Measurements and Instrumentation for Machine Vision

Oleg Sergiyenko, Wendy Flores-Fuentes,
Julio C. Rodríguez-Quiñonez, and
Jesús E. Miranda-Vega (eds.)

CRC Press
Taylor & Francis Group

A SCIENCE PUBLISHERS BOOK

Measurements and Instrumentation for Machine Vision

Editors

Oleg Sergiyenko
Universidad Autónoma de Baja California
México

Wendy Flores-Fuentes
Universidad Autónoma de Baja California
México

Julio C. Rodríguez-Quiñonez
Universidad Autónoma de Baja California
México

Jesús E. Miranda-Vega
Tecnológico Nacional de México/IT de Mexicali
México

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
A SCIENCE PUBLISHERS BOOK

First edition published 2024
by CRC Press
2385 NW Executive Center Drive, Suite 320, Boca Raton FL 33431

and by CRC Press
4 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

CRC Press is an imprint of Taylor & Francis Group, LLC

Library of Congress Cataloging-in-Publication Data (applied for)

ISBN: 978-1-032-38161-9 (hbk)
ISBN: 978-1-032-38163-3 (pbk)
ISBN: 978-1-003-34378-3 (ebk)

DOI: 10.1201/9781003343783

Typeset in Times New Roman
by Radiant Productions

Preface

Measurements and Instrumentation for Machine Vision is the first book in a series from the Taylor & Francis Engineering/Measurement and Instrumentation field at CRC Press. This book is a reference for the scholars' audience, it includes advanced students, early career and established researchers. Readers are provided with selected topics in the state of the art and relevant novelty content in the frontier of knowledge for the engineering implementation, scientific knowledge and technological innovation development related to machine vision. Each chapter is the result of an expert's research and a collaborative work evaluated by a peer review process and consulted by a book editorial board. The importance of this book's topic relies on the fact that machine vision is the basis of cyber-physical systems to be capable of interrelating with humans.

Machine vision is considered the eyes of cybernetics systems and plays a fundamental role in Industry 4.0 and Society 5.0 for the joining of the virtual and real world to coexist in a new era in human lives that integrates the technologies into their daily lives with creativity, and globalization through interconnectivity.

Industry 4.0 implies the development of cyber-physical systems capable of seeing, making decisions and acting; while society 5.0 searches for solutions for real problems to better human life conditions, based on the application of all the benefits and advantages that offers Industry 4.0, this is the last and is also known as the fourth industrial revolution. Although it is not an easy task, it requires the application of measurement fundamentals and methods and the development of instrumentation strategies and technologies for the achievement of machine vision systems for every specific application.

It is a revolution that seeks to replace some of the functions that humans perform with optimal, efficient, automated and interconnected processes; called the Industrial Internet of Things (IIoT). A multidisciplinary integration of mainly electronic and mechatronic devices for signal emitters, sensors and cameras, artificial intelligence algorithms, embedded systems, instrumentation and control, actuators, robotics, interconnectivity, data science and cloud computing, that is the application and development of multiple interconnected disciplines.

In the beginning, one of the dreams of humans has been for machines to do our work, while we enjoy rest and other activities that give us pleasure. This gave rise to the industrial revolution, which brought as a benefit an increase in the world population, an increase in the average rate of living, as well as new lifestyles, with tasks that involve less physical effort and that provide us with better quality of life. But now the dreams of humans are a green, healthy and intelligent life, interconnected,

under continuous monitoring and with accurate control for sustainability, not only for living on the Earth but also to be able to search and live in other places in the universe.

The integration of optoelectronics devices for emitters, sensors and cameras, artificial intelligence algorithms, embedded systems, robust control, robotics, interconnectivity, big data, and cloud computing is the core of machine vision developments for cyber-physical systems to collaborate with humans and their real and virtual environments and activities. It is required to focus on the theory, methods, tools and techniques for the design, instrumentation and measurement methods applied in machine vision systems.

Measurements are values assigned to refer to a physical quantity or phenomenon; they play an important role in science development. One of the most crucial points in the measurement of machine vision is the proper estimation of information transform quality: such parameters as sensitive part calibration, scene size traceability, accuracy/uncertainty, receiver operating characteristic, repeatability, reproducibility, etc.

The use of current technology requires measuring essential attributes from objects, health data, dimensions of a surface and weather, to mention some. These are necessary to do breakthrough innovations in a wide range of fields. The artificial intelligence (AI) field is one of them. This field is aimed at the research for imitating human abilities, above all, how they learn. AI can be divided into two main branches such as machine learning and deep learning. These components are methods and algorithms that are used for prediction and analysis in the case of machine perception. Although deep learning is also used for forecasting as well as machine learning, the way how the model is created is different. This sub-branch is inspired by how the human brain can learn. In other words, this is based on behavior's neurons to make connections among others to solve a particular problem. In other contexts, measurements are known as data or instances and depending on the quality of them a problem can be solved. Convolutional neural networks (CNN) have catalyzed several vision recognition systems.

Accuracy measurement for medical data is valuable information for a doctor because it can prevent a patient's illness. Health monitoring is required for people who need to constantly check for signs of illness. Data generated by applications can be used by specialists or a smart health system to send an alert in real-time. For example, as well as blood pressure and pulse monitoring are common requirements of a doctor, also medical images obtained by machine vision systems are fundamental. These medical diagnostics can be assisted by machine learning algorithms with the purpose to offer better outcomes.

With the advancements in technology, community researchers have access to better computers in comparison to decades ago. Smart applications aided by computer technology are designed to solve many problems. This progress has increased the interest in the development of cognitive machines, especially in industry. Measurements provided by machine vision systems dotted by AI technology help factories become more efficient and productive. The interaction of humans and machines is possible thanks to accurate smart sensors. Robots safely interact with humans by taking care of the distance between them, they are called cobots. Optical sensors and cameras are the main elements that regularly can be used for this purpose.

SPAD (Single Photon Avalanche Diode) sensors open new possibilities for the development of an accurate vision system by counting individual photon particles instead of the volume of the light by pixel (CMOS technology). Cameras RGB-D depict an important advancement that has recently gained attention in the design of vision systems. Depth information can be estimated with this type of camera to solve the most promising vision tasks.

The application of measurements and instrumentation are infinitive, while the submissions of the present book are addressed to machine vision applications, where the challenge to provide the ability to see, measure, track, create models, interconnect, take decisions and self-adaptation of cyber-physical systems involves multidisciplinary and interdisciplinary requirements with always in mind the sustainability and human-centred resolution of local, regional, and global needs and problems.

Each chapter's contribution demonstrates a deep review of the state of the art, as well as reports the most recent theoretical novelty backgrounded with research results of novel proposals that contribute to current knowledge and development challenges toward futurist trends.

Chapter 1 Machine Learning Approaches for Single Photon Direct Time of Flight Imaging: Reviews the state-of-art deep learning techniques based on processing approaches in the context of SPAD (Single-photon avalanche diodes) sensors. These sensors can offer advantages for operation in outdoor environments with satisfactory results and maintaining a high frame rate. Furthermore, a general description of machine vision tasks such as surface and object detection and super-resolution depth mapping are discussed in detail. Spiking Neural Networks (SNN) are analyzed for Direct time-of-flight (dToF) sensors, which can better exploit SPAD's discrete event-based nature.

Chapter 2 Experimental Evaluation of Depth Measurements Accuracy in Indoor and Outdoor Environments: One of the main goals of this chapter is to integrate a TVS with a stereovision system. For the purpose of the work, a red laser is used to be reflected on a surface, and then a camera RGB-D can capture the scene under observation. Color and depth frames were collected from the depth camera. The main idea is to find the laser point in the color frame to get the corresponding depth information. Four different image processing techniques were applied for smoothing the color frames of the beam laser. An overview of the methods for depth measurements and the key concepts of machine vision is presented.

Chapter 3 Design and Evaluation Support System for Convolutional Neural Network, Support Vector Machine and Convolutional Autoencoder proposes a design, training, and application evaluation based on convolutional neural network (CNN), support vector machine (SVM), convolutional autoencoder (CAE). These authors designed a front-end tool implemented on Matlab based on deep learning structures. The application shown in this work can be addressed to industry for material surface defects detection, such as burrs, cracks, protrusions, chippings, spots and fractures.

Chapter 4 Classification of Objects in IR Images using Wavelet Filters based on Lifting Scheme describes an emerging model using a small CNN and wavelet filters based on a lifting scheme (multi-resolution analysis (MRA) to improve the feature extraction process. A brief overview of infrared image processing is presented. Experiments are conducted with four benchmark datasets for training and testing the models.

Chapter 5 Image Dataset Augmentation: A Survey and Taxonomy Highlight that the lack of training data can affect the performance of a machine learning model. This work is focused on image dataset augmentation approaches to enhance a computer vision system. The key concepts of deep learning and data collection for data augmentation are discussed. The main data image augmentation techniques such as color and geometric transformation, as well as a practical application of this technique, are discussed.

Chapter 6 A Filter-based Feature Selection Methodology for Vehicle/Non-Vehicle Classification presents a machine learning-based model to detect vehicles from a dataset having classes: vehicles and non-vehicles. The main focus of the chapter is to implement a Histogram of Oriented Gradients (HOG) as feature extraction. The method proposed is validated with two benchmark datasets, such as GTI and Kaggle.

Chapter 7 Augmented Visual Inertial Wheel Odometry Through Slip Compensation: This chapter proposes a strategy for compensating wheel slip, based on a differential drive robot kinematics model. The mobile robot presented in this work is equipped with wheel encoders which measure the left and right wheel linear velocities. The key components of a visual-inertial wheel odometry (VIWO) scheme are described. Experimental results obtained through the KAIST dataset to verify the method proposed are presented.

Chapter 8 Methodology for Developing Models of Image Color of Terrain with Landmarks for their Detection and Recognition by Autonomous Mobile Robots: This chapter suggests the creation of a virtual dynamic system based on an image that is scanned by an imaginary fan-shaped beam. A general overview of the construction of automobile mobile robots is presented. The authors describe a methodology for creating models of the color of landmarks against the background of arbitrary terrain. These models are addressed to use in robot navigation.

Chapter 9 Machine Vision – A Measurement Tool for Agricultural Automation: Describes how machine vision applications can improve the agriculture sector. Activities such as fruit counting, fruit robotic harvesting and fruit yield estimation can be automated for attending to an important issue for food security. A modeling of the visual feedback control system was proposed for the navigation task of a mobile robot. The vision system for the agriculture approach is based on the Orchard robot (OrBot) that includes an RGB-D camera. A practical fruit harvesting is carried out in which processing time is measured.

Chapter 10 Occlusion-Aware Disparity-based Direct Visual Servoing of Mobile Robots: Refers to position control of a mobile robot taking into account the camera's

velocity and a disparity map computed from the stereo images. This visual servoing system is integrated by a stereo camera. The proposed visual servoing framework is verified by simulations and experiments. Physical experiments with a mobile robot were carried out for positioning tasks, in which occlusion information in the controller design was integrated.

Chapter 11 Development of a Software and Hardware Complex for the Visualization of Sedimentation Inside a Vortex Chamber develops an application system for visualizing sedimentation inside a vortex chamber. This system works under difficult operating conditions and temperatures over 1000ºC. A complete software and hardware package for the operation of the system mentioned is developed for the monitoring process. Image enhancement algorithms such as distortion correction and background subtraction are detailed and validated with a practical application.

Chapter 12 Machine Vision for Astronomical Images Using the Modern Image Processing Algorithms Implemented in the CoLiTec Software: Big data in astronomy requires high-dimensional information. As a consequence, scientific analysis and data mining represent a challenge for astronomers and scientists around the world. This chapter describes a general overview of the different modern image processing algorithms and their implementation of them in the Collection Light Technology (CoLiTec) software. CoLitec was designed to perform MV tasks for astronomical objects.

Chapter 13 Gallium Oxide UV Sensing in Contemporary Applications explores advanced applications of UV-C photodetectors in contemporary computer machine vision systems. Sensing UV light has become challenging for researchers and arrays of a component such as Gallium oxide can be useful in imaging and machine vision applications. This chapter also gives an overview of photosensors based on vacuum and solid-state devices.

