SPAD-Based LiDAR Data Pre-Processing on Embedded Systems Using Machine Learning

Von der Fakultät für Ingenieurwissenschaften,
Abteilung Elektrotechnik und Informationstechnik
der Universität Duisburg-Essen

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

genehmigte Dissertation

von
Gongbo Chen

aus
Fujian, China

Gutachter: Prof. Dr.-Ing. Rainer Kokozinski
Gutachter: Prof. Dr. rer. nat. Gregor Schiele

Tag der mündlichen Prüfung: 15.07.2022

Acknowledgements

I would like to express my heartfelt thanks to all those who supported me. The work would not have been successfully conducted without their help.

Special thanks to Prof. Dr.-Ing. Rainer Kokozinski. He consistently allowed this work to be my own, but would steer me in the right direction when necessary.

I gratefully acknowledge the assistance of Prof. Dr. rer. nat. Gregor Schiele. He read the work carefully and gave very helpful input and good advice.

I am extremely grateful to Dr.-Ing Pierre Gembaczka, who initially encouraged me to conduct this work and gave me invaluable inspiration.

I would also like to extend my deepest gratitude to Burkhard Heidemann, who is always very sincere and shared many valuable work experiences with me.

I would like to express my deepest appreciation to Dr.-Ing Christian Wiede, who always supported and nurtures me. The door to his office was always open whenever I ran into a trouble spot or had a question about my research or writing.

I also had the great pleasure of working with the team of embedded artificial intelligence. Lars Wulfert, Dennis Latoschewski, Martin Hubert Verbunt, Lukas Krupp, Felix Wichum, and Carolin Wuerich, they were always willing to help and were instrumental in conducting my research.

Many thanks to Dr. Bennertz Bernhard, Wolfgang Gröting, Martin Kemmerling, and Cornelia Metz. I will never forget their kindness and support.

I very much appreciate all students who contributed to this work. Giray Atabey Kirtiz, Felix Landmeyer, Kevin Krause, Steven Truong, and Santosh Kumar Kasam, their efforts have enabled the work to progress seamlessly. I will miss those brainstorming times and lunchtime talks.

I would also like to acknowledge the assistance from the colleagues of optical sensor systems. Sara Grollius, Andre Buchner, Dr. Jan Frederik Haase, and Dr. Jennifer Ruskowski, thanks for their practical suggestions and helpful advice.

I cannot begin to express my thanks to my girlfriend Xueying Wang, who profound belief in my abilities and aided with detailed proofreading.

Finally, I would like to express my deepest appreciation to my parents Jianping Chen and Yilan Fan for their support, love, and patience throughout the years.

Gongbo Chen

Abstract

With the rise of **A**dvanced **D**river **A**ssistance **S**ystems (ADAS), range sensors and corresponding data processing methods are becoming more and more important. It plays a decisive role in one of the major future topics, autonomous driving. **Li**ght **D**etection **A**nd **R**anging (LiDAR) sensors are attracting attention due to their unique advantages in terms of high radial distance resolution. Therein, **S**ingle **P**hoton **A**valanche **D**iode (SPAD)-based direct **T**ime-**O**f-**F**light (TOF) LiDAR system shows a powerful ability regarding its simple measurement principle and extremely high energy detection efficiency. This system measures the distances based on the traveling time of laser photons to reach and return to an object. However, one of the greatest challenges of such a system is high background light, which causes a number of false triggers and interferes with the desired signals. Moreover, due to the large volume of LiDAR data, classical methods are only able to perform simple processing on low-level LiDAR data in applications with a high demand on timing performance, such as ADAS. In the subsequent high-level processing, only the depth information (i.e., point cloud) is utilized and the used LiDAR front-end is considered as a black box. In this case, most features in low-level LiDAR data are overlooked, resulting in low system robustness in harsh conditions, such as high background light and large distances.

This work seeks a breakthrough on the distance determination performance of a SPAD-based direct TOF LiDAR system. In the scope of the work, a machine learning-based distance prediction approach, **M**ulti-**P**eak **A**nalysis (MPA), is proposed for LiDAR time-correlated histograms to improve timing performance and the system robustness against background light. First of all, the existing methods based on the LiDAR system are reviewed. Concepts and attempts are designed and carried out on the combination of machine learning and low-level LiDAR data. Two datasets are used to assess the performance of MPA. After theoretical and simulative evaluations, MPA is designed and includes three components: 1) a physics-guided feature extraction, 2) two distance prediction algorithms (fully-connected neural network and naive Bayes classifier), and 3) a correlation analysis in time and space. The evaluation results lead to the conclusion that MPA outperforms a widely-used classical digital processing in terms of timing performance and distance determination with ±5 % error bound with various background light intensities and distances. Finally, MPA is implemented on both a personal computer and a **F**ield-**P**rogrammable **G**ate **A**rray (FPGA) module for runtime verifications. Demonstrations show a high agreement with theoretical and simulative results and confirm the feasibility of MPA.

Zusammenfassung

Mit dem Aufkommen von **A**dvanced **D**river **A**ssistance **S**ystems (ADAS) gewinnen Entfernungssensoren und entsprechende Datenverarbeitungsmethoden immer mehr an Bedeutung. Sie spielen eine wichtige Rolle bei einem der großen Zukunftsthemen, dem autonomen Fahren. **Li**ght **D**etection **A**nd **R**anging (LiDAR)-Sensoren ziehen aufgrund ihrer einzigartigen Vorteile in Bezug auf eine hohe radiale Entfernungsauflösung die Aufmerksamkeit auf sich. Dabei zeichnet sich das **S**ingle **P**hoton **A**valanche **D**iode (SPAD)-basierte direkte **T**ime-**O**f-**F**light (TOF) LiDAR-System durch sein einfaches Messprinzip und seine extrem hohe Energieerfassungseffizienz aus. Dieses System misst die Entfernungen auf der Basis der Laufzeit der Laserphotonen, die ein Objekt erreichen und zurückkehren. Eine der größten Herausforderungen eines solchen Systems ist jedoch das starke Hintergrundlicht, das eine Reihe von Fehlmessungen verursacht und die gewünschten Signale stört. Außerdem können klassische Methoden aufgrund der großen Menge an LiDAR-Daten nur eine grundlegende Verarbeitung von Low-Level-LiDAR-Daten in Anwendungen mit hohen Anforderungen an die zeitliche Leistung, wie z. B. ADAS, durchführen. Bei der anschließenden High-Level-Verarbeitung werden nur die Tiefeninformationen (d. h. die Punktwolke) verwendet und das verwendete LiDAR-Frontend wird als Blackbox betrachtet. In diesem Fall werden die meisten Merkmale in den Low-Level-LiDAR-Daten übersehen, was zu einer geringen Robustheit des Systems unter schwierigen Bedingungen führt, z. B. bei hohem Hintergrundlicht und großen Entfernungen.

In dieser Arbeit wird ein Durchbruch bei der Entfernungsbestimmung mit einem SPAD-basierten direkten TOF-LiDAR-System angestrebt. Im Rahmen der Arbeit wird ein auf maschinellem Lernen basierender Ansatz zur Entfernungsvorhersage, die **M**ulti-**P**eak **A**nalyse (MPA), für zeitkorrelierte LiDAR-Histogramme vorgeschlagen, um die zeitliche Leistung und die Robustheit des Systems gegenüber Hintergrundlicht zu verbessern. Zunächst werden die bestehende Mess- und Verarbeitungsmethoden für LiDAR-Systeme analysiert. Konzepte und Versuche werden entworfen und auf der Kombination von maschinellem Lernen und Low-Level-LiDAR-Daten durchgeführt. Zwei Datensätze werden verwendet, um die Performance der MPA zu bewerten. Nach theoretischen und simulativen Auswertungen wird die MPA entworfen und umfasst drei Komponenten: 1) eine physikbasierte Merkmalsextraktion, 2) zwei Algorithmen zur Entfernungsvorhersage (fully-connected neural network und naive Bayes classifier) und 3) eine Korrelationsanalyse in Bezug auf zeitliche und räumliche Informationen. Die Evaluierungsergebnisse führen zu dem Schluss, dass die MPA eine weit verbreitete klassische digitale Verarbeitung in Bezug auf die zeitliche Leistung und die Entfernungsbestimmung mit einer Fehlergrenze von ±5 % bei verschiedenen Hintergrundlichtintensitäten und Entfernungen übertrifft. Schließlich wurden die MPA sowohl auf einem PC als auch auf einem **F**ield-**P**rogrammable **G**ate **A**rray (FPGA)-Modul implementiert, um Laufzeitüberprüfungen durchzuführen. Die Laufzeitdemonstrationen zeigen eine hohe Übereinstimmung mit den theoretischen und simulativen Ergebnissen und bestätigen die Funktionsfähigkeit der MPA.

Table of Contents

List of Figures

List of Tables

Abbreviations and Symbols

List of Abbreviations

Adam	**Ada**ptive **m**oment estimation
ADAS	**A**dvanced **D**river **A**ssistance **S**ystems
APD	**A**valanche **P**hoto**D**iode
ARM	**A**dvanced **R**ISC **M**achines
AUC	**A**rea **U**nder **C**urve
BES	**B**ackground **E**stimation and **S**ubtraction
BRAM	**B**lock **R**andom **A**ccess **M**emory
CDP	**C**lassical **D**igital **P**rocessing
CMOS	**C**omplementary **M**etal-**O**xide **S**emiconductor
CNN	**C**onvolutional **N**eural **N**etwork
DSP	**D**igital **S**ignal **P**rocessor
ELU	**E**xponential **L**inear **U**nit
FN	**F**alse **N**egative
FNN	**F**ully-Connected Feed-Forwarded **N**eural **N**etwork
FOV	**F**ield-**O**f-**V**iew
FPC	**F**irst **P**hoton **C**ounting
FPGA	**F**ield-**P**rogrammable **G**ate **A**rray
FP	**F**alse **P**ositive
LiDAR	**Li**ght **D**etection **A**nd **R**anging
LUT	**L**ook**U**p-**T**able
MAE	**M**ean **A**bsolute **E**rror
MAP	**M**aximum **A** **P**osteriori estimation
MCM	**M**onte-**C**arlo-**M**ethod
MEMS	**M**icro-**E**lectro-**M**echanical **S**ystems
MLE	**M**aximum **L**ikelihood **E**stimation
MPA	**M**ulti-**P**eak **A**nalysis
MSE	**M**ean **S**quare **E**rror
NBC	**N**aive **B**ayes **C**lassifier
OPA	**O**ptical **P**hase **A**rray
PC	**P**ersonal **C**omputer
PDF	**P**robability **D**ensity **F**unction
PMF	**P**robability **M**ass **F**unction
PPV	**P**ositive **P**redictive **V**alue

ReLU	**Re**ctified **L**inear **U**nit
RMSE	**R**oot **M**ean **S**quare **E**rror
ROC	**R**eceiver **O**perating **C**haracteristic
SNR	**S**ignal-to-**N**oise **R**atio
SPAD	**S**ingle **P**hoton **A**valanche **D**iode
tanh	**tan**gens **h**yperbolicus
TC-Hist	**T**ime-**C**orrelated **Hist**ogram
TCSPC	**T**ime-**C**orrelated **S**ingle-**P**hoton **C**ounting
TDC	**T**ime-to-**D**igital **C**onverter
TN	**T**rue **N**egative
TOF	**T**ime-**O**f-**F**light
TP	**T**rue **P**ositive
TPR	**T**rue **P**ositive **R**ate

List of Symbols

a	Continuous value
A	Event
A_{Pixel}	Pixel area
Acc_{C}	Accuracy of classification
Acc_{D}	Accuracy of distance prediction
$Acc_{\mathrm{D,\,Val}}$	Acc_{D} on validation dataset
$Acc_{\mathrm{D,\,Test}}$	Acc_{D} on test dataset
$Acc_{\mathrm{D,\,Co}}$	Acc_{D} after correlation
Acc_{Th}	Accuracy for distinguishing object and noise
$Acc_{\mathrm{Th,\,B}}$	Acc_{Th} under certain background light level
$\Delta Acc_{\mathrm{D,\,Test}}$	Difference of two $Acc_{\mathrm{D,\,Test}}$
b	Continuous value
b	Bin number
b_{H}	Bin number with the highest soft-decision
b_n	The n^{th} bin number
b_{Obj}	Bin number of object distance
$bias$	Bias value
b'	Bin number of reference feature
Δb	Difference between original and reference bin numbers
Δb_n	Difference between original and reference bin numbers in sub-region n
B	Event
c	Speed of light
c	Class
C	Class group
C_{Max}	Received maximum photon counts
C_{Obj}	Received photon count at object distance
$\overline{C}_{\mathrm{Obj}}$	Received average photon counts at object distance
d_{Obj}	Object distance
d_{Pred}	Predicted distance
d_{TDC}	TDC distance resolution
d_{Vert}	Vertical distance of object

d_{Wall}	Vertical distance of wall
D_{Max}	Maximum detection range
D	Dataset
D1	Dataset with synthetic data
$\mathrm{D1}_{\text{Test}}$	Test dataset
$\mathrm{D1}_{\text{Train}}$	Training dataset
$\mathrm{D1}_{\text{Val}}$	Validation dataset
D2	Dataset with real data
e	Euler's number
ΔE	Tolerable error margin
f_{H}	The highest feature value
f_n	The n^{th} noise-reduced feature value
$f_{n,\,t,\,x,\,y}$	f_n at time t and pixel $(x,\,y)$
f_{NBC}	Function of NBC
$f_{\text{Norm},\,n}$	n^{th} normalized feature in NBC
f^*	Correlated feature value
f_n^*	The n^{th} correlated feature value
$f^*_{n,\,t,\,x,\,y}$	f_n^* at time t and pixel $(x,\,y)$
Δf	Difference between original and reference feature values
$\Delta f_{\text{Time},\,n}$	Difference between original and reference feature values in time within sub-region n
$\Delta f_{\text{Space},\,n}$	Difference between original and reference feature value in space within sub-region n
$\hat{f}_{i,\,t}$	Correlated feature in time
$\hat{f}_{i,\,j}$	Correlated feature in space
F_{Fill}	Fill factor
F_{FOV}	Transmission factor of FOV
F_{G}	Feature group
$F_{\mathrm{G},\,t}$	Feature group at time t
F_n	The n^{th} noise-reduced sub-region feature
F_{Rft}	Attenuation factor during reflection
F_1	Attenuation factor of laser propagation path
F_2	Attenuation factor of background irradiance
F_n'	The n^{th} noise-reduced sub-region feature

F'_{G}	Noise-reduced feature group
F^*_{G}	Correlated feature group
$F^*_{\mathrm{G},\,t}$	Correlated feature group at time t
G	Gain function
h	Plank constant
H	Sub-region number with the highest s
i	Discrete variable
I	Irradiance of light
$I_{\mathrm{B,\,Obj}}$	Irradiance of background light on object surface
$I_{\mathrm{L,\,Obj}}$	Laser irradiance on object surface
$I_{\mathrm{L,\,Sensor}}$	Irradiance of laser on sensor surface
I_{N}	Input vector with N elements
I_{S}	Solar irradiance
I_{Sensor}	Overall light irradiance on sensor surface
I_{Spec}	Spectral irradiance
j	Discrete variable
k	Discrete value
k_1	Constant value
k_2	Constant value
k	Discrete variable
K	Discrete value
K	Discrete variable
$K_{\mathrm{Const,\,b}}$	Constant convolution kernel at bin b
$K_{\mathrm{Exp,\,b}}$	Exponential convolution kernel at bin b
$K_{\mathrm{Gauss,\,b}}$	Gaussian convolution kernel at bin b
$\mathcal{L}$	Likelihood function
$L\mathcal{L}$	Log-likelihood function
$Loss$	Loss function
$Loss_{\mathrm{CE}}$	Cross-entropy as loss function
$Loss_{\mathrm{MAE}}$	MAE as loss function
$Loss_{\mathrm{MSE}}$	MSE as loss function
m	Discrete variable
M	Number of models
M_n	The n^{th} local maximum

n	Discrete variable
N	Number of sub-regions
N_{B}	Number of bins in a TC-Hist
N_{BE}	Number of samples creating visible side-peaks
N_c	Number of samples in class c
N_{Coin}	Coincidence number
N_{Com}	Number of clock cycles for comparison operation
N_{Fetch}	Number of clock cycles for fetching operation
N_{HN}	Number of hidden layer neurons
N_{M}	Number of measurements
N_{NBC}	Number of classes in NBC
N_{Opt}	Number of optimized clock cycles
N_{P}	Number of laser pulse bins
N_{PC}	Number of clock cycles in PC-based implementation
N_{R}	Number of sub-regions
N_{S}	Number of samples
N_{Sum}	Number of clock cycles for adding operation
N_{T}	Number of trials
N_{True}	Number of true feature groups
N_{Write}	Number of clock cycles for writing operation
N'_{BE}	Number of samples creating barely side-peaks
ΔN_{CC}	Saved number of clock cycles
$\mathcal{N}$	Normal distribution
o_c	Soft-decision for class c
o_n	Soft-decision for the n^{th} feature value
o_{H}	Soft-decision with the highest value
o_{Th}	Threshold of soft-decision
o'_n	Soft-decision for the n^{th} reference feature value
o^*_n	Soft-decision for the n^{th} correlated feature value
O	Output
O_t	Output at time t
O^*	Output after correlation
O^*_t	Output after correlation at time t
p	Probability density

p_1	Probability density function of receiving the first photon
P	Probability
P_{B}	Binomial distribution
$P_{\mathrm{Bernoulli}}$	Bernoulli distribution
P_{BE}	Proportion of samples with boundary effect
P_{L}	Laser power
P_{Norm}	Normalized probability group of a TC-Hist
$P_{\mathrm{Norm},\,b}$	Normalized probability at bin b
$P_{\mathrm{Norm},\,b'}$	Normalized probability at bin b'
P_{P}	Poisson distribution
$P_{1,\,b}$	Probability mass function of receiving the first photon at bin b
$P_{1,\,\mathrm{Max}}$	Maximum value of P_1
P_{θ}	Probability function with parameter θ
$\overline{P}_{\mathrm{B},\,b}$	Mean of background at bin b
$\overline{P}_{\mathrm{BE},\,n}$	Mean of $P_{\mathrm{Norm},\,n}$ with boundary effect
$\overline{P}_{n,\,c}$	Mean of $P_{\mathrm{Norm},\,n}$ with class c
$\overline{P}_{\mathrm{Noise},\,n}$	Mean of $P_{\mathrm{Norm},\,n}$ from noise
$\overline{P}_{\mathrm{Norm},\,n}$	Mean of normalized probabilities
$\overline{P}_{\mathrm{Obj},\,n}$	Mean of $P_{\mathrm{Norm},\,n}$ from object
$\overline{P}_{m,\,n,\,c}$	Mean of $P_{\mathrm{Norm},\,n}$ with class c for model m
r	Photon detection rate
r_{B}	Background photon detection rate
r_{BQ}	Quantized background photon detection rate
r_{L}	Laser photon detection rate
r_{LB}	Photon detection rate of laser and background
$s_{\mathrm{BE},\,n}$	Population variance of $P_{\mathrm{Norm},\,n}$ with boundary effect
$s_{n,\,c}$	Population variance of $P_{\mathrm{Norm},\,n}$ with class c
$s_{\mathrm{Noise},\,n}$	Population variance of $P_{\mathrm{Norm},\,n}$ from noise
$s_{\mathrm{Obj},\,n}$	Population variance of $P_{\mathrm{Norm},\,n}$ from object
$s_{m,\,n,\,c}$	Population variance of $P_{\mathrm{Norm},\,n}$ with class c for model m
t	Time variable
t_{Coin}	Coincidence time
T	Time variable

T_{Obj}	Propagation time of laser reflected by object
T_{P}	Laser pulse time duration
T_{TDC}	TDC time resolution
T_{TOF}	Time between laser emission and photon detection
T'_{TOF}	Digital timestamp
Δt	Unit time duration
ΔT	Observation time window
U	Uniform distribution
v	Intermediate value
v_{Obj}	Object velocity
v_{Max}	Maximal intermediate value
v_{Min}	Minimal intermediate value
v'_{Max}	Maximized v_{Max}
v'_{Min}	Minimized v_{Min}
w_i	The i^{th} weight
w_m	Weighting factor of the m^{th} model
w_{Obj}	Object width
w'_i	The i^{th} updated weight
W_{Ph}	Photon energy
W_{R}	Sub-region width
x	Continuous value
x	Variable
x_i	The i^{th} element in a data sequence
x_n	The n^{th} element in a data sequence
x'_i	The i^{th} modified x for maximization/minimization of v
X	Constant value
X_1	Constant value
X_2	Constant value
X	Data sequence
y	Output value
y^*	Label
α	Impact factor of reference amplitude
α_{FNN}	Impact factor of reference amplitude for FNN
α_{NBC}	Impact factor of reference amplitude for NBC

β_{FNN}	Impact factor of Δb for FNN
β_{NBC}	Impact factor of Δb for NBC
η	Update rate
ϕ	Photon flux
η_{PDE}	Photon detection efficiency
θ	Function parameter
$\theta_{n,c}$	Variance of sub-region n with class c
$\hat{\theta}_{\mathrm{MAP}}$	Maximum a posteriori estimation
$\hat{\theta}_{\mathrm{MLE}}$	Maximum likelihood estimation
λ	Wavelength
μ	Expected value
μ_b	Expected value at bin b
$\mu_{\mathrm{B,\,Obj}}$	Expected background photon counts of at object bin
μ_{t}	Expected value at time t
μ_{T}	Expected value within $(0, T]$
μ_{Obj}	Expected photon counts at object bin
σ	Variance
φ	Aperture angle
ε	Step function
ε_1	High-level control signal
ε_2	Low-level control signal
π	Ratio of a circle's circumference

Chapter 1

Introduction

Traffic safety has always been a topic of great concern. In 2016, it is reported that 1,099,032 traffic accidents took place, resulting in 25,651 fatalities and more than 1.4 million injured people in Europe [1]. According to the report, accidents caused by distraction, drugs, alcohol, fatigue, and over-speed driving account for a large proportion. As summarized in [2], human errors, including distraction, fatigue, and immature driving behavior, are the main causes of traffic accidents. In order to reduce the safety hazards caused by human errors, **A**dvanced **D**river **A**ssistance **S**ystems (ADAS) is gaining traction as a solution to assist driving behavior. Therein, depth information for environmental interaction is one of the most important sectors [3], which involves several range sensors and data processing technologies. The development of range sensors has lasted more than 20 years [4]. Common ranging technologies used in ADAS are radar, camera, ultrasonic, and **Li**ght **D**etection **A**nd **R**anging (LiDAR) [2]. An intuitive comparison among radar, camera, and LiDAR is given in [5]. Since these ranging technologies have their own strengths and weaknesses, sensor fusion in automotive receives increasing interest to achieve performance beyond any single sensor technology. Among these technologies, LiDAR provides high range resolution and long depth range [5, 6, 7]. As applied in [8], the precise range information provided by the LiDAR is indispensable for the safety in ADAS. Among all variants in the LiDAR family, **S**ingle **P**hoton **A**valanche **D**iode (SPAD)-based direct **T**ime-**O**f-**F**light (TOF) LiDAR system shows a tendency to outperform other companions due to its mature stage on research and development and its powerful abilities [9].

Research has been carried out both in the hardware and the software of the

LiDAR system. In hardware, although LiDAR technology has not been deployed in production automobiles to date [5], many LiDAR-based product prototypes are built both in the commercial sector [10, 11, 12, 13] and for research use [14, 15, 16, 17, 18, 19, 20, 21]. Besides, several sensors are used for on-road testing and analysis as well. Waymo's fully self-driving technology, along with the LiDAR system developed by themselves, has driven over 20 million miles on real-world roads since 2009 [22]. In addition to the hardware, research on LiDAR data processing is equally important. Therein, the study of the LiDAR data generation is one of the most important fields. For example, CARLA is a well-known open-source LiDAR data simulator for autonomous driving research. It offers a variety of different virtual scenarios for detections. Compared to real measurements, the parameters of synthetic data are highly configurable. Besides, the simulation process is highly automated. This promotes the preliminary validation and optimization of data processing methods. In addition to the synthetic data, real data is important as well, since measurements in ADAS involve a large number of parameters, which are challenging for simulations, e.g., unknown interference, system-induced error, vibration, and deviation in production. One widely-used LiDAR data benchmark is KITTI [23], which contains datasets accumulated by LiDAR systems onboard moving vehicles. Based on these datasets, numerous works are carried out to investigate useful information in LiDAR data, including depth completion, object detection, tracking, and road detection.

1.1 Motivation

In autonomous driving applications, SPAD-based LiDAR systems and on-board processing methods are responsible for environmental sensing and decision making. Typically, the environment to be detected is changing rapidly and a system needs to collect data and to analyze and extract useful information in a short time. Therefore, the ability to quickly locate the position of objects in harsh environments becomes one of key metrics for LiDAR sensor systems. Due to the high dynamic degree of environments, the demand on precision of sensor systems is relatively low compared to information update rate and system robustness.

Modern LiDAR systems can reach precision between centimeter and millimeter for distance measurement. Therefore, they are well suited to meet the precision requirements of autonomous driving. However, system robustness and information update rates impose significant challenges on LiDAR systems. On the one hand, many promising hardware-level approaches, e.g., time-gating [24] and coincidence detection [25], are proposed to improve measurement reliability. However, they have a number of restrictions. Measurement reliability under extreme conditions, e.g., changing ambient light and large distances, are still questionable. On the other hand, although machine learning-based methods have proven their flexibility and adaptability in the field of LiDAR point cloud processing, they consider the sensor front-end as a black box [26] and overlook the low-level data features, e.g., photon statistics, noise features, and sensor characteristics. This makes these methods highly dependent on the quality of point clouds and results in low robustness to environmental factors, such as background light. Therefore, holistic analysis and investigation between sensor front-end and machine learning-based processing methods should be carried out to improve the information utilization and robustness of the system.

1.2 Objectives and Thesis Outline

This work focuses on SPAD-based direct TOF LiDAR systems. The key idea is to investigate detection reliability and timing performance by embedding measurement and sensor internal processes into machine learning framework. Thus, the work pursues two main objectives. First, by constructing the whole LiDAR data processing workflow, different features in low-level LiDAR data are studied and characterized. Second, low-level LiDAR data are combined with the concept of machine learning to investigate a new data pre-processing method for allocating coarse object distance under harsh conditions, with focus on robustness against background light and real-time performance.

To achieve the overall objectives, sub-objectives are specified as follows:

- Studies on decomposition and construction of the LiDAR data workflow.
- Investigation and characterization of LiDAR data forms and features.
- Research on machine learning-based concepts and algorithms.
- LiDAR data generation, including synthetic data and real data.

- Design of a feature extraction process based on the concept of pattern recognition and feature transformation.
- Fusion of machine learning algorithms with low-level LiDAR data features.
- Hypotheses and verifications of machine learning algorithms on SPAD-based direct TOF LiDAR data.
- Evaluation of system performance with the focus on robustness and timing performance.
- Optimization and runtime demonstration.

According to the previously described research objectives, the work is structured as follows:

In chapter 2, the fundamental of related areas is presented, including LiDAR systems and their working principles, related mathematics, and machine learning algorithms in data processing.

In chapter 3, the state-of-the-art in the field of SPAD-based direct TOF LiDAR data processing is provided. The chapter starts with the framework of a data processing workflow. Different data forms as well as the corresponding representative processing methods in LiDAR systems are discussed, including hardware-level processing, classical digital processing, and machine learning. As a result, challenges of the current development are discussed according to their categories, namely LiDAR front-end, data pre-processing, machine learning-related, and ADAS-related challenges. Finally, the uncharted research areas and their values are indicated, and the research gaps are proposed.

In chapter 4, a designed LiDAR data processing approach, i.e., **M**ulti-**P**eak **A**nalysis (MPA) on **T**ime-**C**orrelated **Hist**ogram (TC-Hist)s, is proposed and explained thoroughly. First, a measurement principle for TC-Hist, i.e., **F**irst **P**hoton **C**ounting (FPC), is introduced. Second, a physics-guided feature extraction process is presented, which is governed by the Poisson statistics and focuses on local maxima in a TC-Hist. Third, distance prediction algorithms, including a **F**ully-Connected Feed-Forwarded **N**eural **N**etwork (FNN), a binomial estimation, and a **N**aive **B**ayes **C**lassifier (NBC) in the application of previous extracted features, are described. In addition, a correlation analysis aiming at improving LiDAR data reliability is explained, which considers the information of TC-Hists in spatial and temporal dimensions simultaneously. At the end of this chapter, a summary of the whole approach is given.

In chapter 5, detailed result analysis and discussion are provided. At the beginning of this chapter, the used datasets, including a dataset with synthetic TC-Hists and a dataset with real TC-Hists, are introduced and the corresponding parameters are specified. After that, important evaluation metrics in terms of feature extraction, prediction, comparison, and optimization are presented. Based on the synthetic TC-Hists and the specified metrics, the parameter determination of MPA introduced in chapter 4 is conducted. Subsequently, three optimization approaches, i.e., ablation study of background suppression, stride analysis, and threshold analysis, are introduced aiming at improving the performance and timing. The last part of this chapter deals with the performance on the real TC-Hists.

Chapter 6 devotes to the system implementation and demonstration of the proposed approach. A LiDAR system is used as the demonstration platform and MPA is implemented using software LabVIEW [27] on a **P**ersonal **C**omputer (PC) as well as on a **F**ield-**P**rogrammable **G**ate **A**rray (FPGA) module. In particular, a step-by-step implementation process on the FPGA module is explained and the applied approximations are introduced. Finally, the runtime demonstration results are provided.

Chapter 7 gives the summary of this work with the focus of contributions to LiDAR data processing field. Furthermore, outlooks are indicated to provide information and inspiration for future research.

Chapter 2

Background

2.1 LiDAR Systems

This work focuses on the data generated by the SPAD-based TOF system. In this section, components and working principles in the system are introduced, including SPADs, flash and scanning LiDAR systems, and direct TOF measurements. Finally, the definition of the light intensity is specified.

2.1.1 Single Photon Avalanche Diode and Quenching Circuit

A photodiode is a p-n junction, which is supplied with voltage in the reverse direction, shown in Figure 2.1. In this case, the space charge zone, i.e., a zone without free charge carriers, is created between the p and n regions. When a photon strikes the zone with sufficient energy, the energy is absorbed by the space charge zone and an electron-hole-pair is generated. Due to the presence of the applied electric field, the electron drifts into the n region and the hole drifts into the p region. Since the anode (+) and the cathode (-) are conductively connected, a current flow occurs. However, such a photodiode cannot generate sufficient electrical signals for detection with limited incident photons. Therefore, an internal amplification for the charge carriers generated by incident photons is applied. By using a high electric field within the photon diode, the created electron-hole-pair is accelerated and generates further pairs through atomic collision. Such a diode is called **A**valanche **P**hoto**D**iode (APD).

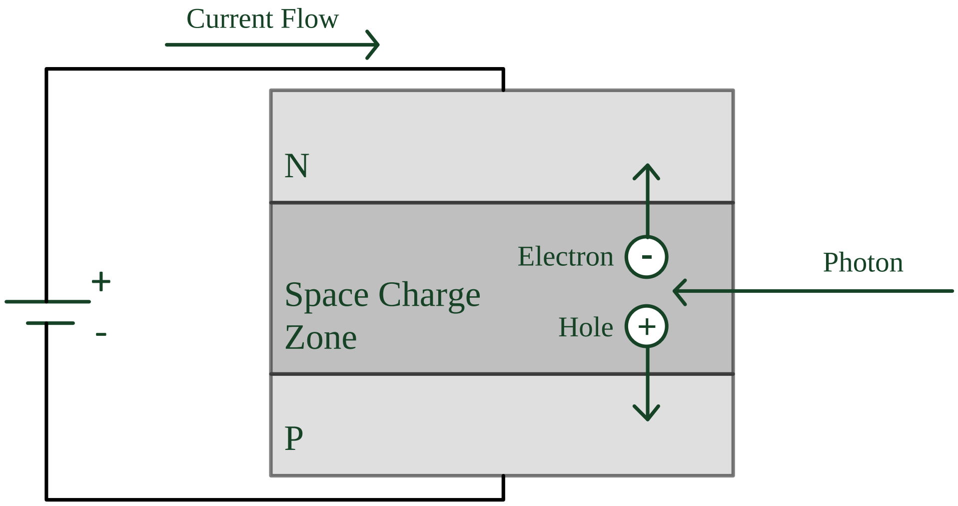


Figure 2.1: Working principle of an APD.

SPAD is a special APD that is biased beyond the breakdown voltage. In this case, the energy of a single photon is able to trigger the avalanche effect and breaks down the photon diode. Afterwards, the current reaches a significant value, which can be detected easily. This allows a SPAD to detect extremely weak photon energy. The operating mode of SPAD is referred to as Geiger mode [28]. After the breakdown, SPAD lost the ability to detect further photons. In practice, a quenching circuit is necessary to initialize a SPAD to the biased state [24]. The initialization is categorized into a quenching phase and a reset phase. They can be constructed either in passive or in active mode. In the passive mode, SPAD is connected to a specific circuit, which automatically releases the current caused by the activation of SPAD and resets it back to the initial state after a relatively long time. In the active quenching phase, the applied voltage on a SPAD is reduced to keep the current flow caused by the breakdown as low as possible until the current is completely exhausted, so that the occurrence of the after-pulsing, heating of the photodiode, and the crosstalk can be minimized. In the active reset phase, the applied voltage is set back to the desired level to activate the functionality of a SPAD. The insensitive time of SPAD is also referred to as dead time [29]. A specific use of quenching circuit called time-gating is followed by section 3.2. Further knowledge of SPAD regarding **C**omplementary **M**etal-**O**xide **S**emiconductor (CMOS) technology is referred to [28].

The detection principle gives SPAD an ideal photon detection efficiency under low background light. Meanwhile, it makes a distinction of the source of an incident photon impossible. Besides, when background light is present, SPAD reaches saturation more easily than other detectors [30]. Therefore, the technology for handling background light is one of the most important components in a SPAD-based system.

2.1.2 Scanning LiDAR Systems

A scanning LiDAR system uses different approaches to steer the direction of a laser source and obtains distances of all pixels in series. Within a measurement period, the laser source only illuminates a single spot (or a sub-area) in a scene. Therefore, this system typically preserves a high laser intensity reflected on each pixel and is suitable for long-range object detection.

A rotating mechanism is one of the most used approaches in a scanning LiDAR system. By mounting the rotating mechanism with a laser emitter, the working principle of such a system is straightforward: Each sub-area of the scene is scanned in sequence. With a uniform scanning speed, it can provide a wide **F**ield-**O**f-**V**iew (FOV). Thus, this kind of LiDAR system becomes the most popular option for many commercial LiDAR sensors [31]. However, an accurate rotating mechanism occupies a relatively large place and requires high power consumption. Moreover, since a moving component is prone to error, its reliability and durability are critical for an application with a high demand on safety.

To avoid the drawbacks caused by moving components, many alternative solutions are proposed, e.g., **O**ptical **P**hase **A**rray (OPA) [32]. An OPA system typically includes light splitters, phase shifters, and antennas [33]. To control the final projection direction of a beam array, the phase shifters perform different phase delays to the beam array split by a light splitter. An OPA-based system doesn’t have any mechanical moving component, and provides an extremely high scanning speed (e.g., 100 kHz) over a large FOV [31]. As a drawback, thermal distortions have a significant impact on the accuracy of steering angles [34]. Since research and development are still in an early stage, OPA is considered as an emerging solution for LiDAR systems in ADAS and requires more investigations in the future [33].

Micro-**E**lectro-**M**echanical **S**ystems (MEMS) is another competitive solution to OPA. A MEMS system applies a scanning silicon mirror, which is first introduced in the 1980s [35]. The mirror is deployed in a LiDAR sensor front-end and reflects the laser into the detection environment. By applying a stimulus, such as an electrostatic, magnetic, thermal, or piezoelectric signal, the deflection angle of the mirror can be precisely controlled to project the laser in a desired direction. Although the fabrication of MEMS mirrors is not fully standardized, it is based on a very mature semiconductor integrated-circuit industry. Therefore, the MEMS industry is growing rapidly with a progressive standardization [36]. Compared to a rotating LiDAR system, the MEMS technology has a smaller FOV, while offering a higher lifetime and anti-vibration performance. At present, the MEMS LiDAR system shows a number of commercial advantages, particularly in terms of balancing lightness, compactness, and power consumption [34].

