9342, San Francisco, California, U.S, Mar. 2015, p. 934221, doi: 10.1117/12.2080181 (cited on page 44)

48. N. Govender, J. Warrell, P. Torr, and F. Nicolls, "Probabilistic models for 2D active shape recognition using Fourier descriptors and mutual information," in *Proc. of the Scientific Cooperations Int. Workshops on Electr. and Comput. Eng. Subfields*. Istanbul, Turkey: Koc University, 2014, Accessed: November 4, 2024. [Online]. Available: https://conf-scoop.org/ACV-2014/5.N.Govender_ACV.pdf (cited on page 172)

49. M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: An overview," *arXiv*, Aug. 2020, Accessed: February 27, 2024. [Online]. Available: http://arxiv.org/abs/2008.05756 (cited on page 46)

50. I. Gyongy, N. A. W. Dutton, and R. K. Henderson, "Direct time-of-flight single-photon imaging," *IEEE Trans. on Electron Devices*, vol. 69, no. 6, pp. 2794–2805, Jun. 2022, Accessed: August 2, 2024, doi: 10.1109/TED.2021.3131430. [Online]. Available: https://ieeexplore.ieee.org/document/9650745/ (cited on pages 14 and 38)

51. I. Gyongy *et al.*, "High-speed 3D sensing via hybrid-mode imaging and guided upsampling," *Optica*, vol. 7, no. 10, pp. 1253–1260, Oct. 2020, Accessed: March 28, 2023, doi: 10.1364/OPTICA.390099. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=optica-7-10-1253 (cited on pages 15, 29, and 30)

52. ——, "High-speed vision with a 3D-stacked SPAD image sensor," in *Proc. SPIE*, M. A. Itzler, K. A. McIntosh, and J. C. Bienfang, Eds., Online Only, Apr. 2021, p. 2, doi: 10.1117/12.2586883 (cited on page 30)

53. ——, "A direct time-of-flight image sensor with in-pixel surface detection and dynamic vision," *IEEE J. of Sel. Topics in Quantum Electron.*, vol. 30, no. 1: Single-Photon Technologies, pp. 1–11, Jan./Feb. 2024, Accessed: August 5, 2024, doi: 10.1109/JSTQE.2023.3238520. [Online]. Available: https://ieeexplore.ieee.org/document/10023481/ (cited on page 29)

54. R. H. Hadfield *et al.*, "Single-photon detection for long-range imaging and sensing," *Optica*, vol. 10, no. 9, pp. 1124–1141, Sep. 2023, Accessed: August 1, 2024, doi: 10.1364/OPTICA.488853. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=optica-10-9-1124 (cited on pages 12 and 13)

55. A. Halimi, P. Ciuciu, A. McCarthy, S. McLaughlin, and G. S. Buller, "Fast adaptive scene sampling for single-photon 3D lidar images," in *Proc. IEEE 8th Int. Workshop on Comput. Advances in Multi-Sensor Adaptive Process. (CAMSAP)*, Le gosier, Guadeloupe, Dec. 2019, pp. 196–200, doi: 10.1109/CAMSAP45676.2019.9022519 (cited on pages 13 and 32)

56. A. Halimi *et al.*, "Restoration of intensity and depth images constructed using sparse single-photon data," in *Proc. IEEE 24th European Signal Process. Conf. (EUSIPCO)*, Budapest, Hungary, Aug. 2016, pp. 86–90, doi: 10.1109/EUSIPCO.2016.7760215 (cited on pages 12, 22, 23, and 32)

57. A. Halimi, A. Maccarone, A. McCarthy, S. McLaughlin, and G. S. Buller, "Object depth profile and reflectivity restoration from sparse single-photon data acquired in underwater environments," *IEEE Trans. on Comput. Imag.*, vol. 3, no. 3, pp. 472–484, Sep. 2017, Accessed: March 7, 2023, doi: 10.1109/TCI.2017.2669867. [Online]. Available: http://ieeexplore.ieee.org/document/7857033/ (cited on pages 25 and 32)

58. A. Halimi and S. McLaughlin, "Robust and guided Bayesian reconstruction of single-photon 3D lidar data: Application to multispectral and underwater imaging," *IEEE Trans. On Comput. Imag.*, vol. 7, pp. 961–974, Sep. 2021, Accessed: November 3, 2024, doi: 10.1109/TCI.2021.3111572. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9534479 (cited on page 15)

59. ——, "Plug-and-play for joint deblurring and guided super-resolution of single-photon 3D lidar data," in *Proc. 31st European Signal Process. Conf. (EUSIPCO)*, Helsinki, Finland, Sep. 2023, pp. 1718–1722, doi: 10.23919/EUSIPCO58844.2023.10289913 (cited on pages 14 and 32)

60. A. Halimi, R. Tobin, A. McCarthy, J. Bioucas-Dias, S. McLaughlin, and G. S. Buller, "Robust restoration of sparse multidimensional single-photon LiDAR images," *IEEE Trans. on Comput. Imag.*, vol. 6, pp. 138–152, Jul. 2020,

Accessed: April 30, 2022, doi: 10.1109/TCI.2019.2929918. [Online]. Available: https://ieeexplore.ieee.org/document/8776646/ (cited on pages 8, 16, 20, 22, 23, 24, 25, 32, 45, and 147)

61. A. Halimi, R. Tobin, D. A. McCarthy, S. McLaughlin, and G. Buller, "Restoration of multilayered single-photon 3D LiDAR images," in *Proc. IEEE 25th European Signal Process. Conf. (EUSIPCO)*, Kos, Greece, 2017, pp. 708–712, doi: 10.23919/EUSIPCO.2017.8081299 (cited on pages 15, 16, 20, 22, 23, and 32)

62. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. on Comput. Vis. and Pattern Recognit. (CVPR)*, Las Vegas, NV, U.S, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90 (cited on pages 75, 76, 78, and 79)

63. R. K. Henderson *et al.*, "A 192 x 128 time correlated SPAD image sensor in 40-nm CMOS technology," *IEEE J. of Solid-State Circuits*, vol. 54, no. 7, pp. 1907–1916, Jul. 2019, Accessed: November 3, 2024, doi: 10.1109/JSSC.2019.2905163. [Online]. Available: https://ieeexplore.ieee.org/document/8681087 (cited on pages 15 and 32)

64. M. Henriksson, H. Larsson, C. Grönwall, and G. Tolt, "Continuously scanning time-correlated single-photon-counting single-pixel 3-D lidar," *Opt. Eng.*, vol. 56, no. 3, p. 031204, Sep. 2016, Accessed: March 24, 2023, doi: 10.1117/1.OE.56.3.031204. [Online]. Available: http://opticalengineering.spiedigitallibrary.org/article.aspx?doi=10.1117/1.OE.56.3.031204 (cited on pages 16 and 32)