Chapter 14 Technical Vision System for Alive Human Detection in an Optically Opaque Environment focuses on the technical problems of creating a highly sensitive vision system for detecting and identifying living people behind optically opaque obstacles by estimation of Doppler phase shifts in reflected signals caused by the process of breathing and heartbeat of the living human. This work also discusses the advantages and disadvantages of hardware implementations for this task.

Chapter 15 The Best Linear Solution for a Camera and an Inertial Measurement Unit Calibration is addressed for applications where one or more sources of noise are present in a system. In this work, a set of experiments were carried out to find a rotational offset between the two sensors as well as to verify the robustness of the solution proposed based on basic least squares. The effect of noise concerning multiple positioning of the camera´s rotational matrix versus IMU′s rotational matrix was analyzed, to measure the offset between the two sensors for geometrical calibration between them to enhance visual perception and pattern recognition capabilities.

Chapter 16 Methods of Forming and Selecting a Reference Image to Provide High-Speed Navigation for Maneuvering Aircraft provides a given speed and accuracy of navigation of high-speed manoeuvring aircraft by reducing the computational complexity of processing superimposed images in combined extremal-correlation flyers navigation systems. This work is focused on eliminating emerging uncertainties when using multispectral sensors as part of combined navigation systems of flyers.

Chapter 17 Application of Fibre Bragg Grating Arrays for Medical Diagnosis and Rehabilitation Purposes: gives an overview of a human health application based on a semiconductor material such as Ga2o3 used on sensors for physiological parameters measurements. This study is based on Fibre Bragg Grating Arrays (FBGA) for physiological pulse detection. The work highlights the advantages of using FBGAs due to it being provided multiple data from a region of interest simultaneously. These applications minimize the effects of sensor location and facilitate locational referencing capabilities.

Acknowledgement

The editors would like to offer our acknowledgement to all the contributors for their time and effort. We are delighted to have this academic product with a global vision. It was a pleasure to work with researchers in the areas of Machine Vision, Navigation, Robotics, Control, and Artificial Intelligence. Seventy-five researchers from around the world have collaborated on this project, representing the participation of fourteen countries: Canada, China, Egypt, France, India, Japan, Malaysia, Mexico, Russia, Spain, Thailand, Ukraine, the United Kingdom and the United States. A whole list of authors with affiliations is available in the "List of Authors" section of this book.

We are grateful for the indispensable role of the following reviewers who have done a wonderful job reading and suggesting improvements for each chapter. Oleksandr Tsymbal, Thomas Tawiah, Andrey Somov, Abdulkader Joukhadar, Swagato Das, Moises Jesus Castro Toscano, Oscar Real Moreno, Javier Sanchez Galan, Giovanni Fabbrocino, Hang Yuan, Huei-Yung Lin, Miti Ruchanurucks, Junzhi Yu, Piotr M. Szczypiński, Tadeusz Uhl, Shahnewaz Ali, Subhajit Maur, Oleksandr Tymochko, Guilherme B. Pintarelli, Sergey V. Dvoynishnikov, Oleksander Poliarus, Dah-Jye Lee, Pr. Azzeddine Dliou, Alberto Aloisio, Dayi Zhang, Junfan Wang, Oleg Starostenko, Jonathan Jesús Sanchez Castro, Ruben Alaniz-Plata, Jacek Izydorczyk, Muaz Khalifa Alradi, Radhakrishna Prabhu, Tanaka Kanji and Sergii Khlamov.

Special thanks also go to the editorial board and the officials at CRC Press/Taylor & Francis Group for their invaluable efforts, great support and valuable advice for this project towards the successful publication of this book. We are also grateful to our institutions Universidad Autónoma de Baja California, and Tecnológico Nacional de México/IT de Mexicali to provide us with a location and time where to develop this project.

Oleg Sergiyenko
Universidad Autónoma de Baja California, México

Wendy Flores-Fuentes
Universidad Autónoma de Baja California, México

Julio C. Rodríguez-Quiñonez
Universidad Autónoma de Baja California, México

Jesús E. Miranda-Vega
Tecnológico Nacional de México/IT de Mexicali, México

Contents

List of Contributors

Afanasiev Volodymyr
Kharkiv National Air Force University named after Ivan Kozhedub, Kharkiv, Ukraine

Andrey Somov Skoltech
Skolkovo Institute of Science and Technology Russia

Arnoldo Díaz-Ramirez
Department of Computer Systems, Tecnológico Nacional de México, IT de Mexicali, Mexicali, BC 21376, México

Arup Roy
School of Computing and Information Technology, Reva University, Bengaluru, Karnataka- 560064, India

Atikul Islam
Department of Computer Science & Engineering, Annex College of Management Studies, BD-91, Salt Lake Bypass, Sector 1, Bidhannagar, Kolkata, West Bengal 700064, India

Azman Jalar
Department of Applied Physics, Faculty of Science and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

Baris Fidan
Department of Mechanical and Mechatronics Engineering University of Waterloo

Brian Stewart
STMicroelectronics

Brice Allen
Department of Engineering and Physics, Northwest Nazarene University

Chaker Tlili
Chongqing Key Laboratory of Multi-scale Manufacturing Technology, Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, People's Republic of China.

Daniel Trevino-Sanchez
Department of Computing, Electronics and Mechatronics Universidad de las Américas Puebla, Sta. Catarina Martir, San Andres Cholula, Puebla 72810, México

Dmitriev Oleg
Flight Academy of National Aviation University, Kropyvnytzkyy, Ukraine

DO Semenov
Kutateladze Institute of thermophysics SB RAS Lavrentyeva avenue 1 , Novosibirsk, Russia

Duke M Bulanon
Department of Engineering and Physics, Northwest Nazarene University

Ehsan Hashemi
Department of Mechanical Engineering University of Alberta

Fusaomi Nagata
Sanyo-Onoda City University, 1-1-1 Daigaku-Dori, Sanyo-Onoda, 756-0884, Japan

Fustii Vadim
Kharkiv National Air Force University named after Ivan Kozhedub, Kharkiv, Ukraine

Garrisen Cizmich
Department of Engineering and Physics, Northwest Nazarene University

GV Bakakin
Kutateladze Institute of thermophysics SB RAS Lavrentyeva avenue 1 , Novosibirsk, Russia

IK Kabardin
Kutateladze Institute of thermophysics SB RAS Lavrentyeva avenue 1 , Novosibirsk, Russia **Iryna Tabakova**
Kharkiv National University of Radio Electronics, 14 Nauki Avenue, 61166 Kharkiv, Ukraine

Isaac Compher
Department of Engineering and Physics, Northwest Nazarene University

Istvan Gyongy
School of Engineering, Institute for Integrated Micro and Nano Systems, The University of Edinburgh

Jack Iain MacLean
School of Engineering, Institute for Integrated Micro and Nano Systems, The University of Edinburgh, STMicroelectronics

Jesús E Miranda-Vega
Department of Computer Systems, Tecnológico Nacional de México, IT de Mexicali, Mexicali, BC 21376, México

Joseph Ichiro Bulanon
Department of Engineering and Physics, Northwest Nazarene University

Julio C Rodríguez-Quiñonez
Universidad Autónoma de Baja California, México

Keigo Watanabe
Okayama University, Okayama, Japan

Kento Nakashima
Sanyo-Onoda City University, 1-1-1 Daigaku-Dori, Sanyo-Onoda, 756-0884, Japan

Kohei Miki
Sanyo-Onoda City University, 1-1-1 Daigaku-Dori, Sanyo-Onoda, 756-0884, Japan

Koki Arima
Sanyo-Onoda City University, 1-1-1 Daigaku-Dori, Sanyo-Onoda, 756-0884, Japan

Maki K Habib
Mechanical Engineering Department, School of Sciences and Engineering, American University in Cairo, AUC Avenue, P.O. Box 74, New Cairo 11835, Egypt

Manish Mishra
School of Electrical Sciences, Indian Institute of Technology, Bhubaneswar, Argul - Jatni Road, Kansapada, Odisha – 752050, India

Marina Kolendovska
Kharkiv National University of Radio Electronics, 14 Nauki Avenue, 61166 Kharkiv, Ukraine

Maroi Agrebi
LAMIH UMR CNRS 8201, Department of Computer Science, Université Polytechnique Hauts-de-France, 59313 Valenciennes, France

Miti Ruchanurucks
Electrical Engineering Department, Kasetsart University, Thailand

Mohd Firdaus Raih
Department of Applied Physics, Faculty of Science and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia/ Institute of Systems Biology, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia

Muhammad Azmi Abdul Hamid
Department of Applied Physics, Faculty of Science and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

Naif H Al-Hardan
Department of Applied Physics, Faculty of Science and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

Naser M Ahmed
School of Physics, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia

Niraj Reginald
Department of Mechanical and Mechatronics Engineering University of Waterloo

Oleg Sergiyenko
Universidad Autónoma de Baja California, México

Oleg Sytnik
O.Ya. Usikov Institute for Radio Physics and Electronics, National Academy of Sciences of Ukraine, 12 Academician Proskura St., Kharkov 61085, Ukraine

Olizarenko Serhij
Flight Academy of National Aviation University, Kropyvnytzkyy, Ukraine

Oleksandr Poliarus
Kharkiv National Automobile and Highway University, Ukraine

Omar Al-Buraiki
Department of Mechanical and Mechatronics Engineering University of Waterloo

Pawan Kumar Singh
Department of Information Technology, Jadavpur University, Jadavpur University Second Campus, Plot No. 8, Salt Lake Bypass, LB Block, Sector III, Salt Lake City, Kolkata-700106, West Bengal, India

Prasant Kumar Sahu
School of Electrical Sciences, Indian Institute of Technology, Bhubaneswar Argul - Jatni Road, Kansapada, Odisha – 752050, India

Ratchakorn Srihera
Ryowa Co., Ltd., 10-5 Torigoe-Cho, Kanda-Machi, Miyako-Gun, Fukuoka, Japan

Saurav Mallik
Center for Precision Health, School of Biomedical Informatics, The University of Texas Health Science Center at Houston, Houston, TX, 77030, USA

Sergey Nesteruk
Skolkovo Institute of Science and Technology Russia

Sergii Khlamov
Kharkiv National University of Radio Electronics, 14 Nauki Avenue, 61166 Kharkiv, Ukraine

Soo Jeon
Department of Mechanical and Mechatronics Engineering University of Waterloo

Sotnikov Oleksandr
Kharkiv National Air Force University named after Ivan Kozhedub, Kharkiv, Ukraine

Stepanenko Dmitryy
Flight Academy of National Aviation University, Kropyvnytzkyy, Ukraine

Surangrak Sutiworwan
Office of Information and Communications Technology, United Nations, Thailand

SV Dvoinishnikov
Kutateladze Institute of thermophysics SB RAS Lavrentyeva avenue 1 , Novosibirsk, Russia

Svetlana Illarionova
Skolkovo Institute of Science and Technology Russia

Tatsuki Shimizu
Sanyo-Onoda City University, 1-1-1 Daigaku-Dori, Sanyo-Onoda, 756-0884, Japan

Tetiana Trunova
Kharkiv National University of Radio Electronics, 14 Nauki Avenue, 61166 Kharkiv, Ukraine

Tiurina Valeriia
Kharkiv National Air Force University named after Ivan Kozhedub, Kharkiv, Ukraine

Trystan Andriy
State Scientific Research Institute of Armament and Military Equipment Testing and Certification, Cherkasy, Ukraine

Tymochko Oleksandr
Flight Academy of National Aviation University, Kropyvnytzkyy, Ukraine

VA Pavlov
Kutateladze Institute of thermophysics SB RAS Lavrentyeva avenue 1 , Novosibirsk, Russia

Vadym Savanevych
Kharkiv National University of Radio Electronics, 14 Nauki Avenue, 61166 Kharkiv, Ukraine

VG Meledin
Kutateladze Institute of thermophysics SB RAS Lavrentyeva avenue 1 , Novosibirsk, Russia

Vicente Alarcon-Aquino
Department of Computing, Electronics and Mechatronics Universidad de las Américas Puebla, Sta. Catarina Martir, San Andres Cholula, Puebla 72810, México

Vladimir Kartashov
Kharkov National University of Radio Electronics,14 Nauky Ave., Kharkov 61085, Ukraine

Vladimir Kartashov
Kharkiv National University of Radio Electronics, 14 Nauki Avenue, 61166 Kharkiv, Ukraine

VV Rakhmanov
Kutateladze Institute of thermophysics SB RAS Lavrentyeva avenue 1 , Novosibirsk, Russia

Wendy Flores-Fuentes
Universidad Autónoma de Baja California, México

Wendy García-Gonzalez
Universidad Autónoma de Baja California, México

Xiule Fan
Department of Mechanical and Mechatronics Engineering University of Waterloo

Yevhen Poliakov
Kharkiv National Automobile and Highway University, Ukraine

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

1

Machine Learning Approaches for Single Photon Direct Time of Flight Imaging

Jack Iain MacLean,[1,*] *Brian Stewart*[2] and *Istvan Gyongy*[1]

1.1 Introduction

1.1.1 LiDAR

LiDAR sensors are becoming increasingly widespread in applications requiring 3D imaging or proximity sensing, such as autonomous navigation or machine vision. These optical sensors typically use one of two types of time-of-flight (ToF) approach, indirect time-of-flight (iToF) and direct time-of-flight (dToF), which emit a wave or light pulse to illuminate a scene of interest and time the returning, back-scattered photons to estimate distance, as in Figure 1.1. As the speed of light in the air is constant, the duration for incident photons to return is directly proportional to the distance of the object from the sensor. A sensor is made up of many pixels which are able to independently measure the ToF of objects within the field-of-view (FoV). The difference between iToF and dToF lies in the methodology of time measurement. iToF-based sensors do not directly measure the time between emitted and received pulses, instead, it measures the delay between signals by integrating the received signal during specific windows of time synchronized with the emitted signal.