Nevertheless, scanning LiDAR systems suffer from the following problems: 1) The received depth information is asynchronous. The degree of asynchrony depends on the scanning pattern, the scanning speed, and the relative motion between the LiDAR system and the object. To cope with this effect precisely, a complex compensation approach is required. 2) In order to keep a reasonable frame rate, the number of measurements on a single point is typically low. This makes scanning LiDAR systems highly rely on the quality of a single measurement, which is challenging to guarantee in a complex detection environment.

2.1.3 Flash LiDAR Systems

A flash LiDAR system has the simplest hardware construction among all systems in the LiDAR family. The laser source in such a system illuminates the whole detection scene at once. After that, the reflected laser photons reach all SPADs within the same measurement observation window. This means each SPAD only receives a small energy portion of the laser source. Since the desired laser photons can be significantly reduced by factors such as the laser attenuation regarding distance, the divergence of the illuminating area, and the backscattering at the object, the number of received laser photons on each SPAD are relatively small. This characteristic limits the maximum detection range of a flash LiDAR system. In order to detect distances reliably, plural measurements are conducted and accumulated to obtain statistical information of the incoming photons.

Flash LiDAR systems offer multiple advantages. First, the measurements on all pixels are obtained with a single laser emission and are therefore highly synchronized. No further calibration algorithm is required for asynchronous effects and moving artifacts. In addition, the system is considered to be more durable and compact to other alternatives, since it has a solid-state construction with minimal controlling units. Because of this reason, the system has a high robustness to vibrations, which is always present in applications of ADAS. As a limitation, a flash LiDAR system typically has a narrow FOV, since a wide FOV will significantly shorten the maximum detection range. Besides, the spatial resolution of the system can only be improved by increasing the alignment density of SPADs and the corresponding circuits, which places high demands on microcircuit processes and increases the cost.

2.1.4 Direct Time-of-Flight Measurement

A SPAD-based direct TOF LiDAR front-end consists of laser emitters, SPAD arrays, and a **T**ime-to-**D**igital **C**onverter (TDC). One measurement of a direct TOF measurement is described as follows: TDC starts with the emission of a laser pulse. The laser pulse travels through the air and is reflected by the object. Afterwards, TDC stops with the detection of an arrived photon. The time duration between a laser emission and a detected photon by a SPAD is defined as T_{TOF}. Since TDC has a resolution of T_{TDC}, T_{TOF} can be represented by a digital timestamp T'_{TOF} with

$$T'_{\mathrm{TOF}} = \lfloor \frac{T_{\mathrm{TOF}}}{T_{\mathrm{TDC}}} \rfloor T_{\mathrm{TDC}} \ . \tag{2.1}$$

Since a direct TOF measurement involves a round trip time, the resulting distance resolution d_{TDC} is given by

$$d_{\mathrm{TDC}} = \frac{\mathrm{c}}{2} T_{\mathrm{TDC}} \ , \tag{2.2}$$

where c represents the speed of light. Normally, a TDC has a picosecond resolution, which is far beyond the precision requirements in this work. Therefore, the following approximation holds:

$$T'_{\mathrm{TOF}} \approx T_{\mathrm{TOF}} \ . \tag{2.3}$$

In particular, T_{TOF} of the photon reflected by a target object is defined as T_{Obj}. Accordingly, the object distance d_{Obj} is calculated by

$$d_{\mathrm{Obj}} = \frac{\mathrm{c}}{2}\, T_{\mathrm{Obj}} \; . \tag{2.4}$$

The direct TOF measurement is simple, efficient, and precise. By specifying the control scheme of SPADs, it is suitable for many different applications, such as autonomous driving, geodesy, aerospace, and medicine.

2.1.5 Light Specification

With an assumption that the incoming light is coherent, the phase, frequency, and amplitude of its electromagnetic field are constant. The intensity of light can be defined by irradiance $I(\lambda)$ with $[I] = \mathrm{W}/\mathrm{m}^2$, which refers to the amount of light radiation with a wavelength of λ obtained per unit area. Accordingly, the irradiance on the detector surface is defined as $I_{\mathrm{Sensor}}(\lambda)$ and is given by the superposition of laser irradiance $I_{\mathrm{L,\,Sensor}}(\lambda)$ and background irradiance $I_{\mathrm{B,\,Sensor}}(\lambda)$, i.e.,

$$I_{\mathrm{Sensor}}(\lambda) = I_{\mathrm{L,\,Sensor}}(\lambda) + I_{\mathrm{B,\,Sensor}}(\lambda) \; . \tag{2.5}$$

In the scope of this work, the light propagation process is represented by a model shown in Figure 2.2. The propagation of a laser starts with a laser emission. By assuming a homogenous irradiation with a point laser source with power P_{L}, the laser irradiance on an object surface at d_{Obj} is calculated by

$$I_{\mathrm{L,\,Obj}}(\lambda) = F_1 \frac{P_{\mathrm{L}}(\lambda)}{d_{\mathrm{Obj}}^2\, F_{\mathrm{FOV}}} \; , \tag{2.6}$$

where F_1 refers to the attenuation factor caused by a dissemination medium between the sensor and the object. F_{FOV} represents a transmission factor determined by the FOV of a LiDAR system. Consequently, $I_{\mathrm{L,\,Obj}}(\lambda)$ is reflected back and the laser irradiance $I_{\mathrm{L,\,Sensor}}(\lambda)$ on the sensor surface is given by

$$I_{\mathrm{L,\,Sensor}}(\lambda) = F_{\mathrm{Det}}\, F_1\, F_{\mathrm{Rft}}\, I_{\mathrm{L,\,Obj}}(\lambda) \; , \tag{2.7}$$

where F_{Rft} refers to the attenuation factor during reflection, including object reflectance and the type of reflection (e.g., diffuse reflection). F_{Det} refers to the attenuation factor at the detector side, including the diameter of receiving aperture, lens transmittance, and the imaging ratio between scene and sensor surface.

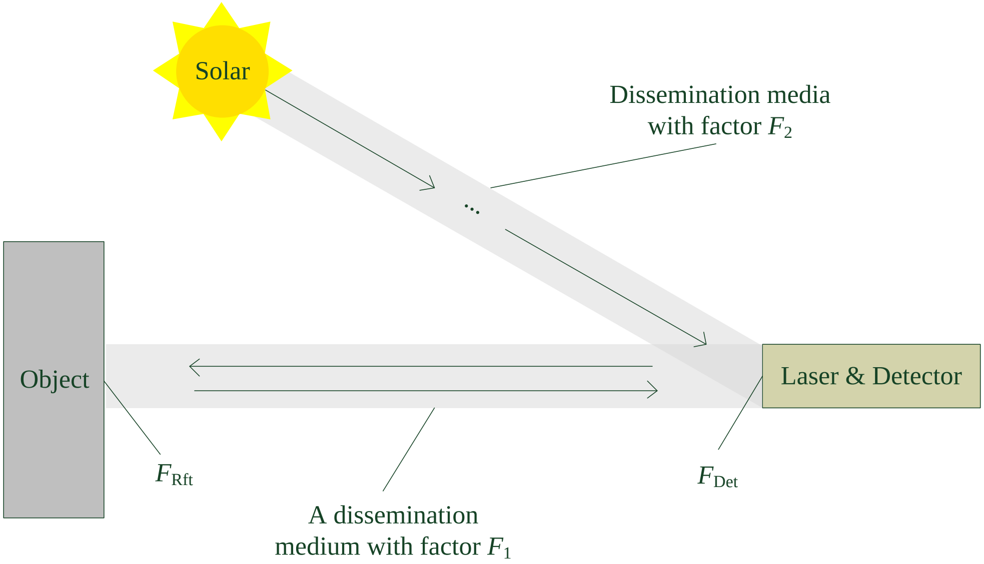


Figure 2.2: Light propagation model in a LiDAR system.

The specification for background light is slightly different from that of a laser source. In the following context, sunlight is used as an example, since it is the most common interference of LiDAR systems in ADAS. Figure 2.3 shows a solar spectrum [37] regarding the international standard ISO 9845-1. Therein, the intensity of sunlight is defined by the spectral irradiance as

$$I_{\mathrm{Spec}} = \frac{I_{\mathrm{S}}(\lambda)}{\lambda} \, , \tag{2.8}$$

where $I_{\mathrm{S}}(\lambda)$ refers to solar irradiance. According to different conditions, background light reaching the sensor surface includes direct irradiance and indirect irradiance (being reflected at least once by other objects). They are summarized by $I_{\mathrm{B,\,Sensor}}(\lambda)$ as

$$I_{\mathrm{B,\,Sensor}}(\lambda) = F_{\mathrm{Det}}\, F_2\, I_{\mathrm{S}}(\lambda) \, , \tag{2.9}$$

where F_2 refers to the attenuation factor of background light caused by corresponding dissemination media shown in Figure 2.2.

In addition to the irradiance, the light intensity can be defined by the number of photons per unit time as well due to its quantum nature. This is of particular concern, since SPAD is a digital detector and has the ability to detect the energy

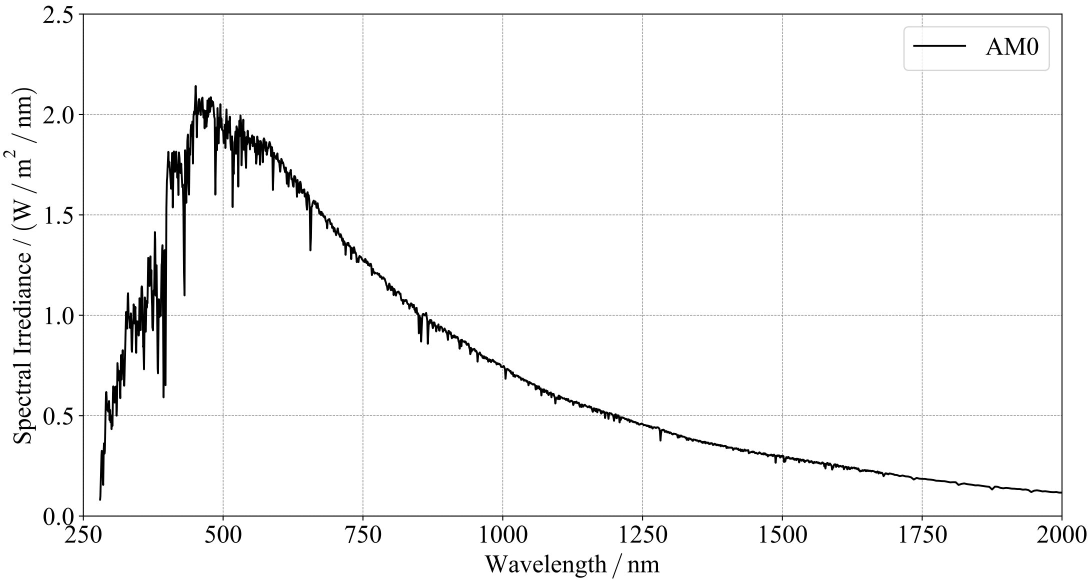


Figure 2.3: Spectral irradiance of sunlight [37]. The standard spectrum for space applications is referred to as AM0. It has an integrated power of 1366.1 W / m^2.

of a single photon. According to Planck, the energy of a photon W_{Ph} is

$$W_{\mathrm{Ph}} = \frac{\mathrm{h\,c}}{\lambda} \tag{2.10}$$

with h $= 6.626 \times 10^{-34}\,\mathrm{W\,s^2}$. Thus, the photon flux ϕ on the sensor surface is given by

$$\phi(\lambda) = \frac{I_{\mathrm{Sensor}}(\lambda)}{W_{\mathrm{Ph}}}\ . \tag{2.11}$$

Afterwards, the corresponding photon detection rate $r(\lambda)$ at a SPAD detector is calculated by considering a range of λ (typically determined by the used optical bandpass filter) using

$$r = \eta_{\mathrm{PDE}}\, F_{\mathrm{Fill}}\, A_{\mathrm{Pixel}} \int \phi(\lambda) d\lambda\ , \tag{2.12}$$

where η_{PDE} is the photon detection efficiency, F_{Fill} refers to the fill factor of a SPAD array, and A_{Pixel} is the area of a pixel. Thus, the laser photon detection rate r_{L} given by

$$r_{\mathrm{L}} = \eta_{\mathrm{PDE}}\, F_{\mathrm{Fill}}\, A_{\mathrm{Pixel}} \int \frac{F_{\mathrm{Det}}\, F_1^2\, F_{\mathrm{Rft}}\, P_{\mathrm{L}}(\lambda)}{W_{\mathrm{Ph}}\, d_{\mathrm{Obj}}^2\, F_{\mathrm{FOV}}}\, d\lambda \tag{2.13}$$

according to (2.6) and (2.7). The background photon detection rate r_B is given by

$$r_\mathrm{B} = \eta_\mathrm{PDE}\, F_\mathrm{Fill}\, A_\mathrm{Pixel} \int \frac{F_\mathrm{Det}\, F_2\, I_\mathrm{S}(\lambda)}{W_\mathrm{Ph}}\, d\lambda \; . \tag{2.14}$$

according to (2.9).

2.2 Mathematical Statistics in LiDAR Systems

In this section, the basics of probability theory and commonly used probability distributions are presented.

2.2.1 Probability Theory

Probability theory describes the occurrence of events as a regular phenomenon through mathematical models. Cumulative distribution function (also mentioned as distribution function) $P(x \leqslant \mathrm{X})$ denotes the probability of taking a value x smaller than or equal to a given value X. If x is a continuous variable, $P(x \leqslant \mathrm{X})$ can be further represented by **P**robability **D**ensity **F**unction (PDF) $p(x)$ as

$$P(x \leqslant \mathrm{X}) = \int_{-\infty}^{\mathrm{X}} p(x) dx \; . \tag{2.15}$$

In the case of discrete events, the function is calculated by summing up all associated values as

$$P(k \leqslant \mathrm{K}) = \sum_{k \leqslant \mathrm{K}} P(k) \; , \tag{2.16}$$

where k represents the discrete event counts and $P(k)$ is a **P**robability **M**ass **F**unction (PMF). In particular, if k is continuous and refers to all values within a range $(\mathrm{X}_1,\ \mathrm{X}_2]$ in x, $P(k)$ can be calculated from $p(x)$ using

$$P(k) = \int_{\mathrm{X}_1}^{\mathrm{X}_2} p(x) dx \; . \tag{2.17}$$

2.2.2 Continuous Probability Distributions

A basic probability model is the uniform distribution $U(x)$

$$U(x) = \begin{cases} \frac{1}{b-a}\,, & a < x < b \\ 0\,, & \text{else} \end{cases} \; . \tag{2.18}$$

The normal distribution is another common mathematical model. PDF of a normal distribution $\mathcal{N}(x \,|\, \mu,\, \sigma^2)$ is defined as

$$\mathcal{N}(x \,|\, \mu,\, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\, \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}} \;, \tag{2.19}$$

where μ and σ denote the expected value and the variance, respectively. According to the central limit theorem [38], many natural processes follow the normal distribution when the observation time tends to infinity.

2.2.3 Discrete Probability Distributions

If outcome y of an event is a Boolean value, where the probability of $y = 1$ is P and the probability of $y = 0$ is $(1 - P)$, it can be described using the Bernoulli distribution $P_{\mathrm{Bernoulli}}$ with

$$P_{\mathrm{Bernoulli}}(y,\, P) = P^y\,(1-P)^{1-y} = \begin{cases} P\,, & y = 1 \\ 1 - P\,, & y = 0 \end{cases} \;. \tag{2.20}$$

The probability of observing k success on n independent Bernoulli trials is known as the binomial distribution $P_{\mathrm{B}}(k, n, P)$, which is calculated as

$$P_{\mathrm{B}}(k,\, n,\, P) = \frac{n!}{k!(n-k)!} P^k (1-P)^{n-k} \;. \tag{2.21}$$

Another common model is the Poisson distribution given by

$$P_{\mathrm{P}}(k,\, \mu) = \frac{\mu^k}{k!}\, \mathrm{e}^{-\mu} \;, \tag{2.22}$$

where μ represents the expected number of random events per unit time (or unit area). In particular, if a binomial distribution has a large n and a small P, the Poisson distribution can be used as an approximation to the binomial distribution, where $\mu = n\,P$. Commonly, when $n \geqslant 20$ and $P \leqslant 0.05$, the difference between these two distributions can be neglected.

2.2.4 Bayes' Theorem

Bayes' theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event [39]. With the given event A and B, the conditional probability is defined by Bayes' theorem as

$$P(A \,|\, B) = \frac{P(B \,|\, A) P(A)}{P(B)} \tag{2.23}$$

with the following definitions:

- $P(A \,|\, B)$: The probability of event A after the occurrence of event B. It is a posterior probability.
- $P(A)$: The probability of event A. It is a prior probability.
- $P(B \,|\, A)$: The probability of event B after the occurrence of event A. It is a posterior probability.
- $P(B)$: The probability of event B. It is a prior probability.

2.2.5 Parameter Estimation Methods

Maximum **L**ikelihood **E**stimation (MLE) is a typical method to estimate the class-conditional probability. With an assumed probability distribution $P(x)$ with a parameter θ and a data sequence $X = \{x_1, ..., x_n\}$, the method estimates θ such that the probability of observing the data sequence X is maximized. In practice, each element x_i is assumed to be independent to each other and follows an identical probability distribution for simplification. In this case, the joint probability of observing X with θ is written as:

$$P(X \,|\, \theta) = \prod_{i=1}^{n} P(x_n \,|\, \theta) \; . \tag{2.24}$$

The probability estimation by θ for a given X is made explicitly by the likelihood function:

$$\mathcal{L}(\theta \,|\, X) = P_\theta(X) \; . \tag{2.25}$$

Afterwards, the maximum probability of observing X with respect to θ is given by

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta} \mathcal{L}(\theta) \; . \tag{2.26}$$

If the likelihood function is continuous and differentiable, the extreme value for θ can be solved by derivation. To prevent the underflow caused by a series of multiplications, the log-likelihood is applied, i.e.,

$$L\mathcal{L}(\theta) = \log P(X \,|\, \theta) = \sum_{x_i}^{n} \log P(x_i \,|\, \theta) \tag{2.27}$$

Maximum **A** **P**osteriori estimation (MAP) is another method used for parameter estimation. Compared to MLE, MAP introduces the prior probability and obtains the maximum probability of X with respect to (2.23). In this case, (2.26) can be rewritten as

$$\hat{\theta}_{\mathrm{MAP}} = \arg\max_{\theta} P(\theta \,|\, X) = \arg\max_{\theta} \frac{P(\theta)P(X \,|\, \theta)}{P(X)} \,. \tag{2.28}$$

Since MAP considers the prior probability distribution $P(\theta)$ in the calculation, it offers advantages over MLE, when the number of samples for observation is limited.

2.3 Machine Learning for Data Processing

The concept of machine learning exists for decades. With the development of computer technology, many machine learning algorithms have been successfully applied and shown powerful ability in different fields. In this section, the principle of machine learning is introduced. Besides, the used machine learning method, i.e., the fully-connected neural network and the naive Bayes classifier are explained in detail.

2.3.1 Concept of Machine Learning

As introduced before, probability theory aims at objective laws behind the event of observation. In the concept of probability and statistics, model building and parameter calculation are rule-based. Thus, there is a clear derivation process between observations and results. Although this process is logical, transparent, and traceable, it often encounters performance bottlenecks in practical applications, if the issues to be addressed are highly complex and contain a large number of internal connections that are difficult to model precisely using probability theory. On the contrary, machine learning is an empirical method based on given datasets. More precisely, it obtains empirical information from given datasets (training data) and focuses on achieving the highest performance in similar datasets (validation data) by optimizing this information. The characteristics of machine learning are summarized as:

- It is a result-oriented method, which creates a mapping from inputs to outputs without establishing mathematical functions explicitly. This results in a highly flexible optimization process with a utilization of all possible connections and has proved to far outperform other rule-based counterparts in some applications. Meanwhile, the result-oriented process with high flexibility often creates unexplainable mapping models from inputs to outputs, which is also mentioned as a black-box model, resulting in low comprehensibility, transparency, and traceability.

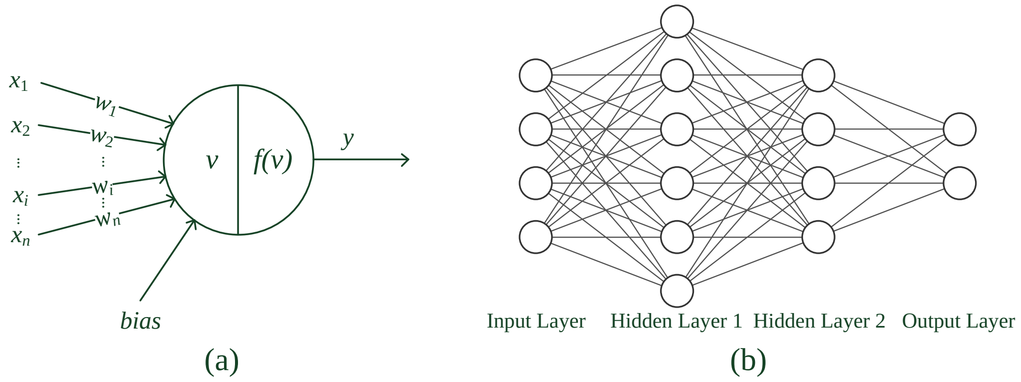
- In machine learning, empirical information (usually referring to parameters in a machine learning model) is obtained automatically based on training datasets. Thus, the performance of machine learning heavily depends on these datasets. Powerful machine learning solutions can be designed only if training datasets with a considerable size and high diversity are available.

- Machine learning models require the availability of an initial model, which will be optimized step-by-step [40]. Although most models are pervasive, they still have their own strengths and weaknesses and tasks they are specialized in. Therefore, the choice of a suitable model and a starting point can critically influence the final performance for a task.

In the following subsections, the working principles of two basic machine learning models used in this work are introduced.

2.3.2 Artificial Neuron and Fully-Connected Neural Network

Artificial neural network (mentioned as neural network in the following context) is defined as "massively parallel interconnected networks of simple elements and their hierarchical organizations which are intended to interact with the objects of the real world in the same way as biological nervous systems do." [41]. Neural network has been proved to be a very general statistical framework for modeling posterior probabilities given a set of samples [42]. It can be constructed to solve both regression and classification tasks. In 1943, the basic model of an artificial neuron, shown in Figure 2.4 (a), has been proposed [43] and is still in use today. The model specifies mathematical computations from an input sequence $\{x_1, ..., x_i, ... x_n\}$ to an output y. First, x_i is weighted by its corresponding weight



(a) (b)

Figure 2.4: Basic principle of neural network. (a) Artificial neuron. (b) FNN with a 4 × 6 × 4 × 2 structure.

value w_i. Afterwards, the intermediate value v is calculated by summing up all weighted x_i and a specified bias to the current neuron, namely

$$v = \sum_{i=1}^{n} w_i \, x_i + bias \; . \tag{2.29}$$

Consequently, the activation function f takes v as input and calculates the output y by

$$y = f(v) \; . \tag{2.30}$$

The commonly used activation functions include **Re**ctified **L**inear **U**nit (ReLU), sigmoid, **tan**gens **h**yperbolicus (tanh), leaky ReLU, **E**xponential **L**inear **U**nit (ELU), and softmax. A neural network is created by connecting plural neurons to each other with a specified structure. Figure 2.4 (b) shows an example of an FNN with the structure of 4 × 6 × 4 × 2. It consists of an input layer, two hidden layers, and an output layer. Neurons in each layer are connected to all neurons of the next layer by weights. The training of an FNN is an error minimization process between a given label y^* and an output y. This process is carried out by updating all weights using the algorithm of error backpropagation. Typically, the error is calculated by a loss function *Loss*. For a regression task, **M**ean **S**quare **E**rror (MSE) given by

$$Loss_{\mathrm{MSE}} = \frac{\sum_{i=1}^{n} (y^* - y)^2}{n} \; . \tag{2.31}$$

or **M**ean **A**bsolute **E**rror (MAE) given by

$$Loss_{\mathrm{MAE}} = \frac{\sum_{i=1}^{n} |y^* - y|}{n} \; . \tag{2.32}$$

can be used. For a classification task, the cross-entropy loss and its variants are often applied. The basic cross-entropy loss is given by

$$Loss_{\mathrm{CE}} = -(y \log(y^*) + (1 - y) \log(1 - y^*)) \; . \tag{2.33}$$

The update of w_i for minimizing $Loss$ is calculated by an optimization strategy according to its partial derivation with a given update rate η using

$$w_i' = w_i - \Delta w_i = w_i - \eta \, \frac{\partial Loss}{\partial w_i} \; , \tag{2.34}$$

where η determines the stride of each update. This optimization strategy is also mentioned as gradient descent. Many variants based on the gradient descent are proposed to improve the learning speed and the prediction performance of a neural network. Therein, **Ada**ptive **m**oment estimation (Adam) is one of the most effective optimization strategies. The detailed implementation of Adam is referred to [44].

2.3.3 Naive Bayes Classifier

NBC is a classification method based on (2.23). Frequently, event B refers to class c and event A refers to an input data sequence X in machine learning. Furthermore, the target to be estimated is the posterior probability $P(c \,|\, X)$. Therefore, Bayes' theorem can be rewritten as

$$P(c \,|\, X) = \frac{P(X \,|\, c) P(c)}{P(X)} \; . \tag{2.35}$$

NBC predicts $P(c \,|\, X)$ by estimating $P(X \,|\, c)$, $P(c)$, and $P(X)$ based on a given dataset. Apparently, the estimation of $P(X \,|\, c)$ suffers from attribute explosion, since it represents a joint probability among all $x \in X$, which contains a massive number of combinations. To cope with this problem, NBC assumes that for given classes c, all $x \in X$ are independent to each other. Therefore, the following equation holds:

$$P(X \,|\, c) = P(x_1, \, x_2, \, ..., \, x_n \,|\, c) = \prod_{i=1}^{n} P(x_i \,|\, c) \; . \tag{2.36}$$

Table 2.1: Confusion Matrix.

	Prediction	
Truth	Positive	Negative
Positive	**T**rue **P**ositive (TP)	**F**alse **N**egative (FN)
Negative	**F**alse **P**ositive (FP)	**T**rue **N**egative (TN)

$P(c)$ is simply estimated by the portion of a corresponding class c in the dataset using

$$P(c) = \frac{N_c}{N_\mathrm{S}} \, , \tag{2.37}$$

where N_c refers to the number of samples in class c and N_S refers to the number of samples in the given dataset. $P(X)$ is calculated as

$$P(X) = \sum_c P(X \mid c) P(c) \, . \tag{2.38}$$

Since $P(X)$ is the same for all possible c, it can be neglected in the calculation. Therefore, NBC is represented by

$$f_\mathrm{NBC} = \arg\max_c (P(X \mid c) \, P(c)) \, . \tag{2.39}$$

It is a generative model. In practice, NBC offers several advantages: 1) If an application has a high demand on real-time performance, the application of a **L**ook**U**p-**T**able (LUT) is possible, since $P(X \mid c)$ and $P(c)$ can be pre-calculated on a given training dataset. 2) If the data to be observed continuously change over time, the concept "lazy learning" can be carried out, which only trains the model when an execution request is received; 3) The application of incremental learning is possible, since new parameters can be easily calculated using old parameters with new data.

2.3.4 Confusion Matrix

The confusion matrix is an important performance measurement for machine learning algorithms. Table 2.1 shows a basic layout of such a matrix for the case of binary classification. Therein, samples in a dataset are classified into positive and negative ones. The four terms in the table are specified as:

- TP: The number of correct predictions, where the predictions are positive.
- FP: The number of false predictions, where the predictions are positive.
- TN: The number of correct predictions, where the predictions are negative.
- FN: The number of false predictions, where the predictions are negative.

The performance of a machine learning algorithm can be assessed by observing these four terms and their relations to each other. In practice, the confusion matrix can be applied to a multi-classification problem as well by creating individual confusion matrices for every two classes.

2.4 Monte-Carlo-Method

Monte-**C**arlo-**M**ethod (MCM) refers to a class of stochastic methods. In some stochastic problems, there are a large number of basic events. The probabilities of these events are given and a joint probability among them is hidden and of interest. Due to exponential correlation properties among the events, it is impractical to obtain the joint probability directly using probability theory. In this case, MCM can be applied. First, a number of samples are generated based on the probabilities of basic events. Afterwards, MCM makes observations and performs statistical modeling on these samples to derive the approximate solution of the joint probability.

For example, with a given probability mass function $P(i)$, where $i \in \{1, ..., n\}$, the task is to determine the probability of a specific i having the maximum occurrences in N_T trials. If $n \gg 1$, probabilistic derivations are extremely challenging, since a large number of arrangements and combinations exist, which are difficult to categorize and simplify. By applying the concept of MCM, one simulation of N_T random trials following $P(i)$ refers to a sample. By accumulating sufficient samples with simulations, the task can be approximated by the statistics of i having the maximum counts among the samples. Apparently, as the number of samples increases, the approximation will gradually converge to the true probability. This is one generic property of MCM.

Chapter 3

SPAD-Based LiDAR Data Processing

A SPAD-based direct TOF LiDAR system can be decomposed into three parts, i.e., hardware, software (processing algorithms), and data forms, as shown in Figure 3.1. Therein, the hardware component includes a LiDAR front-end, a data pre-processing unit, and a data processing unit. The LiDAR front-end is responsible for laser emission, photon detection, and LiDAR measurement data generation. The data pre-processing unit normally includes a data pre-processing algorithm, which makes distance determination from low-level LiDAR data. The data processing unit is installed with powerful processing algorithms for high-level properties in point clouds. These three elements usually have their own encapsulations and collaborate with each other by transmitting processed data.

In the following sections, important research and development in the field of SPAD-based LiDAR data processing are reviewed. First, a LiDAR system structure is proposed. Afterwards, data forms in the LiDAR system and their corresponding processing approaches are introduced. Finally, challenges are summarized and research gaps are proposed.

3.1 LiDAR System Structure

The main goals of ADAS are analyzing environmental information collected through sensors and ultimately obtaining comprehensible high-level properties and decisions. To illustrate different stages of development in data processing, a system

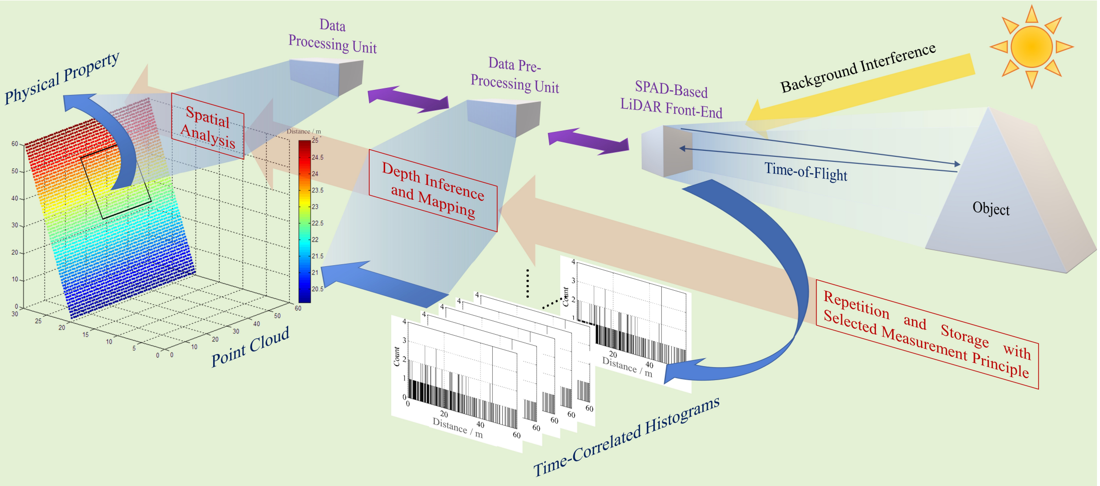


Figure 3.1: LiDAR system and data processing workflow [45]. Therein, purple components refer to different level of hardware, red components refer to software for data processing, and blue components refer to different data forms.

structure is necessary. A generic architecture for simulation of ADAS sensors was proposed in [46]. Based on this, a LiDAR-specific system structure was introduced for object detection in [47]. However, the former focuses on commonality among multiple sensors for ADAS, while the latter places more emphasis on object detection by LiDAR sensors. Neither involves different stages in LiDAR data processing. Therefore, a LiDAR system model focused on data transfer in Figure 3.2 is illustrated to study the characteristics of different data forms and the hierarchy of corresponding analysis approaches. The LiDAR structure is divided into three parts: a LiDAR front-end, a data pre-processing unit, and a data processing unit. The LiDAR front-end includes all necessary hardware units for the execution and delivery of LiDAR measurements. The data pre-processing unit typically refers to an embedded system, which is responsible for data storage and early-stage data analysis. The data processing unit refers to an advanced calculation unit. It can be seen that different data processing procedures involve different data forms as inputs, including timestamp, TC-Hist, and point cloud.

In the following sections, the data processing workflow is divided into three parts regarding these data forms. Each section starts with the formation of the corresponding data form. Afterwards, research related to the data form is presented. Finally, existing challenges are summarized and research gaps are proposed. For consistency and clarity purposes, four important time durations

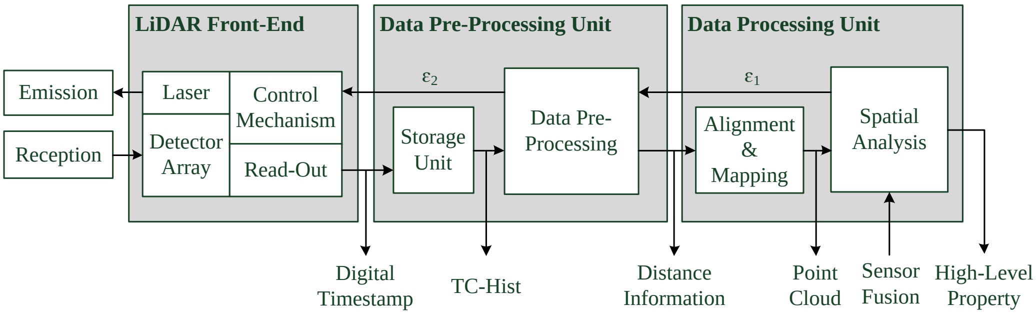


Figure 3.2: LiDAR system structure and data transformation. ε_1 and ε_2 refer to the control signals from master devices.

are specified as follows:

- A measurement cycle refers to the time duration between two consecutive laser pulses, which includes an observation window, a custom function window, data readout, initialization, and a safety margin.
- A TC-Hist generation period refers to the time duration for the generation of a TC-Hist, which includes a plurality of measurement cycles, the time of TC-Hist formation, and a safety margin.
- A data generation frame refers to the time duration for the generation of TC-Hists from all pixels.
- A data processing frame refers to the time duration of distance determination for all pixels, which includes a data generation frame and data processing time.

3.2 Digital Timestamp

As described in 2.1.4, a digital timestamp refers to T_{TOF} with the resolution of T_{TDC}. The timestamp is the first accessible data form in the LiDAR system. Although SPAD is one of the ideal detectors in the direct TOF measurement, it is prone to background light due to its high sensitivity. Thus, one of the greatest obstructions in a SPAD-based LiDAR system is background light [17]. Considerable timestamps with incorrect information are generated during a measurement process, which degrades the system reliability and limits the applicable range in terms of light conditions. Moreover, as the dead time is unavoidable, SPAD suffers from a saturation effect under high background light [30].