65. P. A. Hiskett, K. J. Gordon, J. W. Copley, and R. A. Lamb, "Long range 3D imaging with a 32×32 Geiger mode InGaAs/InP camera," in *Proc. SPIE*, M. A. Itzler and J. C. Campbell, Eds., vol. 9114, Baltimore, Maryland, U.S, May 2014, p. 91140I, doi: 10.1117/12.2050540 (cited on page 32)

66. Y. Hong, Y. Li, C. Dai, J.-T. Ye, X. Huang, and F. Xu, "Image-free target identification using a single-point single-photon LiDAR," *Opt. Exp.*, vol. 31, no. 19, pp. 30 390–30 401, Sep. 2023, Accessed: August 1, 2024, doi: 10.1364/OE.498470. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=oe-31-19-30390 (cited on pages 29, 32, and 74)

67. A. Hosikian, S. Lim, R. Halim, and M. K. Danquah, "Chlorophyll extraction from microalgae: a review on the process engineering aspects," *Int, J. Chem. Eng.*, vol. 2010, no. 391632, pp. 1–11, Mar. 2010, Accessed: November 14, 2019, doi: 10.1155/2010/391632. [Online]. Available: http://www.hindawi.com/journals/ijce/2010/391632/ (cited on page 90)

68. J. D. Howe, "Fundamentals of electro-optical imaging systems analysis," in *The Infrared & Electro-Optical Systems Handbook - vol. 4: Electro-Optical Systems Design, Analysis, and Testing*, M. C. Dudzik, Ed. Michigan and Washington, U.S:

Infrared Inf. Analysis Center, ERIM and SPIE Opt. Eng. Press, 1993, ch. 2, sec. 2.2.3, pp. 62–63. (cited on page 1)

69. S. W. Hutchings *et al.*, "A reconfigurable 3-D-stacked SPAD imager with in-pixel histogramming for flash LIDAR or high-speed time-of-flight imaging," *IEEE J. of Solid-State Circuits*, vol. 54, no. 11, pp. 2947–2956, Nov. 2019, Accessed: April 17, 2023, doi: 10.1109/JSSC.2019.2939083. [Online]. Available: https://ieeexplore.ieee.org/document/8848491/ (cited on page 29)

70. P.-Y. Jiang *et al.*, "Long range 3D imaging through atmospheric obscurants using array-based single-photon LiDAR," *Opt. Exp.*, vol. 31, no. 10, pp. 16 054–16 066, May 2023, Accessed: May 3, 2023, doi: 10.1364/OE.487560. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=oe-31-10-16054 (cited on pages 17, 24, 25, and 32)

71. F. D. Kashani, M. R. H. Rad, and E. Kazemian, "Analyzing the propagation behavior of a gaussian laser beam through seawater and comparing with atmosphere," *Iranian J. of Elect. Electron. Eng.*, vol. 9, no. 4, pp. 197–203, Dec. 2013, Accessed: November 4, 2024, doi: 20.1001.1.17352827.2013.9.4.4.9. [Online]. Available: https://ijeee.iust.ac.ir/browse.php?a_code=A-10-824-2&slc_lang=en&sid=1 (cited on page 44)

72. Keqi Zhang, Shu-Ching Chen, D. Whitman, Mei-Ling Shyu, Jianhua Yan, and Chengcui Zhang, "A progressive morphological filter for removing nonground measurements from airborne LIDAR data," *IEEE Trans. on Geosci. and Remote Sens.*, vol. 41, no. 4, pp. 872–882, Apr. 2003, Accessed: January 4, 2024, doi: 10.1109/TGRS.2003.810682. [Online]. Available: http://ieeexplore.ieee.org/document/1202973/ (cited on page 23)

73. A. Kirmani *et al.*, "First-photon imaging," *Science*, vol. 343, no. 6166, pp. 58–61, Jan. 2014, Accessed: May 7, 2022, doi: 10.1126/science.1246775. [Online]. Available: https://www.science.org/doi/10.1126/science.1246775 (cited on pages 13 and 32)

74. N. J. Krichel, A. McCarthy, and G. S. Buller, "Resolving range ambiguity in a photon counting depth imager operating at kilometer distances," *Opt. Exp.*, vol. 18, no. 9, pp. 9192–9206, Apr. 2010, Accessed: March 23, 2023, doi: 10.1364/OE.18.009192. [Online]. Available: https://opg.optica.org/oe/abstract.cfm?uri=oe-18-9-9192 (cited on pages 13 and 32)

75. N. J. Krichel, A. McCarthy, R. J. Collins, V. Fernández, A. M. Wallace, and G. S. Buller, "Scanning of low-signature targets using time-correlated single-photon counting," in *Proc. SPIE*, G. W. Kamerman, O. K. Steinvall, K. L. Lewis, R. C. Hollins, T. J. Merlet, G. J. Bishop, and J. D. Gonglewski, Eds., vol. 7482, Berlin, Germany, Sep. 2009, p. 748202, doi: 10.1117/12.829362 (cited on page 32)

76. N. J. Krichel, A. McCarthy, A. M. Wallace, J. Ye, and G. S. Buller, "Long-range depth imaging using time-correlated single-photon counting," in *Proc. SPIE*, E. L. Dereniak, J. P. Hartke, P. D. LeVan, A. K. Sood, R. E. Longshore, and

M. Razeghi, Eds., vol. 7780, San Diego, California, Aug. 2010, p. 77801I, doi: 10.1117/12.876144 (cited on page 32)

77. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, Accessed: March 8, 2019, doi: 10.1145/3065386. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3098997.3065386 (cited on pages 75, 76, 78, and 79)

78. R. Kundu, "Confusion Matrix: How To Use It & Interpret Results [Examples]," Sep. 2022, Accessed: January 15, 2024. [Online]. Available: https://www.v7labs.com/blog/confusion-matrix-guide#confusion-matrix-for-multiple-classes (cited on page 71)

79. Q. Legros *et al.*, "Robust 3D reconstruction of dynamic scenes from single-photon lidar using beta-divergences," *IEEE Trans. on Image Process.*, vol. 30, pp. 1716–1727, Dec. 2021, Accessed: March 20, 2023, doi: 10.1109/TIP.2020.3046882. [Online]. Available: https://ieeexplore.ieee.org/document/9311752/ (cited on pages 14 and 32)

80. Z.-P. Li *et al.*, "All-time single-photon 3D imaging over 21 km," in *Proc. Conf. on Lasers and Electro-Opt. (CLEO)*, May 2019, pp. 1–2, doi: 10.1364/CLEO_SI.2019.SM1N.1 (cited on pages 12 and 32)