[1] School of Engineering, Institute for Integrated Micro and Nano Systems, The University of Edinburgh.
[2] STMicroelecronics (R&D) Ltd.
* Corresponding author: s2110140@ed.ac.uk

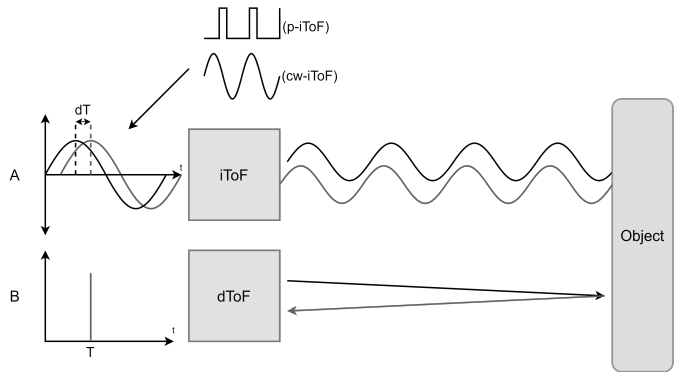


Figure 1.1: The figure illustrates the two different methods of ToF measurement (black: emitted, red: reflected). (A) iToF. The two types of iToF shown are the square waved pulsed-light (p-iToF) and the sinusoidal continuous wave (cw-iToF). (B) dToF.

Sensors based on the dToF principle directly measure the time taken between the laser emission towards the scene and the detection of back-scattered photons. Timing is often performed using an electronic stopwatch called a time-to-digital converter (TDC). The distance (d) can be extracted from the time between the sensor and object using equation 1.1 [Shahnewaz and Pandey, 2020].

$$d = \frac{ct_m}{2} \tag{1.1}$$

where t_m is the time measured by the sensor with a precision of Δt, and c is the speed of light.

Single-photon avalanche diodes (SPADs) are photodetectors that are seeing increased use in 3D depth imaging sensors as they provide a high sensitivity to photons, low timing jitter, and a fast response time to detected photons. SPADs operate in an unstable state called Geiger-mode, where the SPADs are reverse-biased above their breakdown voltage. While operating in Geiger mode, a single photon hitting the SPAD could trigger an avalanche of self-sustaining current flow which can be detected. The resultant current flow caused by photon events activates a Schmitt trigger, as seen in Figure 1.2. The Schmitt trigger outputs a square wave digital pulse which can then be processed by accompanying electronics. The chance of a SPAD detecting a photon event is measured by its photon detection probability (PDP) which, depending on the wavelength of light, can range from a few percent up to fifty percent [Piron et al., 2021]. After the photon is detected and the SPAD avalanches, the bias voltage of the SPAD must be reduced to reset its state. The time taken to reset the SPAD is typically only a few nanoseconds long and is called dead time due to the inability to detect new photons. Two quenching and recharge circuits exist in order to reset the SPAD back to its Geiger-mode operating point. Passive quenching can be seen in Figure 1.2(a) and active quenching in Figure 1.2(b). The difference between these two methods is that active quenching utilizes a MOSFET to connect the SPAD circuit to a voltage source below the SPADs breakdown voltage, which decreases the

time to quench the avalanche current. When the MOSFET is closed the avalanche current is quickly quenched as the circuit is biased to the low voltage source. The addition of active recharge can be seen in Figure 1.2(c) and is achieved by adding a low-capacitance MOSFET in parallel with the quenching resistor. When the switch is closed, due to a photon event being detected, the SPAD circuit voltage quickly returns to its operating voltage thus further reducing the dead time [Vilà et al., 2012].

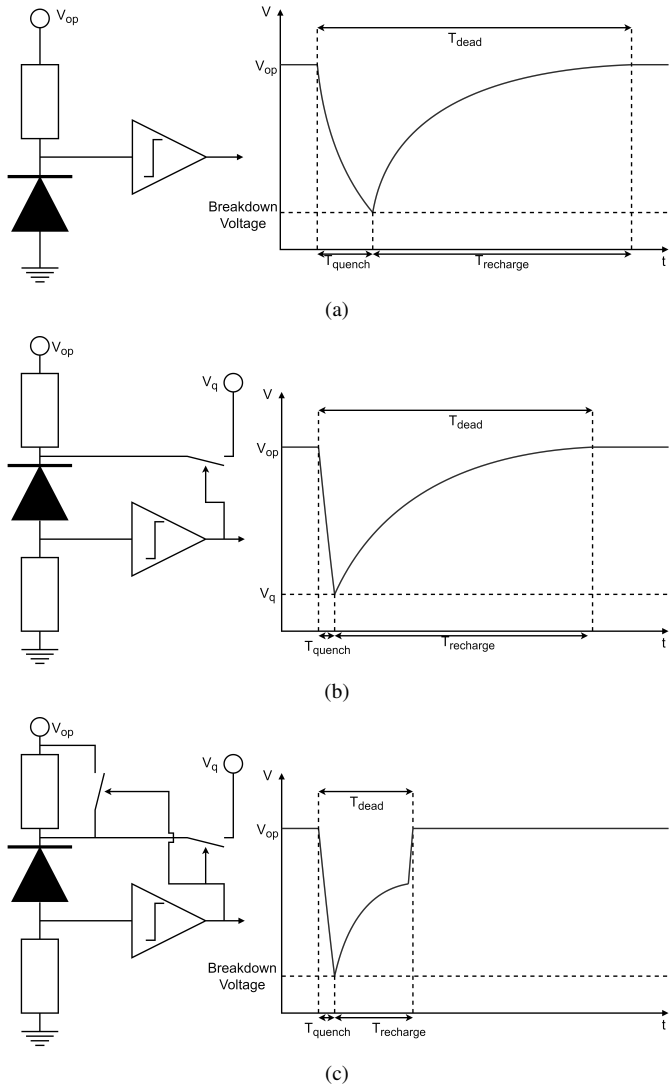


Figure 1.2: (a) SPAD Circuit utilizing passive quenching and recharging, (b) SPAD Circuit utilizing active quenching and passive recharging, (c) Active quenching and active recharge.

The SPAD is also affected by two main types of noise, afterpulsing and the dark count rate (DCR). Afterpulsing is caused by carriers being trapped during an avalanche which are then released after the reset process, causing the SPAD to be triggered again. The DCR is the rate at which SPADs avalanches can be triggered by thermally generated carriers rather than photon events. The main drawbacks of using SPAD-based sensors is that the SPAD pixel requires excess circuitry to operate quenching, counting, timing, and buffering all of which reduce the sensor's fill factor (FF). Typically, this means that SPAD-based sensors have a lower FF when compared to sensors based on other photodetectors. To increase the FF, and reduce the effects of dead time, multiple SPADs are combined via a combination tree, typically OR or XOR, into macro pixels which act as one pixel.

As SPAD-based sensors can generate large quantities of data, the bottleneck inherent in the data readout for off-chip processing is only exacerbated, see Figure 1.3. The most common method for data compression is on-chip per-pixel histogramming, Figure 1.4, which takes advantage of new 3D stacking technology to integrate SPAD and CMOS circuitry onto the same silicon die. The histograms are built up over multiple laser cycles using a TDC which is triggered on each SPAD event to create a timestamp. This timestamp is used to measure increments in the photon count in the corresponding histogram bin in memory. Photon event timestamps are allocated to the appropriate histogram bin in memory, with the histogram being built up over multiple laser cycles until the desired SNR is achieved. The histogram memory is then read out from the sensor and a peak estimation algorithm is applied to extract detected surface depths.

Under high photon fluxes, photon pile-up can occur which prevents the detection of additional photons after the first due to the SPAD quenching circuit being paralyzed by additional detections preventing a recharge. Even at lower photon rates, TDC pile up may occur which distorts the histograms and prevents the detection of back-scattered photons at longer ranges [Gyongy et al., 2022]. One method to prevent this is the use of multi-event TDCs, such as the one implemented in [Al Abbas et al., 2018], which uses a one-hot encoding scheme to encode the photon events combined with a processing pipeline that allows for the recording of time stamps at a much higher rate.

The completed histogram is then read off-chip where a peak detection algorithm is used to extract the depths of the surfaces detected in the pixels FoV. While histogramming does achieve a high degree of data compression, it introduces higher power consumption and requires a significant area of silicon to implement. The general process flow for SPAD-based LiDAR, as seen in Figure 1.5, is that when the reflected incident and ambient photons arrive at the SPAD pixels of the sensor, only a fraction are detected based on the SPAD sensors Photon Detection Efficiency (PDE). The detected photon events cause an avalanche current which activates the Schmitt triggers to output a digital pulse, these pulses are then buffered and optionally shortened. A combination tree, usually consisting of OR or XOR gates, are used to combine multiple SPAD pixel signals into one macro-pixel. A TDC synchronized with a clock then generates timestamps based on the photon events represented in the macro pixel signal and increments the count in the corresponding histogram bin stored in

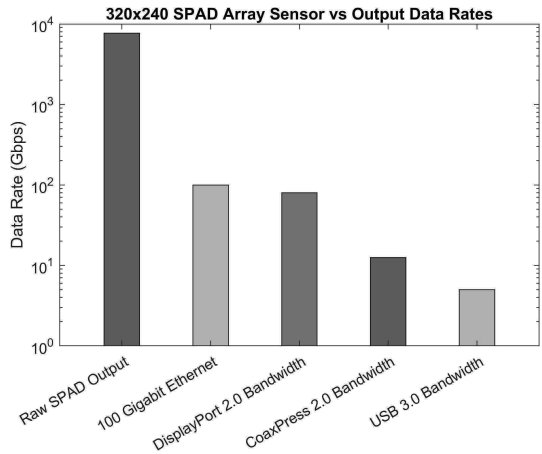


Figure 1.3: Maximum output data rate of a 320x240 SPAD array, assuming each SPAD is firing at 100 Mega counts per second (Mcps), compared to current data transfer solutions.

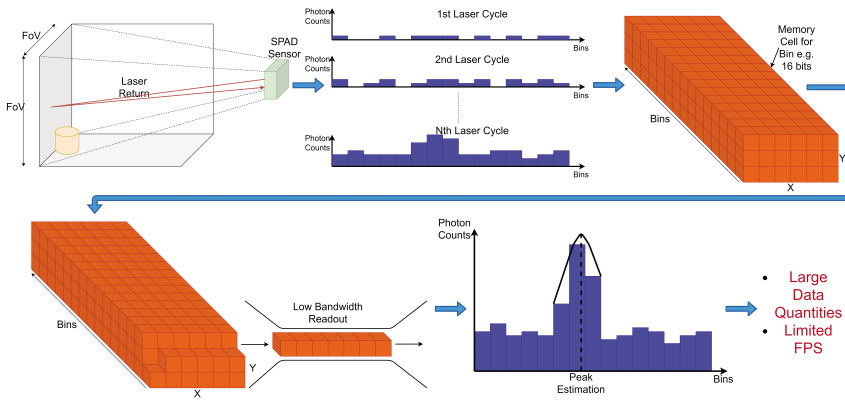


Figure 1.4: On-chip histogram buildup process.

memory. The histogram is then read off the chip and processed into a depth map or 3D point cloud for further processing.