3.2.1 Optical Bandpass Filter

Since laser sources used in LiDAR systems generate laser pulses with a specific wavelength (typically a value between 800 nm and 1550 nm), while solar irradiance includes the full wavelength range, an optical bandpass filter is often applied as a universal solution to remove the irradiance at unwanted wavelengths [48]. In practice, due to the fluctuations in the wavelength of a Fabry-Perot laser source, the allowed bandwidth is typically larger than 40 nm [49]. The use of more complex laser sources can further reduce the bandwidth but increases product price. Nevertheless, since solar irradiance is highly significant in the wavelength range used in LiDAR systems [37, 50], Süss et al. pointed out that the remaining background light can still saturate a SPAD, even when a 50 nm optical bandpass filter is used [51].

3.2.2 Coincidence Counting

Coincidence counting, which originates from [25], is another effective measurement approach to further reduce false timestamps. This approach can be integrated into the read-out unit of a sensor front-end, as shown in Figure 3.2. It involves several SPADs in one pixel. These SPADs work in parallel. During the initialization, a minimum photon number N_{Coin} and a coincidence time interval t_{Coin} are defined. The coincidence counting starts from the first photon-triggered event on a SPAD at time t. Afterwards, the event is retained for t_{Coin}. If at least $N_{\text{Coin}} - 1$ events are further triggered within $(t,\, t + t_{\text{Coin}})$, t is outputted and recorded as a timestamp. Otherwise, the event is released and the activated SPAD is put back into operation after a reboot time. Obviously, N_{Coin} and t_{Coin} work as a threshold. It has a strong inhibitory effect on photon rates below the coincidence threshold, resulting in a significant reduction in the generation rate of timestamps. In this case, the saturation effect of a single SPAD is mitigated and SNR can be improved [51]. LiDAR systems that use this method can operate under strong background light [17, 52, 53, 54]. Furthermore, Beer et al. presented an adaptive coincidence counting method, which enables a runtime adjustment possibility on N_{Coin} and t_{Coin} [14]. Nevertheless, the selection of a coincidence threshold is challenging. An ideal threshold must be exactly between r_{B} and the overall photon detection rate r_{LB}, where

$$r_{\mathrm{LB}} = r_{\mathrm{B}} + r_{\mathrm{L}}. \tag{3.1}$$

A high threshold suppresses the desired laser photons as well, while a low threshold barely reduces the negative effect from background light. However, r_{L} strongly depends on object reflection, object distance, and weather conditions, which cannot be pre-defined in real-time. Although a large r_{L} improves the effective range of the coincidence counting, it is impractical due to restrictions on eye safety [55]. Furthermore, SPADs in the coincidence counting are triggered and rebooted more frequently than it is in FPC. This results in some degree of statistical distortion e.g., missed counts and the after-pulsing effect. While shortening the rebooting time reduces missed count rate, it exacerbates the after-pulsing effect [14]. Besides, the coincidence counting requires a set of SPADs implemented in a single pixel. This will lead to complex readout circuits and sparse distribution of measurement points in space.

3.2.3 Time-Gating

To further reduce timestamps with incorrect information, a technique called time-gating was introduced [24]. In a direct TOF LiDAR system, the emitted laser is a short-pulsed beam with a pulse width of T_{P}. Since the photon detection rate is a superposition of r_{L} and r_{B}, the total photon detection rate regarding subsection 2.1.5 is determined by

$$r(t) = r_{\mathrm{B}} + r_{\mathrm{L}}(\varepsilon(t - T_{\mathrm{Obj}}) - \varepsilon(t - (T_{\mathrm{Obj}} + T_{\mathrm{P}}))) \tag{3.2}$$

with assumptions that r_{B} and r_{L} are constant. Wherein, step function ε is defined as

$$\varepsilon(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geqslant 0 \end{cases} . \tag{3.3}$$

Obviously, the desired signal is present only within $[T_{\mathrm{Obj}}, T_{\mathrm{Obj}} + T_{\mathrm{P}}]$. Since T_{P} is much shorter than an observation window, a SPAD is exposed to background photons most of the time during a measurement. Time-gating improves this situation by shortening the sensitive period of a SPAD. In an ideal case, by activating a SPAD right before T_{Obj}, the risk of a SPAD being triggered prematurely by background photons can be minimized. Kostamovaara et al. demonstrated the effectiveness of this technique in their experiment by applying different gate windows [49]. The technique can be implemented by sending gate schemes to the quenching circuits through ε_2 in Figure 3.2 to control the activation and rebooting of SPADs. The design of a gate scheme is typically twofold: (1) specifying a number of gate windows of interest (prior-knowledge) and (2) measuring within the specified gate windows. For example, Apple Inc. designed a gate scheme to enable coarse and fine scanning to improve memory occupancy and measurement quality [56]. The gate scheme divides a measurement into a coarse and a fine phase. In the coarse measurement phase, SPAD is activated at the beginning of a measurement cycle. Afterwards, a coarse position of an object is estimated using the timestamps acquired in this phase. In the fine measurement phase, SPADs are activated slightly before the determined coarse position to shorten the invalid exposure time of SPADs. Compared to a full-range detection, the memory occupancy using a gate scheme is typically smaller, since only a sub-range detection result is stored.

Although the benefit of the time-gating technique is significant, its limitation is quite obvious: A well-defined gate scheme is inevitable to ensure its effectiveness. It is very challenging to determine such a gate scheme. In [56], the effort of inferring a proper gate window is several times greater than directly inferring an object distance. This results in a significant reduction in frame rate. Moreover, the measurements of the technique highly rely on the prior-knowledge. If the specified observation time window is incorrect, the following measurements fail as well.

3.3 Time-Correlated Histogram

Since the negative effects caused by background light cannot be completely eliminated, the reliability of a single timestamp is low. To cope with this problem, **T**ime-**C**orrelated **S**ingle-**P**hoton **C**ounting (TCSPC) technique [57, 58] is typically used. Nowadays, it can be divided into a synchronous and an asynchronous mode, which correspond to a clock-driven architecture and an event-driven architecture, respectively [59]. Both modes collect plural consecutive timestamps and form a statistical TC-Hist as their output. In such a TC-Hist, discrete timestamps are represented by monospaced adjacent bins on the x-axis. The bin width corresponds to T_{TDC}. Timestamps with the same value are accumulated in the same bin and the y-axis shows the number of timestamps within each bin. In contrast to a timestamp, A TC-Hist contains not only distance information, but also noise characteristics, sensor characteristics, and photon statistics. The variety of information in a TC-Hist makes a reliable and precise analysis possible. Approaches were proposed for TC-Hist and were implemented at the stage of data pre-processing in Figure 3.2.

3.3.1 Maximum Detection and Digital Filters

According to (3.2), r_{L} and r_{B} are superposed at the position of an object in a TC-Hist. Thus, distance information corresponds to a local maximum. Therefore, processing algorithms on TC-Hist revolve around a maximum detection. The simplest solution is global maximum detection. It directly compares the count value of each individual bin and does not consider the influence of noise and a laser waveform. Thus, it is only applied in a noiseless detection scenario with sufficient measurements in a TC-Hist. However, only limited measurements can be carried out due to the timeliness of measurements. Moreover, background noise is often present in ADAS. As a solution, digital filtering is applied. It consists of three steps:

1) The desired distance information of an object is spread over multiple bins. Therefore, a mean filter is often used to highlight high-density bin groups, so that the global maximum detection can be applied on a TC-Hist with a sparse photon distribution as well. In the case of sufficient measurement data in a TC-Hist, this step can usually be skipped.

2) To suppress the significance of background photon counts, noise removal is performed by an estimation of the noise level in a TC-Hist. A common way is background subtraction. It is applied before the global maximum detection.

3) In order to improve the precision of distance information, some modern processes extract a bin group around the global maximum and apply an

interpolation algorithm. After that, the second filtering is implemented to obtain the desired information. For example, a matched-filter aims at the bin group with the highest SNR [60], a center-of-mass algorithm allocates the position of the center of gravity for each bin group [61], and a Gaussian curve fitting matches the given bin group to a Gaussian distribution [62]. However, this step brings no benefit if the global maximum obtained from previous steps deviates significantly from ground truth.

The filtering process enables a comprehensive analysis due to the use of complex filters on a TC-Hist. However, it requires a complete TC-Hist as input and results in relatively long processing time.

3.3.2 Other Approaches

In addition to the widely accepted solutions, research was conducted on some specific problems and corresponding approaches were proposed. In [61], a likelihood ratio test is applied for the estimation of the laser pulse shape based on MLE. Although a superior result was reported over other methods, the experiment in this work was only carried out within 2 m. To the best of our knowledge, the proposed method calculates a joint probability of each bin. Therefore, an exorbitant computational cost is required if the detection range increases. To simplify the computation of MLE, an approximation, i.e., the log-matched filter is applied in [63]. However, for single-pixel estimation, the effectiveness of the MLE-based methods strongly depends on SNR and the number of samples [64]. When SNR is low and the number of samples is small, they give inaccurate estimations.

A determination of distance using the global maximum is error-prone, especially in the case of few measurements accumulated in a TC-Hist and high background light, since a group of concentrated incorrect measurements caused by coincidence can create a maximum as well. Thus, the first and second global maxima from a TC-Hist were observed in [17]. It can be seen from the experiment that the first and second global maxima show different details of the detection environment and there is a certain degree of complementarity. However, maxima selection based on its amplitude alone cannot avoid the negative effects from strong background light. Moreover, an analysis of maxima selection and subsequent data processing are not covered in their work.

Except for background interference, LiDAR system faces another critical problem: excess data amount. High depth resolution means that a TC-Hist contains a massive number of bins. In combination with the number of pixels, a LiDAR system can produce several gigabytes of data in a few seconds. This results in a large memory occupation and a high real-time throughput requirement for data transmission. Zhang et al. proposed a partial histogram readout method to reduce memory occupation of a TC-Hist [19]. The method includes a current and a previous TC-Hist generation period. The current period utilizes a smaller depth resolution than the previous period. The previous period gives a subpart of the complete histogram as a proposal by global maximum detection. The current period only records measurements in this subpart. The process is repeated until the required depth resolution is reached. Since the data acquired in each intermediate TC-Hist generation period cannot be reused, multiple generation periods are required to output one partial histogram, resulting in a degradation of the data frame. Moreover, the premise of this method is low background light, since the method has low noise tolerance due to the effect of histogram folding.

The analysis of system-induced distortions is another topic of concern. The pile-up effect, which is an aberration that existed in a SPAD-based system, is studied and the corresponding post-processing approach was proposed on the synchronous TCSPC process [65]. Besides, Rapp et al. focused their research on the asynchronous TCSPC process and proposed two Markov chain-based methods for histogram modeling and detector dead time compensation [66]. In their work, the positive effect of the presence of dead time was reported as well.

Nevertheless, most existing approaches using TC-Hist for distance determination focus on improving precision while overlooking the robustness of algorithms to interference. The reliability of these approaches is significantly reduced when background light is strong or the number of measurements per TC-Hist is limited. Therefore, it is necessary to develop an approach that focuses on the robustness of distance prediction and the processing frame rate.

3.4 Point Cloud

A TC-Hist only represents measurement information on one pixel, which is not sufficient to restore an entire detection environment. Therefore, a point cloud is created by accumulation, alignment, and mapping of distance information from all pixels. The point cloud is a three-dimensional data form, which contains rich spatial information among pixels. Processing approaches on point cloud take place in the data processing unit in Figure 3.2 and can be categorized into 1) data compression, 2) depth information optimization, and 3) high-level property inference.

3.4.1 Data Compression

In general, LiDAR data generation and processing are conducted in different hardware units. Since a raw point cloud is typically large, a high transmission throughput is required. Hornung et al. presented their Octomap, which is an open-source framework for three-dimensional mapping [67]. The framework is based on octrees and uses a probabilistic occupancy estimation and synthesizes probabilistic representation, modeling of unmapped areas, and efficiency with respect to runtime performance and memory usage. Although it acts more as a mapping structure than a compression algorithm, it substantially improves memory efficiency. Golla et al. argued that the real-time performance of a compression method is necessary for robotics [68]. They introduced a real-time compression method supporting incrementally acquired data and local decompression. The compression method can compress 1.5 million points per second.

3.4.2 Depth Information Optimization

Due to the characteristics of LiDAR measurements, point cloud inherits some limitations, such as low spatial resolution and sparse desired laser photons caused by low reflectivity or high background intensity. These limitations can significantly degrade the performance of point cloud-based processing methods [69]. Altmann et al. proposed a new Bayesian reflectivity and depth model when an incoming photon flux is very low [70]. The model establishes Markovian dependencies according to spatial correlations with neighboring pixels. Subsequently, an adaptive Markov chain Monte Carlo algorithm is applied to compute Bayesian estimates of interest and perform Bayesian inference. Block-matching and three-dimensional filtering (BM3D in short) is an image denoising strategy based on sparse three-dimensional transform-domain collaborative filtering [71]. The strategy first builds blocks by searching for proximity pixels with similar features in an image, which is referred to as grouping. After that, a collaborative filtering by shrinkage in the transform domain is applied to remove noise. Finally, the image is recovered by reverse transformation. The strategy was introduced to LiDAR data in [63]. In addition, Kirmani et al. presented a technique to obtain single-pixel distance information based on the spatial correlations of neighboring pixels [72]. This work was extended in [63] and a better **R**oot **M**ean **S**quare **E**rror (RMSE) was reported in terms of depth recovery compared to the case using BM3D and a log-matched filter. Most recently, a pre-processing filtering step on noisy point clouds was introduced in [73], where a guided three-dimensional point cloud filter and an iterative guidance normal filter were designed for producing high-quality LiDAR point clouds.

Except for solid rule-based approaches, machine learning has achieved great success in depth information optimization as well. In [74], an up-sampling method based on a **C**onvolutional **N**eural **N**etwork (CNN) was presented on LiDAR point clouds to deal with limited spatial resolution. Cheng et al. introduced a feedback-loop to automatically clean noise data in a point cloud [69]. In addition, limitations of LiDAR point clouds can be compensated by depth completion, which includes not only optimization, but also reconstruction of depth information by analyzing different sensor data jointly. This approach is also known as sensor fusion. The most common example is the fusion between LiDAR point clouds and images. In contrast to a LiDAR point cloud, an image from a camera has much higher pixel resolutions, while the depth estimation from it alone is ambiguous and unreliable [75]. Therefore, an image has good complementary properties to a point cloud. The so-called “depth images” generated from them often inherits high resolution from the image and accurate depth information from LiDAR data. For example, Ma et al. fused 100 correlated LiDAR points as guidance of depth information to an encoder-decoder-based CNN [75]. As reported, depth estimation with their fusion method significantly outperforms that of monocular images alone. On the contrary, Ferstl et al. used an anisotropic diffusion tensor, calculated from a high-resolution intensity image, to guide the upsampling of a LiDAR point cloud [76]. However, sensor fusion applications assume that a LiDAR point cloud is noise-free and used as ground truth. Manual annotation is required if a point cloud is noisy, which is barely possible in a dynamic outdoor environment. To cope with this, a semi-supervised learning method using a deep residual network with long-skip connections was introduced in [77]. In this method, a CNN-based unsupervised image alignment and a supervised learning using LiDAR point clouds as labels were seamlessly combined in a loss function. In [69], based on a late-fusion strategy, an unsupervised end-to-end learning network "LidarStereoNet" was proposed. The network correlates LiDAR and stereo image data automatically without manual annotation.

3.4.3 High-Level Property Inference

By applying task-specific algorithms, high-level properties, e.g., object position, size, movement, and category can be inferred from spatial information in a point cloud. Gargoum et al. divided possible features for pattern recognition in LiDAR data into on-road information, roadside information, and in conducting assessment of highways [78]. According to [79], property inference involved in ADAS is categorized into 1) vehicle and pedestrian detection, 2) driver's state, behavior, and identification, 3) traffic sign recognition, and 4) road detection and scene understanding. The following review in this subsection covers mainly 1) and 4).

CNN shows a powerful ability in property inferences for the images [80]. Since both images and point clouds represent digitalized detection environments and are highly similar, CNN is applied to the point cloud for property inference as well. Li et al. generated three-dimensional bounding boxes from LiDAR data using a two-dimensional CNN [81]. In their further work, while retaining the original idea of a two-dimensional CNN, a three-dimensional CNN was proposed and achieved a clear improvement in average precision and orientation similarity on KITTI benchmark [82]. Similar to the depth information optimization, the complementarity between point cloud and image has inspired researchers to investigate joint solutions between them and received satisfactory results. In [83], a point cloud was firstly up-sampled and then was aligned to a stereo image. A deep CNN was used to the reconfigured data for object classification. Zhao et al. pointed out that classical feature extraction methods in a CNN, e.g., a sliding window [84], propose a large number of unnecessary candidate proposals from an image and have a long extraction time [85]. To accelerate the extraction, they employed the point cloud to generate region proposals. The proposals were further mapped to stereo image for the determination of region-of-interest. This method reduces the total calculation time to a tenth of the sliding window. In [86], a method named “multi-view three-dimensional object detection network (MV3D in short)” was proposed, where a point cloud is converted to a front-view image and a bird-view image. Afterwards, the front-view image, the bird-view image, and an additionally-captured stereo image are concurrently processed by individual convolutional layers. A deep fusion network is applied to jointly predict object class and correlated three-dimensional bounding boxes according to the proposals generated by the three images. As reported, MV3D obtains more than 25 % average precision improvements for three-dimensional localization and detection compared to other contemporaneous algorithms.

PointNet [87] is one of the pioneers in processing point clouds based on recursive neural networks. Instead of transforming a point cloud into three-dimensional voxel grids or images, the method directly uses point data as inputs and outputs classification results or single-point-based segmentation. In order to improve the ability to recognize fine-grained patterns and generalizability to complex scenes, an upgraded version PointNet++ was presented [88], which applies PointNet recursively on a nested partitioning of the input point set. Frustum-PointNet [89] is the integrator of all PointNet versions, which generates a frustum point cloud by combining LiDAR point clouds and images for a three-dimensional bounding box estimation. Compared with the previous versions, Frustum-PointNet achieves not only a better mean average precision but also a higher inference speed.

Despite the large scale of a network, a modern deep learning algorithm is designed generally only for one specific task. This limits the reusability and universality of a designed algorithm. Through the concept of transfer learning [90, 91], the encoder-decoder network structure in [92], which shares an encoder while designing individual decoders for different tasks, can reduce repetitive operations arising from simple combinations of algorithms. However, as reported in [69], even a small five-layer network contains 25,000 parameters. Ren et al. argued that real-time performance in autonomous driving is as important as inference accuracy and proposed their sparse blocks networks (SBNet in short) to speed up the training and the calculation in CNN [93].

Although the point cloud-based methods are able to resist a small number of noise by exploiting potential relationships between different measurement points, the noise is not handled adequately. Zhao et al. mentioned that the noise in point cloud constrains the performance of their proposed method in the detection of tiny objects at moderator hard levels of difficulty [85]. Moreover, although it is not explicitly defined, the noise of point clouds in many studies refers to small deviations of measured depth information relative to ground truth. Another type of noise, namely false detection caused by background light, is rarely discussed. It can easily occur in harsh measurement environments and is crucial to high-level property inference. It is intractable to minimize its negative effects by removing false detections with thresholding. The introduction of noise-related features from low-level LiDAR data (e.g., TC-Hists) may be an effective solution to further improve the quality of the point cloud.

3.5 Challenges

Although the development of LiDAR technology is driving the evolution of ADAS, it still faces challenges from different sectors.

3.5.1 Challenges on LiDAR Front-End

A large number of possible components and configurable parameters make the choice of LiDAR front-end quite versatile. This results in the difference in data characteristics and application scenarios. However, there is a lack of a widely-accepted standard to evaluate and guide these variants uniformly. In addition, data optimization approaches in this stage are mainly employed to mitigate the negative effects of background noise and hardware-induced signal distortion. Since their working principles are closely related to physical laws and hardware parameters, a new concept always requires a dedicated hardware implementation, resulting in a long research cycle and low flexibility in terms of concept extension and real-time adjustment. Nevertheless, relying solely on upgrading LiDAR hardware to obtain optimal data is prone to bottlenecks. Meanwhile, a balance between performance and cost is hard to maintain. Compared to hardware, adjustments and adaptations on software are much easier and, in some cases, more efficient. Thus, a holistic approach combining hardware adjustment strategies and software processing algorithms may open up new possibilities.

3.5.2 Challenges of Data Pre-Processing

The data pre-processing approaches are dedicated to removing interference and to determining distance information in TC-Hist. These approaches must be performed on embedded systems as much as possible to mitigate data transmission overload. As a result, their performance is often limited by resources available on embedded systems. Most pre-processing approaches simplify the diversity of information in TC-Hist, i.e., except for distance, other information, such as sensor settings, noise features, and statistical characteristics, are not forwarded to the next processing stage. This turns a sensor front-end and pre-processing procedure together into a black box [26]. The subsequent processing can be significantly misled and thus becomes meaningless, if the quality of distances provided by pre-processing is low. A trade-off must be made on information diversity, processing speed, memory requirement, accuracy, precision, and robustness to maximize the ability of a pre-processing in different scenarios. Besides, a LiDAR system generates a large amount of data, which places a great deal of strain on available data storage and transmission in an embedded system. However, reducing LiDAR data requires a considerable number of additional operations. No research is found to maintain the processing frame rate while reducing the amount of stored data.

3.5.3 Machine Learning-Related Challenges

Recent evidence shows that a simple machine learning network suffers from relatively low accuracy [94] and its performance can be improved by adding additional layers [95]. Thus, in order to achieve a satisfactory result, the scale of a machine learning-based approach in a LiDAR system becomes larger and larger. However, it dramatically increases the number of parameters and the complexity of the approach as well, which makes training of a network and real-time inference challenging. Moreover, the traceability and interpretability of a very deep network are low. Although the concept of ablation study is introduced to verify the necessity of each unit, the exact functionality of each part remains unclear. This makes a network less transparent and does not allow for an intuitive verification of its uniqueness and plausibility.

Another great challenge is training data. In order to prevent a deep network from overfitting, a well-labeled dataset with high diversity is required. However, such a dataset is difficult to obtain in ADAS, since direct road testing to collect real measurements with labels is inefficient and expensive. Although a well-established LiDAR data benchmark KITTI [23] is widely used by point cloud-based methods, low-level data and detailed sensor information are inaccessible. Besides, it is generated by Velodyne LiDAR systems. However, there is a variety of different LiDAR front-ends in addition to the Velodyne systems. Point clouds they generate can vary widely in terms of sparsity, precision, and accuracy. Algorithms that have been proven effective on the KITTI dataset alone may not be universal and generalizable to other datasets. Typically, further tuning and verification are required.

In order to train a deep network adequately, raw data measured at different time points are typically aggregated from distributed measurement front-ends to a central processing module. To avoid transmission overload while to ensure the availability of raw data at all times, most machine learning methods are trained offline. Furthermore, some raw data are privacy protected and only allow to be kept locally, which makes the transmission of raw data limited. Emerging concepts, such as distributed learning and edge artificial intelligence, were proposed to separate a deep network into several parts and train them locally. In theory, these concepts not only reduce the amount of data to be transmitted, but also abstract raw data through pre-processing and feature extraction, so that the "encrypted" data can be transmitted instead of the original private data. Meanwhile, since the processing takes place near sensor front-ends, good information timeliness can be achieved. Nevertheless, these concepts are in the early stage of development. More efforts need to be invested to verify the feasibility and effectiveness of these concepts in different applications.

3.5.4 ADAS-Related Challenges

In ADAS, detection scenarios are highly dynamic and the measured data are directly related to safety. Therefore, the timeliness of a data processing method is salient. In this case, obtaining information within a defined margin of error in real-time is more valuable than striving for maximum precision in exchange for timing performance. Meanwhile, the method must have an adaptive ability to different scenarios and high robustness to counteract both environment-induced and system-induced interference. This could be achieved by a closed-loop system.

However, the feedback signals ε_1 and ε_2 shown in Figure 3.2 are barely enabled in a state-of-the-art LiDAR system. Besides, for a machine learning-based system, the change of parameters for measurements regarding feedback signals means a large dynamic range of inputs, which complicates the learning process of an algorithm and may cause unpredictable results. A possible approach is to design a physics-based feedback rule and to normalize input data before being fed into a black-box model, thus making a dynamic process relatively transparent and predictable.

3.6 Research Gap

High distance precision provided by a LiDAR system is a key to safety. On the one hand, a number of LiDAR hardware balancing cost and performance have been manufactured in the market. They cover a wide range of measurement scenarios and provide attractive features for ADAS applications. On the other hand, point cloud-based approaches are evolving and massive data processing methods are emerging regarding depth completion, high-level property inference, and sensor fusion. Those approaches have demonstrated recognition capabilities far beyond the eye on ideal sensor data. However, there is a gap between LiDAR hardware and high-level data processing. No research was found, which has the ability to combine both fields effectively, and can proceed with the development of LiDAR systems in ADAS. In particular, the following research aspects need to be studied:

- According to subsection 2.1.1, subsection 2.1.4, and subsection 3.2, the greatest interference in a SPAD-based LiDAR system is background light, especially for an outdoor detection scenario. Although various solutions against background light were proposed, they reduce the frame rate of a LiDAR system significantly. Considering an autonomous driving scenario, where high background light is always present and a high frame rate is required, reliable detection is challenging.
- Although machine learning-based approaches show great ability in the field of data processing, they belong to the black-box model and suffer from low transparency and interpretability. This makes prediction behaviors of the

approaches in unknown situation unpredictable and untraceable. As it relates to safety, the use of a black-box model is particularly critical in ADAS. According to the development trend of machine learning proposed in [96], an improvement in transparency and interpretability can be investigated using additional knowledge, e.g., physical laws, finite automata, logical rules, and the formalized knowledge of experts.

- According to section 3.3, a TC-Hist retains multiple useful measurement characteristics. Current TC-Hist processing methods are mostly rule-based, which specify distance information as the only output. It means that the conversion from TC-Hist to point cloud exists information loss and places high demands on the reliability of TC-Hist processing methods. Moreover, TC-Hist-based processing and point cloud-based processing are completely separated. The latter considers the former as a black box and has no access to the measurement information before point cloud, while the former cannot adjust corresponding parameters according to the results of the latter. This results in an underutilization of LiDAR's measurement data and an inefficient processing procedure. Therefore, a more advanced and flexible method, such as machine learning, should be applied to TC-Hist to maximize the utility of the information. In addition, holistic considerations should be made to improve the information exchange between TC-Hist and point cloud.

- The large volume of a TC-Hist makes processing methods impossible to analyze data comprehensively in real-time. A compact and reliable feature extraction method should be developed to reduce data volume and to focus the processing only on important features. In addition, due to the limitations on real data collection and annotation in ADAS, a machine learning-based method that is less dependent on real data without sacrificing its generalization capability should be investigated, for example training a method with a hybrid dataset including real and synthetic data.

Chapter 4

Machine Learning-Based Multi-Peak Analysis

In this chapter, a machine learning-based TC-Hist processing method named MPA is proposed in order to overcome the research gap presented in section 3.6. Preliminary studies of MPA are presented in [97] [98]. In particular, a TC-Hist generation process based on FPC and TCSPC is provided. Using TC-Hist as the input data, the principle of MPA is outlined by dividing it into three parts: 1) physics-guided feature extraction, 2) machine learning-based distance prediction, and 3) correlation analysis.

4.1 First Photon Counting

To determine the photon statistics of direct TOF measurements, the whole observation time window for incoming photons is defined as ΔT. Hence, a fraction $\Delta t \in (0,\, \Delta T]$ is further divided into m small time durations by

$$\Delta t = \frac{\Delta T}{m}\ , \tag{4.1}$$

where $m \to \infty$, resulting in $\Delta t \to 0$. Thus, the probability of receiving more than one photon within Δt can be neglected. In this case, the probability of receiving a photon at time t is approximated by

$$P(t) = r(t)\,\Delta t\ . \tag{4.2}$$

Under the assumptions that an event of receiving a photon is independent and photon detection rate $r(t)$ is constant within $(t,\, t + \Delta t]$, the probability of receiving k photons within $(t,\, t + \Delta t]$ is the binomial function P_B given by

$$P_\mathrm{B}(k \,|\, m,\, P(t)) = \frac{m!}{k!(m-k)!}\, P(t)^k\, (1 - P(t))^{m-k} \; . \tag{4.3}$$

Since $m \to \infty$, the binomial distribution can approximate to the Poisson distribution P_P as

$$\lim_{m\to\infty} P_\mathrm{B}(k \,|\, m,\, P(t)) = \frac{P(t)^k}{k!}\, \mathrm{e}^{-P(t)} = \frac{\mu_t^k}{k!}\, \mathrm{e}^{-\mu_t} = P_\mathrm{P}(k,\, \mu_t) \; , \tag{4.4}$$

where μ_t is equal to $P(t)$. The analogy to the binomial theorem guarantees that the distribution is normalized [99].

In order to further improve the resistance to background light, the TCSPC technique is used to generate TC-Hists. Subsequently, FPC is applied to the measurement process. In FPC, a SPAD is deactivated after the detection of the first received photon and waits for the initialization of the next measurement. This means only one timestamp is obtained in a measurement cycle. With the assumption of an observation beginning at time $t = 0\,\mathrm{s}$, the probability of receiving no photon within $(0,\, T]$ according to (4.4) is

$$P_\mathrm{P}(k = 0,\, \mu_T) = \mathrm{e}^{-\mu_T} = \mathrm{e}^{-\int_0^T r(t)dt} = \mathrm{e}^{-\sum_{t=0}^{T} \mu_t} = \prod_{t=0}^{T} P_\mathrm{P}(k = 0,\, \mu_t) \; . \tag{4.5}$$

Therefore, the probability of receiving at least one photon within $(0,\, T]$ is given by

$$P_\mathrm{P}(k > 0,\, \mu_T) = 1 - P_\mathrm{P}(k = 0,\, \mu_T) = 1 - \mathrm{e}^{-\mu_T} = \int_0^T p_1(t)dt \; , \tag{4.6}$$

where $p_1(t)$ represents PDF of receiving the first photon at time t. By considering the total photon detection rate $r(t)$ according to (3.2), $P_\mathrm{P}(k > 0, \mu_T)$ becomes a three-stage function given by

$$P_\mathrm{P}(k > 0, \mu_T) = \begin{cases} 1 - \mathrm{e}^{r_\mathrm{B}\, T} \, , & 0 \leqslant T < T_\mathrm{Obj} \\ 1 - \mathrm{e}^{-(r_\mathrm{B}\, T_\mathrm{Obj} + r_\mathrm{LB}\, (T - T_\mathrm{Obj}))} \, , & T_\mathrm{Obj} \leqslant T < T_\mathrm{Obj} + T_\mathrm{P} \\ 1 - \mathrm{e}^{-(r_\mathrm{B}\, (T - T_\mathrm{P}) + r_\mathrm{LB}\, T_\mathrm{P})} \, , & T \geqslant T_\mathrm{Obj} + T_\mathrm{P} \end{cases} \; . \tag{4.7}$$

Afterwards, $p_1(t)$ is derived by taking the derivative of T in (4.7):

$$p_1(t) = \begin{cases} r_\mathrm{B}\, \mathrm{e}^{-r_\mathrm{B}\, t} \, , & 0 \leqslant t < T_\mathrm{Obj} \\ r_\mathrm{LB}\, \mathrm{e}^{-r_\mathrm{B}\, T_\mathrm{Obj}}\, \mathrm{e}^{-r_\mathrm{LB}\, (t - T_\mathrm{Obj})} \, , & T_\mathrm{Obj} \leqslant t < T_\mathrm{Obj} + T_\mathrm{P} \\ r_\mathrm{B}\, \mathrm{e}^{r_\mathrm{L}\, T_\mathrm{P}}\, \mathrm{e}^{-r_\mathrm{B}\, t} \, , & t \geqslant T_\mathrm{Obj} + T_\mathrm{P} \end{cases} \; . \tag{4.8}$$

In a TC-Hist, the x-axis is discrete due to TDC resolution T_{TDC}. Therefore, t is represented by the bin number b using

$$b = \lfloor \frac{t}{T_{\mathrm{TDC}}} \rfloor \ . \tag{4.9}$$

In this case, the probability of receiving the first photon in b is calculated by the integration of $p_1(t)$ across the bin width with

$$P_{1,b} = \int_{b\,T_{\mathrm{TDC}}}^{(b+1)\,T_{\mathrm{TDC}}} p_1(t)dt \ . \tag{4.10}$$

According to the principle of TCSPC, N_{M} individual measurement cycles are carried out and the received first photons are accumulated in a TC-Hist. By assuming that measurement cycles included in a TC-Hist follow the same $p_1(t)$, the expected value μ_b is given by

$$\mu_b = N_{\mathrm{M}}\, P_{1,b} \ . \tag{4.11}$$

4.2 Physics-Guided Feature Extraction

Based on (4.10), an example of $P_{1,b}$ and one corresponding TC-Hist with specific settings are shown in Figure 4.1. It is observed that an object distance corresponds to a local maximum, because r_{L} is only present and superposed on r_{B} within $[T_{\mathrm{Obj}}, T_{\mathrm{Obj}} + T_{\mathrm{P}}]$. It also indicates that in a TC-Hist, signals outside this region are noise. Thus, to avoid wasting a massive number of computations on noise, a subsequent analysis can be focused on several significant local maxima instead of the whole TC-Hist. For this purpose, a physics-guided feature extraction is designed as a three-step process: 1) convolution, 2) region division & local maxima selection, and 3) feature preparation, shown in Figure 4.2.