81. ——, "Single-photon imaging over 200 km," *Optica*, vol. 8, no. 3, pp. 344–349, Mar. 2021, Accessed: March 21, 2023, doi: 10.1364/OPTICA.408657. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=optica-8-3-344 (cited on pages 12 and 32)

82. Z.-P. Li and others, "Single-photon computational 3D imaging at 45 km," *Photon. Res.*, vol. 8, no. 9, pp. 1532–1540, Sep. 2020, Accessed: November 3, 2024, doi: 10.1364/PRJ.390091. [Online]. Available: https://opg.optica.org/prj/fulltext.cfm?uri=prj-8-9-1532&id=437808 (cited on pages 24 and 32)

83. P. Lif, F. Bissmarck, G. Tolt, and P. Jonsson, "User performance for vehicle recognition in three- dimensional point clouds," in *Proc. of the Human Factors and Ergonom. Soc. Europe Chapter 2017 Annual Conf.*, 2017, pp. 127–140, Accessed: November 7, 2024. [Online]. Available: https://www.hfes-europe.org/wp-content/uploads/2017/10/Lif2017.pdf (cited on page 3)

84. H. Lin, G. Ao, and Y. Li, "A single photon imaging method for solving maximum likelihood estimation to improve photon utilisation," in *Proc. SPIE*, Hefei, China, Apr. 2023, p. 126176A, doi: 10.1117/12.2666698 (cited on pages 13 and 32)

85. D. B. Lindell, M. O'Toole, and G. Wetzstein, "Single-photon 3D imaging with deep sensor fusion," *ACM Trans. on Graph.*, vol. 37, no. 4, p. 113, Aug. 2018,

Accessed: April 30, 2022, doi: 10.1145/3197517.3201316. [Online]. Available: https://dl.acm.org/doi/10.1145/3197517.3201316 (cited on pages 13 and 32)

86. D. Liu *et al.*, "Single-parameter estimation construction algorithm for Gm-APD ladar imaging through fog," *Opt. Commun.*, vol. 482, p. 126558, Mar. 2021, Accessed: November 3, 2024, doi: 10.1016/j.optcom.2020.126558. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0030401820309767 (cited on pages 17, 18, 19, 32, and 144)

87. R. Lussana, F. Villa, A. D. Mora, D. Contini, A. Tosi, and F. Zappa, "Enhanced single-photon time-of-flight 3D ranging," *Opt. Exp.*, vol. 23, no. 19, pp. 24 962–24 973, Sep. 2015, Accessed: March 17, 2019, doi: 10.1364/OE.23.024962. [Online]. Available: https://www.osapublishing.org/abstract.cfm?URI=oe-23-19-24962 (cited on pages 13, 14, 15, 16, and 32)

88. A. Maccarone *et al.*, "Underwater depth imaging using time-correlated single-photon counting," *Opt. Exp.*, vol. 23, no. 26, pp. 33 911–33 926, Dec. 2015, Accessed: May 5, 2022, doi: 10.1364/OE.23.033911. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=oe-23-26-33911 (cited on pages 3, 25, 26, 32, and 37)

89. ——, "Depth imaging in highly scattering underwater environments using time-correlated single-photon counting," in *Proc. SPIE*, K. L. Lewis and R. C. Hollins, Eds., vol. 9992, Edinburgh, United Kingdom, Oct. 2016, p. 99920R, doi: 10.1117/12.2242710 (cited on pages 25, 26, 32, and 45)

90. ——, "Underwater three-dimensional imaging using single-photon detection," in *Proc. Conf. on Lasers and Electro-Opt.*, San Jose, California, 2017, p. SF2M.2, doi: 10.1364/CLEO_SI.2017.SF2M.2 (cited on pages 25 and 32)

91. ——, "Underwater depth imaging using time-correlated single-photon counting at video frame rates," in *Proc. SPIE*, G. W. Kamerman and O. Steinvall, Eds., vol. 11160, Strasbourg, France, Oct. 2019, p. 111600I, doi: 10.1117/12.2534303 (cited on pages 25 and 32)

92. ——, "Custom-technology single-photon avalanche diode linear detector array for underwater depth imaging," *Sensors*, vol. 21, no. 14, p. 4850, Jul. 2021, Accessed: March 20, 2023, doi: 10.3390/s21144850. [Online]. Available: https://www.mdpi.com/1424-8220/21/14/4850 (cited on pages 25, 26, and 32)

93. ——, "Real-time underwater single-photon three-dimensional imaging," in *Proc. SPIE*, vol. 12274, Berlin, Germany, Dec. 2022, p. 1227403, doi: 10.1117/12.2638494 (cited on pages 14 and 32)

94. ——, "Submerged single-photon LiDAR imaging sensor used for real-time 3D scene reconstruction in scattering underwater environments," *Opt. Exp.*, vol. 31, no. 10, pp. 16 690–16 708, May 2023, Accessed: May 10, 2023, doi: 10.1364/OE.487129. [Online]. Available: https:

//opg.optica.org/abstract.cfm?URI=oe-31-10-16690 (cited on pages 14, 15, 25, 26, 27, and 32)

95. A. Maccarone, F. Mattioli Della Rocca, A. McCarthy, R. Henderson, and G. S. Buller, "Three-dimensional imaging of stationary and moving targets in turbid underwater environments using a single-photon detector array," *Opt. Exp.*, vol. 27, no. 20, pp. 28 437–28 456, Sep. 2019, Accessed: March 20, 2023, doi: 10.1364/OE.27.028437. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=oe-27-20-28437 (cited on pages 25, 32, and 40)

96. J. S. Massa, A. M. Wallace, G. S. Buller, S. J. Fancey, and A. C. Walker, "Laser depth measurement based on time-correlated single-photon counting," *Opt. Lett.*, vol. 22, no. 8, pp. 543–545, Apr. 1997, Accessed: March 8, 2023, doi: 10.1364/OL.22.000543. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=ol-22-8-543 (cited on page 40)

97. J. S. Massa, G. S. Buller, A. C. Walker, S. Cova, M. Umasuthan, and A. M. Wallace, "Time-of-flight optical ranging system based on time-correlated single-photon counting," *Appl. Opt.*, vol. 37, no. 31, pp. 7298–7304, Nov. 1998, Accessed: April 30, 2022, doi: 10.1364/AO.37.007298. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=ao-37-31-7298 (cited on page 40)

98. D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. on Intell. Robots and Syst. (IROS)*, Hamburg, Germany, Sep. 2015, pp. 922–928, doi: 10.1109/IROS.2015.7353481 (cited on pages 75, 76, and 78)

99. J. Mau, V. Devrelis, G. Day, J. Trumpf, and D. V. Delic, "The use of statistical mixture models to reduce noise in SPAD images of fog-obscured environments," in *Proc. SPIE*, vol. 11525, Online Only, 2020, p. 115250P, doi: 10.1117/12.2580251 (cited on pages 8, 9, 32, 148, 155, 161, 164, 165, 184, 187, and 188)