However, in recent years some of these steps have begun to be replaced with neural network implementations. As will be demonstrated in this chapter, the depth estimation algorithms and point cloud generation steps which were integral to machine vision can be replaced with deep neural network implementations, improving latency and reducing power consumption. Even histogramming could be replaced with an appropriate neural network architecture. Such implementations which bypass point cloud generation and output a form of metadata using neural networks can be seen in sensors such as the Sony IMX500 [imx,].

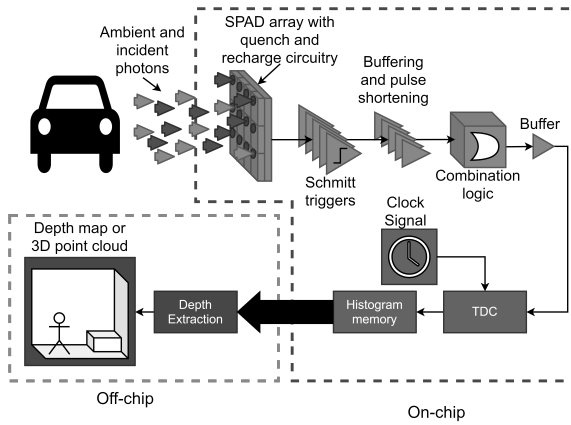
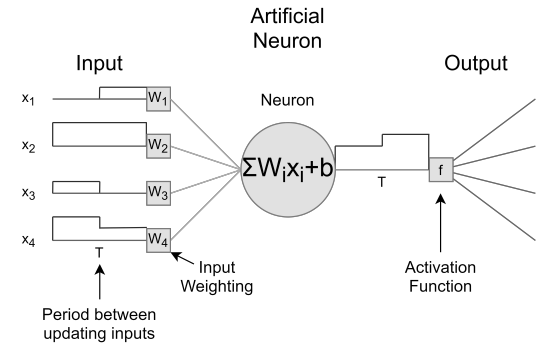


Figure 1.5: Typical SPAD-based dToF processing pipeline.

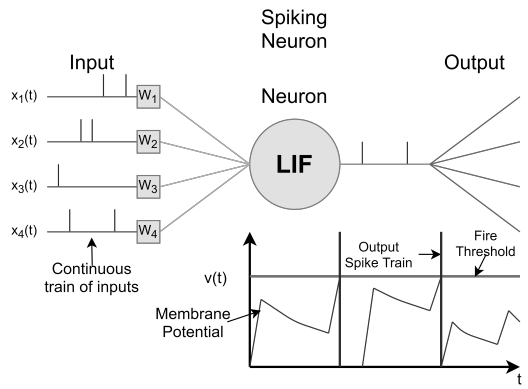
1.1.2 Deep neural and neuromorphic networks

Artificial Neural Networks (ANNs) are seeing increased use in 3D LiDAR applications to perform tasks such as object detection, autonomous navigation, and upsampling. Recently it can be seen that deep learning approaches are outperforming classical machine learning methods. A prime example is in object detection where Convolutional Neural Networks (CNNs) exceed the accuracy of methods such as k-nearest neighbors or gradient boosting trees. The concept of deep learning networks has been around for decades but has only recently become the predominant machine learning method. The main causes for the growing use of deep neural networks are the increase in cheap and powerful computational resources, larger datasets, and more refined training methods for the models [Goodfellow et al., 2016]. Previous issues which plagued deep networks such as vanishing and exploding gradients have effectively been solved with the improvement of the gradient descent optimization algorithm.

As the improvement of training techniques and resources for deep neural networks have grown, and so too has the use of Spiking Neural Networks (SNNs). These neuromorphic networks are predisposed to take advantage of event-based nature of the SPAD. SNNs operate asynchronously and encode information as spikes allowing for better integration of time, frequency, and phase information. Other appealing qualities offered are the greater data sparsity and relatively lower power consumption when compared to traditional synchronous networks. The difference in SNNs architecture when compared to analogue networks can be seen in Figure 1.6, where a Leaky Integrate-and-Fire (LIF) neuron [Koch and Segev, 1998] is compared with traditional artificial neurons. Traditional artificial neurons operate with fixed update periods with constant activation levels. These inputs are then summed with each other and a bias term, with the result being passed through an activation function (typically a ReLU). Spiking neurons operate asynchronously, with each neuron maintaining its own membrane potential. The membrane potential is increased by the arrival of an



(a)



(b)

Figure 1.6: Difference between traditional (a) and spiking neuron (b) architecture.

input spike and, in the case of an LIF neuron, decreases over time in the absence of input spikes. When the membrane potential reaches a set threshold an output spike is produced and the membrane potential is reset.

While the advantages of spiking architecture are clear, the main problem preventing their widespread use is the lack of a developed training method [Taherkhani et al., 2020] which also limits their size to a few layers. One method to circumvent this is to convert a traditionally trained ANN network into a spiking equivalent [Rueckauer et al., 2016]. However, the process for conversion does put limitations on the layer and activation types that can be used in the architecture design. As the activations of the original network are approximated using spiking equivalents, the overall accuracy deteriorates for every layer in the network. Despite these drawbacks, SNNs have still seen some tentative use in dToF based object detection.

1.2 Surface detection

Surface detection is the process of extracting the depth of objects encoded in the timing information output by the dToF SPAD sensors, typically in histogram format. Peak Detection (PD), as in Figure 1.7, locates the detected surfaces based on the reflected photon rates, which is the superposition of the background photons and incident photons. Various algorithms have been developed to perform PD such as Gaussian curve fitting and Continuous wavelet transform [Nguyen et al., 2013]. Digital filters have also been previously applied to perform PD, such as a Center-of-Mass filter [Tsai et al., 2018], but demand a higher computational cost to process the entire histogram using small sliding steps to maintain high precision. Neural network approaches have seen increased use in this field over the last few years, with some approaches outlined below.

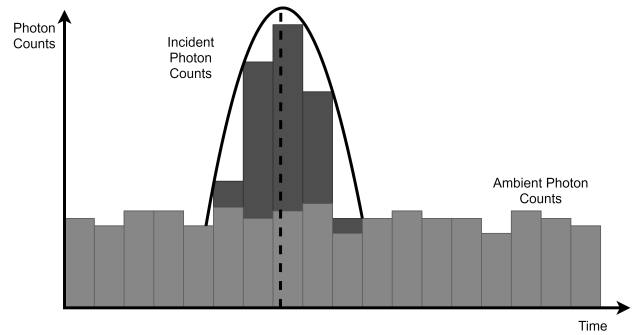


Figure 1.7: Peak detection using a histogram with a Gaussian pulse shape. Light blue represents the ambient photon bed, and dark blue is the incident back-scattered photons detected by a SPAD sensor. The solid black line represents the beam shape, in this case, a Gaussian, and the dashed black line is the estimated peak.

1.2.1 Artificial Neural Networks

The work of [Chen et al., 2022] explores a method to perform feature extraction on SPAD histogram data using a neural network-based multi-peak analysis (NNMPA) to improve the robustness of the distance measurement under harsh environmental conditions with the goal to predict the target distance's coarse position in a noisy histogram. The datasets used to train the system can be seen in Table 1.1, the first being a synthetic dataset based on the Montecarlo principle and the second being real data captured using the "OWL" flash LiDAR sensor by Fraunhofer IMS [owl,]. The first stage of the system is to extract features from the histogram, this is done by first convolving over the raw histogram and then splitting it into multiple regions with a feature extracted from each, 12 being shown as the optimum number. The local maximum M_n and the corresponding bin number b_n from each region are extracted and the background is estimated and subtracted from this value. The features are then normalized, f_n, and combined into a feature group $F_n = \{f_n, b_n\}$. These features are then processed by the NN which uses the softmax function to assign soft decisions

to each feature where the highest scoring feature is chosen as the final prediction. As the resolution of the classification is lower than the histogram resolution, a distance recovery process is used to reduce loss from the low precision in the final distance estimation. The paper provides a detailed comparison between the proposed NNMPA network and the chosen classical method, showing that the former outperforms the latter on both datasets. However, under certain conditions, i.e., short distance and low ambient, in the synthetic dataset the classical method outperforms the NNMPA network. The classical comparison method is an average filter with background estimation and subtraction with a final global maximum detection applied. For instance, in the synthetic dataset at a distance of 50–60 m and ambient rates of 4Mcps, the NNMPA has an accuracy of 57.17% compared with 18.33% for the classical. However, at a distance of 0–10m at the same ambient, the NNMPA method has an accuracy of 83.17% compared with 89.00% for the classical. The overall performance of the NNMPA exceeds that of the classical method in most situations where either the ambient is >3Mcps and the distance is <30m. No detailed analysis was provided as to why the CDP outperforms NNMPA at shorter distances and lower ambient levels, so it remains an open question. When tested on the real dataset the NNMPA reaches accuracies approaching 100%, exceeding the CDP across all measured distances. The NNMPA method also outperforms the classical in computation time, 0.26ms vs 0.32ms when simulated using LabView [ni,] on a PC (not specified whether CPU or GPU utilized), with the largest contribution of 0.23s from the initial convolution over the histogram.

An FPGA implementation of this system is also outlined by [Chen et al., 2021] which is compared to the performance of the PC implementation. To increase the efficiency of the networks FPGA implementation, fixed point arithmetic replaces floating point to reduce the computational demands. Approximated values in a Look-Up-Table (LUT) replaces the background light estimation for increased computational efficiency, however, this caused a large quantization error in low ambient levels with the maximum error for ambient estimation being 39.28% at 1–1.5 Mcps ambient photon rate. The conversion to an FPGA implementation caused a maximum accuracy loss of 2.76% with a negligible accuracy reduction at low ambient levels despite the high quantization error, as can be seen in Figure 1.8. The FPGA used, an Enclustra Mas ZX3, cannot stabily use the full 384 pixels of the SPAD sensor due to timing violations, instead only 96 pixels are used resulting in the resource occupation in Table 1.2. The end result for the FPGA implementation was a marked reduction in computation time by 0.17 ms per histogram in each pixel while having an overall accuracy drop of 1.21% (maximum of 2.76%).

The work outlined in [Aßmann et al., 2021] first presents an approach to perform super-resolution on LiDAR waveforms using a CNN. It is trained using 25,000 synthetic histograms where one reflector is separated from another with random separation distance, amplitude, and noise within experiment parameters. This dataset is designed to replicate the experiment first presented in [Hernández-Marín et al., 2008] which proposes the statistical Reverse Jump Markov Chain Monte Carlo (RJMCMC) method. The proposed CNN details can be seen in Table 1.3. The goal of the network is to process the synthetic histograms to extract the principal peak location compo-

Table 1.1: Details of datasets used in [Chen et al., 2022].

	Synthetic dataset	Real dataset
Dataset size	96,000	7,000
Histogram size	1310 bins	1310 bins
TDC resolution	312.5ps	312.5ps
Max ranges	60m	25m
Incident rate	10 Mcps	
Background Rate	1-8 Mcps	7-9 Mcps
Sensor array size		2x192
Pulse width	5ns	18.75ns
Peak power		75W

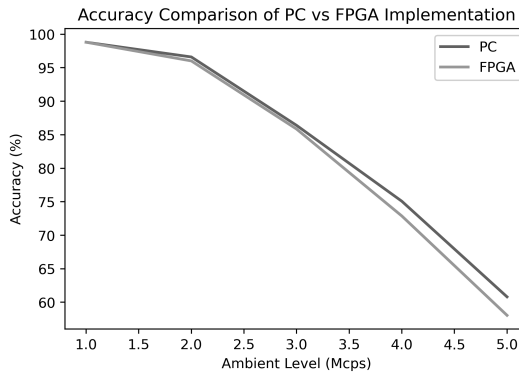


Figure 1.8: Accuracy comparison of PC and FPGA implementation on 2360 bin histograms where the PC configuration is a simulation run using the software LabView [Chen et al., 2021].