4.2.1 Convolution

In Figure 4.1 (b), the measurement distribution is sparse with only 400 measurements per TC-Hist and under a certain amount of background noise, resulting in large jitters. In this case, the selection of maxima becomes critical. Since T_{P} is typically larger than T_{TDC}, the received laser photons are spread over N_{P} adjacent bins. Therefore, a one-dimensional convolution kernel with size N_{P} is applied to the TC-Hist to strengthen the bin group containing the desired laser

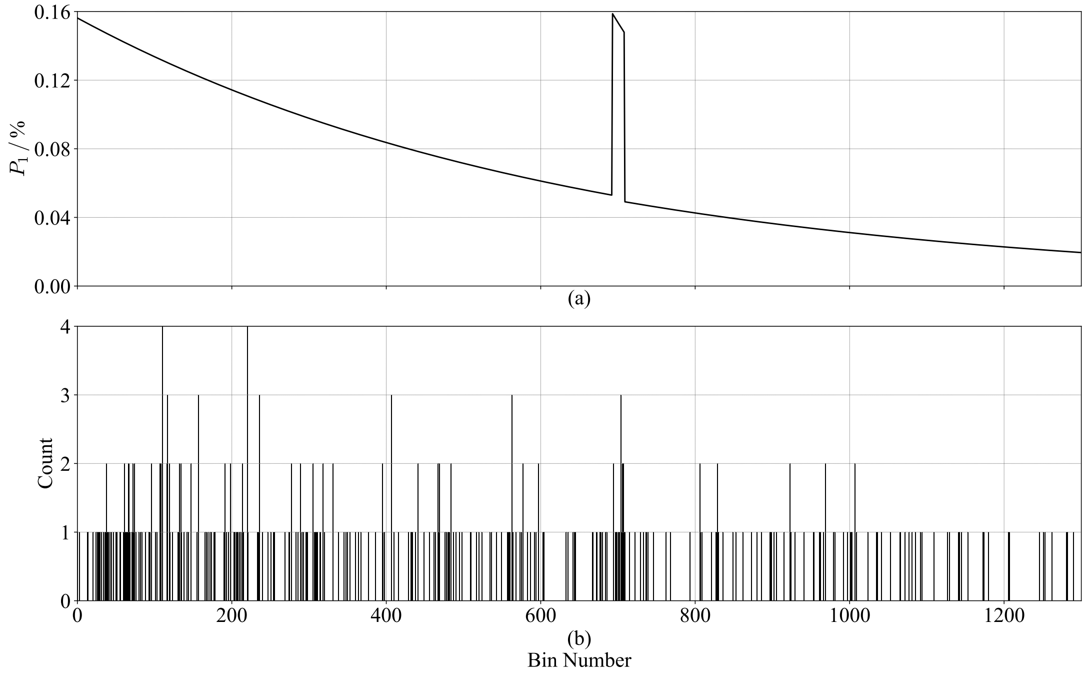


Figure 4.1: PMF and TC-Hist with specific parameter settings. (a) PMF of the first received photon. (b) One of the corresponding TC-Hists. Therein, T_{TDC} is equal to 312.5 ps, r_{B} is 5 MHz, r_{L} is 10 MHz, T_{P} is 5 ns, and N_{M} is equal to 400. T_{Obj} is set to 216.82 ns, i.e., the object bin number b_{Obj} is 693.

photons. Accordingly, the bin in a convolved TC-Hist is given by the first bin of each convolution group. Similar to template matching, the optimal convolution kernel must be determined in order to maximize the desired laser pulse shape. The most basic convolution kernel is the constant kernel K_{Const}, defined as

$$K_{\mathrm{Const},\,b} = 1, \;\; b \in [1, \, ..., \, N_{\mathrm{P}}] \, . \tag{4.12}$$

In addition, according to (4.8), the statistical characteristics of laser photons are governed by the Poisson distribution as well. Therefore, an exponential kernel K_{Exp} is designed as

$$K_{\mathrm{Exp},\,b} = \mathrm{e}^{-r_{\mathrm{LB}}\,T_{\mathrm{TDC}}\,b}, \;\; b \in [1, \, ..., \, N_{\mathrm{P}}] \tag{4.13}$$

to match the ideal laser pulse shape. However, (4.8) assumes the emitted laser is a rectangular pulse, which is impractical due to the response time and physical limitations of a laser emitter. Normally, the laser is approximated by a Gaussian function. Accordingly, a Gaussian kernel K_{Gauss} is chosen and defined as

$$K_{\mathrm{Gauss},\,b} = \mathrm{e}^{\frac{-b^2}{2\sigma^2}}, \;\; b \in [-3\sigma, \, ..., \, 3\sigma] \, , \tag{4.14}$$

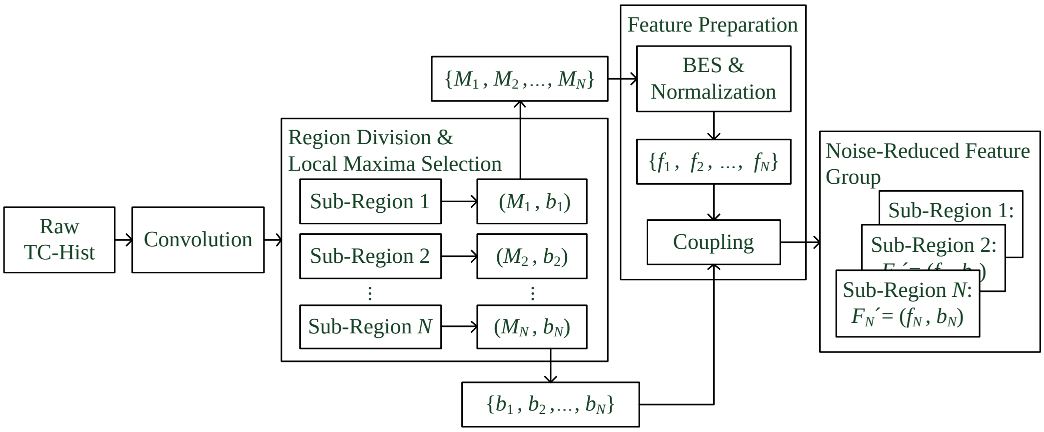


Figure 4.2: Process of physics-guided feature extraction. A raw TC-Hist is first convolved and then divided into N sub-regions. Local maximum M_n and its corresponding bin number b_n are extracted from each sub-region. Subsequently, f_n is formed by applying **B**ackground **E**stimation and **S**ubtraction (BES) on M_n. By coupling f_n and b_n, a sub-region feature F_n is created. Finally, feature group F'_{G} is obtained based on F_n of all sub-regions.

where

$$3\sigma = \lfloor \frac{N_{\mathrm{P}}}{2} \rfloor \, . \tag{4.15}$$

Figure 4.3 (a) shows a convolved TC-Hist using a constant kernel. It can be seen that the convolved TC-Hist is dense and thus more suitable for maxima extraction compared to a raw TC-Hist with limited N_{M}.

4.2.2 Region Division and Local Maxima Selection

In the next step, a convolved TC-Hist is divided into multiple sub-regions for local maxima selection. The following aspects are considered:

- To guarantee the effectiveness of feature extraction, the probability of selecting the desired object information should be maximized.
- Using FPC, the count value of a small bin number is likely to be larger than that of a large bin number due to the presence of background noise. Therefore, the tendency for selecting local minima with small bin numbers should be avoided.

- A laser pulse may create multiple local maxima within a bin group. The portion of selected local minima originating from the same laser pulse should be minimized in order to avoid information redundancy.

- The objective of feature extraction is to improve processing efficiency. Therefore, the local maxima search should be as simple as possible while achieving the performance requirements.

As a result, the concept of a sub-region division is proposed. First, a TC-Hist is divided into N adjacent sub-regions. Each sub-region has the same sub-region width W_{R} calculated by

$$W_{\mathrm{R}} = \frac{\mathrm{D}_{\mathrm{Max}}}{N} , \tag{4.16}$$

where $\mathrm{D}_{\mathrm{Max}}$ refers to a given maximum detection range. Afterwards, a local maximum M_n is selected from each sub-region. This guarantees an even selection of features on the x-axis. Obviously, N is equal to the number of selected features N_{F} and affects the accuracy of feature extraction. For example, if a large N_{F} is chosen, the probability of the maximum value being the laser pulse in the sub-region where the laser pulse locates is considerably high. If a small N_{F} is chosen, this probability is relatively low. Examples are given in Figure 4.3 (b) and (c). Therein, the desired local maximum is included in the selected features when $N_{\mathrm{F}} = 12$, while it is overlooked when $N_{\mathrm{F}} = 2$. However, the increase of N_{F} increases the number of features from noise. On the one hand, these noise features can interfere with distance predictions. On the other hand, the number of inputs for the subsequent processing increases, resulting in an exponential growth in computations. Therefore, the optimal distance division requires the selection of as few sub-regions as possible while ensuring the highest possible success rate of extracting the desired information. Since a direct determination of the optimal N_{F} suffers from the attribute explosion due to interdependence among a large number of bins, the determination of W_{R} is carried out using MCM. After the extraction of N_{F} local maxima from a TC-Hist, a local maximum M_n and its correlated bin number b_n are coupled as a region feature F_n, i.e., $F_n = \{M_n, b_n\}$. Finally, a feature group $F_{\mathrm{G}} = \{F_1, ..., F_N\}$ is formed to represent a complete TC-Hist.

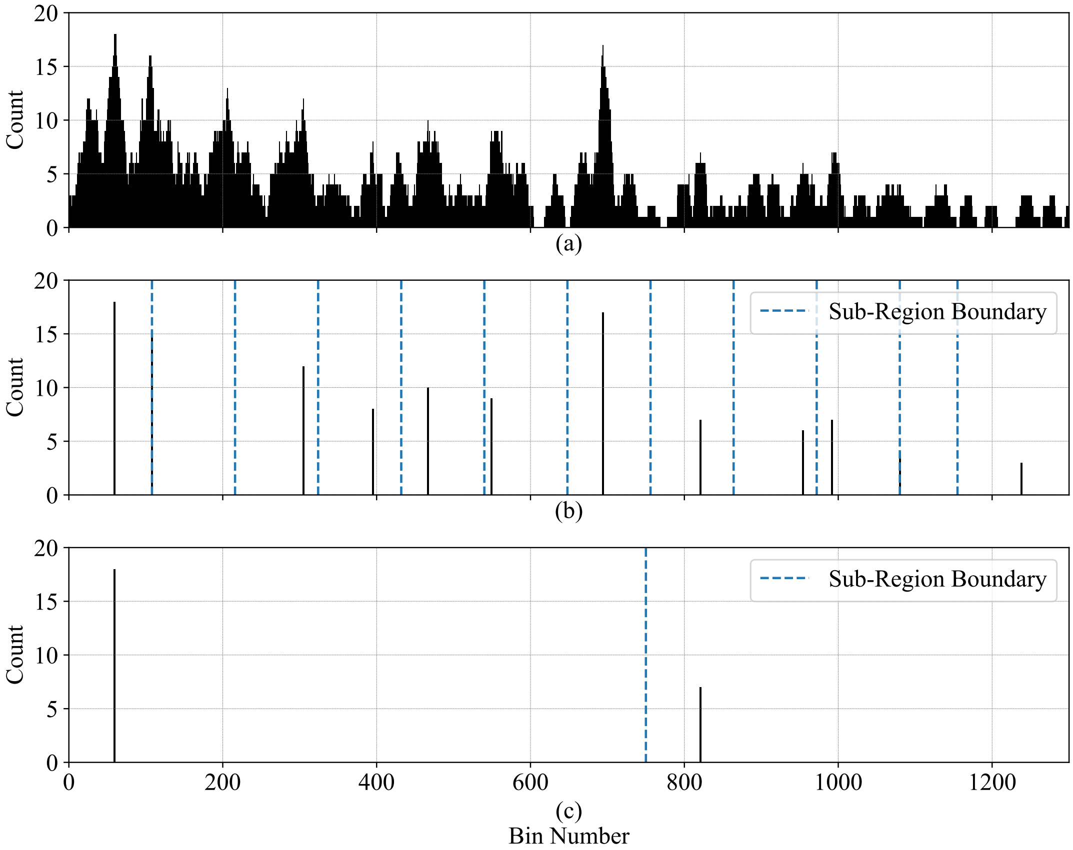


Figure 4.3: Example of feature selection with different N_{F}. The desired bin number is around bin 693. (a) A convolved TC-Hist by applying a constant kernel on the raw TC-Hist in Figure 4.1. (b) Selected local maxima with $N_{\mathrm{F}} = 12$. The selected bin number in sub-region 7 is 694, which corresponds to the desired bin number. (c) Selected local maxima with $N_{\mathrm{F}} = 2$. In this case, the selected local maxima do not contain the desired bin number.

4.2.3 Feature Preparation

In this subsection, the extracted local maxima are pre-processed by removing noise and normalization. Based on measured r_{B}, the expected background count is calculated regarding (4.11) by

$$\mu_{\mathrm{B},\,b} = N_{\mathrm{M}}\, P_{1,\,b}\,|_{\,r_{\mathrm{L}}=0} = N_{\mathrm{M}}\, r_{\mathrm{B}} \int_{b\,T_{\mathrm{TDC}}}^{(b+1)\,T_{\mathrm{TDC}}} \mathrm{e}^{-r_{\mathrm{B}}\,t} dt \,. \tag{4.17}$$

Subsequently, a noise-reduced feature value f_n is determined by a BES method on M_n using

$$f_n = \frac{M_n - \sum_{b=b_n}^{b_n+N_\mathrm{P}} \mu_{\mathrm{B},\,b}}{\max_n (M_n - \sum_{b=b_n}^{b_n+N_\mathrm{P}} \mu_{\mathrm{B},\,b})}\ , \tag{4.18}$$

where $n \in \{1,\ ...,\ N_\mathrm{F}\}$. Similarly, a noise-reduced feature group $F'_\mathrm{G} = \{...,\ F'_n,\ ...\}$ is created, where $F'_n = \{f_n,\ b_n\}$.

4.3 Distance Prediction Algorithms

In this section, the designs of different distance prediction algorithms for the noise-reduced feature group F'_G are presented.

4.3.1 Fully-Connected Feed-Forward Neural Network

The powerful representation and generalizability of neural networks make them one of the optimal choices among machine learning algorithms. However, training a neural network directly on TC-Hists to infer exact distances is challenging. On the one hand, if a classification network is used, a large number of classes is required in order to maintain a high distance resolution, which results in an attribute explosion. On the other hand, if a regression network is applied, a large number of input attributes and a deep network are required to accurately model the regression function. Besides, a regression typically provides only one output, which makes a subsequent analysis on further useful information impossible.

To cope with these difficulties, a classification FNN and a distance recovery process are designed to combine with the feature extraction introduced in section 4.2, shown in Figure 4.4. First, sub-region feature f_n is separated from noise-reduced feature group F'_G obtained in subsection 4.2.3. Afterwards, the feature values are formed as an input vector $I_N = \{f_1,\ ...,\ f_N\}$. The FNN is trained, validated, and tested by I_N. To prevent the attribute explosion, a class group is created with N_F classes, which corresponds to the sub-regions and the extracted features in section 4.2. Accordingly, an output vector $\{o_1,\ ...,\ o_N\}$ is calculated using softmax function. o_n represents the score of the feature in the n^th sub-region being the object distance. The sub-region H with the highest score o_H will be chosen as the final prediction. Since the resolution provided by N_F classes is much lower than T_TDC, a distance recovery process is introduced to restore

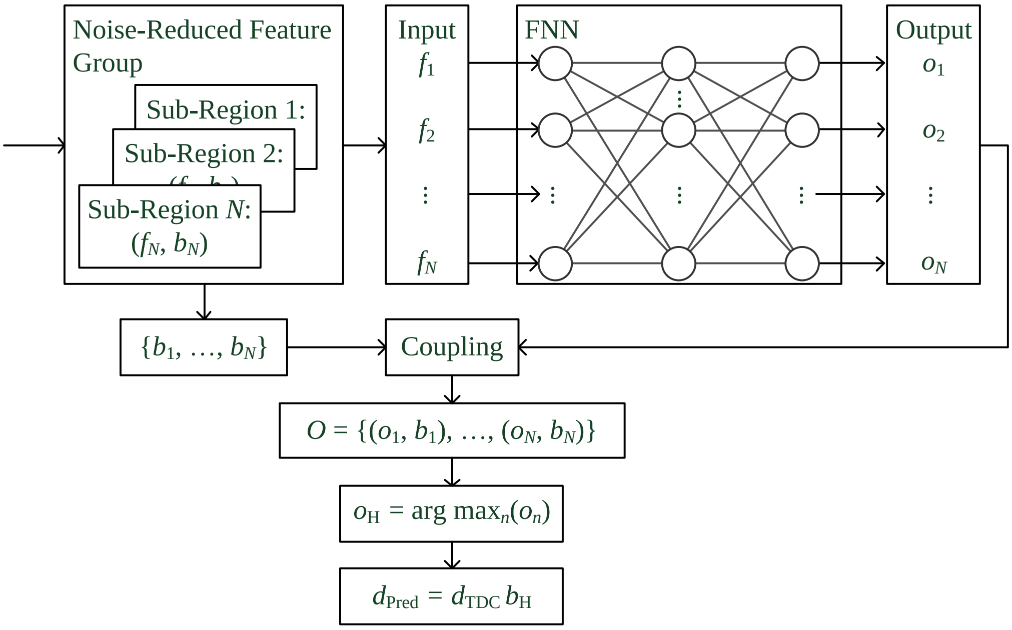


Figure 4.4: Distance prediction with FNN. Based on F'_{G} in Figure 4.3, feature sequence $\{f_1, \ldots, f_N\}$ is inputted into an FNN and output vector $\{o_1, \ldots, o_N\}$ is obtained. Final output O is formed by coupling $\{o_1, \ldots, o_N\}$ and $\{b_1, \ldots, b_N\}$ correspondingly. Predicted distance d_{Pred} is calculated using b_{H} and d_{TDC}.

distance precision. At first, each o_n is coupled to the corresponding b_n and an output $O = \{(o_1, b_1), \ldots (o_n, b_n)\}$ is formed. Subsequently, the distance prediction d_{Pred} is calculated using bin number b_{H} from sub-region H as

$$d_{\mathrm{Pred}} = d_{\mathrm{TDC}}\, b_{\mathrm{H}} \ . \tag{4.19}$$

The configuration of FNN parameters is conducted in the following steps:

1) In order to exclude irrelevant parameters and to narrow the scope of parameter tuning, a preliminary grid search is carried out.

2) After the grid search, a number of the most relevant parameters are selected and analyzed jointly.

Before training, the data for training and validation are shuffled so that the classes in each training mini-batch are balanced. To minimize the random factor introduced by weight initialization, each network structure is trained multiple times individually. As a result, the optimal variant of FNN is determined.

4.3.2 Binomial Estimation

Since r_{L} defines the magnitude of a laser pulse, it directly influences the significance of laser information (i.e., object distance) in a TC-Hist. However, according to subsection 2.1.5, the propagation process of a laser pulse is highly uncertain and involves a number of environmental and hardware-induced factors. Therefore, an accurate estimation for r_{L} is impractical. In contrast, r_{B} can be well estimated using the average photon counts during an observation window. Thus, instead of pursuing laser information directly, an approach can be designed from another perspective: analyzing whether features come from background light or not. This is done by conducting an analysis for the statistical behavior of background light. To this end, r_{B} is estimated by counting the number of received photons during a specific observation window. Afterwards, by assuming $r_{\mathrm{L}} = 0\,\mathrm{MHz}$, the probability of receiving a photon in bin number b is given by

$$P_{1,b}\,|_{\,r_{\mathrm{L}}=0} = \int_{b\,T_{\mathrm{TDC}}}^{(b+1)\,T_{\mathrm{TDC}}} p_1(t)\,|_{\,r_{\mathrm{L}}=0}\,dt = \int_{b\,T_{\mathrm{TDC}}}^{(b+1)\,T_{\mathrm{TDC}}} r_{\mathrm{B}}\,\mathrm{e}^{-r_{\mathrm{B}}\,t}dt \tag{4.20}$$

according to (4.10). Hereby, each probability $P_{1,b}$ is independent to $P_{1,b'}$, where $b' \neq b$. Moreover, since each detected photon in a TC-Hist generation period is independent and is assumed to be identically distributed, the probability of obtaining k photons in b with N_{M} trials can be calculated using (4.3) as

$$P_{\mathrm{B}}(k\,|\,N_{\mathrm{M}},\, P_{1,b}\,|_{\,r_{\mathrm{L}}=0}) = \frac{N_{\mathrm{M}}!}{k!(N_{\mathrm{M}}-k)!}\, P_{1,b}^{k}\,|_{\,r_{\mathrm{L}}=0}\,(1-P_{1,b}\,|_{\,r_{\mathrm{L}}=0})^{N_{\mathrm{M}}-k}\,. \tag{4.21}$$

According to (4.8), the presence of r_{L} causes $P_{1,b}$ to rise when $t \in [T_{\mathrm{Obj}},\, T_{\mathrm{Obj}} + T_{\mathrm{P}})$. The influence of r_{L} on $P_{1,b}$ when $t \geqslant T_{\mathrm{Obj}} + T_{\mathrm{P}}$ is neglected, since T_{P} is small. These characteristics mean that the bins containing laser photons deviate significantly from the background photon distribution, resulting in $P_{\mathrm{B}} \to 0$ and thus allowing T_{Obj} to be allocated. An application of the binomial estimation on a complete TC-Hist is extremely computational-expensive, because each bin must be calculated using P_{B} individually. Therefore, it is applied to the extracted local maxima M_n introduced in subsection 4.2.2 to reduce unnecessary computation on noise.

An example of $P_{\mathrm{B}}(k\,|\,N_{\mathrm{M}} = 400,\, P_{1,b}\,|_{\,r_{\mathrm{L}}=0} = 0.0247)$ is shown in Figure 4.5. Apparently, the maximum probability is obtained around the expected value μ. For the purpose of explanation, the distribution is divided into two parts according to μ. Since k corresponds to a local maximum, which is generally larger than μ, the result tends to be located in the black area with a relatively small P_B. In particular, if r_L is present, k will deviate from μ significantly, resulting in an extremely small P_B. Thus, the key to determining the object distance (i.e., k with laser photons) is to identify the local maxima with small probabilities. To improve the discriminatory ability at small probabilities and to transform them to the probability of being the desired object information, P_B is normalized by

$$P_{\mathrm{Norm},\,b} = \frac{-\ln(P_\mathrm{B}(k \,|\, N_\mathrm{M},\, P_{1,\,b}\,|\,_{r_\mathrm{L}=0}))}{\sum -\ln(P_\mathrm{B}(k \,|\, N_\mathrm{M},\, P_{1,\,b}\,|\,_{r_\mathrm{L}=0}))} \,. \tag{4.22}$$

$P_{\mathrm{Norm},\,b}$ directly represents the probability of the object distance at the associated bin. The result of the binomial estimation using the input from Figure 4.3 (b) is shown in Table 4.1. It can be seen that the local maxima at bin 59, 694, and 992 have P_B lower than 1 %, which are most likely to be the bin representing the object distance. However, these probabilities are small, which may be indistinguishable due to the underflow. Compared with $P_{\mathrm{B},\,b}$ of {0.291074, 0.000008, 0.433654}, the corresponding $P_{\mathrm{Norm},\,b}$ of {9.686889, 27.118243, 9.025539} are more suitable either as final outputs or as inputs for subsequent processing.

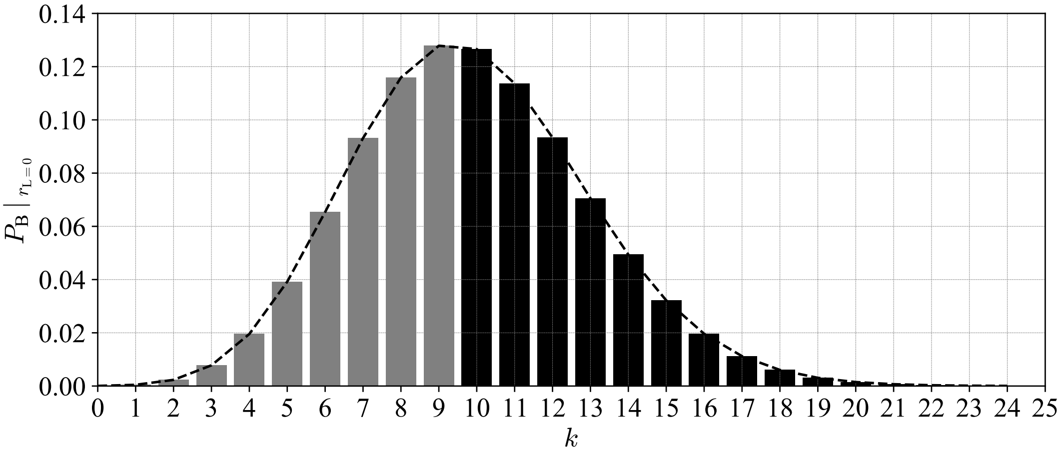


Figure 4.5: Binomial probability with $N_\mathrm{M} = 400$ and $P_{1,\,b}\,|\,_{r_\mathrm{L}=0} = 0.0247$. The distribution is divided into a grey area and a black area according to the expected value $\mu = N_\mathrm{M}\, P_{1,\,b}\,|\,_{r_\mathrm{L}=0} = 9.88$.

The binomial estimation is conducted for each feature F_n individually and outputs a vector P_Norm, which includes N_F normalized probabilities $P_{\mathrm{Norm},\,b}$.

Table 4.1: Binomial estimation on the example in Figure 4.3. The object distance is at 32.50 m. b_n refers to the bin number of the selected feature in the n^{th} sub-region. $P_{\mathrm{B},\,b}$ and $P_{\mathrm{Norm},\,b}$ refer to corresponding probabilities with $b = b_n$.

b_n	$d_n\,/\,\mathrm{m}$	M_n	$P_{\mathrm{B},\,b}\,/\,\%$	$P_{\mathrm{Norm},\,b}\,/\,\%$
59	2.763712	18	0.291074	9.686889
108	5.058998	15	1.201604	7.334844
305	14.286984	12	1.281738	7.227746
395	18.502816	8	7.815941	4.228542
467	21.875481	10	1.410607	7.068819
549	25.716572	9	1.657963	6.800791
694	32.508745	17	0.000008	27.118243
821	38.457751	7	1.481113	6.987907
954	44.687813	6	1.819096	6.646928
992	46.467831	7	0.433654	9.025539
1080	50.589977	4	7.467957	4.304095
1238	57.991104	3	11.627201	3.569657

4.3.3 Naive Bayes Classifier

Since TC-Hists with the same measurement parameters, e.g., background photon rate r_B and object distance d_Obj, follows the same probability distribution given by (4.8), P_Norm of TC-Hists exist a degree of similarity to each other. In this subsection, NBC is introduced to model the sample distribution of P_Norm and to determine object distance accordingly.

The prerequisite of NBC is the attribute conditional independence. This is guaranteed by the weak dependence of $P_{\mathrm{B},\,b}$ to $P_{\mathrm{B},\,b' \neq b}$ regarding (4.21) and the binomial estimation, which minimizes the effect of background noise floor. According to (2.39), NBC on the normalized probability is written as

$$f_\mathrm{NBC}(P_\mathrm{Norm}) = \arg\max_c (P(P_\mathrm{Norm}\,|\,c)\,P(c)) \,, \tag{4.23}$$

where c refers to one of the classes created for NBC. Therein, $P(c)$ represents the portion of class c in a given dataset D, which is easily obtained by

$$P(c) = \frac{N_c}{N_\mathrm{S}} \;, \tag{4.24}$$

where N_c is the number of TC-Hists in class c and N_S refers to the total number of samples (TC-Hists) in D. $P(P_\mathrm{Norm} \,|\, c)$ refers to the probability distribution of P_Norm in class c. In the following calculation, the bin-wised probability $P_{\mathrm{Norm},\, b} \,|\, c$ is transformed to the sub-region-wised probability $P_{\mathrm{Norm},\, n} \,|\, c$. Hereby, n refers to the sub-region where b is located.

The derivation process using NBC is decomposed into four steps, shown in Figure 4.6. Each step is specified as follows:

1) **Categorization**: First, the effects of different parameters, e.g., r_B and d_Obj, are specified based on the observation of a given dataset D. Accordingly, a class group with N_NBC classes is created by grouping the dataset D regarding the degree of similarity between P_Norm of different TC-Hists. In contrast to FNN, where the output classes are equal to N_F, N_NBC is not necessarily equal to N_F.

2) **Modeling**: Based on observation results of the categorization, the modeling of the posterior probability $P(P_{\mathrm{Norm},\, n} \,|\, c)$ is carried out using a suitable probability distribution. The first attempt is assuming that $P_{\mathrm{Norm,\, n}} \,|\, c$ follows a normal distribution. In this case, (2.19) is applied and $P(P_{\mathrm{Norm},\, n} \,|\, c)$ is given by

$$P(P_{\mathrm{Norm},\, n} \,|\, c) = \mathcal{N}(P_{\mathrm{Norm},\, n} \,|\, \overline{P}_{n,\, c},\; s^2_{n,\, c}) = \frac{1}{\sqrt{2\pi s^2_{n,\, c}}} \mathrm{e}^{-\frac{(P_{\mathrm{Norm},\, n} - \overline{P}_{n,\, c})^2}{2 s^2_{n,\, c}}} \;. \tag{4.25}$$

Here, $\overline{P}_{n,\, c}$ and $s_{n,\, c}$ are empirical values estimated based on samples in each class, using

$$\overline{P}_{n,\, c} = \frac{\sum_{i=1}^{N_c} P_{\mathrm{Norm},\, n,\, i} \,|\, c}{N_c} \quad \text{and} \tag{4.26}$$

$$s_{n,\, c} = \sqrt{\frac{\sum_{i=1}^{N_c} (P_{\mathrm{Norm},\, n,\, i} \,|\, c - \overline{P}_{n,\, c})^2}{N_c}} \;. \tag{4.27}$$

However, P_Norm may not follow a standard normal distribution. Therefore, two distributions are further considered: (1) If a skewed distribution is observed on $P_{\mathrm{Norm},\, n} \,|\, c$, the modeling is carried out using a log-normal distribution by

$$P(P_{\mathrm{Norm},\, n} \,|\, c) = \mathcal{N}(ln(P_{\mathrm{Norm},\, n} \,|\, \overline{P}_{n,\, c},\; s^2_{n,\, c}) \;. \tag{4.28}$$

(2) If a multimodal distribution is observed, the modeling is conducted using a Gaussian mixture distribution by

$$P(P_{\mathrm{Norm},\,n}\,|\,c) = \sum_{m=1}^{M} w_m \mathcal{N}(P_{\mathrm{Norm},\,n}\,|\,\overline{P}_{m,\,n,\,c},\; s^2_{m,\,n,\,c}) \;, \tag{4.29}$$

where M represents the number of Gaussian models and w_m refers to the weight of the m^{th} model. Finally, the models form an identifier of the associated class.

3) **Joint probability**: After the modeling, the probability distribution is created for each input $P_{\mathrm{Norm},\,n}\,|\,c$. To determine the general similarity of a given input data to a certain class, the joint probability $P(P_{\mathrm{Norm}}\,|\,c)$ is given by

$$P(P_{\mathrm{Norm}}\,|\,c) = ln(\prod_{n=1}^{N} P(P_{\mathrm{Norm},\,n}\,|\,c)) = \sum_{n=1}^{N} ln(P(P_{\mathrm{Norm},\,n}\,|\,c)) \;. \tag{4.30}$$

4) **Normalization**: Finally, f_{NBC} is normalized and the corresponding soft decision is given by

$$o_c = \frac{f_{\mathrm{NBC},\,c}}{\sum_{c=1}^{N_{\mathrm{NBC}}} f_{\mathrm{NBC},\,c}} \;. \tag{4.31}$$

o_{H} with the highest value is given by

$$o_{\mathrm{H}} = \arg\max_{c}(o_c) \;. \tag{4.32}$$

Meanwhile, N_{NBC} classes are converted to N_{F} bin numbers accordingly. Similar to the design of distance recovery in subsection 4.3.1, $O = \{(o_1, b_1), \ldots (o_N, b_N)\}$ is formed and d_{Pred} of NBC is calculated using (4.19) as well.

4.4 Correlation Analysis

Both FNN and NBC generate soft decisions o_n for N_{F} potential object distances. Each soft decision represents the confidence level that the corresponding distance is the object distance. Compared to hard decisions, which only provide an absolute value as output, soft decisions are better suited to the optimization and fine-tuning process with additional knowledge and reasonable assumptions, e.g., correlations in time and space. In this section, the design of correlations for MPA is introduced.

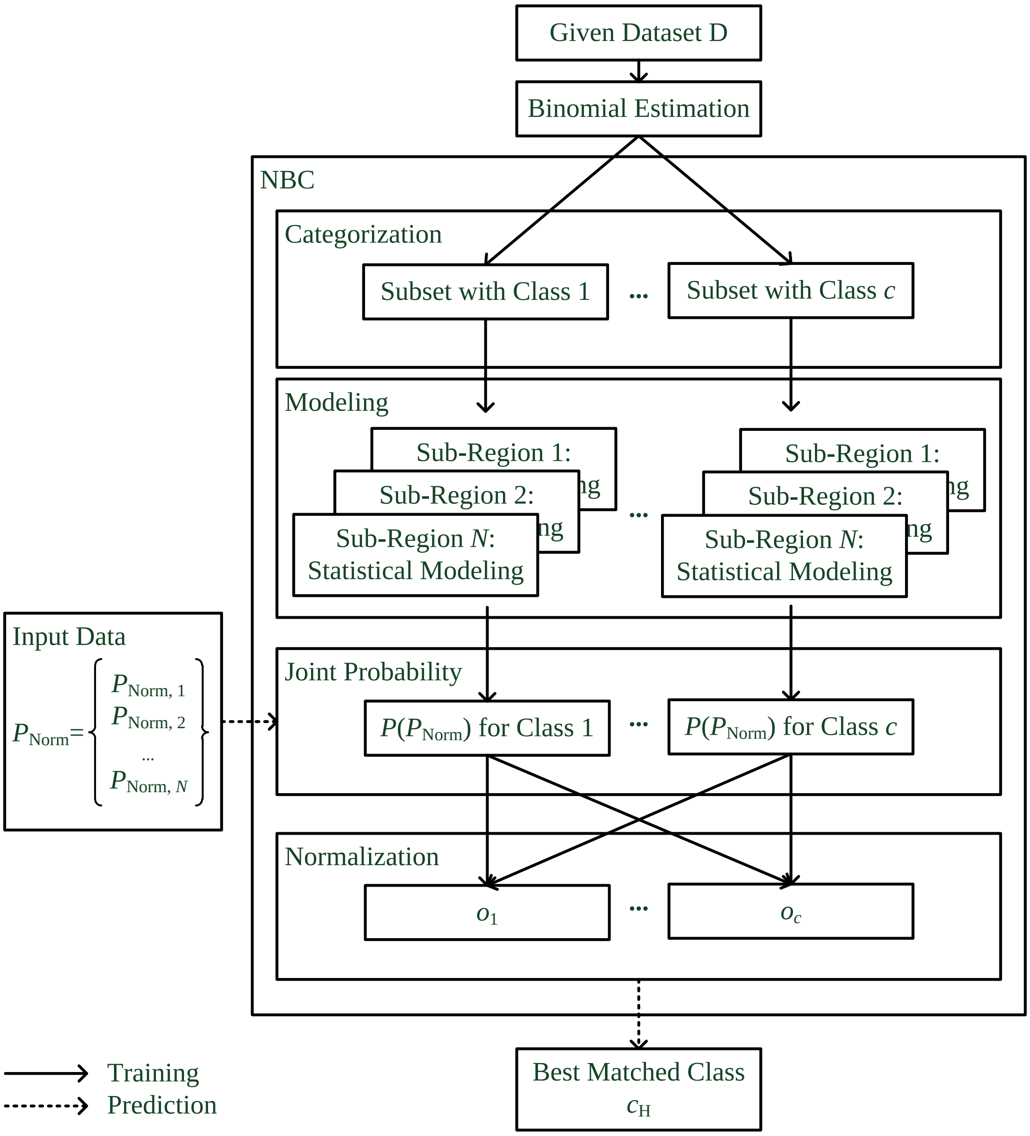


Figure 4.6: Process of NBC on P_{Norm} from binomial estimation. During the training process, samples in dataset D is categorized into multiple subset according to their similarities and a statistical modeling is carried out on each subset for obtaining the corresponding class identifier. In the prediction process, the joint probability between an input and each class identifier is calculated and the best-matched class is determined correspondingly.

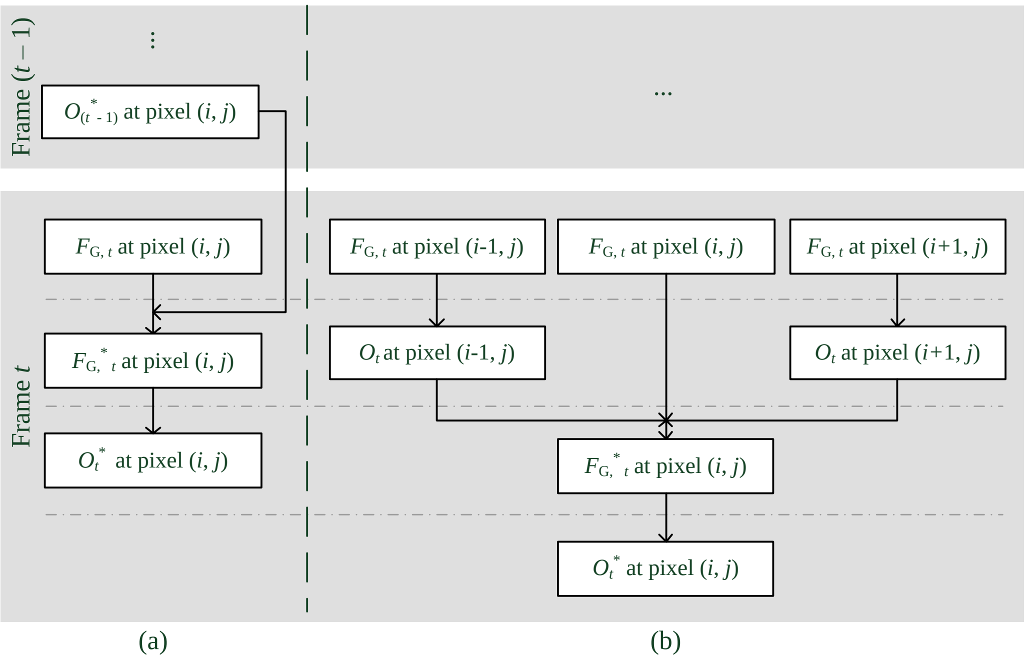


Figure 4.7: Correlation strategy. (a) Time correlation with three processing steps. The output from the previous frame $O_{(t-1)}$ is correlated to the feature group of the current frame $F_{\mathrm{G},t}$. (b) Spatial correlation with four processing steps. The outputs of neighboring pixels $(i \pm 1, j)$ is correlated to the feature group of the central pixel $F_{\mathrm{G},i,j}$. New predictions are obtained by inputting the correlated feature $F^{*}_{\mathrm{G},t}$ into FNN / NBC.