100. J. Mau, V. Devrelis, G. Day, G. Nash, J. Trumpf, and D. Delic, "Through thick and thin: Imaging through obscurant using SPAD array," in *Proc. IEEE Sensors*, Rotterdam, Netherlands, 2020, pp. 1–4, doi: 10.1109/SENSORS47125.2020.9278706 (cited on pages 9, 32, 148, 150, 153, and 181)

101. J. Mau, V. Devrelis, G. Day, J. Trumpf, and D. Delic, "Impact of water quality on Single Photon Avalanche Diode direct time-of-flight imaging," in *Proc. Global Oceans 2020: Singapore – U.S. Gulf Coast*, Biloxi, MS, U.S, 2020, pp. 1–8, doi: 10.1109/IEEECONF38699.2020.9389293 (cited on pages 7, 9, 32, 44, 84, 85, 89, 101, 112, 140, and 200)

102. J. Mau *et al.*, "Embedded implementation of a random feature detecting network for real-time classification of time-of-flight SPAD array recordings," in *Proc. SPIE*, vol. 11005, Baltimore, U.S, May 2019, p. 1100505, doi: 10.1117/12.2517875 (cited on pages xvii, xxiii, 6, 7, 9, 32, 51, 56, 57, 60, 65, 66, 67, 69, 70, 71, 72, 73, 74, 80, and 81)

103. ——, "Embedded implementation of a random feature detecting network for real-time classification of time-of-flight SPAD array recordings," Baltimore, U.S, May 2019, Accessed: August 7, 2019 [Online Video]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11005/2517875/Embedded-implementation-of-a-random-feature-detecting-network-for-real/10.1117/12.2517875.full (cited on pages xvii and 66)

104. J. Mau, J. Trumpf, G. Day, and D. Delic, "An image feature-based approach to improving SPAD Flash LiDAR imaging through fog," in *Proc. SPIE*, vol. 12274, Berlin, Germany, Nov. 2022, p. 1227405, doi: 10.1117/12.2633941 (cited on pages 8, 9, 32, 148, and 192)

105. A. McCarthy, R. J. Collins, N. J. Krichel, V. Fernández, A. M. Wallace, and G. S. Buller, "Long-range time-of-flight scanning sensor based on high-speed time-correlated single-photon counting," *Appl. Opt.*, vol. 48, no. 32, pp. 6241–6251, Nov. 2009, Accessed: April 30, 2022, doi: 10.1364/AO.48.006241. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=ao-48-32-6241 (cited on page 32)

106. A. McCarthy *et al.*, "Kilometer-range depth imaging at 1550 nm wavelength using an InGaAs/InP single-photon avalanche diode detector," *Opt. Exp.*, vol. 21, no. 19, pp. 22 098–22 113, Sep. 2013, Accessed: May 7, 2022, doi: 10.1364/OE.21.022098. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=oe-21-19-22098 (cited on pages 12, 32, and 37)

107. P. F. McManamon, "Context of lidar," in *LiDAR Technologies and Systems*. Washington, U.S: SPIE, 2019, ch. 1, sec. 1.1, p. 3, Accessed: March 8, 2023, doi: 10.1117/3.2518254. [Online]. Available: https://www.spiedigitallibrary.org/ebooks/PM/LiDAR-Technologies-and-Systems/eISBN-9781510625402/10.1117/3.2518254 (cited on page 2)

108. C. Mobley, "References-brief definitions," in *Ocean Optics Web Book*. Creative Commons Attribution, Aug. 2020, Accessed: November 24, 2020. [Online]. Available: https://www.oceanopticsbook.info/view/references/brief-definitions (cited on page 43)

109. ——, "The physics of scattering," in *Ocean Optics Web Book*. Creative Commons Attribution, Oct. 2021, Accessed: December 12, 2023. [Online]. Available: https://oceanopticsbook.info/view/scattering/physics-scattering (cited on page 83)

110. V. Molebny, P. McManamon, O. Steinvall, T. Kobayashi, and W. Chen, "Laser radar: historical prospective—from the East to the West," *Opt. Eng.*, vol. 56, no. 3, p. 031220, Dec. 2016, Accessed: March 8, 2023, doi: 10.1117/1.OE.56.3.031220. [Online]. Available: http://opticalengineering.spiedigitallibrary.org/article.aspx?doi=10.1117/1.OE.56.3.031220 (cited on page 2)

111. T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, Nov. 1996, Accessed: February 15, 2024, doi:

10.1109/79.543975. [Online]. Available: https://ieeexplore.ieee.org/document/543975 (cited on page 157)

112. G. Mora Martín *et al.*, "High-speed object detection using SPAD sensors," in *Proc. SPIE*, Y. Soskind and L. E. Busse, Eds., vol. 11693, Online Only, Mar. 2021, p. 116930L, doi: 10.1117/12.2577545 (cited on pages 29, 30, 32, and 74)

113. G. Mora-Martín *et al.*, "High-speed object detection with a single-photon time-of-flight image sensor," *Opt. Exp.*, vol. 29, no. 21, pp. 33 184–33 196, Oct. 2021, Accessed: March 23, 2023, doi: 10.1364/OE.435619. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=oe-29-21-33184 (cited on pages 27, 29, 30, 31, 32, 36, and 74)

114. G. Mora-Martín, S. Scholes, R. K. Henderson, J. Leach, and I. Gyongy, "Human activity recognition using a single-photon direct time-of-flight sensor," *Opt. Exp.*, vol. 32, no. 10, pp. 16 645–16 656, May 2024, Accessed: August 1, 2024, doi: 10.1364/OE.516681. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=oe-32-10-16645 (cited on pages 29, 30, 31, 32, and 74)

115. G. Mora-Martín, S. Scholes, A. Ruget, R. Henderson, J. Leach, and I. Gyongy, "Video super-resolution for single-photon LIDAR," *Opt. Exp.*, vol. 31, no. 5, pp. 7060–7072, Feb. 2023, Accessed: August 2, 2024, doi: 10.1364/OE.478308. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=oe-31-5-7060 (cited on pages 14 and 32)

116. D. Morrison, S. Kennedy, D. Delic, M. R. Yuce, and J.-M. Redoute, "A 64 × 64 SPAD flash LIDAR sensor using a triple integration timing technique with 1.95 mm depth resolution," *IEEE Sensors J.*, vol. 21, no. 10, pp. 11 361–11 373, May 2021, Accessed: April 30, 2022, doi: 10.1109/JSEN.2020.3030788. [Online]. Available: https://ieeexplore.ieee.org/document/9222158/ (cited on page 6)