Table 1.2: NNMPA FPGA resource occupation [Chen et al., 2021].

Resources	**384 Pixels**	**96 Pixels**
LUT as Logic	6594 (12.39%)	6373 (11.98%)
LUT as memory	496 (2.85%)	496 (2.85%)
Slice registers as flip-flop	9549 (8.97%)	9416 (8.85%)
Slice registers as latch	0 (0%)	0 (0%)
BRAM	101.5 (72.50%)	29.50 (21.07%)
DSP blocks	15 (6.82%)	15 (6.82%)

nents present and attach a confidence value to each peak detected. The histograms consist of 4096 bins representing a max distance of 65m. The trained network is tested on the same real data used in [Hernández-Marín et al., 2008] and compared to its proposed method RJMCMC, with the results presented in Table 1.4. While LiDARNet has an overall worse performance than the classical method, LiDARNet is better able to detect multiple surfaces in the input waveform while RJMCMC either mislocates or fails to identify the secondary surfaces.

[Aßmann et al., 2021] also presented an automotive LiDAR model which uses a similar CNN as the super-resolution, with the same application to extract the depths

Table 1.3: Super resolution and automotive LiDAR networks parameters EB stands for Encode Block which consists of a 1D convolutional layer, a dropout layer, and a max-pooling layer [Aßmann et al., 2021].

	Super Resolution		Automotive		Activation
	L, W	#Params	L, W	#Params	
EB1	64, 64	4,160	96, 48	4704	ReLU
EB2	64, 32	131,136	96, 48	442,464	ReLU
EB3	-	-	64, 24	147,520	ReLU
Conv1D	32, 32	65,568	32, 24	49,184	ReLU
Conv1D	16, 32	16,400	16, 24	12,304	ReLU
	C	#Params	C	#Params	
Dense	128	2,097,280	256	3,842,304	ReLU
Dense	4096	528,384	7500	1,927,500	SoftMax
Total		2,842,928		6,425,980	

Table 1.4: Evaluation of a real Super-Resolution Benchmark using the synthetically trained LiDARNet [Aßmann et al., 2021] [Hernández-Marín et al., 2008].

	RJMCMC		LiDARNet	
Ground Truth (cm)	Mean (cm)	Error (cm)	Mean (cm)	Error (cm)
1.7	1.462	0.238	1.281	0.419
3.2	3.281	0.081	3.843	0.643
5.2	5.086	0.114	5.489	0.289
7.2	7.053	0.147	7.136	0.064
9.2	9.108	0.092	9.332	0.132
11.2	11.092	0.108	11.345	0.145
13.2	13.155	0.045	13.357	0.157

of all surfaces present in the histogram data. This model is also trained using synthetic data, which was proven to be viable in the super-resolution model. The automotive synthetic datasets utilize existing labeled scenes in [Ros et al., 2016] and [Gaidon et al., 2016] to provide detailed ground truth data that acts as a base scene for a SPAD sensors FoV to be projected. Beam expansion is simulated by employing a spatial down-sampling routine to emulate each SPAD pixel detecting multiple returns, while this results in the loss of objects spatial location their depth is retained. The paper sets the maximum number of separate surfaces present in each pixel as 9 with the surface reflectivity determined using generic values for each label type scaled by their brightness in the RGB image. 14,000 Synthetic waveforms of distances up to 300m are represented with 7500 bin histograms. The parameters of the trained network can also be seen in Table 1.3. The final network is then compared to RJMCMC on 1000 test waveforms, with the results being presented in Table 1.5. The main advantage that can be seen is the total time required to process a single waveform, with LiDARNet having total run times order lower than that of RJMCMC while also providing a slightly better Peak Signal-to-Noise Ratio (PSNR).

The work presented by [Sun et al., 2020] presents a network that fuses SPAD histogram data and monocular (RGB) data for robust depth estimation to overcome noisy or corrupted data. The final depth is estimated by using a CNN to combine

Table 1.5: Evaluation of LiDARNet in automotive configuration for 1,000 simulated waveforms compared to ideal waveforms [Aßmann et al., 2021] [Hernández-Marín et al., 2008].

	Signal	JMCMC	LiDARNet
PSNR (dB)	8.25	40.43	43.50
MSE	0.2935	0.0008	0.0006
time (ms)		>5000	3.4

the noisy output of the SPAD sensor with the depth information extracted from an RGB image using a pre-trained monocular depth estimation neural network. The network is trained on synthetic data generated using the NYUv2 [Silberman et al., 2012] with depths of under 10m where the SPAD histograms consist of 1,024 bins with a temporal resolution of 80ps, and a total input resolution of 512x512x1,024 (processed in smaller patches and recombined at the output due to memory constraints). Pre-processing is used to convert the histogram bins from a linear to a logarithmic scale, reducing the bin count from 1,024 to 128 in order to reduce the runtime and memory consumption by a factor of 7. In total, there are 7,600 training scenes and 766 testing scenes. Real-depth data is captured using the LinoSPAD [Burri et al., 2017] sensor with an array size of 256x256 SPADs and 1,536 bin histograms with a temporal resolution of 26ps. Monocular data is captured at 5Hz using a PointGrey camera. The network structure can be seen in Figure 1.9. The monocular depth estimator used in SPADnet is pre-trained (DORN [Fu et al., 2018] and DenseDepth [Alhashim and Wonka, 2018]) with the resultant depth estimation being converted into the z axis index for each x,y spatial coordinate where the corresponding indice is set to 1 and all other locations set to 0. The process is referred to as "2D to 3D up-projection". The results of the trained SPADnet network compared to other solutions can be seen in Table 1.6. It can be seen that the SPADnet model using the log-scaled bins produces the best results, with under half the RMSE of the linear implementation. It also produces the highest accuracy within 1.25% of the ground truth, with 99.6% of results falling within the accuracy metric. When tested on the captured data with cases where areas of the scene has low reflectivity, optical misalignment, and multipath interference, the proposed log SPADnet has a greater ability to reconstruct the depth maps than the other approaches. In examples with extremely weak returns, SPADnet produces an RMSE of 7.15cm versus the 24.01cm and 11.68cm of [Rapp and Goyal, 2017] and [Lindell et al., 2018] (log) respectively. The main issue with the log scale re-binned histogram architecture is that the resolution decreases with distance, so further depths will be combined into the larger time bins increasing the quantization error. The monocular depth estimator also has a tendency to fail when used on previously unseen datasets or when objects are too close, causing SPADnet to not benefit from the monocular depth in these cases. The benefits of utilising ToF data with the monocular depth data is demonstrated when the DORN network [Fu et al., 2018] is swapped for the DenseDepth [Alhashim and Wonka, 2018] monocular network. When trained on the NYUv2 dataset DORN and DenseDepth have a RMSE (cm) error of 53.7 and 71.2 respectively, however, when combined into SPADnet this RMSE reduces to 14.4 and

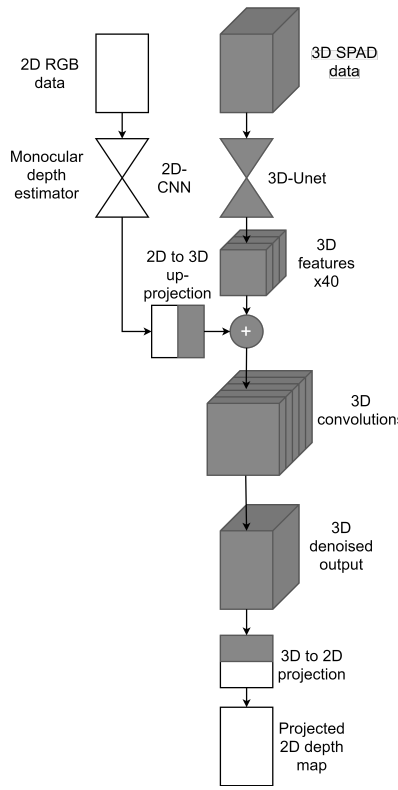


Figure 1.9: Network diagram of proposed network SPADnet [Sun et al., 2020].

13.3 respectively. The addition of SPAD ToF data is able to compensate for monocular depths shortcomings in prediction when objects are too close or there is low texture data available [Ali and Pandey, 2022]. While the monocular data provides accurate of relative depth which can be used to de-noise SPAD LIDAR's high accuracy but low resolution depth data. These results demonstrate that the combination of monocular RGB data and SPAD histograms provide a more robust solution than either in isolation.

The work presented by [Zang et al., 2021] proposes a 3D CNN to process ToF data utilizing multi-dimensional spatial and temporal features into depth maps under low photon flux and SNR conditions. The architecture is also compressed using low-bit parametric quantization to allow for implementation on FPGA while maintaining high reconstruction quality. The network is trained on synthetic data generated using the NYUv2 [Silberman et al., 2012] and Middlebury [Scharstein and Pal, 2007] datasets. The synthetic dataset consists of 13,000 and 1,300 ToF tensors for training and validation respectively, with input data being 64x64x1,024 bins. Real data for testing is collected using the LinoSPAD [Burri et al., 2017] which provides SPAD data at a resolution of 256x256x1,536 with a bin width of 26ps. The network uses a 3D version of U-net++ [Zhou et al., 2019] as the core, with the main network being

Table 1.6: Results of proposed and existing methods on synthetic data. Note that the different image sizes depended on whether the bins were in linear or log scale due to memory constraints during training. $*^1$ [Fu et al., 2018] (Pre-trained Monocular depth estimator), $*^2$ [Rapp and Goyal, 2017], $*^3$[Lindell et al., 2018]. [Sun et al., 2020].

	Signal photons = 2, background photons = 50, SBR = 0.04				
	Patch size	$\delta < 1.25$	$\delta < 1.25^2$	RMSE (cm)	Abs rel
$*^1$		0.881	0.976	53.7	0.117
$*^2$		0.965	0.986	43.9	0.032
$*^3$ (linear)	64x64	0.935	0.952	72.1	0.058
$*^3$ (log)	128x128	0.993	0.998	18.2	0.011
SPADnet (linear)	64x64	0.970	0.993	34.7	0.031
SPADnet (log)	128x128	**0.996**	**0.999**	**14.4**	**0.010**

split into a feature extraction module and a refinement module. The network learns to filter the noise during the training to output a 2D depth map without using any guiding images such as monocular or intensity data. As the main obstacle to embedding the network on FPGA is the large quantity of memory used for data transfer and the low on-chip memory available for parameters, a 2D low-bit quantization method outlined in [Zhou et al., 2016] was utilized to quantize the 3D data parameters to compress the model, with the floating point parameters converted into a fixed point format. In order to reduce the loss from quantization, the weight parameters of the first and last layers were not changed. The bit widths for the weights of other hidden layers are reduced to 2 bits while the bit widths of outputs from activation functions are reduced to 4 bits. The result is an impressive reduction in network size compared to the original floating point model, allowing for implementation on FPGA. The compressed version of the network is referred to as W2A4, and relative compression to other existing networks can be seen in Table 1.7. The proposed and existing networks were tested on 7 indoor scenes of the Middlebury dataset using three different SNRs, with the results shown in Table 1.8. These results demonstrate that despite the small size of the network, it produces high accuracy with a low error rate. The Absolute relative difference (ABS rel) shows a difference of 2.94×10^{-5} between the proposed method and [Sun et al., 2020] and 16.2×10^{-3} with [Peng et al., 2020].

Table 1.7: Comparison of different existing and proposed networks in terms of No. of parameters, training time, and the relative compression compared to W2A4. $*^1$ without utilizing intensity data. $*^2$ utilizing intensity data [Zang et al., 2021].

	No. parameters	Training time	Compression rate
[Lindell et al., 2018] $*^1$	3.95M	24h	21.99x
[Lindell et al., 2018] $*^2$	3.93M	24h	21.83x
[Sun et al., 2020]	3.95M	24h	21.99x
[Peng et al., 2020]	1.01M	36h	5.61x
32-bit floating point	2.19M	17h	12.17x
W2A4	0.18M	16h	-

Table 1.8: Results of proposed and existing algorithms over three different SNR levels [Zang et al., 2021].