4.4.1 Correlation Strategies in Time and Space

In the field of object detection, a basic rule holds: The behavior of an object is continuous and does not suddenly disappear and appear. This leads to two hypotheses: 1) If the measurements of multiple TC-Hists for the same pixel are temporally continuous, the object in a new TC-Hist will locate close to the position of the object in a previous TC-Hists. 2) If the object is flat and occupies multiple pixels, object distances among these pixels will locate close to each other. To take the advantage of the information behind these hypotheses, two correlation strategies are designed, shown in Figure 4.7. Both correlation strategies use the output O (including N_{F} potential distances and soft decisions) as the reference in order to minimize irrelevant correlations from noise. However, the process of time correlation is slightly different than that of spatial correlation. In the time domain, $O_{(t-1)}$ from the previous frame is used as a reference and is correlated to the feature group $F_{\mathrm{G},t}$ of the current frame. Afterwards, a new prediction result O_t^* is obtained. To realize this, $O_{(t-1)}$ must be stored additionally for each pixel. In contrast to the time correlation, which utilizes output results from the previous time frame without adding further operations, O_t of all pixels without correlation must be calculated additionally in the spatial correlation. Subsequently, O_t from neighboring pixels $(i-1,\, j)$ and $(i+1,\, j)$ are correlated to $F_{\mathrm{G},t}$ of the central pixel (i, j). Finally, $O_{t,i,j}^*$ is obtained by the second distance prediction on $F_{\mathrm{G},t}^*$. This difference is illustrated in Figure 4.7, i.e., the time correlation has three main processing steps, while the spatial correlation has four main processing steps.

4.4.2 Gain Function

Gain functions are determined in each sub-region individually. A key to an effective correlation algorithm is generating significant gain under desired conditions, while minimizing negative influences under irrelevant conditions. To this end, object behaviors and feature characteristics are studied. In sub-region n, a feature f_n or a soft decision o_n originates from

(1) a static/continuous moving object,

(2) the flat surface of an object,

(3) an object leaving/coming into FOV,

(4) the boundary of an object, or

(5) noise.

Apparently, the position (i.e., bin number b_n) of a feature originating from an object is relatively consistent. On the contrary, the position of a feature originating from noise is occurred by coincidence with a constrained N_M. This results in a randomly distributed b_n of a noise f_n (and the corresponding o_n). Based on the information, critical detection scenarios and estimated characteristics in the correlation analysis are summarized in Table 4.2. It can be seen that an ideal correlation scenario in time (scenario 1) or in space (scenario 2) is represented by a small difference in b_n. A large difference in b_n is caused by the state change of the rear object between blocked and visible (scenario 3). The detection of a

Table 4.2: Critical detection scenarios and estimated characteristics in the correlation analysis. Five scenarios are listed regarding different conditions of data from the original sample (f_n) and data from the reference sample (o_n). Accordingly, the estimated characteristics of b_n in MPA are specified.

	Condition		
Scenario	f_n	o_n	Estimated Characteristics in MPA
1	(1)	(1)	Small $(b_{n,t} - b_{n,t-1})$
2	(2)	(2)	Small $(b_{n,i,j} - b_{n,i\pm1,j})$
3	(3)	(3)	Large $(b_{n,i,j} - b_{n,i\pm1,j})$
4	(4)	(4)	Large/Random $(b_{n,t} - b_{n,t-1})$ Large/Random $(b_{n,i,j} - b_{n,i\pm1,j})$
5	(5)	(1)-(5)	Random $(b_{n,t} - b_{n,t-1})$ Random $(b_{n,i,j} - b_{n,i\pm1,j})$
	(1)-(5)	(5)	Random $(b_{n,t} - b_{n,t-1})$ Random $(b_{n,i,j} - b_{n,i\pm1,j})$

boundary (scenario 4) involves two situations. If two pixels detect two sides of a boundary, a large difference in b_n between these pixels is observed. If a boundary locates exactly in the laser transmission path of a pixel, the reflected laser echo is weak and difficult to be detected due to the scattering at large angles, resulting in a random difference in b_n. Similarly, if at least a noise feature is involved (scenario 5), a random difference in b_n should be observed as well.

Clearly, a distinction of gain behaviors is required, i.e., a significant gain is desired for scenario 1 and 2, while a gain for scenario 3 - 5 should be avoided. In addition, it can be indicated that the desired gain behaviors in both time and spatial correlations are identical for the given five scenarios, i.e., the gain should increase with the decrease of the difference in b_n. For the purpose of clarity, two unified definitions are given: 1) The output from the previous frame or from the neighboring pixel is defined as the reference data, which is represented by (o'_n, b'_n). 2) The feature of the current frame or from the central pixel is defined as the original data, which is represented by (f_n, b_n). Subsequently, the gain function

G is given by

$$G(o'_n,\, \Delta b_n) = \alpha\, o'_n\, \mathrm{e}^{-\frac{\Delta b_n^2}{2\beta^2}} \ , \tag{4.33}$$

where $\Delta b_n = (b_n - b'_n)$. α refers to the impact factor of the reference amplitude and β refers to the impact factor of Δb_n. Obviously, G is centered on the position of b_n and is proportional to o'_n. With these characteristics, a significant gain is generated only with a small Δb_n (i.e., the peak position in the n^{th} sub-region of the reference TC-Hist is close to the peak position in the same sub-region of the original TC-Hist) and a noticeable o'_n (i.e., the position of the reference peak has a high probability to be the object distance). An example of the correlation in MPA is shown in Figure 4.8. Therein, three sub-regions are assumed. In terms of sub-region 1, although o'_1 is significant, the gain is small, since Δb_n is large. Similarly, the gain in sub-region 3 is small as well with a small o'_3 and a small Δb_n. In sub-region 2, since Δb_n is small and o'_2 is large, a significant gain is generated. Thus, a noticeable gain is created only if the position of the reference information is close to the feature of the original TC-Hist and the reference information has a high probability to be the object distance. By the determination of the optimal α and β for specific detection scenarios, the requirements based on Table 4.2 could be fulfilled.

According to Figure 4.7, the time and the spatial correlation are determined based on (4.33) separately. In terms of the time correlation, the feature gain $\Delta f_{\text{Time},\,n}$ is calculated by

$$\Delta f_{\text{Time},\,n} = G(o_{n,\,t-1}, b_{n,\,t} - b_{n,\,t-1}) \ . \tag{4.34}$$

In terms of the spatial correlation, the feature gain $\Delta f_{\text{Space},\,n}$ is expressed as

$$\Delta f_{\text{Space},\,n} = G(o_{n,\,i-1,\,j},\, b_{n,\,i,\,j} - b_{n,\,i-1,\,j}) + G(o_{n,\,i+1,\,j},\, b_{n,\,i,\,j} - b_{n,\,i+1,\,j}) \ . \tag{4.35}$$

According to the process in Figure 4.7 and (4.33), $\Delta f_{n,\,t}$ and $\Delta f_{n,\,i,\,j}$ are determined independently of each other. Therefore, they can be applied to $f_{n,\,t,\,i,\,j}$ jointly using

$$f^*_{n,\,t,\,i,\,j} = f_{n,\,t,\,i,\,j} + \Delta f_{\text{Time},\,n} + \Delta f_{\text{Space},\,n} \ . \tag{4.36}$$

4.4.3 Scenario Analysis

The analysis of the correlation strategies in time and space requires a detection scenario with a certain dynamic degree. For this purpose, a virtual scene is

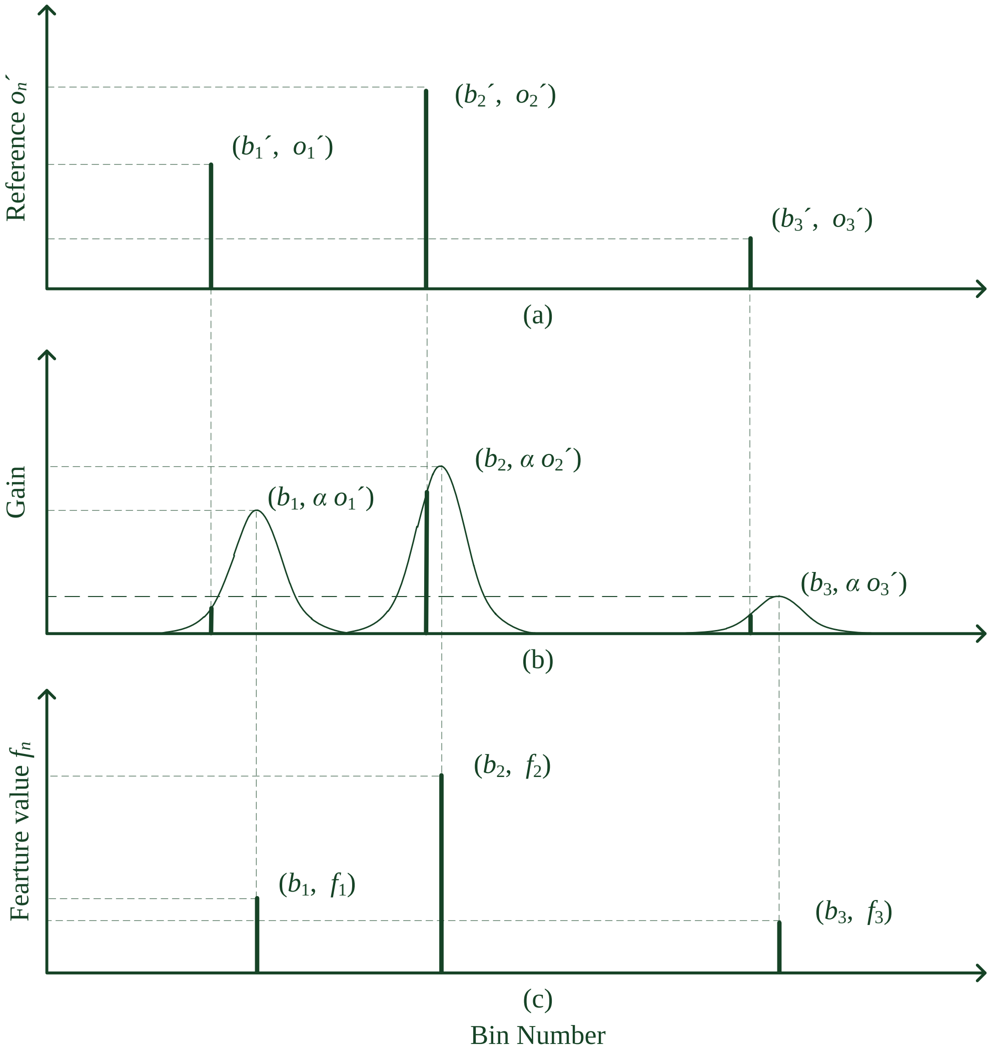


Figure 4.8: A correlation example. (a) Prediction results of a reference TC-Hist . (b) Calculated gains for the original TC-Hist. (c) Extracted features of the original TC-Hist.

created, shown in Figure 4.9. It consists of a LiDAR front-end, a background wall, and multiple square objects. The LiDAR front-end is located in the lower center and creates a FOV with an aperture angle of φ. The background wall covers the whole FOV with a distance of d_{Wall} to reduce random signals caused by the loss of laser echoes. The objects have the same size of w_{Obj} and are lined up horizontally. In addition, they are moving to the right with a constant velocity of v_{Obj}. The vertical distance of the objects to the LiDAR detector plane is d_{Vert}.

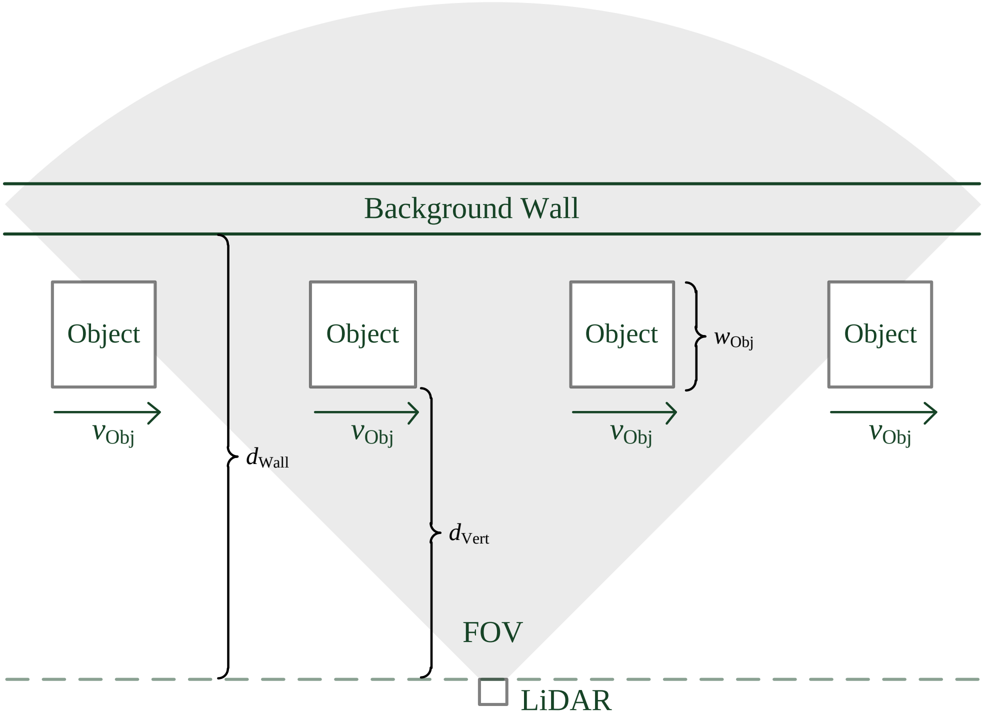


Figure 4.9: Bird's-eye view of virtual detection scene for correlation analysis. It includes multiple objects, a background wall, and a LiDAR front-end. The objects are lined up in a row and have a constant velocity v_{Obj} to the right. The LiDAR front-end locates at the lower center of the scene. It emits artificial photons continuously within its FOV. The photons are reflected back once they reach the surface of the objects or the background wall. Accordingly, d_{Obj} is determined. By specifying other parameters, e.g., r_{B}, r_{L}, and T_{P}, the TC-Hist simulation for the scene is carried out.

The simulation starts with the generation of virtual lasers. The lasers are emitted outward with the LiDAR front-end as the center of the circle in the range of FOV. Afterwards, each laser is reflected back along the emission path when it encounters the first object boundary. Accordingly, the object distance d_{Obj} is determined. By specifying r_{B} and r_{L} for each laser path, a corresponding TC-Hist group for the scene is created. In this simulation, the light propagation follows the model shown in Figure 2.2. Finally, The correlation strategies are applied to TC-Hist groups of neighboring pixels and at consecutive time frames simulated in the scene.

4.5 Other Optimization Approaches

In previous sections, MPA is designed to maximize the utility of the potential information in TC-Hists and to achieve the optimal prediction accuracy. In this section, further aspects are taken into consideration, including 1) real-time performance, 2) ablation study, 3) result optimization.

4.5.1 Ablation Study for Background Suppression

As introduced in section 4.3, both FNN and NBC utilize the noise-reduced feature as the input. In terms of FNN, the background suppression is carried out based on the estimation of r_{B}, which involves a real-time exponential approximation. In terms of NBC, the binomial estimation is applied, which involves not only exponential calculations but also factorials. Although they are only carried out on N_{F} selected features, it is still challenging on embedded systems.

Since the background information is already included in the local maximum M_n, a machine learning-based algorithm could have the ability to learn the background pattern directly from M_n. Therefore, a proof of necessity for background suppression is carried out by excluding BES in FNN variants and by excluding the binomial estimation in NBC variants. The evaluation is conducted regarding timing performance and prediction accuracy.

4.5.2 Convolution Stride

The value of a convolution stride affects the precision of extracted features and the real-time performance of MPA. With a small stride, the distance resolution in a feature TC-Hist is fine and a long processing time is required. Increasing the stride decreases the distance resolution while the overall processing time is reduced. Thus, the studies of applying different strides in MPA is of interest in order to provide a guidance on stride settings for different application scenarios. To this end, the stride is gradually increased and combined with the optimal FNN and NBC obtained in section 4.3. The impact of different strides is discussed by observing the decreasing trends of the prediction accuracy.

4.5.3 Threshold Analysis

MPA predicts the final distance according to the value of o. However, if a reflected laser signal is weak or lost, the corresponding TC-Hist contains barely desired object information. In order to reduce the misclassification rate and the overall reliability of distance predictions, it is necessary to classify this case as an indiscernible result rather than imposing a false distance prediction. A weak reflected laser signal means that $r_{\mathrm{L}} \rightarrow 0$. Based on (4.8), it results in a convergence of the object features with the noise features. Based on the fact that determining object distance distinctively is challenging under this condition, it is assumed that the corresponding confidence level (i.e., the value of o) of the distance prediction algorithm is in general lower than that with strong laser signals. Thus, a threshold o_{Th} is introduced to separate the distance predictions with respect to the value of o. In this case, MPA is changed to a binary classification problem, i.e., with a given o_{Th},

- if the highest prediction value $o_{\mathrm{H}} > o_{\mathrm{Th}}$, o_{H} is predicted as an object,
- if the highest prediction value $o_{\mathrm{H}} \leqslant o_{\mathrm{Th}}$, o_{H} is predicted as an indiscernible result (noise).

For this specific problem, the optimization task is to determine a o_{Th} that minimizes the number of indiscernible results, while maximizing the accuracy of discernible results.

Chapter 5

Results and Discussion

In this chapter, results of the proposed MPA variants in chapter 4 are evaluated and discussed. At the beginning of this chapter, two datasets are introduced, including a dataset with synthetic TC-Hists and a dataset with real TC-Hists measured by a LiDAR system. Afterwards, performance matrices used in the evaluation are presented. Subsequently, MPA with different components and parameters is evaluated and the optimal configuration is determined based on the dataset with synthetic TC-Hists. In addition, a lateral comparison among different distance prediction algorithms and possible optimization approaches are provided and analyzed. Finally, the performance of different MPA variants on the dataset with real TC-Hists is provided and discussed.

5.1 Datasets

Two datasets are created and specified in this section.

5.1.1 Dataset D1

The first dataset D1 is based on MCM using a simulation tool. The tool generates TC-Hists according to (4.8) and (4.10). The distribution is derived with the given background photon rate r_{B}, the object distance d_{Obj}, and the received laser photon rate r_{L}. D1 consists of 96,000 TC-Hists. T_{P}, r_{L}, and N_{M} in each TC-Hist is set to 5 ns, 10 MHz, and 400, respectively. The given r_{B} are 1, 2, ..., 8 MHz and the simulated d_{Obj} are from 0.5 m to 60 m, with an interval of 0.5 m. Each TC-Hist consists of 1,310 bins with a bin width of $T_{\mathrm{TDC}} = 312.5\,\mathrm{ps}$ resulting in a maximum detection distance of 61.26 m. The TC-Hists are evenly distributed under each combination of the aforementioned parameters, i.e., each combination includes 100 TC-Hists. In the following work, D1 is divided into training dataset $\mathrm{D1}_{\mathrm{Train}}$ (50 %), validation dataset $\mathrm{D1}_{\mathrm{Val}}$ (20 %), and test dataset $\mathrm{D1}_{\mathrm{Test}}$ (30 %) for evaluation. The division is carried out manually to ensure that each dataset includes all simulation conditions.

5.1.2 Dataset D2

The second dataset D2 originates from the LiDAR system “Owl” developed by Fraunhofer IMS (mentioned as the “Owl” in the following context). Relevant parameters of the “Owl” are specified as follows: The used lasers emit at 905 nm wavelength with 75 W peak power, 10 kHz pulse repetition rate, and 18.75 ns pulse width resulting in a mean optical emission power of 11.25 mW. The bandpass of the equipped optical filter is 905 ± 30 nm. D2 was accumulated in a basement tunnel. The measurement environment is specified as: The total length of the tunnel is 25 m. The target object is a 1.5 m×1.5 m-sized white board. One experimental setup is shown in Figure 5.1. During the data collection process, the target object was located at seven different distances. Since the objective of this work is to infer the object distance from the raw TC-Hists, the calibration and the offset of the LiDAR sensor are not part of the discussion. Therefore, after setting up the object, the true object distances refer to $\{3.19, 6.32, 11.38, 13.44, 16.77, 18.78, 21.59\}$ m, which are calculated based on 20,000 measurements in the environment with minimized background light ($r_{\mathrm{B}} < 1$ kHz). During the accumulation of D2, a halogen floodlight was applied and was aimed at the object in order to simulate high background light conditions. The detected background photon rate r_{B} is within $(7, 9)$ MHz. For each distance, 1000 TC-Hists were taken and each TC-Hist contains 400 measurements.

5.2 Performance Metrics

In this section, performance metrics for the evaluation of the MPA components are introduced.

Figure 5.1: Front view of experimental setup. The total length of the basement tunnel is 25 m. A 1.5 m×1.5 m-sized white board is used as the target object. The LiDAR system "Owl" has a two-line SPAD detector array. The lines are positioned 1 degree apart vertically to the ground. Each line is horizontal to the ground and contains 192 pixels. A halogen floodlight is aimed at the object, which creates a maximum r_{B} of 9 MHz. During the experiment, the object was placed at different distances. The measurement data was accumulated from the 96th pixel of the upper detector line.

5.2.1 Metrics of Feature Extraction

In the physics-guided feature extraction, raw TC-Hists are pre-processed by three convolution kernels. To characterize the proportion of the desired signals, SNR after the convolution is observed. According to [100], SNR in a TC-Hist is defined as

$$\mathrm{SNR} = \frac{C_{\mathrm{Obj}} - \mu_{\mathrm{B,\,Obj}}}{C_{\mathrm{Obj}}}\ , \tag{5.1}$$

where C_{Obj} represents the received photon count and $\mu_{\mathrm{B,\,Obj}}$ represents the expected count of background photons at the object distance. Another important criterium is the accuracy of the feature extraction Acc_{F}, defined as

$$Acc_{\mathrm{F}} = \frac{N_{\mathrm{True}}}{N_{\mathrm{S}}}\ , \tag{5.2}$$

where N_{True} denotes the number of true feature groups. A true feature group must meet the following condition: $\exists\ b_{\mathrm{Nst}}$ in this feature group, that

$$(1-\Delta E)d_{\mathrm{Obj}} \leqslant d_{\mathrm{TDC}}\, b_{\mathrm{Nst}} \leqslant (1+\Delta E)d_{\mathrm{Obj}} \ \text{ with } \ \Delta E = 0.05\ , \tag{5.3}$$

where b_{Nst} represents the bin nearest to the object distance d_{Obj} in a feature group F_{G}. ΔE refers to the factor of a tolerable error margin. According to (5.3), short-range objects have a narrower tolerable error margin than long-range objects, which is suitable for a coarse object distance determination mentioned in section 1.2. In addition, RMSE between b_{Nst} and d_{Obj} given by

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N_{\mathrm{S}}} (d_{\mathrm{TDC}}\, b_{\mathrm{Nst},\,i} - d_{\mathrm{Obj},\,i})^2}{N_{\mathrm{S}}}} \tag{5.4}$$

is used to evaluate the influence of feature extraction on distance precision.

5.2.2 Metrics of Prediction Algorithms

The assessment of a prediction algorithm involves performance metrics in multiple aspects. In terms of the proposed FNN in MPA, which performs as a classifier, the classification accuracy Acc_{C} measures the capacity of FNN to locate the object sub-region and is given by

$$Acc_{\mathrm{C}} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}\ . \tag{5.5}$$

Besides, based on the confusion matrix introduced in subsection 2.3.4, **P**ositive **P**redictive **V**alue (PPV) (also mentioned as precision or specificity), defined as

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}\ , \tag{5.6}$$

True **P**ositive **R**ate (TPR) (also mentioned as recall or sensitivity), defined as

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \ , \tag{5.7}$$

and the F1-Score, which measures the overall performance of PPV and TPR, defined as

$$\text{F1-Score} = \frac{2\,\mathrm{PPV} \times \mathrm{TPR}}{\mathrm{PPV} + \mathrm{TPR}} = \frac{2\,\mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \ , \tag{5.8}$$

are used to assess algorithm performance as well. Moreover, to evaluate the ordering ability of the algorithm, **A**rea **U**nder **C**urve (AUC) of **R**eceiver **O**perating **C**haracteristic (ROC) curve is applied.

In addition to the aforementioned classification metrics, the distance prediction accuracy Acc_{D}, defined as

$$(1 - \Delta E) d_{\mathrm{Obj}} \leqslant d_{\mathrm{Pred}} \leqslant (1 + \Delta E) d_{\mathrm{Obj}} \ \text{ with } \ \Delta E = 0.05 \ , \tag{5.9}$$

represents the proportion of correct distance predictions d_{Pred} with the given ΔE. It is applied to measure the capacity of the algorithms to locate the exact object position. Each MPA variant is evaluated individually using these metrics and the optimal parameters are determined correspondingly.

Finally, the comparison between different prediction algorithms is carried out regarding Acc_{C}, F1-Score, AUC, and Acc_{D}. Except for the prediction performance, the timing performance and the memory occupancy of different algorithms are equally important. They are compared as well in order to assess the implementation ability of each variant.

5.2.3 Metrics of Optimization Methods

Optimization methods are dedicated to improving the efficiency and reliability of MPA variants. These methods have different emphases and therefore require different metrics. In the ablation study of the background suppression and the stride analysis, Acc_{D} and the processing time are observed easily obtained. In the threshold analysis, the impact of a threshold o_{Th} is studied based on o_{H}. As mentioned in subsection 4.5.3, the evaluation of MPA changes to a binary classification problem. Under this specific context, positive samples refer to object information, while negative samples refer to noise. Thus, TP, FP, TN, and FN becomes

- TP′: The number of correct predictions, where o_{H} are predicted as object information.
- FP′: The number of false predictions, where o_{H} are predicted as object information.
- TN′: The number of correct predictions, where o_{H} are predicted as noise.
- FN′: The number of false predictions, where o_{H} are predicted as noise.

Apparently, one optimization approach is to maximize TP′ and TN′. Therefore, an accuracy metric Acc_{Th} depending on o_{Th} is defined as

$$Acc_{\mathrm{Th}} = \frac{\mathrm{TP}' + \mathrm{TN}'}{\mathrm{TP}' + \mathrm{FP}' + \mathrm{TN}' + \mathrm{FN}'} \ . \tag{5.10}$$

Table 5.1 summarizes the applied metrics for different MPA processes.

Table 5.1: Performance metrics for different MPA processes.

	MPA Process			
Metric	Feature Extraction	Prediction	Correlation	Optimization
SNR	√			
RMSE	√			
Acc_{F}	√			
Acc_{C}		√		
F1-Score		√		
AUC		√	√	
Acc_{D}		√	√	√
Acc_{Th}				√
Timing	√	√	√	√
Memory	√	√	√	√

5.3 Parameter Determination

In this section, components in MPA, including the physics-guided feature extraction, two prediction algorithms (FNN and NBC), and the correlation analysis are evaluated with a group of parameters and the optimal settings are determined according to their performance.

5.3.1 Parameters of Feature Extraction

Three kernels introduced in subsection 4.2.1 are evaluated regarding SNR, Acc_F, and RMSE. First, calculated SNRs of each kernel are divided into 20 sub-groups and plotted as a histogram, as shown in Figure 5.2. The statistic of the exponential and the constant kernel are almost identical, while SNR of the Gaussian kernel is smaller. In addition, the first SNR groups on the left of all kernels have considerable values. They originate mostly from the TC-Hists with high r_B and large d_Obj. These TC-Hists have extremely few laser photons reflected by the object and thus contain barely any object information. Afterwards, Acc_F and RMSE of three kernels according to (5.2) and (5.4) are demonstrated in Figure 5.3. It is observed that all kernels perform similarly. In general, Acc_F increases rapidly in the initial period and gradually stabilizes as N_F increases. Besides, RMSE decreases significantly when $N_\mathrm{F} < 5$ and starts to saturate when $N_\mathrm{F} > 10$. Furthermore, RMSE is smaller than the pulse width ($N_\mathrm{P}\, d_\mathrm{TDC} = 0.75\,\mathrm{m}$) when

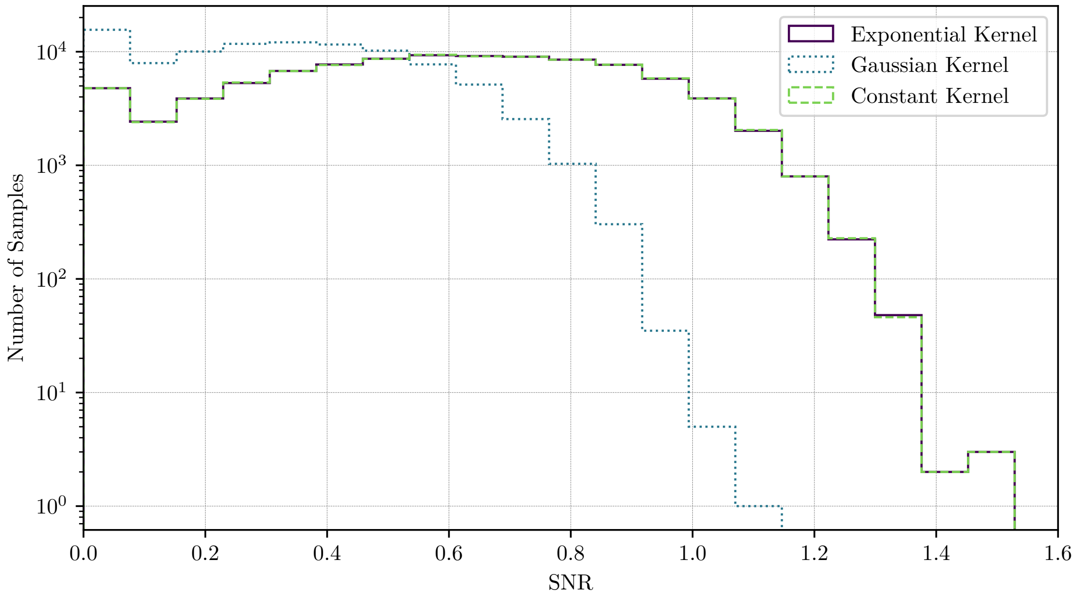


Figure 5.2: SNR statistics with different kernels. The x-axis refers to the value of SNR and the y-axis refers to the number of samples within each value range in D1.

$N_\mathrm{F} > 12$. This implies that most feature groups have at least one f_n containing laser photons. It should be noted that an increase in N_F leads to a linear increase in feature extraction time and an exponential increase in the computational complexity of a subsequent processing method. Therefore, a trade-off between N_F and Acc_F is required in practice.

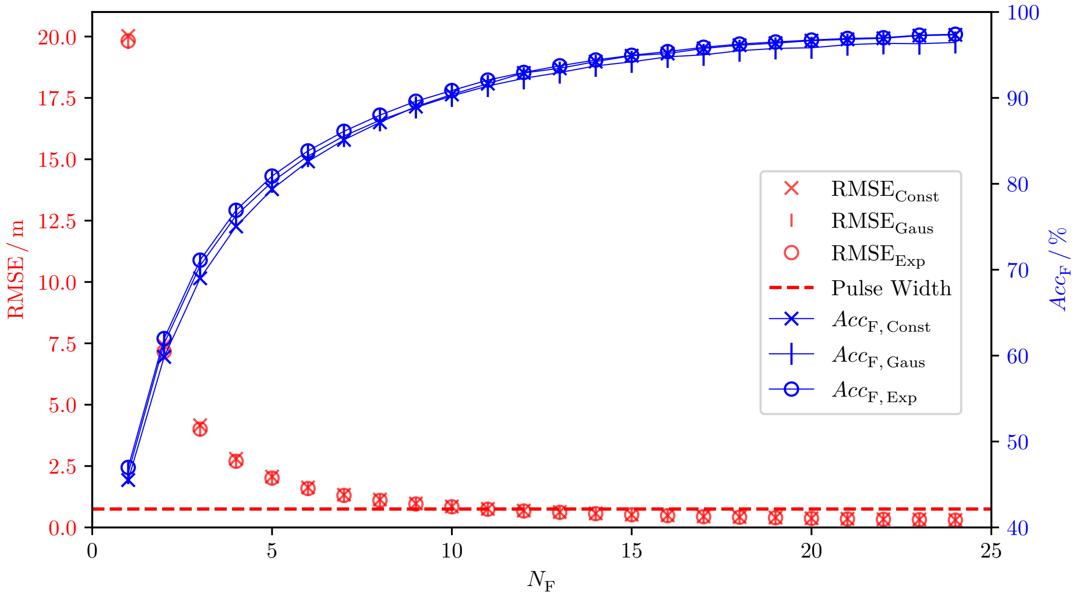


Figure 5.3: SNR comparison of three convolution kernels based on D1. The left y-axis refers to RMSE and the right y-axis refers to the accuracy of feature extraction. The x-axis refers to the number of features.

As a result, the constant kernel is selected, since it achieves a good general SNR and has the best real-time adaptability. A detailed SNR of the constant kernel regarding distance and background light is given in Table 5.2. The minimum average of SNR is 0.16. It means d_Obj can be still represented by a local maximum and explains the reason that Acc_F reaches 92.90 % with only 12 features out of 1310 values. The following analysis focuses on the excess zone between rising and saturated regions, i.e., $N_\mathrm{F} \in \{8, 12, 16, 24\}$, to obtain the most efficient parameters with the consideration of prediction accuracy and timing performance.

5.3.2 Parameters of Neural Network

The parameter determination of an FNN starts with a preliminary grid search. Hyperparameters of FNN are selected and listed in Table 5.3. A discussion on the

Table 5.2: Average SNR after convolution by constant kernel. They are divided into 48 sub-groups regarding d_{Obj} and r_{B}.

	r_{B} / MHz							
d_{Obj} / m	1	2	3	4	5	6	7	8
$(0, 10]$	1.00	0.93	0.87	0.82	0.77	0.73	0.69	0.65
$(10, 20]$	0.97	0.88	0.79	0.72	0.66	0.59	0.54	0.50
$(20, 30]$	0.94	0.81	0.71	0.62	0.55	0.48	0.42	0.37
$(30, 40]$	0.91	0.76	0.64	0.54	0.46	0.39	0.32	0.27
$(40, 50]$	0.88	0.71	0.58	0.47	0.38	0.31	0.25	0.21
$(50, 60]$	0.85	0.66	0.52	0.41	0.31	0.26	0.20	0.16

learning rate, the batch size, the optimizer, the number of hidden layers, and the regularization method are excluded in the following discussion, since they show little relevance to the performance of MPA during the grid search. Therefore, they are pre-determined to fixed values. The focuses of this subsection are N_{F}, the number of hidden layer neurons N_{HN}, and the type of activation functions. A set of values for each is covered in Table 5.3 and their impacts on FNN based on $\mathrm{D1}_{\mathrm{Val}}$ are discussed.

In Table 5.4, the F1-Score and AUC of the FNNs with different N_{F} are shown. It is observed that the standard deviations of both metrics are extremely small. This means that N_{HN} and the activation functions have barely influence on the F1-Score and AUC. However, the average F1-Score decreases, while the average of AUC increases with the increase of N_{F}.

To discuss the impacts of N_{F}, N_{HN}, and activation functions on the prediction accuracy, the transitions and trends of $Acc_{\mathrm{C,\,Val}}$ and $Acc_{\mathrm{D,Val}}$ are shown in Figure 5.4. In general, the FNN without activation function and hidden layer achieves the best overall accuracy and stability with respect to both $Acc_{\mathrm{C,\,Val}}$ and $Acc_{\mathrm{D,\,Val}}$. Besides, in Figure 5.4 (a) and (b), it is clearly seen that $Acc_{\mathrm{C,\,Val}}$ decreases with the increase of N_{F}. However, this trend is not observed in Figure 5.4 (c) and (d). Except for $N_{\mathrm{F}} = 8$, with which $Acc_{\mathrm{D,Val}}$ is distinctly low, FNNs with $N_{\mathrm{F}} = \{12,\ 16,\ 24\}$ have only minor difference on $Acc_{\mathrm{D,\,Val}}$. It should be mentioned that

Table 5.3: FNN hyperparameters for evaluation of MPA. The parameters in the left column are fixed values, while the parameters in the right column correspond to multiple possible values.