117. G. Nash and V. Devrelis, "Flash LiDAR imaging and classification of vehicles," in *Proc. IEEE Sensors*, Rotterdam, Netherlands, Oct. 2020, pp. 1–4, doi: 10.1109/SENSORS47125.2020.9278655 (cited on pages 10, 27, 28, 32, 74, and 98)

118. F. Oh, "What is CUDA?" Nvidia, Sep. 2012, Accessed: February 21, 2023. [Online]. Available: https://blogs.nvidia.com/blog/2012/09/10/what-is-cuda-2/ (cited on page 62)

119. J. Opitz, "From Bias and Prevalence to Macro F1, Kappa, and MCC: A structured overview of metrics for multi-class evaluation," Heidelberg University, Accessed: February 27, 2024. [Online]. Available: https://www.cl.uni-heidelberg.de/~opitz/pdf/metric_overview.pdf (cited on pages 46 and 47)

120. A. M. Pawlikowska, A. Halimi, R. A. Lamb, and G. S. Buller, "Single-photon three-dimensional imaging at up to 10 kilometers range," *Opt. Exp.*, vol. 25, no. 10, pp. 11 919–11 931, May 2017, Accessed: April 30, 2022, doi:

10.1364/OE.25.011919. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=oe-25-10-11919 (cited on pages 32 and 36)

121. J. Peng, Z. Xiong, H. Tan, X. Huang, Z.-P. Li, and F. Xu, "Boosting photon-efficient image reconstruction with a unified deep neural network," *IEEE Trans. on Pattern Analysis and Mach. Intell.*, vol. 45, no. 4, pp. 4180–4197, Aug. 2022, Accessed: October 18, 2023, doi: 10.1109/TPAMI.2022.3200745. [Online]. Available: https://ieeexplore.ieee.org/document/9864284/ (cited on pages 13, 24, and 32)

122. F. Piron, D. Morrison, M. R. Yuce, and J.-M. Redoute, "A review of single-photon avalanche diode time-of-flight imaging sensor arrays," *IEEE Sensors J.*, vol. 21, no. 11, pp. 12 654–12 666, Jun. 2021, Accessed: May 5, 2022, doi: 10.1109/JSEN.2020.3039362. [Online]. Available: https://ieeexplore.ieee.org/document/9264255/ (cited on page 14)

123. S. Plosz, I. Gyongy, J. Leach, S. McLaughlin, G. S. Buller, and A. Halimi, "Fast multiscale 3D reconstruction using single-photon lidar data," in *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Rhodes Island, Greece, Jun. 2023, pp. 1–5, doi: 10.1109/ICASSP49357.2023.10095844 (cited on pages 15 and 32)

124. S. Plosz, A. Maccarone, S. McLaughlin, G. S. Buller, and A. Halimi, "Real-time reconstruction of 3D videos from single-photon LiDAR data in the presence of obscurants," *IEEE Trans. on Comput. Imag.*, vol. 9, pp. 106–119, Feb. 2023, Accessed: March 20, 2023, doi: 10.1109/TCI.2023.3241547. [Online]. Available: https://ieeexplore.ieee.org/document/10034858/ (cited on pages 12, 15, 25, 26, and 32)

125. R. W. Preisendorfer, "Secchi disk science: Visual optics of natural waters: Secchi disk science," *Limnology and Oceanogr.*, vol. 31, no. 5, pp. 909–926, Sep. 1986, Accessed: December 5, 2020, doi: 10.4319/lo.1986.31.5.0909. [Online]. Available: http://doi.wiley.com/10.4319/lo.1986.31.5.0909 (cited on page 45)

126. Quadripartite Working Group on Army Operational Research, "Search and target acquisition nomenclature," Quadripartite Advisory Publication, Special Working Part on the Modeling of Target Acquisition, Tech. Rep., May 1990. (cited on page 3)

127. J. Rapp and V. K. Goyal, "A few photons among many: Unmixing signal and noise for photon-efficient active imaging," *IEEE Trans. on Comput. Imag.*, vol. 3, no. 3, pp. 445–459, Sep. 2017, Accessed: September 30, 2020, doi: 10.1109/TCI.2017.2706028. [Online]. Available: https://ieeexplore.ieee.org/document/7932527/ (cited on pages 8, 13, 20, 24, 32, and 147)

128. X. Ren *et al.*, "High-resolution depth profiling using a range-gated CMOS SPAD quanta image sensor," *Opt. Exp.*, vol. 26, no. 5, pp. 5541–5557, Mar.

2018, Accessed: March 20, 2023, doi: 10.1364/OE.26.005541. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=oe-26-5-5541 (cited on page 32)

129. A. Rochas *et al.*, "First fully integrated 2-D array of single-photon detectors in standard CMOS technology," *IEEE Photon. Technol. Lett.*, vol. 15, no. 7, pp. 963–965, Jul. 2003, Accessed: May 8, 2023, doi: 10.1109/LPT.2003.813387. [Online]. Available: http://ieeexplore.ieee.org/document/1206776/ (cited on page 14)

130. A. Rochas, A. Pauchard, P.-A. Besse, D. Pantic, Z. Prijic, and R. Popovic, "Low-noise silicon avalanche photodiodes fabricated in conventional CMOS technologies," *IEEE Trans. on Electron Devices*, vol. 49, no. 3, pp. 387–394, Mar. 2002, Accessed: April 9, 2024, doi: 10.1109/16.987107. [Online]. Available: http://ieeexplore.ieee.org/document/987107/ (cited on page 37)

131. C. Roesler, "Absorption-measurement of absorption," in *Ocean Optics Web Book*. Creative Commons Attribution, Aug. 2017, Accessed: October 31, 2019. [Online]. Available: http://www.oceanopticsbook.info/view/absorption/measurement_of_absorption (cited on page 44)

132. ——, "Absorption by oceanic constituents," in *Ocean Optics Web Book*. Creative Commons Attribution, Mar. 2021, Accessed: December 12, 2023. [Online]. Available: https://oceanopticsbook.info/view/absorption/absorption-by-oceanic-constituents (cited on page 83)

133. A. Ruget *et al.*, "Pixels2Pose: Super-resolution time-of-flight imaging for 3D pose estimation," in *Proc. Imag. and Appl. Opt. Congr. 2022 (3D, AOA, COSI, ISA, pcAOP)*, Vancouver, British Columbia Canada, Jul. 2022, p. ITh5D.5, doi: 10.1364/ISA.2022.ITh5D.5 (cited on page 14)

134. ——, "Pixels2Pose: Super-resolution time-of-flight imaging for 3D pose estimation," *Sci. Advances*, vol. 8, no. 48, p. eade0123, Nov. 2022, Accessed: November 3, 2024, doi: 10.1126/sciadv.ade0123. [Online]. Available: https://www.science.org/doi/10.1126/sciadv.ade0123 (cited on pages 27, 28, 32, and 74)