	Accuracy		Error	
	$\delta < 1.25$	$\delta < 1.25^2$	RMSE (m)	ABS rel
Signal photons = 2, background photons = 10, SBR = 0.2				
[Lindell et al., 2018]	0.9962	0.9982	0.066	0.0110
[Sun et al., 2020]	0.9966	0.9981	0.062	0.0070
[Peng et al., 2020]	0.9966	**0.9987**	0.064	0.0087
32-bit floating point	**0.9968**	0.9983	**0.059**	0.0069
W2A4	0.9967	0.9980	0.061	**0.0056**
Signal photons = 2, background photons = 50, SBR = 0.04				
[Lindell et al., 2018]	0.9827	0.9951	0.149	0.0260
[Sun et al., 2020]	0.9948	0.9971	0.073	0.0082
[Peng et al., 2020]	0.9961	**0.9980**	0.064	0.0087
32-bit floating point	0.9961	**0.9980**	**0.063**	0.0067
W2A4	**0.9962**	**0.9980**	0.064	**0.0060**
Signal photons = 2, background photons = 100, SBR = 0.02				
[Lindell et al., 2018]	0.9357	0.9729	0.321	0.0580
[Sun et al., 2020]	0.9952	0.9978	0.069	0.0081
[Peng et al., 2020]	0.9961	**0.9981**	0.065	0.0087
32-bit floating point	**0.9963**	0.9980	**0.064**	**0.0060**
W2A4	**0.9963**	**0.9981**	0.065	**0.0060**

1.2.2 Comparison

[Chen et al., 2022] demonstrated a system that can operate well in adverse conditions, though its accuracy deteriorates in ideal conditions when compared to classical methods. It has the lowest computation time at only 0.26ms per histogram vs LiDARNET's 3.4ms. Its FPGA implementation only has a minor drop in depth prediction despite the high quantization error in the ambient estimation. Sadly it doesn't have any detailed information on the exact number of parameters used in the NNMPA fully connected structure, but as the FPGA BRAM utilization is high so to will be the network size. [Aßmann et al., 2021] demonstrates a very low error when detecting multiple close surfaces at distances of up to 300m using a convolutional structure without any feature extraction being performed on the histograms. While the network has relatively few parameters due to its convolutional architecture, making it easier to implement on a chip, it is still a larger network than [Zang et al., 2021] which uses 180,000 parameters. [Zang et al., 2021] presented a small network that is able to outperform multiple state-of-the-art schemes under various conditions, as seen in Table 1.8. One of the key results is the minimal effect quantization had on the performance of the network compared to its 32-bit floating-point equivalent. While [Sun et al., 2020] presented good results utilizing both depth and RGB, it was outperformed by [Zang et al., 2021] in accuracy prediction on similar datasets. A key takeaway from this is that while additional guiding information, such as monocular, is useful in detecting the surfaces present in a scene, it is not a necessary step as proven by the presented systems. Quantisation has also been shown to reduce the size of a network dramatically, Table 1.7, to allow for easier implementation on hardware with minimal increase in error. The main advantage that [Chen et al., 2022] and

Table 1.9: Comparison table of surface detection schemes. $*^{1}$ also uses RGB data 1: [Chen et al., 2022], 2: [Aßmann et al., 2021], 3: [Sun et al., 2020], and 4: [Zang et al., 2021].

	1	2	3	4
Implementation level	FPGA	Simulation	Simulation	Simulation
Accuracy metric	Accuracy	MSE	$\delta < 1.25$	$\delta < 1.25$
Reported result	83.54%	0.0006	0.9960	0.9962
No. parameters		6,425,980		180,000
Input data type	Histograms	Histograms	Histograms$*^{1}$	Histograms
Data resolution	1x1,310	1x7,500	1x128	1x1,024
SPAD array size	2x192		256x256	64x64
Max range	61.26m	300m	10m	
Peak laser power	75W			
Pulse width	18.75ns			
Repetition rate	10KHz			
Temporal resolution	312.5ps			
Frame rate	20			
Computation time	0.26ms	3.4ms		
Power Consumption				

[Aßmann et al., 2021] have over the other schemes presented is the ability to output multiple detected surfaces from the same pixel. The more detailed output of these networks make them ideal for use in combination with spatial super-resolution schemes and algorithms such as that presented in [Gyongy et al., 2020].

1.3 Super-resolution depth mapping

Super-resolution imaging is a series of techniques used to increase the resolution of an imaging system, such as in 1.10. Traditional approaches such as linear and bicubic interpolation [Keys, 1981] can be fast but yield results that have been overly smoothed. Recently deep learning approaches, such as SR-GAN [Ledig et al., 2016], are achieving state-of-the-art performances driving them into becoming more common in super-resolution tasks [Yang et al., 2019]. These techniques have also seen increased use in the upsampling of low spatial resolution depth maps captured using SPAD LiDAR for use in applications such as automotive vehicles.

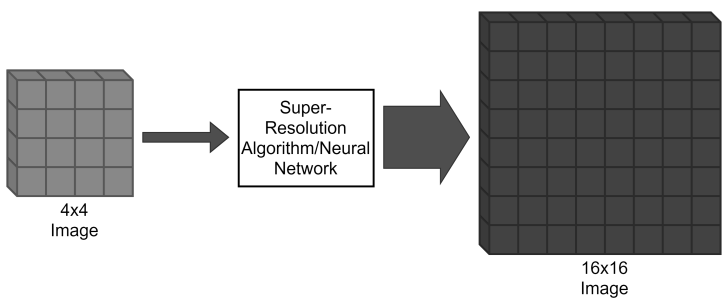


Figure 1.10: Diagram of basic super-resolution output.

1.3.1 Artificial Neural Networks

The work presented by [Ruget et al., 2021a] proposes a deep neural network that de-noises and up-samples a depth map from 64x32 to 256x128 using multi-scale photon count histogram information and exploiting high-resolution intensity images for guided up-sampling. The network is modeled on use with the Quantic 4x4 [Hutchings et al., 2019] sensor which provides both the intensity and depth information at resolutions of 256x128 and 64x32 respectively. The training and testing of the neural network uses synthetic histogram and intensity data generated using the MPI Sintel Depth dataset [Butler et al., 2012], [Wulff et al., 2012] (training) and Middlebury dataset [Scharstein and Pal, 2007], [Hirschmuller and Scharstein, 2007] (test) as the ground truth. The designed network utilizes U-net architecture [Ronneberger et al., 2015] and incorporates guidance information as in [Guo et al., 2019]. The network uses as an input the concatenation of two depth maps (up-sampled from 64x32 to 256x128 using the nearest neighbor algorithm), with the first depth map being comprised of the surfaces with the highest photon counts. The second depth map consists of any pixels which contain secondary surfaces where the photon counts exceed a certain threshold, if not then that pixel entry remains 0. Multi-scale information is also utilized to de-noise the data and is included in both the decoder and encoder using guiding branches, with depth features being incorporated in the encoder and intensity in the decoder. The four multiscale depth features connected to the encoder consist of D1, D2, D3, and D4. D1 is the first depth map down sampled using nearest-neighbor interpolation from 256x128 to 128x64. D2 is computes the depths using center of mass on the source 64x32x16 histograms. D3 and D4 are the source histograms down-sampled by a factor of 2 and 4 respectively by summing adjacent pixels. The intensity image used to guide the upsampling process has a resolution of 256x128. The size of each component of the network can be seen in Table 1.10. The results of the trained network are compared to four other methods, first the nearest-neighbor interpolation, Guided Image Filtering [He et al., 2013], DepthSR-Net (retrained on the same dataset) [Guo et al., 2019], and an algorithm presented in [Gyongy et al., 2020]. Each system was tested under three different conditions, High SNR of 2 and signal photon counts per pixel (ppp) of 1200 with secondary surfaces present, medium SNR of 0.02 and ppp of 4 with no secondary surfaces, and low SNR of 0.006 and ppp of 4 with no secondary surfaces. The results can be seen in Table 1.11 where Absolute Depth Error (ADE) is calculated as $ADE = |R + d - d^{ref}|$, with R being the residual map predicted by Histnet, d being an up-scaled low-resolution depth map, d^{ref} being ground truth. Additional results can also be found in [Ruget et al., 2021b]. The results show that guided upsampling using intensity images and multi-scale depth features significantly improves the accuracy of the final HR image. The main drawback, however, is that even when implemented on a NVidia RTX 6000 GPU the processing time for one frame is significant. With the total processing time reaching 7 seconds, Histnet would be unlikely to be usable in any live scenarios.

Another super-resolution and denoising scheme by [Martín et al., 2022] devises a method to upscale and denoise dToF video sequences using past, present, and future

Table 1.10: No. of parameters per network component of Histnet [Ruget et al., 2021a].

Network section	No. of parameters
Encoder	25,108,992
Decoder	31,058,368
Depth guidance	9,600
Intensity guidance	1,549,824
Total	57,726,784

Table 1.11: Quantitive comparison of the different methods of reconstruction for the 4x upsampling of the MPI Sintel dataset [Ruget et al., 2021a]. ADE stands for Absolute Depth Error. $*^1$ is Nearest neighbour interpolation, $*^2$ is [He et al., 2013], $*^3$ is [Guo et al., 2019], $*^4$ is [Gyongy et al., 2020], and $*^5$ is Histnet.

	$*^1$	$*^2$	$*^3$	$*^4$	$*^5$
Time per frame	1ms	0.4s	7s (on GPU)	4s	7s (on GPU)
Training on high SNR with secondary surface					
Scene	ADE	ADE	ADE	ADE	ADE
Art	0.038	0.039	0.008	0.0076	0.0027
Reindeer	0.035	0.035	0.0051	0.004	0.0018
Training on medium SNR without secondary surface					
Art	0.22	0.17	0.023	0.05	0.019
Reindeer	0.21	0.16	0.024	0.06	0.019
Training on low SNR without secondary surface					
Art	0.276	0.22	0.064	0.187	0.055
Reindeer	0.272	0.206	0.053	0.168	0.05

depth frames based on the network presented in [Li et al., 2022]. It is trained using 15,500 images using synthetic depth data generated from high-resolution depth maps (256x128) and RGB data (256x128 converted to grayscale) recorded in Airsim [Shah et al., 2017]. As these depth maps don't contain the Poisson noise inherent in photonics, histograms are created from the Airsim data using a Poisson distribution to generate incident and background photon counts which form a 16-bin histogram. The FoV of the scenes is 30°with different frame rates simulated to vary the object movement per frame, the SNR is also varied between video sequences, and finally, the depth is varied between 0 and 35m. From these generated histograms, new depth maps are created using center of mass peak extraction [Gyongy et al., 2020]. The depth maps are normalized and consecutive frames with a set temporal radius T_R are concatenated, i.e., in groups of $2T_R + 1$ frames such that the input to the network is of the shape 64x32x($2T_R$+1). When the temporal radius is 0, only a single depth frame is used as input as opposed to if the radius is set to one, where it uses one frame from the past and one from the future. If the frame is at the start or end of the video sequence, then the system replicates the current frame to fulfill the required input size. The network output is a super-solved depth frame of the size 256x128 and is compared with the normalised ground truth depth map from Airsim and evaluated using the metrics Peak Signal-to-Noise Ratio (PSNR) [Fardo et al., 2016] and Structure Similarity Index (SSIM) [Nilsson and Akenine-Möller, 2020]. The network architecture

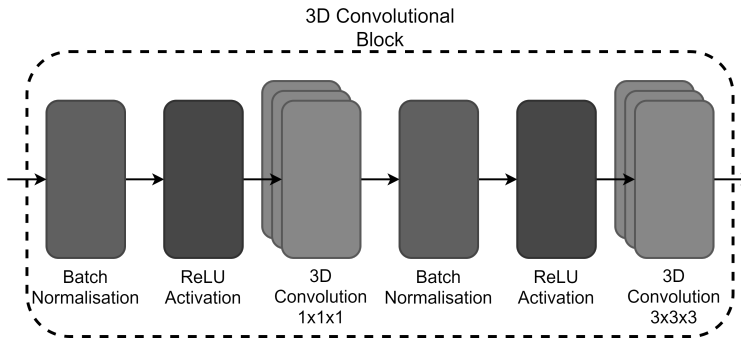


Figure 1.11: Diagram of 3D convolutional blocks used in super-resolution scheme [Martín et al., 2022].

itself is based on blocks of 3D convolutions and dynamic upsampling filters. The structure of the 3D blocks can be seen in Figure 1.11, with the full network shown in Figure 1.12. The number of convolutional blocks varies with the size of the temporal resolution such that there are $3+T_R$ blocks per network. Initial testing to investigate the impact of different temporal resolutions was completed using a dataset of 1500 frames (3 recordings of 500 frames). The results show that while a higher T_R improves PSNR and SSIM, the trade-off is a large decrease in FPS from 43.4 FPS at $T_R = 1$ to 30.8 FPS at $T_R = 4$. An investigation was also made into an object's speed and the network's ability to exploit the temporal information, and was found that as long as an object didn't move more than 2-3 pixels in between frames then there would be no degradation in accuracy. The proposed system is also compared to other contemporary methods such as Bicubic [Keys, 1981], Histnet [Ruget et al., 2021a], and iSeeBetter [Chadha et al., 2020] using the same 1,500 frame dataset. It can be seen in Table 1.12 that the proposed approach has a greater accuracy at upsampling the depth maps with only a slightly lower frame rate than other approaches presented. The images produced from this scheme are better able to replicate the flat surfaces of an object, but have a tendency to blur the edges of an object with the background.