Parameter	Value	Parameter	Value
Learning rate	0.001	N_{F}	$\{8,\ 12,\ 16,\ 24\}$
Batch size	64	Hidden layers	0 or 1
Optimizer	Adam	N_{HN}	$\{0,\ N_{\mathrm{F}},\ N_{\mathrm{F}}+10,\ N_{\mathrm{F}}+20\}$
Regularization	None	Activation function	Sigmoid, ReLU, None

Table 5.4: Average and standard deviation of classification metrics regarding N_{F}. Each value is calculated with a corresponding N_{F} and all combinations of hidden layers, N_{HN}, and activation functions specified in Table 5.3.

	F1-Score / %		AUC / %	
N_{F}	Average	Standard Deviation	Average	Standard Deviation
8	71.24	0.07	94.05	0.07
12	69.77	0.08	94.24	0.02
16	69.12	0.09	95.02	0.04
24	66.85	0.11	95.34	0.03

in exchange for a 0.3 % increment on $Acc_{\mathrm{D,Val}}$, the FNN with $N_{\mathrm{F}} = 24$ has four times as much computation and memory occupation as the FNN with $N_{\mathrm{F}} = 12$.

To further clarify the cause of different characteristics between Figure 5.4 (a) (b) and Figure 5.4 (c) (d), a specific example is provided in Figure 5.5. The figure shows a segment of a TC-Hist after convolution and normalization in D1. In this segment, the object bin b_{Obj} lies exactly on the boundary of two sub-regions. This means each bin around the feature border (between bin 565 and 581) contains at least a portion of laser photons. In this case, a small random fluctuation can cause a position deviation of the maximum value among these bins. As shown in Figure 5.5, the maximum value is shifted from bin 566 (26.51 m) to bin 568 (26.61 m).

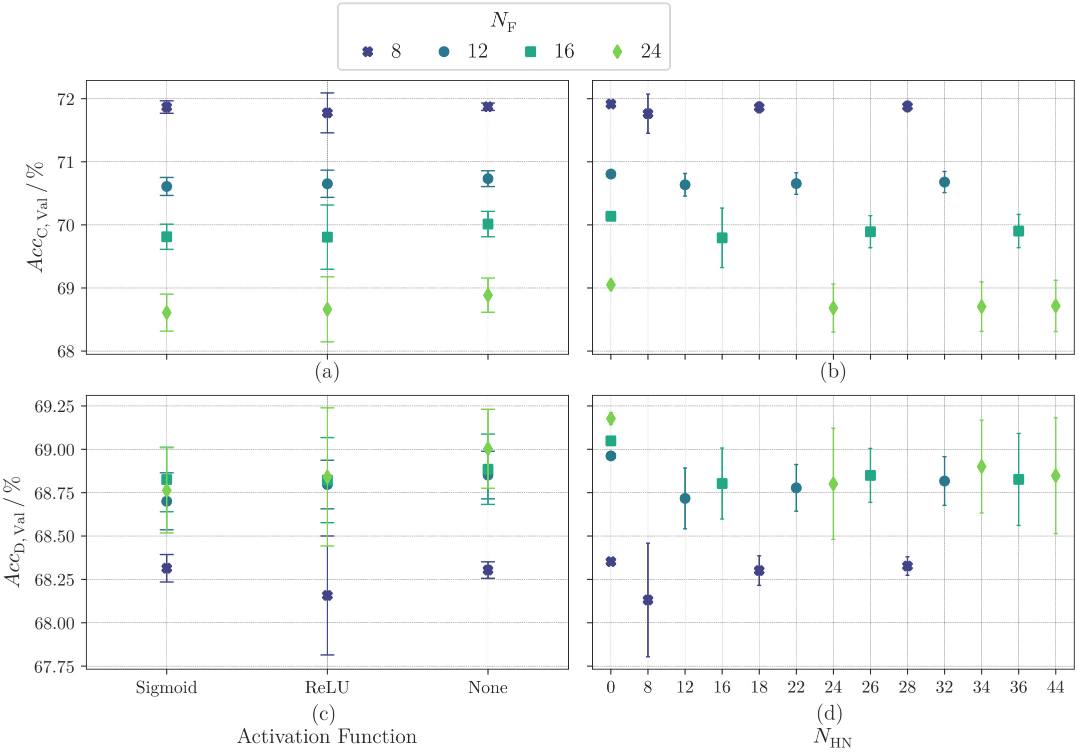


Figure 5.4: Accuracy of FNN in D1$_{\mathrm{Val}}$ with respect to different activation functions, N_{HN}, and N_{F}. (a) Classification accuracy regarding activation functions. (b) Classification accuracy regarding N_{HN}. (c) Distance prediction accuracy regarding activation functions. (d) Distance prediction accuracy regarding N_{HN}. Each point refers to the average accuracy and the corresponding boundary lines refer to the maximal/minimal accuracy. In (a) and (b), the points and the boundary lines are calculated using all N_{HN}. In (c) and (d), they are calculated using all activation functions.

After the subsequent processing, the right region is inferred as the region where the object locates, and 26.61 m is inferred as d_{Pred}. Thus, the classification result is annotated as false, while the distance prediction result is still counted as correct, since the deviation is still within the defined ΔE according to (5.9).

In fact, the divergence between classification and distance prediction can be separated into two cases:

- Case 1: The classification is false, while the distance prediction is correct.
- Case 2: The classification is correct, while the distance prediction is false.

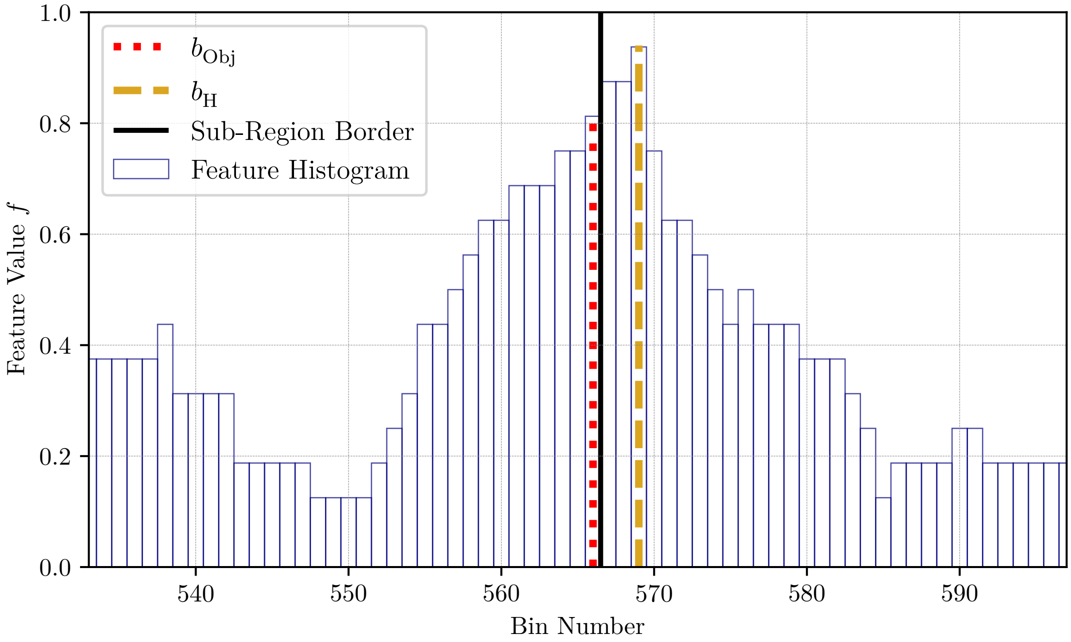


Figure 5.5: A section of TC-Hist after convolution and normalization. Its classification result is false, while its distance prediction result is correct. The object bin number b_{Obj} locates at the left sub-region and is represented by a dotted vertical line, while the predicted bin number b_{H} locates in the right sub-region and is marked by a dashed vertical line.

The example in Figure 5.5 belongs to case 1. A statistical overview of these cases in $\mathrm{D1}_{\mathrm{Val}}$ is provided in Figure 5.6. As shown in Figure 5.6 (a) and (b), the subset of case 1 becomes larger with the increase of N_{F}. Besides, most of the TC-Hists appear with large distances and small errors. For example, with $N_{\mathrm{F}} = 12$, the proportion of case 1 is 0.97 % in D1. Therein, 46.40 % of the distance predictions on this subset have errors within 0.43 m. The statistics of the subset with case 2 is illustrated in Figure 5.6 (c) and (d). In contrast to case 1, it becomes smaller with the increase of N_{F}. Apparently, most of the TC-Hists with case 2 appear within the object distances of 5 m. The corresponding error bound of these TC-Hists is smaller than ±0.25 m according to (5.9). Since T_{P} in D1 is 5 ns, i.e., the laser pulse is about 0.75 m wide, the derivation of the local maximum can be easily greater than the error bound, resulting in an occurrence of case 2.

In the field of ADAS, the distance prediction result of MPA is of more interest compared to the classification results. Therefore, the general system performance

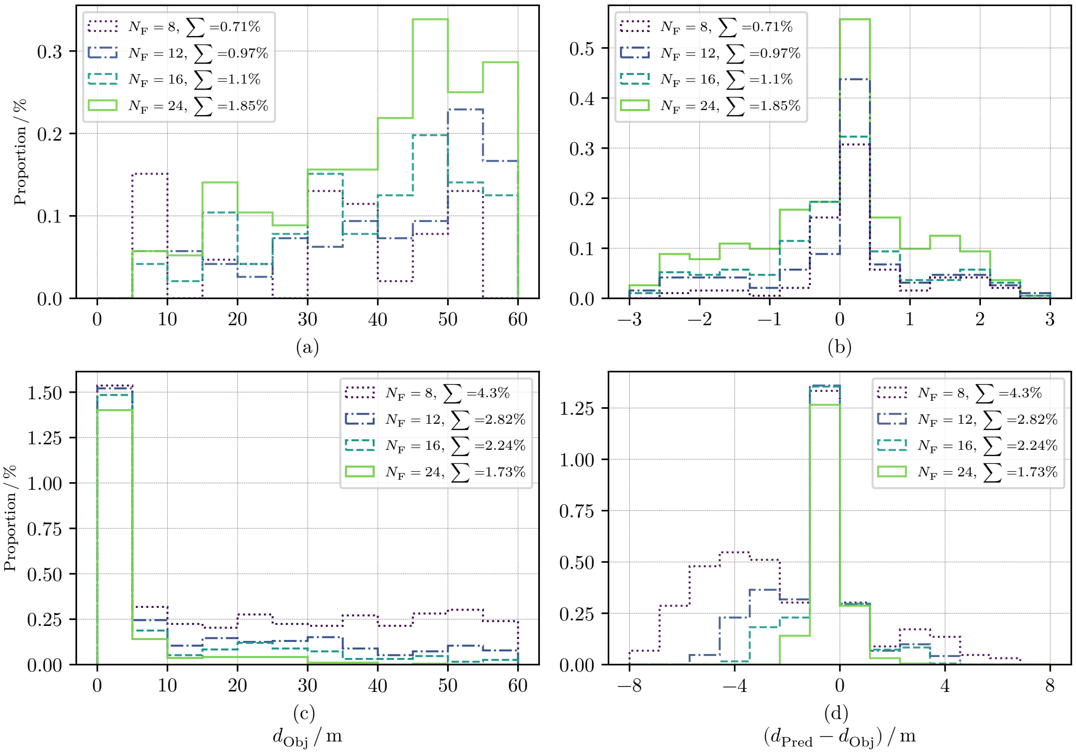


Figure 5.6: Divergence of classification and distance prediction results. The symbol $\sum$ represents the corresponding proportion of the subset in $D1_{Val}$. (a) Statistics of case 1 regarding distance, bin size = 5 m. (b) Statistics of case 1 regarding errors, bin size = 0.43 m. (c) Statistics of case 2 regarding distance, bin size = 5 m. (d) Statistics of case 2 regarding errors, bin size = 1.14 m. The y-axis of each subplot represents the correlated proportion of TC-Hists in $D1_{Val}$.

assessment is only conducted regarding Acc_D. Hence, the best parameters and evaluation criteria of FNN are summarized as: $N_F = 12$, $N_{HN} = 0$, and activation function except the output layer is discarded. In the following context, MPA using FNN with these parameters is mentioned as FNN-MPA.

5.3.3 Parameters of Naive Bayes Classifier

Parameter optimization of NBC starts with an observation on the statistics of the input $P_{Norm,n}$. By taking $N_F = 12$ as an example, the statistics of sub-regions when the object locates within sub-region 1 is shown in Figure 5.7. It is observed that the statistics of $P_{Norm,n}$ approximate normal-like distributions. According to Figure 5.7, the average value μ_1 of the sub-region 1 is higher than

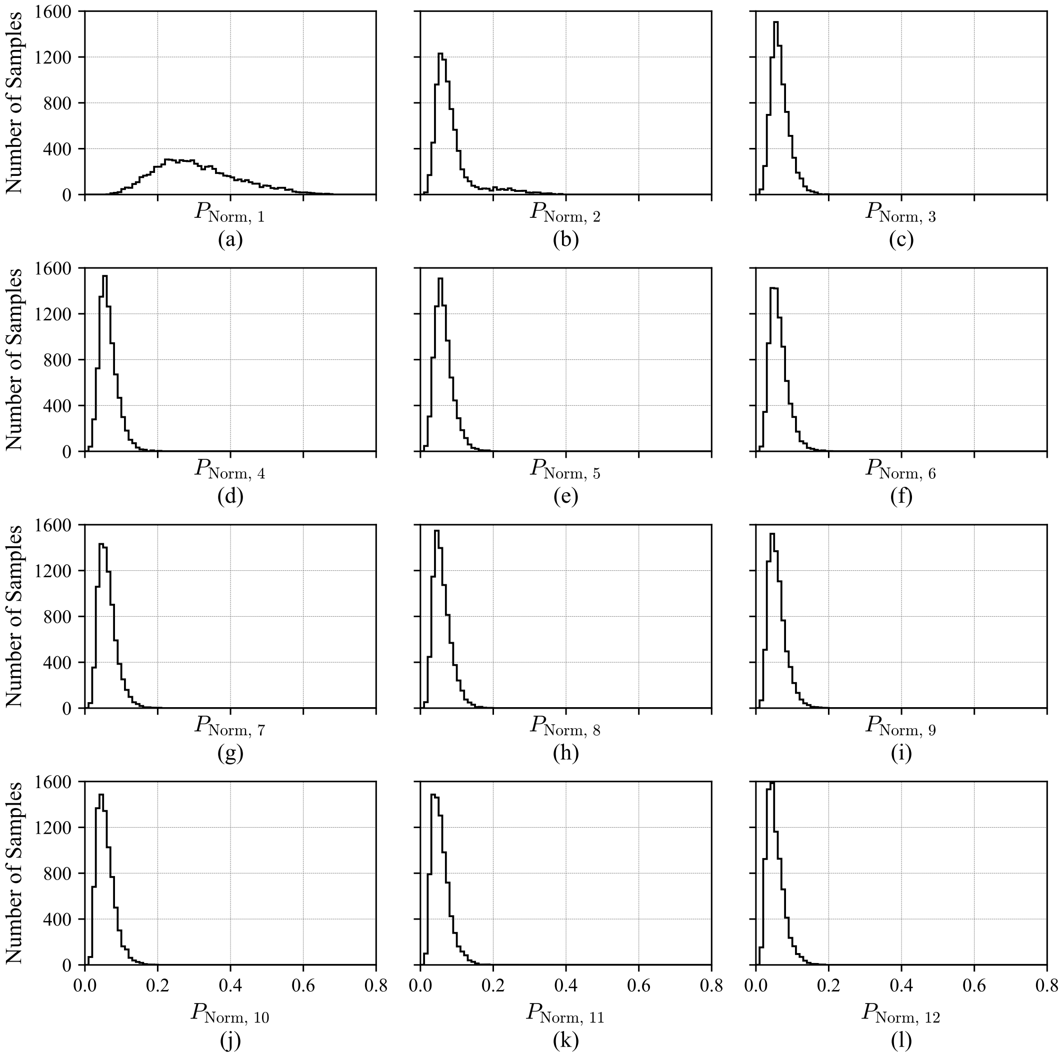


Figure 5.7: Magnitude distribution of $P_{\text{Norm},n}$ from sample group with class 1 in each sub-region. The bin width is equal to 0.01. Sub-plots (a) - (l) represent the corresponding statistics in sub-region 1 - 12.

that of other sub-regions due to the presence of the object. In addition, an abnormal rebound in the value region of [0.2, 0.4] is seen in 5.7 (b), resulting in the shape of a bimodal distribution. To explain this, samples in Figure 5.7 (b) are separated into two sub-groups, as shown in Figure 5.8. Since the width of a laser pulse is 0.75 m and the boundary of sub-region 1 and sub-region 2 is at 5.06 m, the laser pulse reflected by $d_{\text{Obj}} = 5\,\text{m}$ is distributed in both sub-region 1 and 2. Thus, although $d_{\text{Obj}} = 5\,\text{m}$ locates in sub-region 1, it brings impacts to the left of sub-region 2 as well. $P_{\text{Norm},n}$ caused by this case is typically higher than $P_{\mathrm{Norm},\,n}$ from noise, resulting in a side-peak in the statistics. This situation is called the "boundary effect". As shown in Figure 5.8, by separating out the distances causing the boundary effect, the statistics are well decomposed into two single-peak distributions. Although the aforementioned examples and statistics are based on a part of D1, where the objects locate within sub-region 1, their characteristics are representative. After traversing all samples in D1, the general characteristics are summarized as:

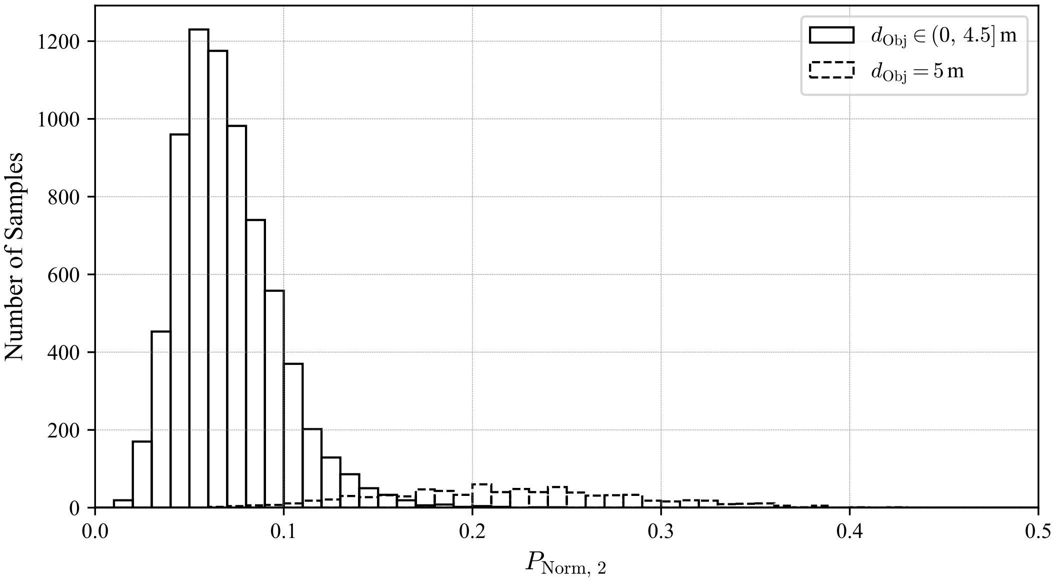


Figure 5.8: Decomposition of statistics in Figure 5.7 (b). The samples are separated into a subset causing the boundary effect ($d_{\mathrm{Obj}} = 5\,\mathrm{m}$) and a subset irrelevant to the boundary effect ($d_{\mathrm{Obj}} \in \{0.5,\ 1.0,\ 1.5,\ 2.0,\ 2.5,\ 3.0,\ 3.5,\ 4.0,\ 4.5\}\,\mathrm{m}$).

- If d_{Obj} of samples are within sub-region n, the statistic of the corresponding $P_{\mathrm{Norm},\,n}$ approximates a normal-like distribution with a relatively high $\overline{P}_n$ and s_n.
- If d_{Obj} of samples are outside sub-region n, the statistic of the corresponding $P_{\mathrm{Norm},\,n}$ approximates a normal-like distribution with a relatively low $\overline{P}_n$ and s_n.
- If d_{Obj} of samples are within sub-region n, the boundary effect is observed at sub-region $n-1$ and sub-region $n+1$. In this case, the statistics of the

corresponding $P_{\mathrm{Norm},\,n-1}$ and $P_{\mathrm{Norm},\,n+1}$ approximate the bimodal distributions.

To further specify the statistical characteristics with different r_{B}, samples in D1 are divided into

$$N_{\mathrm{NBC}} = N_{\mathrm{F}}\,N_{\mathrm{B}} = 96 \tag{5.11}$$

subgroups, where N_{B} refers to the number of background conditions. Afterwards, observations are carried out in three aspects: 1) The statistical parameters in the sub-region where the object locates, i.e., $\overline{P}_{\mathrm{Obj},\,n}$ and $s_{\mathrm{Obj},\,n}$. 2) The statistical parameters of noise, i.e., $\overline{P}_{\mathrm{Noise},\,n}$ and $s_{\mathrm{Noise},\,n}$. 3) The statistical parameters of the boundary effect, i.e., $\overline{P}_{\mathrm{BE},\,n}$ and $s_{\mathrm{BE},\,n}$.

Figure 5.9 illustrates $\overline{P}_{\mathrm{Obj},\,n}$ and $s_{\mathrm{Obj},\,n}$ of 96 classes. In general, $\overline{P}_{\mathrm{Obj},\,n}$ slowly decreases with the increase of the sub-region number. Besides, as r_{B} increases, $\overline{P}_{\mathrm{Obj},\,n}$ decreases from 0.48 to 0.07. Its decreasing rate is rapid at the beginning and gradually slows down. In addition, $s_{\mathrm{Obj},\,n}$ with low r_{B} fluctuates around 0.09 with the increase of the sub-region number. However, its decreasing rate becomes evidently with the increase of r_{B} and $s_{\mathrm{Obj},\,n}$ is finally smaller than 0.04.

Figure 5.10 summarizes $\overline{P}_{\mathrm{Noise},\,n}$ and $s_{\mathrm{Noise},\,n}$ of 96 classes. Compared to $\overline{P}_{\mathrm{Obj},\,n}$, $\overline{P}_{\mathrm{Noise},\,n}$ decreases at a clearly slower rate (from 0.10 to 0.05) with the increase of the sub-region number. Furthermore, in contrast to $\overline{P}_{\mathrm{Obj},\,n}$, $\overline{P}_{\mathrm{Noise},\,n}$ is inversely proportional to r_{B}. Besides, $s_{\mathrm{Noise},\,n}$ remains small and stable in all sub-regions, while it increases as r_{B} increases and finally reaches 0.03. Combined with aforementioned characteristics of the object features, it is noticed that $\overline{P}_{\mathrm{Obj},\,n}$ converges to $\overline{P}_{\mathrm{Noise},\,n}$ and $s_{\mathrm{Obj},\,n}$ converges to $s_{\mathrm{Noise},\,n}$ with the increase of r_{B} and the sub-region number. The decreasing statistical difference between object features and noise features is due to the decrease of SNR in the raw TC-Hist. In particular, $\overline{P}_{\mathrm{Obj},\,n}$ with $r_{\mathrm{B}} \in \{6,\ 7,\ 8\,\}$ MHz in sub-region 12 are highly similar to each other and close to $\overline{P}_{\mathrm{Noise},\,n}$. This brings challenges to distinguish them and makes a distance prediction crucial.

To observe the boundary effect, two distance groups with the object distances close to boundaries are formed. Therein, group 1 refers to the first distances on the left side of boundaries, and group 2 refers to the first distances on the right side of boundaries. Figure 5.11 illustrates the statistics of group 1. It is observed that $\overline{P}_{\mathrm{BE},\,n}$ and $s_{\mathrm{BE},\,n}$ strongly relates to the difference between d_{Obj} and sub-region boundaries. As the object distance moves away from a sub-region boundary,

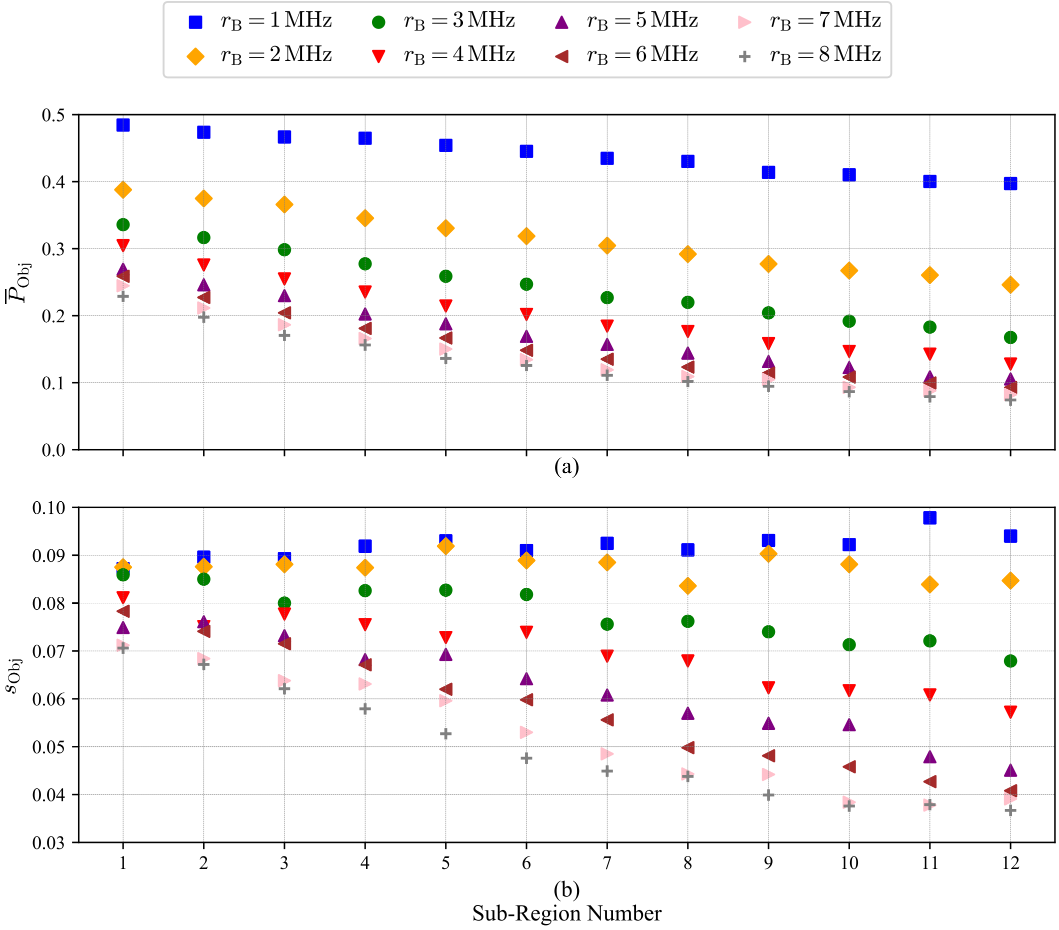


Figure 5.9: Statistical parameters of object features $P_{\mathrm{Norm},\,n}$ in each sub-region. (a) Average of object features $P_{\mathrm{Norm},\,n}$. (b) Standard deviation of object features $P_{\mathrm{Norm},\,n}$.

$\overline{P}_{\mathrm{BE},\,n}$ and $s_{\mathrm{BE},\,n}$ decrease. With $r_\mathrm{B} = 1\,\mathrm{MHz}$, it is clearly seen that $\overline{P}_{\mathrm{BE},\,n}$ to the left side of boundaries changes periodically as the sub-region number increases. This trend is observed with the rest of r_B as well. The reason is that the interval of the distances in D1 is not equal to d_TDC, which generates periodic changes between d_Obj and sub-region boundaries. Besides, $\overline{P}_{\mathrm{BE},\,n}$ is inverse-proportional to r_B and the decrease of $\overline{P}_{\mathrm{BE},\,n}$ due to r_B is more significant with large sub-region numbers. As a result, not all samples in group 1 create visible side-peaks. In Figure 5.11 (a), $\overline{P}_{\mathrm{BE},\,n}$ of 40 out of 88 points drop to around 0.10, which is the value range of noise features as well. This means that half of the samples in group 1 have similar characteristics as noise and cause barely side-peaks. Obviously,

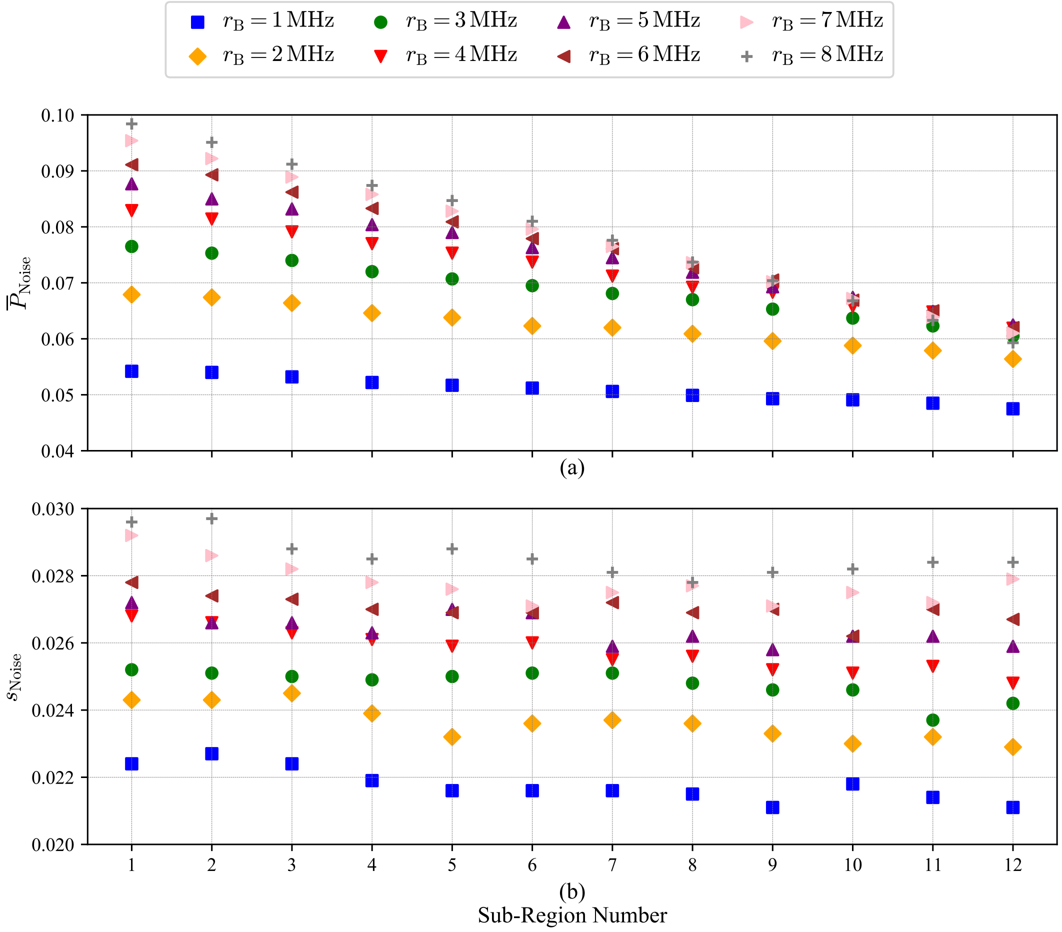


Figure 5.10: Statistical parameters of noise $P_{\text{Norm},\,n}$ in each sub-region. (a) Average of noise $P_{\text{Norm},\,n}$. (b) Standard deviation of noise $P_{\text{Norm},\,n}$.

the parameter characteristics of group 2 are symmetrical to group 1. Thus, the proportion P_{BE} of both groups creating visible side-peaks is estimated by

$$P_{\text{BE}} \approx \frac{N_{\text{BE}}}{N_{\text{BE}} + N'_{\text{BE}}} \frac{2\,(N_{\text{F}} - 1)}{120} \,. \tag{5.12}$$

where N_{BE} refers to the number of samples creating visible side-peaks and N'_{BE} refers to the rest number of samples, which are barely different from noise. 120 refers to the number of different distances in D1. With $N_{\text{F}} = 12$, $N_{\text{BE}} = 48$, and $N'_{\text{BE}} = 40$, P_{BE} accounts for 10.00 % of D1.

Based on the aforementioned observations, the following aspects are considered for parameter optimization:

- N_{F}: Similar to the FNN, the influence of $N_{\text{F}} = \{8, 12, 16, 24\}$ in NBC is

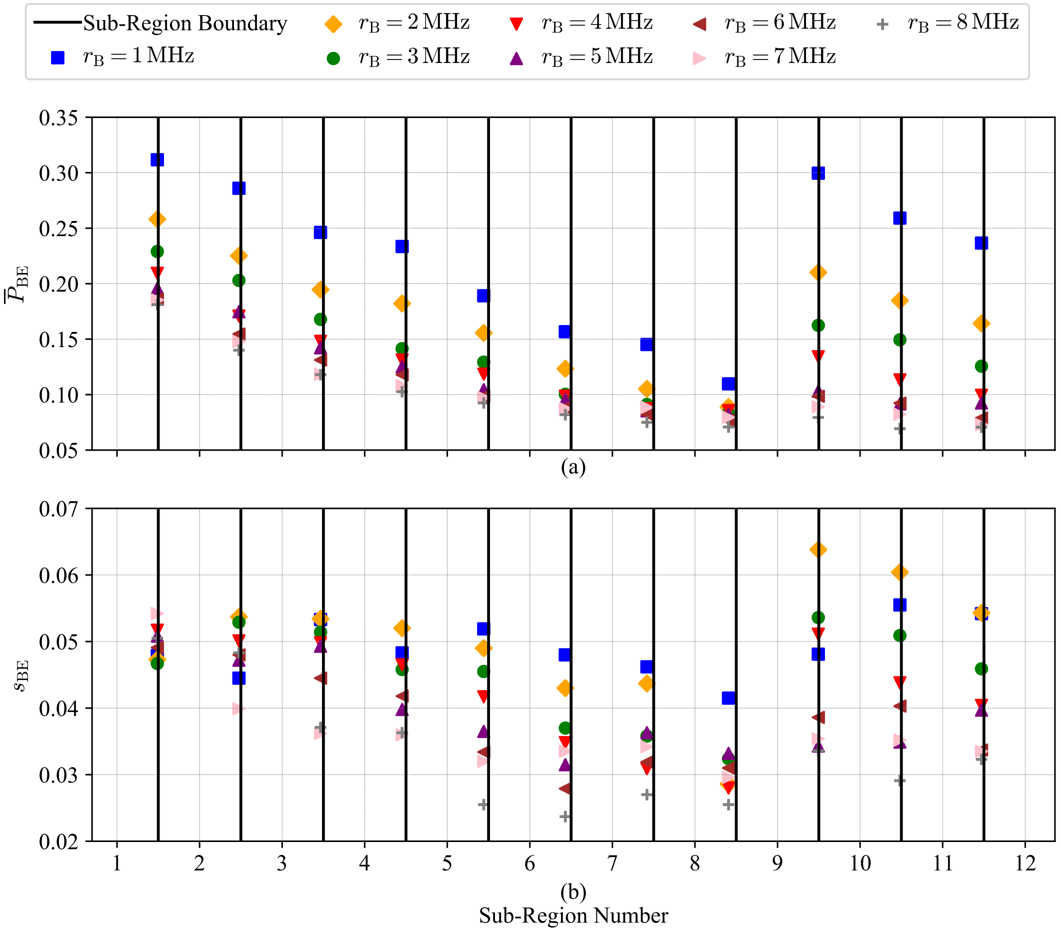


Figure 5.11: Statistical parameters of boundary features $P_{\mathrm{Norm},n}$ caused by distances in group 1. (a) Average of boundary features $P_{\mathrm{Norm},n}$. (b) Standard deviation of boundary features $P_{\mathrm{Norm},n}$. Each point refers to 30 samples. Vertical lines represent the positions of sub-region boundaries. Distances in group 1 = $\{5, 10, 15, 20, 25, 30, 35, 40, 45.5, 50.5, 55.5\}$ m. These distances create side-peaks in the statistics of the adjacent right-handed sub-regions.

evaluated. It should be noted that N_F in NBC influences not only Acc_F, but also the number of sub-region boundaries, which further affects the boundary effect.