135. A. Ruget, S. McLaughlin, R. K. Henderson, I. Gyongy, A. Halimi, and J. Leach, "Robust super-resolution depth imaging via a multi-feature fusion deep network," *Opt. Exp.*, vol. 29, no. 8, pp. 11 917–11 937, Apr. 2021, Accessed: March 20, 2023, doi: 10.1364/OE.415563. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=oe-29-8-11917 (cited on page 14)

136. A. D. Ruvalcaba-Cardenas, T. Scoleri, and G. Day, "Object classification using deep learning on extremely low-resolution time-of-flight data," in *Proc. Digit. Image Comput.: Techn. and Appl. (DICTA)*, Canberra, Australia, Dec. 2018, pp. 1–7, doi: 10.1109/DICTA.2018.8615877 (cited on pages 10, 27, 28, 32, 73, 75, 76, 77, and 80)

137. T.-H. Sang, S. Tsai, and T. Yu, "Mitigating effects of uniform fog on SPAD lidars," *IEEE Sensors Lett.*, vol. 4, no. 9, pp. 1–4, Sep. 2020, Accessed: March 15, 2022, doi: 10.1109/LSENS.2020.3018708. [Online]. Available: https://ieeexplore.ieee.org/document/9174830/ (cited on pages 20, 21, 32, 144, and 145)

138. G. Satat, M. Tancik, and R. Raskar, "Imaging through volumetric scattering with a single photon sensitive camera," in *Proc. Imag. and Appl. Opt. 2018 (3D, AO, AIO, COSI, DH, IS, LACSEA, LS&C, MATH, pcAOP)*, Orlando, Florida, 2018, p. MM5D.2, doi: 10.1364/MATH.2018.MM5D.2 (cited on pages 25 and 32)

139. ——, "Towards photography through realistic fog," in *Proc. IEEE Int. Conf. on Comput. Photography (ICCP)*, Pittsburgh, PA, U.S, 2018, pp. 1–10, doi: 10.1109/ICCPHOT.2018.8368463 (cited on pages 2, 18, 19, 32, 144, and 145)

140. S. Scholes, A. Ruget, G. Mora-Martin, F. Zhu, I. Gyongy, and J. Leach, "DroneSense: The identification, segmentation, and orientation detection of drones via neural networks," *IEEE Access*, vol. 10, pp. 38 154–38 164, Mar. 2022, Accessed: March 28, 2023, doi: 10.1109/ACCESS.2022.3162866. [Online]. Available: https://ieeexplore.ieee.org/document/9743918/ (cited on pages 29, 30, and 32)

141. N. Sharma, V. Jain, and A. Mishra, "An analysis of convolutional neural networks for image classification," *Procedia Comput. Sci.*, vol. 132, pp. 377–384, 2018, Accessed: November 4, 2024, doi: https://doi.org/10.1016/j.procs.2018.05.198. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918309335 (cited on pages xxiii, 75, 76, 78, and 79)

142. M. S. A. Shawkat, M. M. Adnan, R. D. Febbo, J. J. Murray, and G. S. Rose, "A single chip SPAD based vision sensing system with integrated memristive spiking neuromorphic processing," *IEEE Access*, vol. 11, pp. 19 441–19 457, Feb. 2023, Accessed: March 28, 2023, doi: 10.1109/ACCESS.2023.3244793. [Online]. Available: https://ieeexplore.ieee.org/document/10044095/ (cited on pages 28, 32, and 74)

143. D. Shin, A. Kirmani, V. K. Goyal, and J. H. Shapiro, "Photon-efficient computational 3-D and reflectivity imaging with single-photon detectors," *IEEE Trans. on Comput. Imag.*, vol. 1, no. 2, pp. 112–125, Jun. 2015, Accessed: May 7, 2022, doi: 10.1109/TCI.2015.2453093. [Online]. Available: http://ieeexplore.ieee.org/document/7150537/ (cited on page 20)

144. D. Shin, F. Xu, F. N. C. Wong, J. H. Shapiro, and V. K. Goyal, "Computational multi-depth single-photon imaging," *Opt. Exp.*, vol. 24, no. 3, pp. 1873–1888, Jan. 2016, Accessed: May 7, 2022, doi: 10.1364/OE.24.001873. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=oe-24-3-1873 (cited on pages 15 and 32)

145. H. Sizun and M. Al Naboulsi, "Appendix - optical contrast and koschmieder's law," in *Measurements Using Opt. and RF Waves*, P.-N. Favennec and F. de Fornel, Eds. London, UK and Hoboken, U.S: ISTE Ltd and John Wiley & Sons, 2010, ch. 3, sec. 3.9, p. 77, Accessed: February 5, 2024, doi: 10.1002/9781118586228. [Online]. Available: https://onlinelibrary.wiley.com/doi/book/10.1002/9781118586228 (cited on page 46)

146. L. Sjöqvist, M. Henriksson, P. Jonsson, and O. Steinvall, "Time-correlated single-photon counting range profiling and reflectance tomographic imaging," *Advanced Opt. Technol.*, vol. 3, no. 2, pp. 187–197, Apr. 2014, Accessed: March 28, 2023, doi: 10.1515/aot-2014-0003. [Online]. Available: https://www.degruyter.com/document/doi/10.1515/aot-2014-0003/html (cited on pages 13 and 32)

147. P. Soan *et al.*, "Comparative assessment of different active imaging technologies for imaging through obscurants," in *Proc. SPIE*, G. W. Kamerman and O. Steinvall, Eds., vol. 10796, Berlin, Germany, 2018, p. 107960C, doi: 10.1117/12.2326999 (cited on pages 16, 17, and 32)

148. M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. & Manage.*, vol. 45, no. 4, pp. 427–437, Jul. 2009, Accessed: February 27, 2024, doi: 10.1016/j.ipm.2009.03.002. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0306457309000259 (cited on pages 46 and 47)

149. K. R. Spring, B. O. Flynn, and M. W. Davidson, "Photomultiplier Tubes," Olympus, Accessed: October 13, 2019. [Online]. Available: https://www.olympus-lifescience.com/en/microscope-resource/primer/flash/photomultiplier/ (cited on page 36)

150. D. F. Swinehart, "The beer-lambert law," *J. of Chem. Educ.*, vol. 39, no. 7, pp. 333–335, Jul. 1962, Accessed: February 15, 2024, doi: https://doi.org/10.1021/ed039p333. [Online]. Available: https://pubs.acs.org/doi/10.1021/ed039p333 (cited on pages 43 and 44)

151. C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. on Comput. Vis. and Pattern Recognit. (CVPR)*, MA, U.S, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594 (cited on pages 75, 76, 78, and 79)

152. J. Tachella *et al.*, "Bayesian 3D reconstruction of complex scenes from single-photon lidar data," *SIAM J. on Imag. Sci.*, vol. 12, no. 1, pp. 521–550, Jan. 2019, Accessed: March 14, 2022, doi: 10.1137/18M1183972. [Online]. Available: https://epubs.siam.org/doi/10.1137/18M1183972 (cited on pages 16, 20, 22, and 32)

153. ——, "Real-time 3D reconstruction from single-photon lidar data using plug-and-play point cloud denoisers," *Nature Commun.*, vol. 10, no. 4984, Nov. 2019, Accessed: March 20, 2023, doi: 10.1038/s41467-019-12943-7. [Online].