Table 1.12: Comparison of work presented in [Martín et al., 2022] at different temporal resolutions with other contemporary methods. Scenes 1 and 2 have an SNR of 1.3 and scene 3 has a SNR of 3, with different SNR values used to demonstrate the robustness of the network.

	Scene 1		Scene 2		Scene 3		
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	FPS
Bicubic	15.82	0.538	17.60	0.607	26.80	0.841	**185**
iSeeBetter	20.47	0.784	21.96	0.837	28.74	0.843	33
Histnet	19.14	0.812	20.14	0.858	27.18	0.881	0.25
$T_R = 1$	21.20	0.890	22.72	0.910	31.17	0.901	43.4
$T_R = 4$	**22.05**	**0.909**	**23.00**	**0.916**	**31.31**	**0.906**	30.8

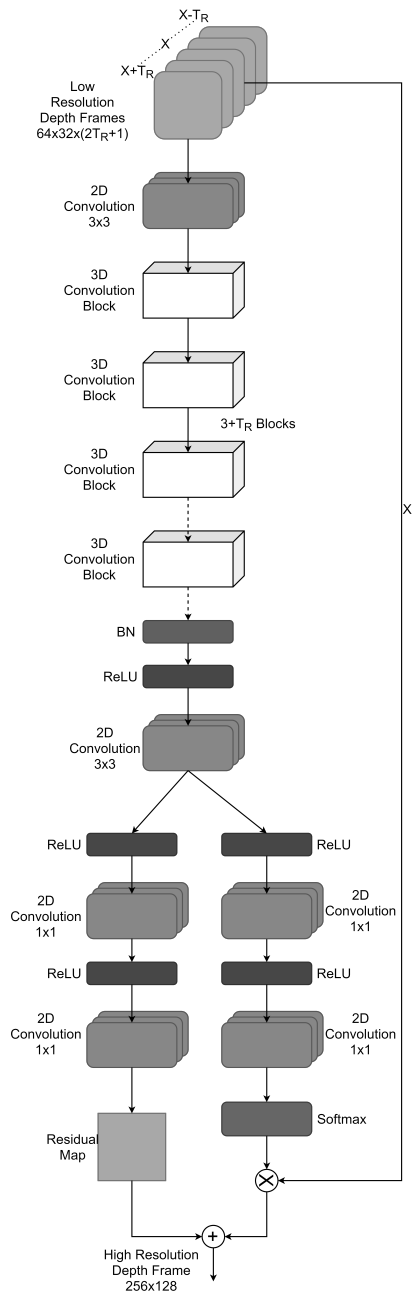


Figure 1.12: Diagram of the full network used in the super-resolution scheme [Martín et al., 2022].

1.3.2 Spiking Neural Networks

The work presented by [Kirkland et al., 2020] presents a single pixel SPAD sensor that utilizes a spiking neural network to upscale the output 1x8,000 bin ToF histogram into a 64x64 resolution depth map. A fully spiking convolutional architecture is used where the depth histogram is encoded using 1D convolutions, then decoded into a 2D depth map using 2D convolutional layers. The network is trained on synthetic data created by simulating a flood illuminated single point SPAD sensor with depths ranging from 2 to 4m. 4 different experiments comprised the 11,600 sample dataset, with the variance of photon count and IRF from 1,000 to 9,500 photon counts and 20 to 100 ps time windows and histogram bin widths of 2.3ps. The synthetic scenes are generated using object silhouettes, with the background objects being static throughout all images and the foreground object moving between 200 locations which only vary along the x and z axis. In total, 29 different object silhouettes are used.

The structure of the devised Spiking Convolutional Neural Network (SCNN) and the network it is compared to can be seen in Table 1.13. The SCNN is trained as a synchronous CNN but is converted into its spiking equivalent using Nengo [Bekolay et al., 2014]. The results of the conversion versus the traditional ANN approach can be seen in Table 1.14. It can be seen that Spike-SPI performs better on all reported metrics except SNR and RMSE. Spike-SPI spatial reconstruction of the scene is superior to the ANN as seen in the Intersection over Union (IoU) and accuracy metrics. While 87% of all depth estimates made by Spike-SPI are accurate, the predictions for the background are considerably noisy which is the root of the low SNR and RMSE. Spike-SPI also benefits from lower processing power over the ANN implementation. While the average firing rate of the neurons is around 1Hz, with a max of 150Hz, only around 11% of the neurons are active as low connection weights inherited from the CNN version don't significantly increase the spiking neurons' membrane potentials to their threshold essentially filtering them out. This reduction in information propagation can reduce the accuracy of the reconstruction, but also reduces the power consumption. The paper reports however that there was no tangible difference between the results of the CNN and SCNN implementations. There is also a marked reduction in processing time for each frame due to the asynchronous nature of the network. By varying the firing rates of the neurons, the time taken to converge on a result reduces with the trade-off of more neural activity. The main flaw of this reconstruction scheme is its reliance on a the background remaining static, with only the foreground changing.

1.3.3 Comparison

The largest difference between the presented schemes is the input data that they use, shown in Table 1.15. Histnet utilizes the most data inputs but does not achieve greater SSIM over the method of [Martín et al., 2022], as seen in Table 1.12, which uses multiple depth frames. Histnet also presents a far slower computation time than the scheme presented in [Martín et al., 2022] of 7s vs 32ms respectively. This shows that there does not appear to be an added advantage in utilizing Depth, intensity,

Table 1.13: Details of Spike-SPI and its comparison network [Kirkland et al., 2020].

	Spike-SPI							
Layer	Ce1	Ce2-3	Ce4-10	Cd8-5	Cd4-3	Cd2	Cd1	Up
Kernel size	7	7	7	5	5	5	5	2
Feature no.	64	128	256	256	128	64	1	
	ANN							
Layer	FC1		FC2		FC3			
Kernel size								
Feature no.	1024		512		256			

Table 1.14: Results of the Spike-SPI vs the ANN solution [Kirkland et al., 2020] where δ relates to the accuracy of the predictions within 1.25% of the ground truth. IoU results performed on a mask of the Spike-SPI depth map outputs where the data is converted into a binary value to represent the presence of an object in the scene.

	Photon count	IRF	IoU	SNR	RMSE	$\delta < 1.25$
ANN	1,000	100ps	0.650	**14.844**dB	**0.189**	0.853
Spike-SPI			**0.783**	14.284dB	0.201	**0.871**
ANN		20ps	0.650	**14.708**dB	**0.192**	0.853
Spike-SPI			**0.760**	14.155dB	0.202	**0.868**
ANN	9,500	100ps	0.637	**15.076**dB	**0.187**	0.856
Spike-SPI			**0.780**	14.391dB	0.198	**0.871**
ANN		20ps	0.631	**15.070**dB	**0.188**	0.856
Spike-SPI			**0.778**	14.424dB	0.198	**0.870**

and histogram data over just depth. Histograms however are shown in the Spike-SPI to provide a wealth of information, providing the most impressive performance in super-resolution. [Kirkland et al., 2020] up-scales the input histogram into a 64x64 depth map. The catch is that it relies on a known static background, so has limited applications in the real world. Its spiking nature compounded with the low neural activity does imply it would have the lowest power consumption out of the presented methods. A comparison in computation time was not investigated in Spike-SPI, but it would be interesting to see the time taken to process the histogram using the ANN vs the SNN. An interesting network would seem to be a structure similar to [Martín et al., 2022] using the raw histograms instead of the depth maps, which could then be converted into a spiking equivalent for the reduction in power consumption and potential computation time.

1.4 Object detection

Object detection is the image processing task that detects and extracts semantic objects of a predefined class (such as boat, airplane, human, etc.) in digital images or videos, such as in Figure 1.13. Every object class defined has its own set of features that is used to identify it, for instance, a square class would be identified b perpendicular corners. However, while in algorithms such as Scale-invariant feature transform (SIFT) [Lowe, 2004] where each class is predefined using reference images, in neural network approaches based on convolutional layers the features of a class are learned

Table 1.15: Comparison table of reviewed super resolution depth mapping solutions $*^{!}$also utilises intensity (256x128) and histogram (64x32x16) data. $*^{2}$ approximated from paper. 1: [Ruget et al., 2021a], 2: [Martín et al., 2022], and 3: [Kirkland et al., 2020].

	1	2	3
Implementation level	Simulation	Simulation	Simulation
Accuracy metric	ADE	SSIM	RMSE
Reported result	0.00225	0.91	≈ 0.20
No. parameters	57,726,784		n/a
Input data type	Depth maps$*^{1}$	Depth maps	Histograms
Input resolution	64x32x2	64x32	1x8000
Output data type	Depth maps	Depth maps	Depth maps
Output resolution	256x128	256x128	64x64
SPAD array size	256x128/64x32	64x32	1x1
Computation time	7s	32ms$*^{2}$	54ms$*^{2}$

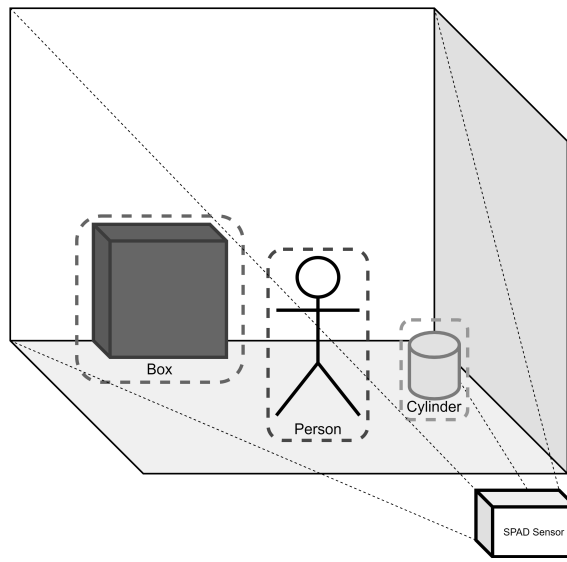


Figure 1.13: Diagram of basic object detection.

and not defined. Object detection applications include face recognition, object tracking, and image annotation. Over time CNN implementations of object detection have seen increased use and has now become state of the art.

1.4.1 Artificial Neural Networks

The first system [Ruvalcaba-Cardenas et al., 2018] is a flash LiDAR which compares a modified version of the VGG-16 [Simonyan and Zisserman, 2014] 2D CNN and compares it to a 3D CNN at identifying three classes (airplane, chair, and drone). These two systems are trained using a limited dataset of only 1,983 and 1,900 images

Table 1.16: Details of datasets used in [Ruvalcaba-Cardenas et al., 2018].