- Modeling: The statistics of $P_{\mathrm{Norm},n}$ show normal-like distributions. Therefore, the optimal modeling is selected based on the evaluation among normal, log-normal, and multimodal functions.
- Boundary effect: In order to evaluate the degree of its impact, three strate-

gies are considered for the boundary effect, namely: single-peak modeling including boundary distances (S1), single-peak modeling excluding boundary distances (S2), and bimodal modeling with additional modeling for the boundary distances (S3).

- N_{NBC}: It is observed that $\overline{P}_n$ and s_n of object features, noise features, and the boundary effects change noticeably with the change of sub-region numbers and r_{B}. Therefore, two categorizations are investigated: (1) categorization regarding different sub-region numbers, i.e., $N_{\mathrm{NBC}} = N_{\mathrm{F}}$ and (2) categorization regarding both sub-region numbers and r_{B}, i.e., $N_{\mathrm{NBC}} = N_{\mathrm{F}}\,N_{\mathrm{B}}$. It should be mentioned that in the prediction phase, step 3 introduced in subsection 4.3.3 is only carried out with the classes corresponding to r_{B} of each input sample. This means that although the categorization with $N_{\mathrm{NBC}} = N_{\mathrm{F}}\,N_{\mathrm{B}}$ requires more computations in the modeling phase, it has the same computations in the prediction phase compared to the categorization with $N_{\mathrm{NBC}} = N_{\mathrm{F}}$.

Finally, the parameters are summarized in Table 5.5, where the rules are specified based on the defined models (M) and strategies (S) above as:

- M1-S1: Modeling using normal distributions and the strategy S1.
- M2-S1: Modeling using log-normal distributions and the strategy S1.
- M1-S2: Modeling using normal distributions and the strategy S2.
- M2-S2: Modeling using log-normal distributions and the strategy S2.
- M1-S3: Modeling using bimodal distributions and the strategy S3.
- M2-S3: Modeling using log-bimodal distributions and the strategy S3.

The results are shown in Figure 5.12 and discussions are separated into the following four aspects:

1) **Normal function (M1) vs. Log-normal function (M2)**: In general, the variants using log-normal functions (M2-S1, M2-S2, and M2-S3) have higher $Acc_{\mathrm{D,\,Val}}$ than the variants using normal functions (M1-S1, M1-S2, and M1-S3), if N_{F} and the applied strategy S are the same. This corresponds to the observation in Figure 5.7, where all sub-region statistics are slightly right-skewed. In this case,

Table 5.5: Selected NBC parameters for MPA.

Parameter Name	Value
N_F	$\{8,\ 12,\ 16,\ 24\}$
Rule	M1-S1, M2-S1, M1-S2, M2-S2, M1-S3, M1-S3
N_NBC	$\{N_\mathrm{F},\ N_\mathrm{B}\,N_\mathrm{F}\}$

a log-normal function normalizes the statistics to a standard normal distribution, which leads to a better model matching than using a normal function directly. The only exception is M1-S2 with $N_\mathrm{F} = 12$ having an accuracy of 69.99 %, which is 0.14 % higher than M2-R2 with $N_\mathrm{F} = 12$.

2) **S1 vs. S2 vs. S3**: The variants with S2 outperform the variants with S1. The reason is that the boundary distances cause deviations in $\overline{P}_n$ and s_n for single-peak modeling (i.e., with normal and log-normal functions). As shown in Figure 5.8, calculations of model parameters including the boundary distances increase both $\overline{P}_n$ and s_n, which in turn leads to a mismatch between the model and the statistics. In terms of bimodal distributions, the advantages of M1-S3 and M2-S3 are barely observed compared to M1-S2 and M2-S2, respectively. This result correlates to the observation in Figure 5.11, where side-peaks are only visible with low r_B and small sub-region numbers. An important fact is that a distance prediction is straightforward under these conditions, since statistics of object features are already unique to each other and easy to identify from noise. In terms of the harsh conditions (i.e., high r_B and large sub-region number), under which a distance prediction is challenging, impacts of the boundary effect is minor and can be neglected. This results in only minimal difference between S2 and S3. However, in the case of $N_\mathrm{F} = 24$, which involves the most boundaries, the impact of the boundary effect reaches its maximum. In this case, M2-S3 with $N_\mathrm{NBC} = N_\mathrm{F}\,N_\mathrm{B}$ shows a slightly better performance than M2-S2 and achieves the highest $Acc_{\mathrm{D,\,Val}}$ among all variants. Considering that a bimodal function requires twice the computations compared to a normal function, it is more efficient to achieve a reliable performance by applying S2.

3) $\boldsymbol{N_\mathrm{F}}$: $Acc_{\mathrm{D,\,Val}}$ of M1-S1 and M2-S1 increase slowly in the initial period with

the increase of N_F. As N_F further increases, they start to decrease except for the variant with $N_\mathrm{NBC} = N_\mathrm{F}\,N_\mathrm{B}$. In particular, $Acc_\mathrm{D,\,Val}$ of M1-S1 reduces to 67.65 %, when $N_\mathrm{NBC} = 24$. This is because the number of boundaries is proportional to N_F, resulting in the increase of P_BE and the deviation of statistical parameters mentioned in aspect (2). Except for M1-S1 and M2-S1, $Acc_\mathrm{D,\,Val}$ of other variants are proportional to N_F. However, similar to the case in FNN, a large N_F not only requires large memory space, but also involves additional calculations for the binomial estimation, model parameters, and joint probabilities.

4) $\boldsymbol{N_\mathrm{NBC}}$: Apparently, $Acc_\mathrm{D,\,Val}$ with $N_\mathrm{NBC} = N_\mathrm{F}\,N_\mathrm{B}$ is not only higher, but also more stable than that with $N_\mathrm{NBC} = N_\mathrm{F}$. This means that N_F classes are insufficient to generalize the statistical characteristics in D1. Thus, it is necessary to perform models separately regarding different r_B with $N_\mathrm{NBC} = N_\mathrm{F}\,N_\mathrm{B}$.

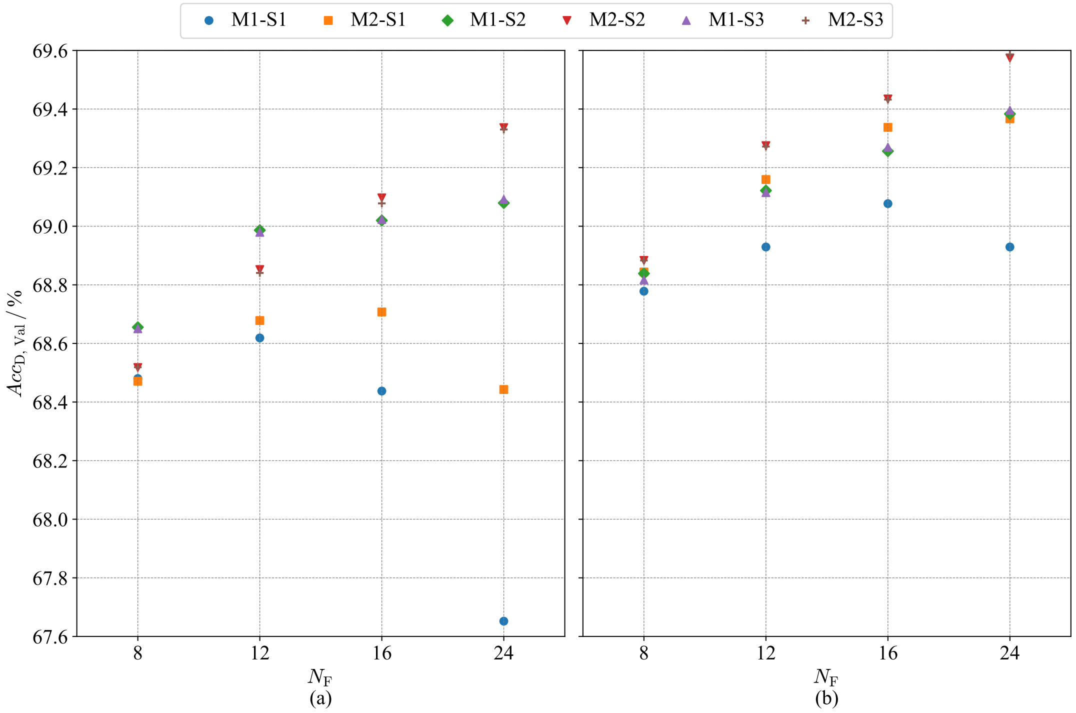


Figure 5.12: Distance prediction accuracy of NBC with different parameters in $\mathrm{D1_{Val}}$. (a) $Acc_\mathrm{D,\,Val}$ with $N_\mathrm{NBC} = N_\mathrm{F}$. (b) $Acc_\mathrm{D,\,Val}$ with $N_\mathrm{NBC} = N_\mathrm{F}\,N_\mathrm{B}$.

In summary, M1-S1 and M2-S2 with $N_\mathrm{NBC} = N_\mathrm{F}$ have the lowest $Acc_\mathrm{D,\,Val}$, since they are strongly influenced by the negative impacts of the boundary effect. Therefore, they are excluded as potential options. Besides, the variants with normal functions are excluded as well, because they are clearly inferior to the variants with log-normal functions for modeling of statistics in D1. Among the remaining options, the variants with large N_F achieve at most 0.82 % accuracy improvement while increasing both required memory space and computations. However, the variants with $N_\mathrm{NBC} = N_\mathrm{F}\, N_\mathrm{B}$ require the same number of calculations for prediction as the variants with $N_\mathrm{NBC} = N_\mathrm{F}$ and have similar performance to the variants with large N_F. This means that an increase in N_B is more efficient than an increase in N_F. Therefore, the optimal parameters for NBC are specified as:

- N_F: 12,
- Rule: M2-S2,
- N_NBC: 96.

It achieves $Acc_{\mathrm{D,\,Val}} = 69.31\,\%$ in $\mathrm{D1}_\mathrm{Test}$. In the following context, MPA using NBC with these parameters is mentioned as NBC-MPA.

5.3.4 Parameters of Correlation Analysis

To specify the impacts of α and β introduced in subsection 4.4.2, the following conditions are considered:

1. The time and space correlations are applied simultaneously. Three output groups are involved as reference data, i.e., $O_{t-1,\,i,\,j}$, $O_{t,\,i-1,\,j}$, and $O_{t,\,i+1,\,j}$.
2. The target object is assumed to be stationary and flat during the correlation, i.e., the object distances of $O_{t,\,i,\,j}$, $O_{t-1,\,i,\,j}$, $O_{t,\,i-1,\,j}$, and $O_{t,\,i+1,\,j}$ are the same.
3. The background light condition is assumed to be constant during the correlation.
4. Randomness of correlation is guaranteed by $O_{t,\,i,\,j} \neq O_{t-1,\,i,\,j} \neq O_{t,\,i-1,\,j} \neq O_{t,\,i+1,\,j}$.
5. To guarantee a fair comparison, the involved TC-Hists in different MPA variants are identical.
6. $\alpha \in [0.001,\ 1000]$.

7. $\beta \in \{16, 32, 64\}$.

The correlation is carried out in $\mathrm{D1}_{\mathrm{Test}}$. Figure 5.13 shows the accuracy difference $\Delta Acc_{\mathrm{D,\,Test}}$ after the correlation analysis. Therein, $\Delta Acc_{\mathrm{D,\,Test}}$ is given by

$$\Delta Acc_{\mathrm{D,\,Test}} = Acc_{\mathrm{D,\,Co}} - Acc_{\mathrm{D,\,Test}} \;, \tag{5.13}$$

where $Acc_{\mathrm{D,\,Co}}$ refer to the distance prediction accuracy in $\mathrm{D1}_{\mathrm{Test}}$ caused by the correlation analysis. According to (4.33), G is proportional to α and β determines the width of the gain area. When $\alpha = 0.001$ and $\beta = 16$, the influence of the reference is limited, resulting in small $\Delta Acc_{\mathrm{D,\,Test}}$. As α starts to increase, $\Delta Acc_{\mathrm{D,\,Test}}$ of both FNN-MPA and NBC-MPA increases as well. In terms of FNN-MPA, $\Delta Acc_{\mathrm{D,\,Test}}$ reaches the maximum at $\alpha = 1.0$. As α further increases, $\Delta Acc_{\mathrm{D,\,Test}}$ decreases and gradually stabilizes. In addition, the impact of different β is small. A slight advantage is observed with $\beta = 32$. In terms of NBC-MPA, the maximum of $\Delta Acc_{\mathrm{D,\,Test}}$ is reached at $\alpha = 0.2$. A significant drop-off is seen as α further increases. The difference regarding $\alpha > 0.2$ between FNN-MPA and NBC-MPA is due to the used normalization functions and their working principles. In FNN-MPA, the normalization is carried out using (4.18) and FNN determines the prediction results mainly according to the amplitudes of input features in $\mathrm{D1}_{\mathrm{Train}}$. In NBC-MPA, the normalization is carried out using (4.22) and NBC determines the prediction results directly based on the statistics in $\mathrm{D1}_{\mathrm{Train}}$. As α increases, the impact of the reference data increases as well, resulting in significant variations in feature amplitudes. However, after the normalization in FNN-MPA, most amplitudes of object features are still similar compared with the case without the correlation, while the statistical distributions used in NBC-MPA are completely changed (i.e., overamplified μ_{Obj}). Therefore, NBC-MPA is more sensitive to α compared with FNN-MPA and $\Delta Acc_{\mathrm{D,\,Test}}$ of NBC-MPA becomes even negative when $\alpha \geqslant 10$.

In practice, the selection of α and β highly depends on the dynamic degree of detection scenarios. Under the detection conditions specified at the beginning of this subsection, the optimal parameters for the correlation analysis are determined as $\alpha_{\mathrm{FNN}} = 1.0$, $\alpha_{\mathrm{NBC}} = 0.2$, and $\beta_{\mathrm{FNN}} = \beta_{\mathrm{NBC}} = 32$. In the following context, the correlated FNN-MPA and NBC-MPA with these parameters are mentioned as CO-FNN-MPA and CO-NBC-MPA. $\Delta Acc_{\mathrm{D,\,Test}}$ of CO-FNN-MPA and CO-NBC-MPA are 14.35 % and 13.15 %, respectively.

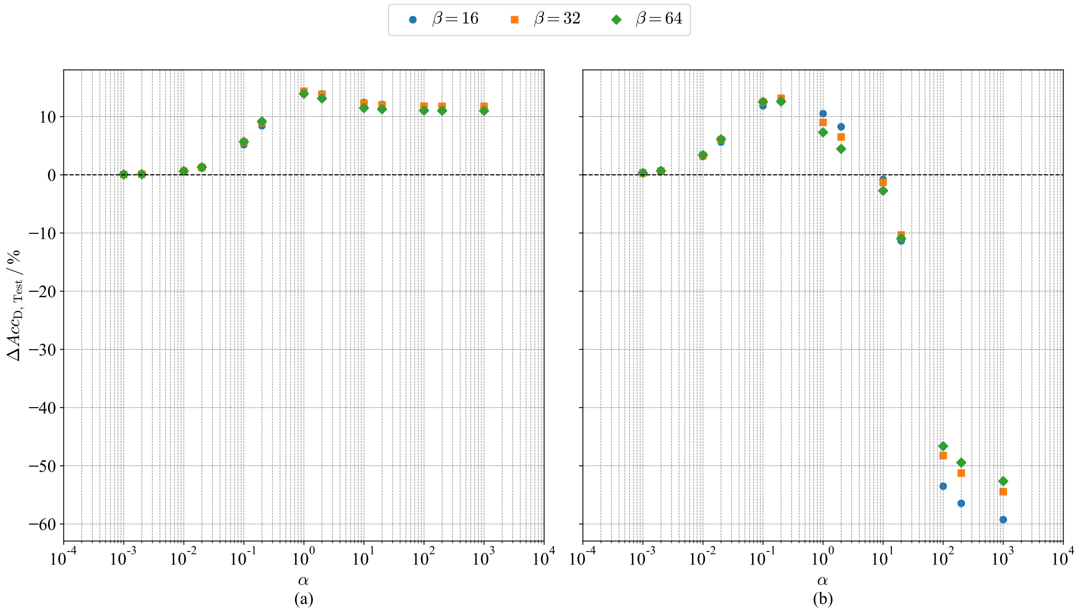


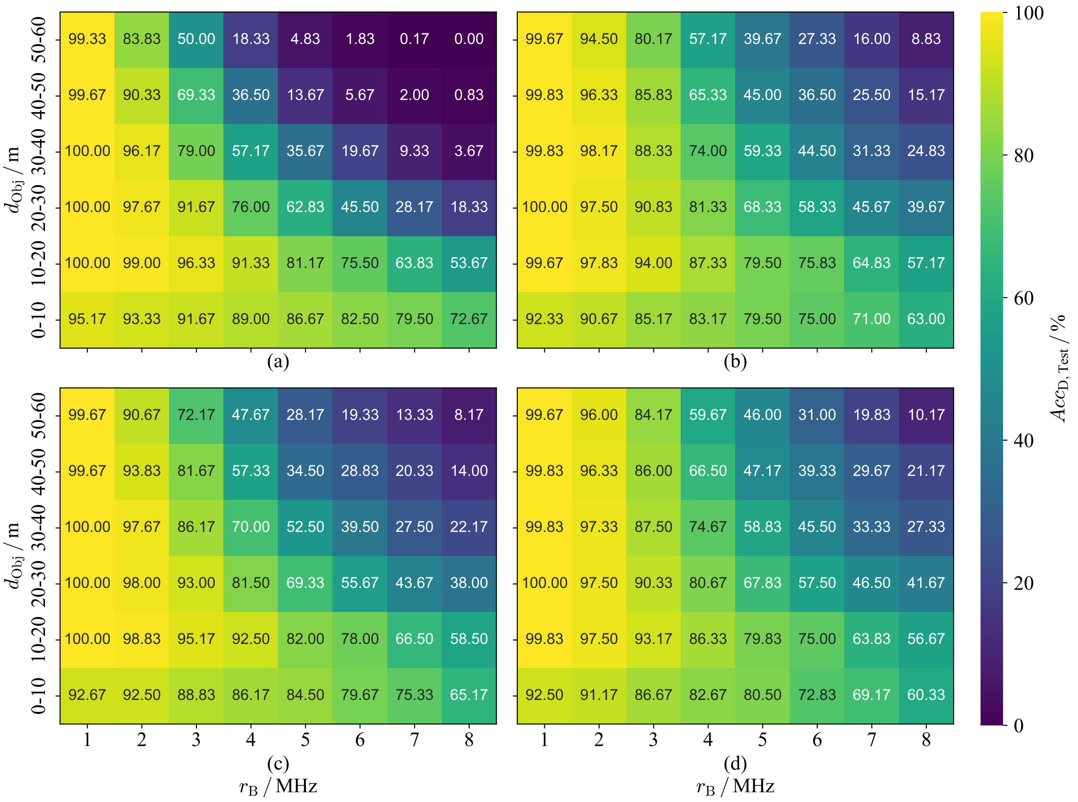
Figure 5.13: $\Delta Acc_{\mathrm{D,\,Test}}$ after correlation analysis. (a) $\Delta Acc_{\mathrm{D,\,Test}}$ between CO-FNN-MPA and FNN-MPA. (b)$\Delta Acc_{\mathrm{D,\,Test}}$ between CO-NBC-MPA and NBC-MPA.

5.3.5 Comparative Discussion

In the following discussion, a widely-used **C**lassical **D**igital **P**rocessing (CDP) method, including an average filter, BES, and the global maximum detection, is configured as a reference for comparison. CDP, FNN-MPA, the binomial estimation, and NBC-MPA specified in the previous sections are compared regarding Acc_{D}, timing performance, and memory occupancy.

Accuracy of Distance Prediction

In general, $Acc_{\mathrm{D,\,Test}}$ of FNN-MPA, the binomial estimation, and NBC-MPA are higher than that of CDP. Since the binomial estimation only serves as a denoising and pre-processing process for NBC, its $Acc_{\mathrm{D,\,Test}}$ is lower than that of FNN-MPA and NBC-MPA. To compare their prediction performance in detail, $Acc_{\mathrm{D,\,Test}}$ is further separated with respect to different d_{Obj} and r_{B}, as shown in Figure 5.14. In terms of low background light ($r_{\mathrm{B}} = 1\,\mathrm{MHz}$), $Acc_{\mathrm{D,\,Test}}$ of all approaches reach almost 100 % and have minor difference. Therefore, the following analysis focuses on $r_{\mathrm{B}} \geqslant 2\,\mathrm{MHz}$.



(a)

d_{Obj} / m	1	2	3	4	5	6	7	8
50-60	99.33	83.83	50.00	18.33	4.83	1.83	0.17	0.00
40-50	99.67	90.33	69.33	36.50	13.67	5.67	2.00	0.83
30-40	100.00	96.17	79.00	57.17	35.67	19.67	9.33	3.67
20-30	100.00	97.67	91.67	76.00	62.83	45.50	28.17	18.33
10-20	100.00	99.00	96.33	91.33	81.17	75.50	63.83	53.67
0-10	95.17	93.33	91.67	89.00	86.67	82.50	79.50	72.67

(b)

d_{Obj} / m	1	2	3	4	5	6	7	8
50-60	99.67	94.50	80.17	57.17	39.67	27.33	16.00	8.83
40-50	99.83	96.33	85.83	65.33	45.00	36.50	25.50	15.17
30-40	99.83	98.17	88.33	74.00	59.33	44.50	31.33	24.83
20-30	100.00	97.50	90.83	81.33	68.33	58.33	45.67	39.67
10-20	99.67	97.83	94.00	87.33	79.50	75.83	64.83	57.17
0-10	92.33	90.67	85.17	83.17	79.50	75.00	71.00	63.00

(c)

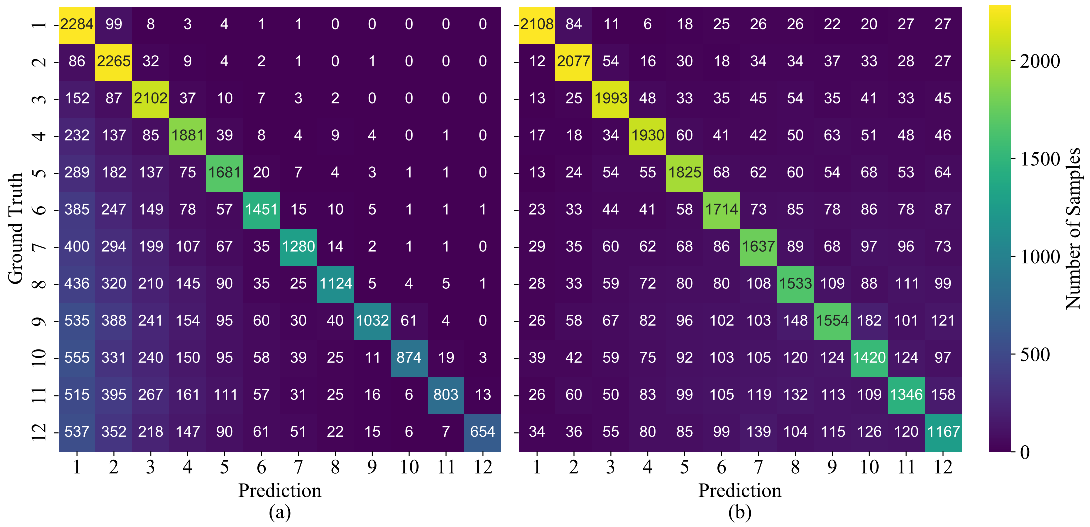
d_{Obj} / m	1	2	3	4	5	6	7	8
50-60	99.67	90.67	72.17	47.67	28.17	19.33	13.33	8.17
40-50	99.67	93.83	81.67	57.33	34.50	28.83	20.33	14.00
30-40	100.00	97.67	86.17	70.00	52.50	39.50	27.50	22.17
20-30	100.00	98.00	93.00	81.50	69.33	55.67	43.67	38.00
10-20	100.00	98.83	95.17	92.50	82.00	78.00	66.50	58.50
0-10	92.67	92.50	88.83	86.17	84.50	79.67	75.33	65.17

(d)

d_{Obj} / m	1	2	3	4	5	6	7	8
50-60	99.67	96.00	84.17	59.67	46.00	31.00	19.83	10.17
40-50	99.83	96.33	86.00	66.50	47.17	39.33	29.67	21.17
30-40	99.83	97.33	87.50	74.67	58.83	45.50	33.33	27.33
20-30	100.00	97.50	90.33	80.67	67.83	57.50	46.50	41.67
10-20	99.83	97.50	93.17	86.33	79.83	75.00	63.83	56.67
0-10	92.50	91.17	86.67	82.67	80.50	72.83	69.17	60.33

Figure 5.14: Comparison of distance prediction accuracy. $\mathrm{D1}_{\mathrm{Test}}$ is divided into 48 subsets regarding d_{Obj} and r_{B}. (a) CDP (overall $Acc_{\mathrm{D,\,Test}} = 59.34\,\%$). (b) FNN-MPA (overall $Acc_{\mathrm{D,\,Test}} = 68.56\,\%$). (c) Binomial estimation (overall $Acc_{\mathrm{D,\,Test}} = 67.09\,\%$). (d) NBC-MPA (overall $Acc_{\mathrm{D,\,Test}} = 69.31\,\%$).

As r_{B} increases, the advantages of CDP are becoming evident in short-range detections ($d_{\mathrm{Obj}} \in (0, 10]\,\mathrm{m}$). To explain this, the confusion matrices of CDP and NBC-MPA are shown in Figure 5.15. In Figure 5.15 (a), most samples with false predictions are in the lower-left region of the confusion matrix and the samples in the first columns take a large portion. This means CDP tends to make predictions as small distances, which improves its $Acc_{\mathrm{D,\,Test}}$ with $d_{\mathrm{Obj}} < 10\,\mathrm{m}$ at the substantial expense of its prediction ability on samples with large distances. This drawback is indicated by unbalanced precision and recall of CDP in Table 5.6 as well. It compromises the overall performance and thus results in a low F1-Score. On the contrary, false predictions are relatively evenly distributed. This means that prediction results of NBC-MPA have no bias towards distance. Except for the short-range detections, MPA variants outperform CDP under all other situations. In



(a)

	1	2	3	4	5	6	7	8	9	10	11	12
1	2284	99	8	3	4	1	1	0	0	0	0	0
2	86	2265	32	9	4	2	1	0	1	0	0	0
3	152	87	2102	37	10	7	3	2	0	0	0	0
4	232	137	85	1881	39	8	4	9	4	0	1	0
5	289	182	137	75	1681	20	7	4	3	1	1	0
6	385	247	149	78	57	1451	15	10	5	1	1	1
7	400	294	199	107	67	35	1280	14	2	1	1	0
8	436	320	210	145	90	35	25	1124	5	4	5	1
9	535	388	241	154	95	60	30	40	1032	61	4	0
10	555	331	240	150	95	58	39	25	11	874	19	3
11	515	395	267	161	111	57	31	25	16	6	803	13
12	537	352	218	147	90	61	51	22	15	6	7	654

(b)

	1	2	3	4	5	6	7	8	9	10	11	12
1	2108	84	11	6	18	25	26	26	22	20	27	27
2	12	2077	54	16	30	18	34	34	37	33	28	27
3	13	25	1993	48	33	35	45	54	35	41	33	45
4	17	18	34	1930	60	41	42	50	63	51	48	46
5	13	24	54	55	1825	68	62	60	54	68	53	64
6	23	33	44	41	58	1714	73	85	78	86	78	87
7	29	35	60	62	68	86	1637	89	68	97	96	73
8	28	33	59	72	80	80	108	1533	109	88	111	99
9	26	58	67	82	96	102	103	148	1554	182	101	121
10	39	42	59	75	92	103	105	120	124	1420	124	97
11	26	60	50	83	99	105	119	132	113	109	1346	158
12	34	36	55	80	85	99	139	104	115	126	120	1167

Figure 5.15: Confusion matrix based on classification results of $\mathrm{D1}_{\mathrm{Test}}$. (a) CDP. (b) NBC-MPA.

Table 5.6: Performance comparison regarding precision, recall, F1-Score, and AUC.

Variant	Metric / %			
	Precision	Recall	F1-Score	AUC
CDP	75.21	60.46	60.00	-
FNN-MPA	69.72	70.14	69.60	94.16
Binomial Estimation	69.51	68.38	67.42	92.26
NBC-MPA	70.22	70.50	70.30	95.43

particular, FNN-MPA, the binomial estimation, and NBC-MPA reach the accuracy of 58.60 %, 51.49 %, and 60.53 % with $d_{\mathrm{Obj}} \in (30, 60]$ m and $r_{\mathrm{B}} \in (3, 6]$ MHz, while CDP has the accuracy of only 32.64 %. This shows the clear advantage of the proposed variants with high background light in middle-range detections over CDP. In the case that $d_{\mathrm{Obj}} \in (40, 60]$ m and $r_{\mathrm{B}} \in (7, 8]$ MHz, all variants reach their detection limits. Therein, NBC-MPA achieves the best $Acc_{\mathrm{D,\,Test}}$ (20.21 %), while FNN-MPA has the second highest $Acc_{\mathrm{D,\,Test}}$ (16.38 %). The superiority of NBC-MPA in this case is provided by its categorization process. NBC-MPA has considerably more classes than FNN-MPA and calculates models for different r_B separately, which gives it a superior discriminative ability under strong background light. Finally, by considering the observation in Table 5.2, it indicates that with SNR $= 0.16$ and $N_\mathrm{M} = 400$, the received laser photons are insufficient for making reliable distance predictions, since the random behaviors of photon statistics are considerably high.

In summary, three variants proposed in this work outperform CDP, especially with high r_B and large d_Obj. Therein, NBC-MPA achieves the best overall accuracy. Besides, FNN-MPA provides a comparable prediction ability to NBC-MPA.

Correlation Results

To discuss detailed impacts of correlation analysis, $\Delta Acc_\mathrm{D,Test}$ defined in subsection 5.3.4 is further separated regarding different d_Obj and r_B, as shown in Figure 5.16. In general, the characteristics of $\Delta Acc_\mathrm{D,Test}$ using FNN-MPA and NBC-MPA are fairly similar. Discussions are divided into four parts:

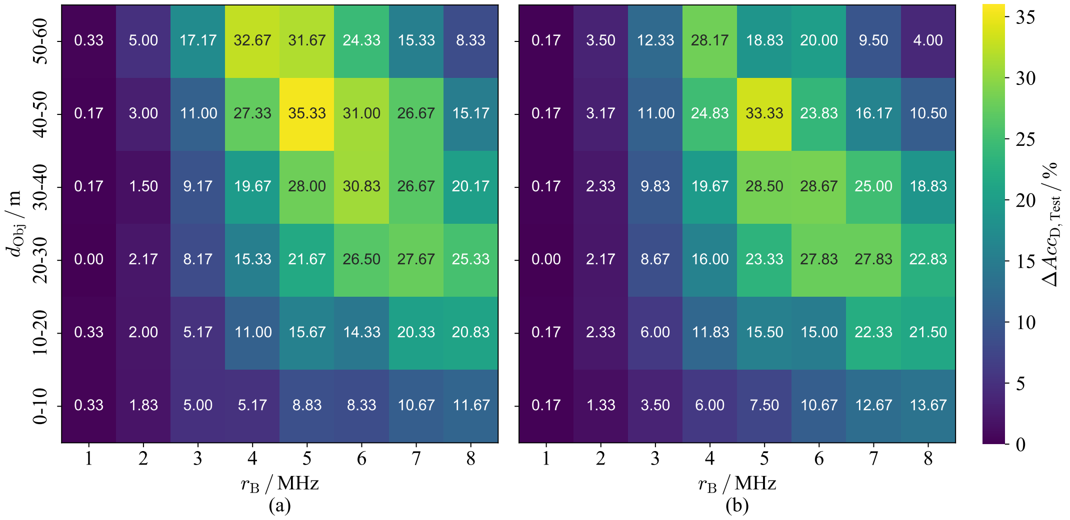


Figure 5.16: Accuracy difference between distance prediction with and without correlation. (a) $\Delta Acc_\mathrm{D,Test}$ between $Acc_\mathrm{D,Correlation}$ of FNN and Figure 5.14 (b). (b)$\Delta Acc_\mathrm{D,Test}$ between $Acc_\mathrm{D,Correlation}$ of NBC and Figure 5.14 (d).

1) $\boldsymbol{r_\mathrm{B} \in [1, 2]}$ **MHz**: In this sub-area, $Acc_\mathrm{D,Test}$ without the correlation is extremely high (average $Acc_\mathrm{D,Test} > 97\,\%$). Thus, there exists little space for improvement. Nonetheless, $Acc_\mathrm{D,Test}$ of FNN-MPA and NBC-MPA still increases

slightly. This implies that the correlation analysis has positive impacts in the detection case with low background light.

2) $\boldsymbol{r_{\mathrm{B}} \in [3,8]}$ **MHz** $\boldsymbol{\bigcap d_{\mathrm{Obj}} \in (0,10]}$ **m**: Considerable increments in $Acc_{\mathrm{D,Test}}$ are observed in this sub-area. With the improvements of the correlation, the inferiority of FNN-MPA and NBC-MPA are compensated and $Acc_{\mathrm{D,\,Test}}$ of both variants exceed CDP without sacrificing the performance on large d_{Obj}.

3) $\boldsymbol{r_{\mathrm{B}} \in [3,6]}$ **MHz** $\boldsymbol{\bigcap d_{\mathrm{Obj}} \in (10,60]}$ **m and** $\boldsymbol{r_{\mathrm{B}} \in [7,8]}$ **MHz** $\boldsymbol{\bigcap d_{\mathrm{Obj}} \in (10,40]}$ **m**: The correlations bring significant improvements in this sub-area. For example, in Figure 5.16 (a), with $r_{\mathrm{B}} = 5\,\mathrm{MHz}$ and $d_{\mathrm{Obj}} \in (40,\ 50]\,\mathrm{m}$, the accuracy is improved from 45.00 % to 80.33 %. However, according to condition 2 and 3 proposed at the beginning of subsection 5.3.4, both reference data and the original data have similar $Acc_{\mathrm{D,\,Test}}$ of around 45 %. This means that the correlation provides notable improvements even if the involved data are less reliable. To explain it, a specific correlation example is described as follows. First, the involved sample A and B chosen from $\mathrm{D1}_{\mathrm{Test}}$ are illustrated in Figure 5.17. Clearly, both distance predictions according to the original output results of sample A and B are false. However, it is observed that the prediction values at the correct object positions are considerably high. According to (4.33), the correlation is carried out using the output of sample A as the reference data and the feature group of sample B as the original data. Figure 5.18 illustrates the correlation process. In Figure 5.18 (a), only one significant gain is generated at bin 907. Subsequently, the gains are added to the original data and the correlated feature group is formed, as shown in Figure 5.18 (b). Finally, by applying the same FNN to the correlated features, the resulting prediction is corrected, as shown in Figure 5.18 (c). This example shows that correlations with positive effects rely not only on the highest prediction value, but also on multiple potential object positions. This characteristic gives the correlation analysis the ability to obtain the correct prediction even when both the original and reference data are wrongly predicted. It benefits a large number of samples and explains the significant improvement observed in Figure 5.16.