Available: https://www.nature.com/articles/s41467-019-12943-7 (cited on pages 14, 16, 26, 32, and 40)

154. A. Talwalkar and V. Smith, "10-701 Introduction to Machine Learning - The EM Algorithm," 2019, [PowerPoint slides], Accessed: March 26, 2024. [Online]. Available: https://www.cs.cmu.edu/~lwehbe/10701_S19/files/18-EM.pdf (cited on pages 158 and 160)

155. R. Tobin, "Long-range depth profiling of camouflaged targets using single-photon detection," *Opt. Eng.*, vol. 57, no. 3, p. 1, Dec. 2017, Accessed: April 30, 2022, doi: 10.1117/1.OE.57.3.031303. [Online]. Available: https://www.spiedigitallibrary.org/journals/optical-engineering/volume-57/issue-03/031303/Long-range-depth-profiling-of-camouflaged-targets-using-single-photon/10.1117/1.OE.57.3.031303.full (cited on pages 16 and 32)

156. R. Tobin, A. Halimi, A. McCarthy, M. Laurenzis, F. Christnacher, and G. S. Buller, "Depth imaging through obscurants using time-correlated single-photon counting," in *Proc. SPIE*, Orlando, U.S, doi: 10.1117/12.2305369 (cited on page 45)

157. ——, "Three-dimensional single-photon imaging through obscurants," *Opt. Exp.*, vol. 27, no. 4, pp. 4590–4611, 2019, Accessed: June 2, 2020, doi: 10.1364/OE.27.004590. [Online]. Available: https://www.osapublishing.org/abstract.cfm?URI=oe-27-4-4590 (cited on pages 2, 3, 16, 17, 23, 24, 25, 32, 35, 40, 45, and 46)

158. R. Tobin, A. Halimi, A. McCarthy, P. J. Soan, and G. S. Buller, "Robust real-time 3D imaging of moving scenes through atmospheric obscurant using single-photon LiDAR," *Scientific Reports*, vol. 11, no. 11236, 2021, Accessed: March 14, 2022, doi: 10.1038/s41598-021-90587-8. [Online]. Available: http://www.nature.com/articles/s41598-021-90587-8 (cited on pages 3, 13, 14, 16, 17, 21, 22, 25, 32, 35, 37, 45, and 178)

159. A. Tokmakoff, "Absorption cross-sections," in *Time Dependent Quantum Mechanics and Spectroscopy*. Open Educ. Resour. (OER) LibreTexts Project, Jun. 2019, Accessed: November 4, 2024. [Online]. Available: https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Time_Dependent_Quantum_Mechanics_and_Spectroscopy_(Tokmakoff)/07%3A_Interaction_of_Light_and_Matter/7.05%3A_Absorption_Cross-Sections (cited on pages 43 and 44)

160. Z. Tong *et al.*, "TSDSR: Temporal–spatial domain denoise super-resolution photon-efficient 3D reconstruction by deep learning," *Photonics*, vol. 10, no. 7, p. 744, Jun. 2023, Accessed: August 1, 2024, doi: 10.3390/photonics10070744. [Online]. Available: https://www.mdpi.com/2304-6732/10/7/744 (cited on pages 14 and 32)

161. S.-Y. Tsai, Y.-C. Chang, and T.-H. Sang, "SPAD LiDARs: modeling and algorithms," in *Proc. 14th IEEE International Conf. on Solid-State and Integrated Circuit*

Technology (ICSICT), Qingdao, China, Oct. 2018, pp. 1–4, doi: 10.1109/ICSICT.2018.8565812 (cited on pages 23 and 32)

162. A. C. Ulku *et al.*, "A 512 × 512 SPAD image sensor with integrated gating for widefield FLIM," *IEEE J. of Sel. Topics in Quantum Electron.*, vol. 25, no. 1, pp. 1–12, Jan. 2019, Accessed: December 15, 2020, doi: 10.1109/JSTQE.2018.2867439. [Online]. Available: https://ieeexplore.ieee.org/document/8449092/ (cited on page 6)

163. B. D. Vaughan, "Soldier-in-the-loop target acquisition performance prediction through 2001: integration of perceptual and cognitive models," U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, U.S, Tech. Rep. ARL-TR-3833, Jul. 2006, Accessed: November 7, 2024. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=08cedb53b4c4de68b4709b6d085e242f470789eb (cited on page 3)

164. F. Villa *et al.*, "SPAD smart pixel for time-of-flight and time-correlated single-photon counting measurements," *IEEE Photon. J.*, vol. 4, no. 3, pp. 795–804, Jun. 2012, Accessed: October 6, 2023, doi: 10.1109/JPHOT.2012.2198459. [Online]. Available: https://ieeexplore.ieee.org/document/6198279/ (cited on page 41)

165. ——, "CMOS imager with 1024 SPADs and TDCs for single-photon timing and 3-D time-of-flight," *IEEE J. of Sel. Topics in Quantum Electron.*, vol. 20, no. 6, pp. 364–373, Nov. 2014, Accessed: March 17, 2019, doi: 10.1109/JSTQE.2014.2342197. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6862847 (cited on pages 37 and 41)

166. F. Villa, F. Severini, F. Madonini, and F. Zappa, "SPADs and SiPMs arrays for long-range high-speed light detection and ranging (LiDAR)," *Sensors*, vol. 21, no. 11, p. 3839, Jun. 2021, Accessed: April 30, 2022, doi: 10.3390/s21113839. [Online]. Available: https://www.mdpi.com/1424-8220/21/11/3839 (cited on page 14)

167. P. Vines *et al.*, "High performance planar germanium-on-silicon single-photon avalanche diode detectors," *Nature Commun.*, vol. 10, no. 1086, Mar. 2019, Accessed: April 8, 2024, doi: 10.1038/s41467-019-08830-w. [Online]. Available: https://www.nature.com/articles/s41467-019-08830-w (cited on page 37)