	Indoor	Outdoor
Range	25m	400m, 600m, and 700m
Ambient	<1 lux	
SPAD array	64x64	
Pulse width	7ns	
Pulse energy	18mj	
Repetition rate	100Hz	
Depth resolution	0.5m	

captured on a SPAD LiDAR setup for the 2D and 3D network respectively, with an example shown in Table 1.16. For the indoor images, a 16mm to 160mm zoom lens was used with different zooms for each object. The outdoor data also used a meade telescope with the SPAD sensor while the laser was mounted using a telescopic sight and a beam expander which was set to give a 3m diameter beam at each of the 3 ranges. The sensor outputs histograms which are processed into depth maps, with distance thresholding, and then median and spatial filters are applied to remove any background data or noise until finally, the image is binarised. A nearest-neighbor interpolation algorithm is used to upsample the binarised images to 320x320, with the data for the 2D network being resized again down to 224x224 while the 3D network converts the images into a 16x16x16 voxel grid where the binary image is projected along the z-axis. The 2D modified VGG-16 network is retrained using transfer learning [Yosinski et al., 2014], where the first 14 layers of VGG-16 are frozen while the last 4 are replaced with 4 unfrozen layers for a total of 18 layers where the final two are a dense and a classification layer. The new network is then trained on 1,923 images (641 per class) and 60 images for testing. The 3D network structure can be seen in Figure 1.14 (using ReLU activation functions) and was trained on a dataset of 1,900 voxel grids with 1,615 training, 285 validation, and 60 testing voxel grids. The results of these networks are given in F1 score over the 60 test images, not accuracy or any other metric. The results show that the modified VGG-16 network provides an average F1 score of 0.95 while the 3D network provides an F1 score of 0.97, indicating that there are clear advantages utilizing the 3D information even when using a smaller network. While the results are impressive, the source dToF data applies significant pre-processing and the depth information is lost before being used with the NN.

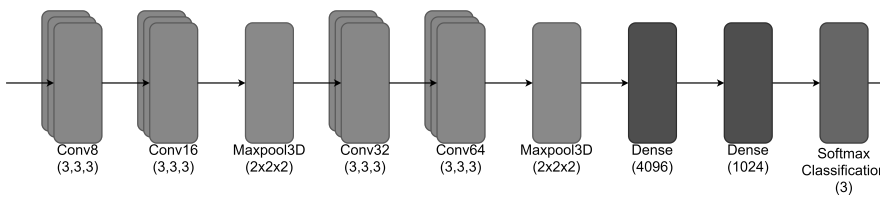


Figure 1.14: Proposed 3D Convolutional Network Architecture [Ruvalcaba-Cardenas et al., 2018].

The work performed by [Nash and Devrelis, 2020] presents a flash LiDAR system that is used to classify 6 different vehicles as if mounted to a drone using a CNN which is contrasted with other traditional methods such as Gaussian naive Bayes Classifier and a nearest neighbor classifier [Goldberger et al., 2004]. The training data consists of 116 vehicle datasets varying from a few to several thousand frames captured using the a 32x32 SPAD array outdoors during twilight. The sensor itself is mounted on a tower 16m above the ground to simulate a drone's elevation with a DJI gimbal used to provide movement and stability, while the target vehicles are placed 30m from the base. The SPAD array is illuminated by a 35mj, 5ns laser with a repetition rate of 20Hz and a beam divergence of 115°providing a depth resolution of 0.75m. The SPAD array uses a first photon TDC, so to reduce the effect of noise histogram averaging of 15-50 frames is used to filter out ambient photon detections. The depth images are pre-processed to remove the ground plane, convert the depth into meters, and reorient the vehicle. The structure of the CNN used to process the final depth images can be seen in Figure 1.15, unfortunately, no detailed information on the parameters was given. The CNN is compared to two traditional approaches, principle component analysis (PCA) with a Gaussian Classifier and PCA with a Nearest Neighbour classifier. The results of the work can be seen in Table 1.17. The CNN-based approach is shown to outperform the traditional methods but only by a small margin. The performance of the system overall is impressive given the low spatial and depth resolution. The low resolution reduces the data throughput making it more applicable for embedding on a drone, however, the network has only been tested using the CPU of an NVIDIA Jetson TX2 board and took multiple seconds to process and single frame. The work did express its desire to implement the system on a GPU which would improve the computation time.

The work presented by [Ito et al., 2017] is a small SPAD LiDAR system that utilizes depth, monocular, and intensity data to perform localization in an environment using a Deep Convolutional Neural Network (DCNN). The LiDAR system, named SPAD LiDAR, is targeted as a low-cost and compact sensor and is used with two CNN schemes: SPAD DCNN and Fast SPAD DCNN. The SPAD LiDAR sensor has a range of up to 80m using two SPAD arrays on the same chip, with one used to detect backscattered photons and the other to measure the ambient light. The main specifications can be seen in Table 1.18. The SPAD DCNN network uses all three data inputs to determine if there is a target and then resolve its 3D coordinates. The network is small, consisting of only three convolutional layers, three pooling layers, and two fully connected layers. Fast SPAD DCNN is a version with improved runtime performance over the original and integrates peak intensity data with the depth data so that results with low certainty are filtered out. This is done pixel-wise by binarising the peak intensity data and multiplying the result with the depth data. The data and ground truth data were collected indoors with trajectories being captured using a vicon motion capture system [vic, 2022] and later resized from the native 202x96 into 112x112. The networks are compared to the same network structure, but as if only depth data was used instead of all 3 inputs. The datasets used to train consist of three recordings of the sensor slowly moving towards a wooden pallet, with two used for training and one for testing. Three experiments were carried out to

Table 1.17: Classification accuracy of Algorithms [Nash and Devrelis, 2020].

Classification Algorithm	Accuracy (%)
PCA with Gaussian	63.0
PCA with Nearest Neighbour	85.2
CNN	86.3

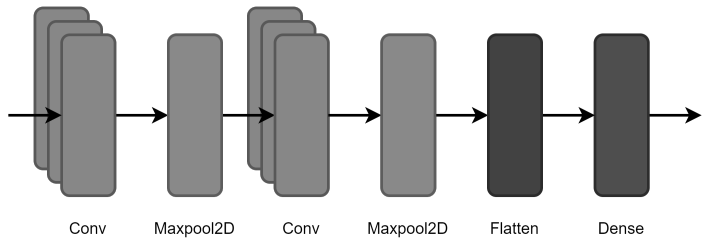


Figure 1.15: Proposed 3D Convolutional Network Architecture [Nash and Devrelis, 2020].

test SPAD DCNN and Fast SPAD DCNN. The first experiment compared the ability of the conventional method vs the SPAD DCNN at localising the target in the scene, with the conventional achieving an error of 6.7cm and SPAD DCNN achieving an error of 4.4cm. As the depth resolution of the SPAD LiDAR is 3.5cm, the result is not only an improvement over the conventional method but close to the resolution limitation. The 2nd experiment compares the runtime of SPAD DCNN and Fast SPAD DCNN. It reuses the dataset from the first experiment and measures the computational demands and runtime to process one frame when the networks are implemented on a CPU (core i7) and a GPU (GTX Titan X), with the results shown in Table 1.19. The third test examines the increase in localization error as the guided vehicle moves off course from the training data. Five trajectories with 5 variations are used to test the system's ability to cope with variance in sensor movement. The results are that for a deviation of 50cm off the path of the training set, the average localization error increases to 0.15m. Overall, it can be seen that by providing the neural network multiple sources of data to leverage, it was able to reduce the error by 2.3cm when compared to depth data alone.

Table 1.18: SPAD LiDAR sensor specifications [Ito et al., 2017].

Pixel resolution	202×96 pixels
Field of view	55°×9°
Frame rate	10fps
Size	W 0.067 × H 0.073 × D 0.177 m
Range	80m (reflectivity 9%)
Laser	Class 1 laser
Distance resolution	0.035m

The work of [Scholes et al., 2022] presents a SPAD LiDAR system called Dronesense which is capable of determining the type, orientation, and segmentation

Table 1.19: Runtime Results of Fast SPAD DCNN and SPAD DCNN [Ito et al., 2017].

	SPAD DCNN	Fast SPAD DCNN
CPU runtime	36.03ms	25.5ms
GPU runtime	4.1ms	2.9ms

(body, engines, camera) of drones in flight using a CNN that utilizes a decision tree and an ensemble structure. The system utilizes the SPAD sensor Quantic 4x4 [Hutchings et al., 2019] which is 256x256 SPAD pixel array able to provide both intensity data and depth histograms (64x64 resolution). The sensor outputs 16 bin histograms with 500ps temporal resolution. In this work, the sensor resolution was set differently from the default at 80x240 pixels for intensity and depth at 20x60 pixels. The network is trained on a synthetic dataset of 72,000 images generated by placing two drones in an Unreal Engine environment at random positions, orientations, and distances within the camera FoV. The simulated intensity images are created using a Poisson filter and are resized to 80x240, while the depth maps are downsampled to 20x60 and converted into 15 bin histograms. The proposed large network structure can be found in the original paper. The core of the network is referred to as the Drone Feature Encoder (DFE), and it extracts features from depth and intensity data into a space of 1x3x32 filters. The remainder consists of either the segmentation or the decision trees for orientation and identification predictions. The testing uses two angular regimes, "full angle" and "reduced angle" as seen in Table 1.20, with the reduced angle having the drone constrained to the specifications of the manufacturer. The trained network was then tested using real data captured by the Quantic 4x4 camera of a drone in flight, with an example of the results shown in Table 1.21. The network was able to correctly distinguish between the two drone types with average accuracies of over 90%. The training and testing data seem to not contain any background noise, so either some pre-processing is performed or the synthetic data does not take into account any ambient light and background objects. The work also experiments with the network's ability to perform using only the depth histograms or only the intensity data. When using only the histograms, the average reduction in accuracy for orientation prediction was less than 0.5% while the reduction when using only intensity data was just under 2%. The small loss when only using histograms suggests the benefit is minimal with the intensity included. The impact of a reduction in resolution was also investigated, with the resolution being halved and quartered. The main result was an overall reduction in the segmentation accuracy, with the largest reduction being the segmentation of engines resulting in a loss of accuracy of 7% and 23% for the half and quarter resolution respectively. The ability to detect the camera was not greatly impacted, however, with only a reduction of 2% and 3% for half and quarter resolution respectively. The resolution also impacted the orientation prediction with the yaw predictions accuracy reducing by 1.7% and 7.6% for the half and quarter resolution respectively.

The work by [Mora-Martín et al., 2021] presents a short-range 3D depth imager that uses CNNs to investigate the accuracy of gesture recognition using 3D depth data with low lateral resolution at high speeds. It uses the Quantic 4x4

Table 1.20: Full angle and reduced angle regimes for drone orientation [Scholes et al., 2022].

	Full angle	Reduced angle
Yaw range	0°, 360°	0°, 360°
Roll range	0°, 360°	140°, 220°
Pitch range	0°, 180°	140°, 220°

Table 1.21: Dronesense predictions on real data example [Scholes et al., 2022].

Metric	Ground truth	Prediction, accuracy %
Classification	Mavic2	Mavic2, 100
Orientation		
Yaw	31	19, 93
Roll	180	177, 96
Pitch	90	92, 96

[Hutchings et al., 2019] sensor to capture both depth and intensity data at resolutions of 64x32 and 256x128 respectively. The laser is a 2W max power flood illumination source with a FoV of 20°. It has a pulse length of 10ns and a repetition rate of 6MHz. The sensor itself is set with a temporal resolution of 4ns per bin and captured data at approximately 200FPS. 4 different combinations of input data was tested, specifically depth (64x32), intensity (256x128), histogram (64x32x16), and intensity and depth (I+D) (256x128x2). I+D stacks depth data up-scaled to 256x128 using nearest neighbor interpolation onto the third axis of the intensity data for a final resolution of 256x128x2. The system comprises two networks, the first is a U-net [Ronneberger et al., 2015] network version that was modified to handle the 4 different data schemes, which outputs a binary mask with 0 being the background and 1 being an object of interest in the pixel. The second is a simpler classification network that predicts the gesture. The network was trained using a RTX2070 GPU using three datasets, the first dataset being a hand in front of a plain background without ambient light, the second introducing objects into the background, and the third introducing objects and ambient light. 1,000 frames for each of the 3 gestures are captured for each dataset, resulting in a 9,000 frame dataset. The network training results can be seen in Table 1.22. The Histogram and I+D network proved to get the best results, though the histogram-based network has the added advantage of faster processing time. The faster processing time can be attributed to only needing one data frame from the sensor which halves the data acquisition time and requires no additional pre-processing to upscale and combine the data. It also shows that while the histogram data has a lower spatial resolution, it benefits from information such as object reflectivity, ambient level, object edges, and the depth that the histogram contains resulting in similar or better performance than that of higher resolution data.

1.4.2 Spiking Neural Networks

The work [Ara Shawkat et al., 2020] presents a SPAD-based vision system that utilizes on-chip memristive spiking neural networks for object detection. It is currently