168. E. Wade, A. McCarthy, R. Tobin, A. Halimi, J. Garcia-Armenta, and G. S. Buller, "Micro-scanning of a focal plane detector array in a single-photon LiDAR system for improved depth and intensity image reconstruction," in *Proc. SPIE*, R. A. L. M. L. Gerald S. Buller, Richard C. Hollins, Ed., vol. 12274, Berlin, Germany, Dec. 2022, p. 1227404, doi: 10.1117/12.2639979 (cited on page 14)

169. E. Wade, R. Tobin, A. McCarthy, and G. S. Buller, "Sub-pixel micro scanning for improved spatial resolution using single-photon LiDAR," in *Proc. SPIE*, M. A. Itzler, K. A. McIntosh, and J. C. Bienfang, Eds., vol. 11721, Online Only, Apr. 2021, p. 1172106, doi: 10.1117/12.2588766 (cited on page 14)

170. A. M. Wallace, J. Ye, N. J. Krichel, A. McCarthy, R. J. Collins, and G. S. Buller, "Full waveform analysis for long-range 3D imaging laser radar," *EURASIP J. on Advances in Signal Process.*, no. 896708, Dec. 2010, Accessed: March 20, 2023, doi: 10.1155/2010/896708. [Online]. Available: https://asp-eurasipjournals.springeropen.com/articles/10.1155/2010/896708 (cited on pages 15 and 32)

171. H. Wang, S. Qiu, and T. Lu, "Reconstruction of high-resolution depth profiling from single-photon data based on PCA," in *Proc. SPIE*, M. Li, H. Asghari, K. Shi, and N. Shi, Eds., vol. 12772, Beijing, China, Nov. 2023, p. 127720J, doi: 10.1117/12.2688550 (cited on pages 13 and 32)

172. R. Wang, B. Liu, Z. Li, H. Yi, Z. Guo, and Z. Chen, "Bayesian model for 3D undulating terrain depth estimation using photon counting LiDAR," *Remote Sens.*, vol. 15, no. 19, p. 4719, Sep. 2023, Accessed: August 1, 2024, doi: 10.3390/rs15194719. [Online]. Available: https://www.mdpi.com/2072-4292/15/19/4719 (cited on pages 13 and 32)

173. W. F. Woods *et al.*, "Object detection and recognition using laser radar incorporating novel SPAD technology," in *Proc. SPIE*, M. D. Turner and G. W. Kamerman, Eds., vol. 11005, Baltimore, U.S, May 2019, p. 1100504, doi: 10.1117/12.2517869 (cited on pages 13 and 32)

174. D. Xie, C. Wang, K. Yuan, X. Wei, and X. Liu, "Adaptive fading Kalman depth estimation with chi-square hypothesis testing for Geiger-mode avalanche photo diode LIDAR," *Opt. Eng.*, vol. 62, no. 7, Jul. 2023, Accessed: August 1, 2024, doi: 10.1117/1.OE.62.7.073104. [Online]. Available: https://www.spiedigitallibrary.org/journals/optical-engineering/volume-62/issue-07/073104/Adaptive-fading-Kalman-depth-estimation-with-chi-square-hypothesis-testing/10.1117/1.OE.62.7.073104.full (cited on pages 13 and 32)

175. R. Xu *et al.*, "Single-photon lidar depth imaging based on sparsity adaptive matching pursuit method," *Opt. and Lasers in Eng.*, vol. 180, p. 108314, Sep. 2024, Accessed: August 2, 2024, doi: 10.1016/j.optlaseng.2024.108314. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0143816624002938 (cited on pages 12 and 32)

176. X. Yang *et al.*, "Deep-learning based photon-efficient 3D and reflectivity imaging with a 64x64 single-photon avalanche detector array," *Opt. Exp.*, vol. 30, no. 11, pp. 32 948–32 964, Aug. 2022, Accessed: November 3, 2024, doi: 10.1364/OE.465918. [Online]. Available: https://opg.optica.org/oe/fulltext.cfm?uri=oe-30-18-32948&id=495569 (cited on pages 13 and 32)

177. F. Zappa, "32x32 SPAD+TDC camera," SPAD lab, Accessed: October 6, 2023. [Online]. Available: http://www.everyphotoncounts.com/files/Datasheet_32x32_SPAD+TDC_camera.pdf (cited on page 41)

178. F. Zappa and others, "Microelectronic 3d imaging and neuromorphic recognition for autonomous UAVs," in *Advanced Technologies for Security*

Applications: Proceedings of the NATO Science for Peace and Security'Cluster Workshop on Advanced Technologies', 17-18 September 2019, Leuven, Belgium, C. Palestini, Ed. Dordrecht, The Netherlands: Springer, 2020, ch. 17, pp. 185–194, Accessed: February 15, 2024, doi: https://doi.org/10.1007/978-94-024-2021-0_17. [Online]. Available: https://link.springer.com/chapter/10.1007/978-94-024-2021-0_17 (cited on pages 6, 9, 32, 51, and 57)

179. C. Zhang, Y. Wang, Y. Yin, and B. Sun, "High precision 3D imaging with timing corrected single photon LiDAR," *Opt. Exp.*, vol. 31, no. 15, pp. 24 481–24 491, Jul. 2023, Accessed: August 1, 2024, doi: 10.1364/OE.493153. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=oe-31-15-24481 (cited on page 32)

180. X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. on Comput. Vis. and Pattern Recognit.*, Salt Lake City, UT, U.S, Jun. 2018, pp. 6848–6856, doi: 10.1109/CVPR.2018.00716 (cited on pages 75, 76, 78, 79, and 80)

181. Y. Zhang, "Dual-parameter estimation algorithm for Gm-APD Lidar depth imaging through smoke," *Measurement*, doi: 10.1016/j.measurement.2022.111269. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224122005127 (cited on pages 17, 18, 19, 32, and 144)

182. Y. Zhang *et al.*, "Three-dimensional imaging of ships in the foggy environment using a single-photon detector array," *Optik*, vol. 272, p. 170310, Nov. 2022, Accessed: March 29, 2023, doi: 10.1016/j.ijleo.2022.170310. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0030402622015686 (cited on pages 18, 19, 29, 32, and 144)

183. ——, "Three-dimensional single-photon imaging through realistic fog in an outdoor environment during the day," *Opt. Exp.*, vol. 30, no. 19, pp. 34 497–34 509, Sep. 2022, Accessed: March 29, 2023, doi: 10.1364/OE.464297. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=oe-30-19-34497 (cited on pages 3, 17, 18, 19, 32, 143, 144, and 178)

184. Y. Zhang, S. Li, J. Sun, X. Zhang, X. Zhou, and H. Zhang, "Noise-tolerant depth image estimation for array Gm-APD LiDAR through atmospheric obscurants," *Opt. & Laser Technol.*, vol. 175, p. 110706, Aug. 2024, Accessed: August 2, 2024, doi: 10.1016/j.optlastec.2024.110706. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0030399224001646 (cited on pages 19 and 32)