4) $\boldsymbol{r_{\mathrm{B}} \in [7,8]}$ **MHz** $\boldsymbol{\bigcap d_{\mathrm{Obj}} \in (40,60]}$ **m**: In this sub-area, the correlation analysis brings a relatively small boost. It is observed that the improvement of FNN-MPA is distinctly higher than that of NBC-MPA. Besides, the effects of the correlation in both variants are positive even with $r_{\mathrm{B}} = 8\,\mathrm{MHz}$ and $d_{\mathrm{Obj}} \in$

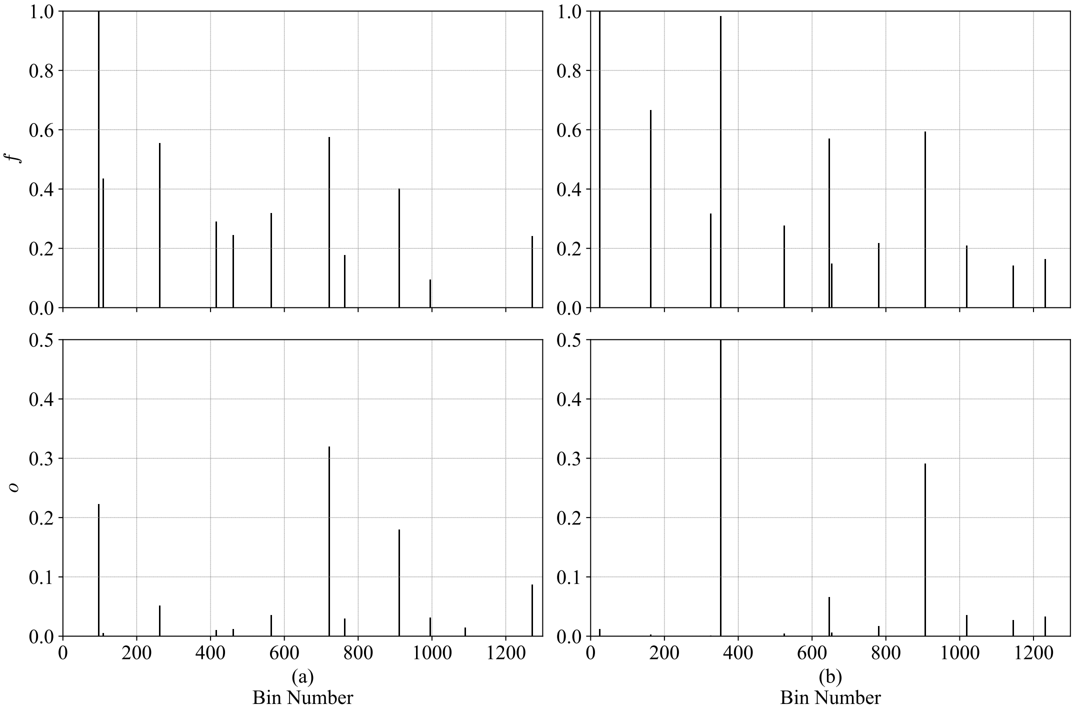


Figure 5.17: Samples and original predictions of FNN-MPA involved in the correlation example. (a) The feature group and the original output results of sample A. The object locates at bin 911, while the highest prediction value without correlation is at bin 721. (b) The feature group and the original output results of sample B. The object locates at bin 907, while the highest prediction value without correlation is at bin 352.

(40, 60] m, where the original prediction accuracies are only 8.83 % and 10.17 %, respectively. This indicates that gains between noise features are well suppressed by the correlation analysis and the induced negative effects can be neglected.

Finally, the MPA variants with the correlation are tested in the virtual scenario introduced in subsection 4.4.3. The parameters of the corresponding TC-Hist generation are identical to that of D1. In addition, the parameter settings for the virtual scenario are listed in Table 5.7. Prediction results of CDP, FNN-MPA, and CO-FNN-MPA are shown in Figure 5.19. Therein, each point represents the predicted distance of the corresponding variant. The color of the point refers to the prediction certainty from 0 % (deep blue) to 100 % (deep red). Since CDP provides only hard decisions, certainties of its predictions are assumed to be 100 %, as shown in Figure 5.19 (a). In the given scenario, CDP only achieves

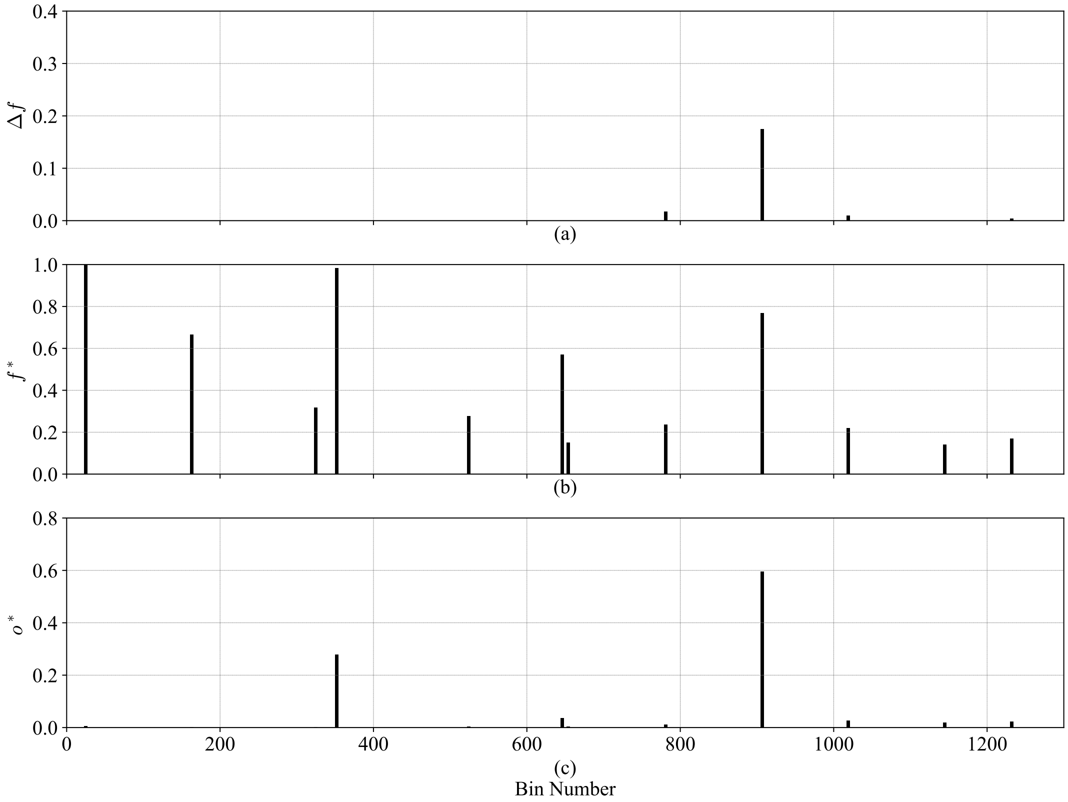


Figure 5.18: Gain values and correlation results using sample A and B. (a) Gain value Δf. (b) Correlated feature group F^*_{G} of sample B. (c) Prediction results on F^*_{G}. Gain values Δf are calculated based on the correlations between reference data A and original data B. Subsequently, Δf are superimposed on the feature values of B and the correlated feature values f^*_n are obtained. Finally, a new prediction o^*_n is made by taking f^*_n as the input of FNN.

47.92 % prediction accuracy. Due to high background light, most of its false predictions are within the vertical distance range of (0, 20] m. It is barely possible to obtain precise scenario information based on these measurement points. As shown in Figure 5.19 (b), the overall prediction accuracy of FNN-MPA reaches 75.52 %. Furthermore, it is observed that the prediction points close to the objects and the background wall have relatively high prediction certainties, while most false prediction points have low prediction certainties. In terms of CO-FNN-MPA, the overall prediction accuracy is further improved by 15.62 %, as shown in Figure 5.19 (c). The certainty difference between correct prediction points and false prediction points is further increased (Most points close to the objects

Table 5.7: Parameter settings of virtual scenario.

Parameter	Unit	Value
v_{Obj}	[m/s]	5
w_{Obj}	[m]	4
d_{Obj}	[m]	25
d_{Wall}	[m]	35
FOV	[°]	36
Number of Pixels	-	192
r_{B}	[MHz]	5

and the background wall have more than 80 % of prediction certainties, while the prediction certainties of the rest of the points are less than 40 %). Besides, it is noticed that the right sides of both point groups close to the background wall are slightly jittered. This is presumed to be induced by the residual effect of the objects. As objects move to the right, the parts of the background wall that are exposed to the LiDAR system shift to the right as well. In terms of the newly exposed parts of the background wall (where the jitters occur), correlations from the last time frame and from the right-side pixel are less effective. Nevertheless, both FNN-MPA and CO-FNN-MPA show their capabilities in the given dynamic scenario and improve the reliability of distance predictions significantly.

Timing Performance

The computation time of each variant is recorded in Table 5.9. Therein, although an average filter used in CDP and a constant convolutional kernel used in MPA variants are different in definitions, their computations and functionality in the scope of this work are the same. Thus, these two calculation processes are collectively referred to as convolution in the following context. It is seen that the distance prediction of CO-FNN-MPA is faster than that of CDP. In addition, the convolution time takes a large portion of the whole processing time. Due to the feature extraction, BES in MPA only needs to be performed on $N_{\mathrm{F}} = 12$ features, which requires considerably less computation time compared to BES in

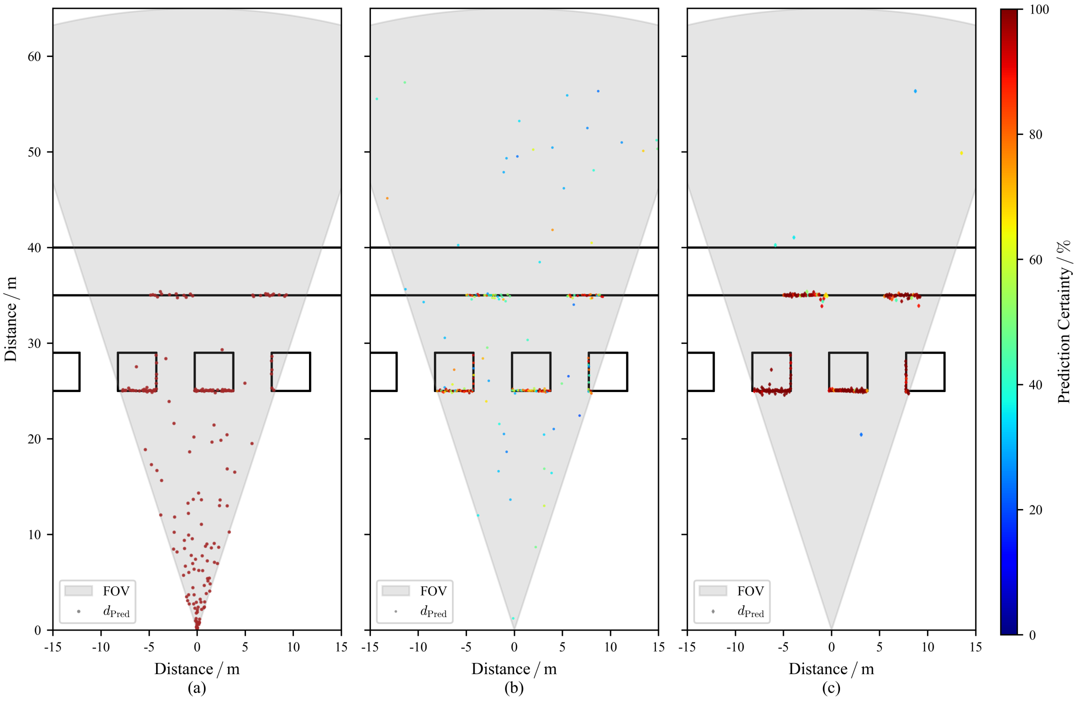


Figure 5.19: Prediction result demonstration in virtual scenario. (a) Predictions of CDP, overall $Acc_{\mathrm{D}} = 47.92\,\%$. (b) Predictions of FNN-MPA, overall $Acc_{\mathrm{D}} = 75.52\,\%$. (b) Predictions of CO-FNN-MPA, overall $Acc_{\mathrm{D}} = 91.15\,\%$. Colors of the prediction points represent the corresponding prediction certainties.

CDP. However, although the binomial estimation is carried out for $N_{\mathrm{F}} = 12$ features as well, the computation time of 200 μs is required, since it involves a large number of operations with factorials and exponents. In terms of distance prediction, CDP only extracts the global maximum as the final prediction, thus taking an extremely short time to complete. Since NBC calculates log-normal functions for predictions, the computation time is clearly longer than that of FNN requiring mainly matrix operations.

In summary, CO-FNN-MPA achieves the best timing performance (289 μs), while CDP, the binomial estimation, and CO-NBC-MPA require more time (320 μs, 440 μs, and 2507 μs) to complete a distance prediction on one pixel.

Memory Occupancy

The number of values required for data storage in each variant is listed in Table 5.9. Normally, photon counts in a TC-Hist are represented by unsigned integers,

Table 5.8: Average computation time for one pixel based on Intel(R) i7-3612QM CPU. Only the most complex MPA variants are listed. The computation time of other variants can be obtained by removing the time of irrelevant processes.

	Time / μs		
Process	CDP	CO-FNN-MPA	CO-NBC-MPA
Convolution	230	230	230
Feature Extraction	-	10	10
BES	90	10	-
Binomial Estimation	-	-	200
Correlation	-	29	29
Distance Prediction	0	10	2038
Overall	320	289	2507

while parameters and prediction results are represented by floating-point numbers. After the convolution, the raw TC-Hist with photon counts is replaced by convolved TC-Hist. Thus, the memory occupation for the former can be released. Although CDP has no pre-stored parameters, it requires considerably more memory to store the feature value f_n compared to MPA variants, since it relies on the complete TC-Hist after the convolution. In terms of prediction results, the MPA variants provide up to 12 potential distances and corresponding certainties, which can be stored and further utilized to improve the result of high-level property inference in subsequent processing.

As a result, CDP requires in total 1311 floating-point numbers, while MPA variants require at most 264 floating-point numbers and 60 integers for processing. Nevertheless, the minimum memory space must be equal to the raw TC-Hists from pixels, since the processing can only be carried out after these TC-Hists are saved. This places high demands on the memory space of a LiDAR system with a large number of pixels.

Table 5.9: Required memory space of data processing for a pixel. Only the most complex MPA variants are listed, the required data amount of other MPA variants can be obtained by removing data from irrelevant processes.

		Required Amount of Data & Parameters		
Data	Type	CDP	CO-FNN-MPA	CO-NBC-MPA
Raw TC-Hist	Integer	1310	1310	1310
f_n	Float	1310	12	12
b_n	Integer	-	12	12
w_n	Float	-	144	-
$\overline{P}_n$	Float	-	-	96
s_n	Float	-	-	96
b'_n	Integer	-	36	36
o'_n	Float	-	36	36
d_{Pred}	Float	1	12	12
o^*_n	Float	-	12	12

5.4 Optimization Approaches

5.4.1 Ablation Study for Background Suppression

In this section, the necessity of background suppression in MPA is studied.

BES in FNN-MPA

Although FNN-MPA requires only the background subtraction on N_{F} selected features, the background estimation is still challenging for embedded systems, since a real-time exponential approximation with a small performance degradation is required. Therefore, an ablation study for BES is carried out. Instead of using (4.18), the local maxima M_n are directly used as the input of FNN. In this case, the patterns of background light are directly included in the input and FNN must learn them additionally by training.

Figure 5.20 illustrates the differences in the training process. It can be seen that the training speed of FNN without BES is clearly slower than that with

BES. However, their performance finally converges in both $\mathrm{D1}_{\mathrm{Train}}$ and $\mathrm{D1}_{\mathrm{Val}}$. Besides, the performance regarding $Acc_{\mathrm{D,\,Test}}$ is assessed. For this purpose, three background conditions are specified as follows:

- $r_{\mathrm{All}} = \{1, 2, 3, 4, 5, 6, 7, 8\}\,\mathrm{MHz}$,
- $r_{\mathrm{Low}} = \{1, 2, 3, 4\}\,\mathrm{MHz}$,
- $r_{\mathrm{High}} = \{5, 6, 7, 8\}\,\mathrm{MHz}$.

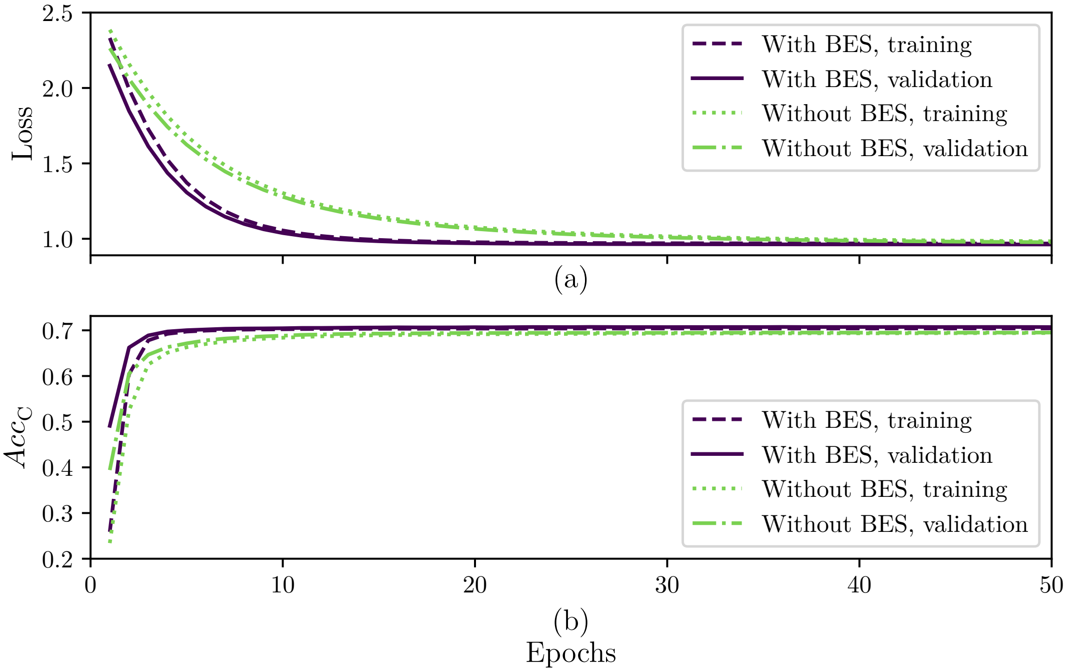


Figure 5.20: Training process of FNN with and without BES. (a) Training loss. (b) Classification accuracy.

Consequently, in addition to FNN-MPA, three variants are introduced in order to assess their general performance and generalization ability, namely:

- FNN_1-MPA: FNN is trained without BES and with $r_{\mathrm{B}} \in r_{\mathrm{All}}$,
- FNN_2-MPA: FNN is trained with BES and with $r_{\mathrm{B}} \in r_{\mathrm{Low}}$,
- FNN_3-MPA: FNN is trained without BES and with $r_{\mathrm{B}} \in r_{\mathrm{Low}}$.

All training samples with the specified conditions are taken from $\mathrm{D1}_{\mathrm{Train}}$. The cross-comparison analysis based on the result shown in Table 5.10 is as follows:

Table 5.10: Prediction accuracies on $D1_{\text{Test}}$.

(a) $Acc_{\text{D, Test}}$ / % of FNN-MPA.

	r_{B} / MHz			
T_{TOF} / m	1 & 2	3 & 4	5 & 6	7 & 8
(0, 20]	95.12	87.42	77.46	64.00
(20, 40]	98.88	83.62	57.62	35.38
(40, 60]	97.58	72.12	37.12	16.38

(b) $Acc_{\text{D, Test}}$ / % of FNN_1-MPA.

	r_{B} / MHz			
T_{TOF} / m	1 & 2	3 & 4	5 & 6	7 & 8
(0, 20]	94.75	84.38	72.96	59.71
(20, 40]	98.62	83.25	57.54	35.62
(40, 60]	97.62	72.50	37.54	18.00

(c) $Acc_{\text{D, Test}}$ / % of FNN_2-MPA.

	r_{B} / MHz			
T_{TOF} / m	1 & 2	3 & 4	5 & 6	7 & 8
(0, 20]	95.38	88.96	79.83	66.92
(20, 40]	98.62	82.54	54.29	30.29
(40, 60]	97.42	71.25	32.62	11.71

(d) $Acc_{\text{D, Test}}$ / % of FNN_3-MPA.

	r_{B} / MHz			
T_{TOF} / m	1 & 2	3 & 4	5 & 6	7 & 8
(0, 20]	94.75	88.25	78.83	58.17
(20, 40]	98.62	83.29	50.58	14.12
(40, 60]	97.88	70.62	20.62	1.46

1) **Comparison between FNN-MPA and FNN_1-MPA**: In general, the overall $Acc_{\text{D, Test}}$ in Table 5.10 (a) is only 0.85 % higher than that in Table 5.10 (b). With $r_{\text{B}} \in r_{\text{High}}$ and $d_{\text{Obj}} \in (0, 20]$ m, table 5.10 (a) shows superior results than Table 5.10 (b). Except for these, they are in a small range of accuracy difference between each other. This indicates that FNN can relatively self-learn the background light information well with a comprehensive $D1_{\text{Train}}$ and the final performance is comparable to FNN-MPA.

2) **Comparison between FNN_2-MPA and FNN_3-MPA**: As expected, minor differences can be observed on $Acc_{\text{D, Test}}$ in Table 5.10 (c) and (d) with $r_{\text{B}} \in r_{\text{Low}}$. However, since TC-Hists with $r_{\text{B}} \in r_{\text{High}}$ are excluded in $D1_{\text{Train}}$, $Acc_{\text{D, Test}}$ of FNN_3-MPA drops off significantly when background light is high. On the contrary, as the background light has been subtracted, FNN_2-MPA still retains its prediction ability even without training on samples with high background light.

3) **Comparison between FNN-MPA and FNN_2-MPA**: A clear drop-off can be observed in Table 5.10 (c) with $r_{\text{B}} \in r_{\text{High}}$ and $d_{\text{Obj}} \in (20, 60]$ m. Surprisingly, a noticeable advantage is present on FNN_2-MPA over FNN-MPA with $r_{\text{B}} \in r_{\text{High}}$ and $d_{\text{Obj}} \in (0, 20]$ m. A possible reason is that under high background

light, samples with large d_{Obj} have low SNRs, resulting in a characteristic dominated by noise (local maxima decrease broadly exponentially with increasing d_{Obj}). The lack of these samples in the training dataset makes FNN$_2$-MPA to treat them as samples with small d_{Obj} and high r_{B}, whose local maxima show the trend of an exponential decay as well. This results in a falsely high $Acc_{\text{D, Test}}$ on short-range samples in exchange for the prediction accuracy on long-range samples.

4) **Comparison between FNN$_1$-MPA and FNN$_3$-MPA**: Compared to Table 5.10 (d), $Acc_{\text{D, Test}}$ in Table 5.10 (b) with $r_{\text{B}} \in r_{\text{High}}$ and $d_{\text{Obj}} \in (20, 60]$ m is much higher. Similar to the comparison between FNN-MPA and FNN$_2$-MPA, the difference of $Acc_{\text{D, Test}}$ between Table 5.10 (b) and Table 5.10 (d) with $r_{\text{B}} \in \{5, 6\}$ MHz and $d_{\text{Obj}} \in (0, 20]$ m reaches 7.08 % due to the biased predictions on short-range and long-range samples. However, $Acc_{\text{D, Test}}$ in Table 5.10 (d) with $r_{\text{B}} \in \{7, 8\}$ MHz and $d_{\text{Obj}} \in (0, 20]$ m reduces rapidly and becomes slightly worse than that in Table 5.10 (b). Clearly, due to both limitations (the lack of training data with high r_{B} and the absence of BES), FNN$_3$-MPA is inferior to other variants in most detection conditions.

Accordingly, it is concluded that in the scope of D1$_{\text{Test}}$, BES can only be discarded if the training data completely cover real measurement conditions.

Binomial Estimation in NBC-MPA

In NBC-MPA, the binomial estimation is performed as a background suppression and normalization method. As shown in Table 5.9, the binomial estimation requires a relatively long processing time. In the following experiment, instead of using the binomial estimation, the extracted feature f_n is simply normalized by

$$f_{\text{Norm}, n} = \frac{f_n}{\sum_{n=1}^{N_{\text{F}}} f_n} \, . \tag{5.14}$$

and used as the input of NBC-MPA. This variant is mentioned as NBC*-MPA in the following context. Figure 5.21 shows $Acc_{\text{D, Test}}$ of NBC*-MPA. In general, $Acc_{\text{D, Test}}$ of NBC*-MPA is 1.06 % lower than NBC-MPA. In conditions with low r_{B}, NBC*-MPA shows a comparable prediction ability with respect to NBC-MPA. However, a high r_{B} results in a significant noise level in the TC-Hist and increases the randomness of features. As expected, the robustness of NBC*-MPA

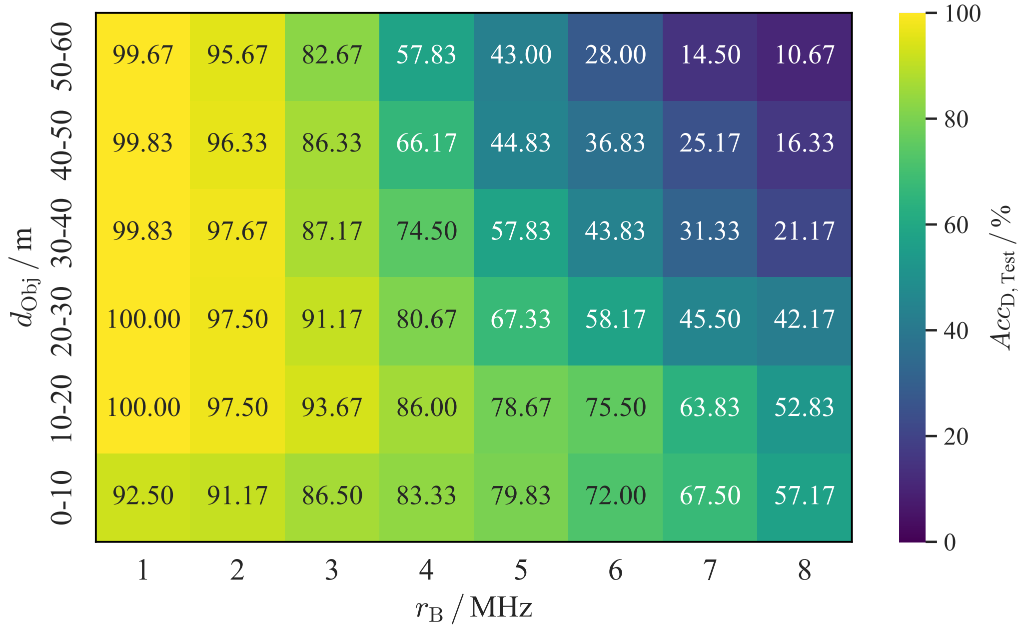


Figure 5.21: Accuracy of NBC*-MPA. $\mathrm{D1}_{\mathrm{Test}}$ is divided into 48 subsets regarding d_{Obj} and r_{B}. Overall $Acc_{\mathrm{D,Test}} = 68.25\,\%$.

is worse than that of NBC-MPA as r_{B} increases. With $r_{\mathrm{B}} = 8\,\mathrm{MHz}$, the accuracy difference between NBC*-MPA and NBC-MPA reaches 2.83 %.

5.4.2 Stride Analysis

Achieving the optimal real-time performance of a LiDAR system has always been a topic of interest. As shown in Table 5.9, the time for convolution takes a large portion, where the convolutional stride is 1. Therefore, the influence of different strides on CDP, FNN-MPA, and NBC-MPA is investigated. The overall accuracy is shown in Table 5.11. With a big stride, both MPA variants have smaller drop-off on $Acc_{\mathrm{D,Test}}$ than CDP. In terms of FNN-MPA, by increasing the stride to 3, the convolution time is reduced to one-third of the original time with only a decrease of 0.47 % in $Acc_{\mathrm{D,Test}}$. Interestingly, both MPA variants with stride = 5 have a lower $Acc_{\mathrm{D,Test}}$ than that with stride = 7. This is caused by a mismatch of the resulting resolution due to stride = 5 and distances in D1, where $d_{\mathrm{Obj}} \in \{0.5, 1, ..., 60\}\,\mathrm{m}$.

With the consideration that the stride mainly affects the precision of the

Table 5.11: Overall accuracy on $\mathrm{D1}_{\mathrm{Test}}$ with different strides.

	Stride				
Accuracy / %	1	2	3	5	7
$Acc_{\mathrm{C,\,Test}}$, FNN-MPA	70.47	70.23	69.97	68.51	67.35
$Acc_{\mathrm{D,\,Test}}$, FNN-MPA	68.56	68.16	68.09	65.82	66.28
$Acc_{\mathrm{D,\,Test}}$, NBC-MPA	69.31	68.92	68.83	66.44	67.23
$Acc_{\mathrm{D,\,Test}}$, CDP	59.34	58.95	58.10	55.12	51.46

distance, the accuracies with $d_{\mathrm{Obj}} < 10\,\mathrm{m}$ (corresponding to the error bound smaller than $\pm 0.5\,\mathrm{m}$) are demonstrated in Table 5.12. Similar to the results with $d_{\mathrm{Obj}} < 10\,\mathrm{m}$ in Figure 5.14, CDP outperforms both MPA variants with stride = 1 due to the unbalance of predictions. However, with the increase of the stride, $Acc_{\mathrm{D,\,Test}}$ of both MPA variants remain relatively stable, while the accuracy of CDP decreases rapidly. Compared to Table 5.11, the drop-off on $Acc_{\mathrm{D,\,Test}}$ with stride = 5 is more obvious due to the mismatch in distances. It should be noticed that $Acc_{\mathrm{C,\,Test}}$ of FNN-MPA is barely affected by the stride. Even with stride = 7, $Acc_{\mathrm{C,\,Test}}$ still maintains at 89.07 %. Thus, with a large stride, the Acc_{C} can be used as a guide for a sub-region in which the object distance locates. Afterwards, subsequent processing and measurements with higher precisions can be conducted only within this sub-region to restore the exact distance.

Table 5.12: Accuracy on $\mathrm{D1}_{\mathrm{Test}}$ with $d_{\mathrm{Obj}} < 10\,\mathrm{m}$.

	Stride				
Accuracy / %	1	2	3	5	7
$Acc_{\mathrm{C,\,Test}}$, FNN-MPA	90.78	90.39	90.12	89.18	89.07
$Acc_{\mathrm{D,\,Test}}$, FNN-MPA	80.41	77.64	78.13	65.42	73.73
$Acc_{\mathrm{D,\,Test}}$, NBC-MPA	79.34	76.93	77.21	63.97	71.91
$Acc_{\mathrm{D,\,Test}}$, CDP	86.22	82.11	75.80	56.56	36.61

5.4.3 Threshold Analysis

The evaluation of the threshold is carried out in two aspects, i.e., (1) Acc_{Th} is optimized by the global adjustment of o_{Th} on $\mathrm{D1}_{\mathrm{Test}}$ and (2) Acc_{Th} is optimized by weighted $Acc_{\mathrm{Th,\,B}}$, where $Acc_{\mathrm{Th,\,B}}$ is adjusted locally on the data group with the same r_{B} in $\mathrm{D1}_{\mathrm{Test}}$.

Figure 5.22 shows the relationship between Acc_{Th} and o_{Th} of four variants regarding aspect (1). It is seen that Acc_{Th} of the correlated variants are higher than that of uncorrelated variants. This indicates that the correlation increases the discrepancy of o_{H} associated with objects and noise. However, the uncorrelated variants have a higher rise in Acc_{Th} and a smaller optimal o_{Th} compared to the correlated variants. Besides, Acc_{Th} of FNN-MPA and CO-FNN-MPA drop off rapidly with $o_{\mathrm{Th}} > 0.9$, while Acc_{Th} of NBC-MPA and CO-NBC-MPA are relatively stable even with $o_{\mathrm{Th}} = 0.99$. This means that most o_{H} of the NBC variants locate at the boundary of the value range (i.e., around 0.0 and 1.0). Presumably, this is due to the fact that the NBC variants apply joint probability distributions for predictions and are therefore more deterministic than that of the FNN variants.

Table 5.13 shows locally adjusted o_{Th} and the weighted Acc_{Th} based on aspect (2). Although the optimal o_{Th} regarding each r_{B} is different from each other, the weighted Acc_{Th} is only improved minimally in contrast to that of the global optimization, especially when correlated variants are used. It indicates that the distributions of o_{H} under different r_{B} have only little difference.

Table 5.13: Locally optimized o_{Th} regarding $r_{\mathrm{B}} \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ MHz and weighted Acc_{Th}.

Variant	o_{Th}	$Acc_{\mathrm{Th}}\,/\,\%$
FNN-MPA	{0.49, 0.23, 0.35, 0.44, 0.51, 0.61, 0.67, 0.79}	81.97
CO-FNN-MPA	{0.00, 0.70, 0.61, 0.52, 0.58, 0.62, 0.67, 0.77}	89.48
NBC-MPA	{0.00, 0.48, 0.49, 0.54, 0.56, 0.58, 0.64, 0.63}	82.43
CO-NBC-MPA	{0.00, 0.55, 0.61, 0.55, 0.57, 0.59, 0.67, 0.79}	88.07

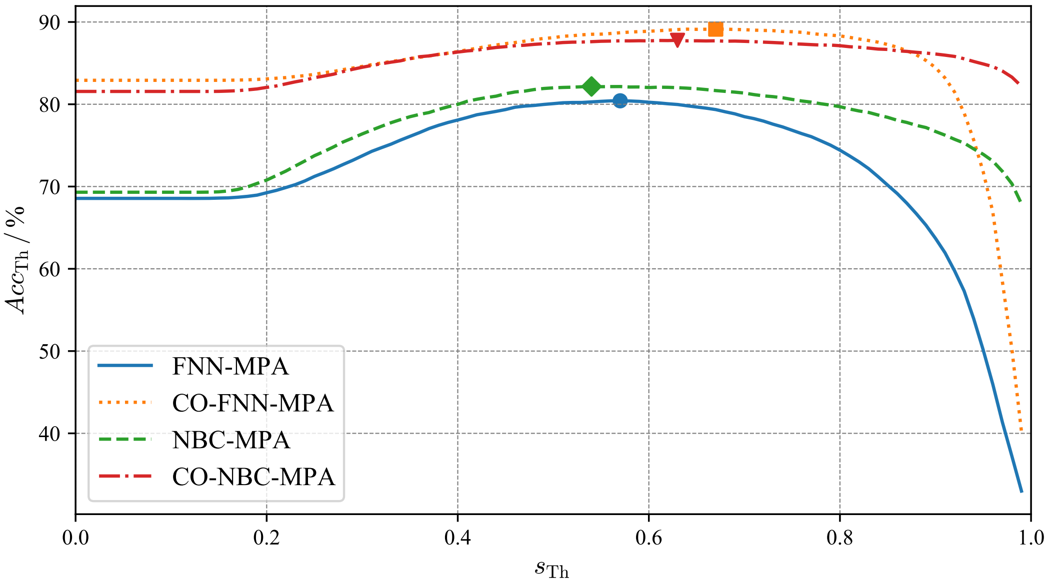


Figure 5.22: Acc_{Th} with global threshold adjustment. The maximum Acc_{Th} of FNN-MPA, CO-FNN-MPA, NBC-MPA, and CO-NBC-MPA are achieved at point (0.57, 80.42), (0.67, 89.13), (0.54, 82.16), and (0.63, 87.74), respectively.

5.5 Performance on Dataset D2

Finally, six variants introduced in previous sections are tested on D2. It should be mentioned that r_{L} in D1 and in D2 exists a degree of difference. In D1, r_{L} is assumed to be constant with the value of 10 MHz. In practice, r_{L} is influenced by the measurement environment. With the object distances in D2, r_{L} with a small distance is significantly higher than 10 MHz. A comparison between the average received photon counts $\overline{C}_{\mathrm{Obj}}$ in D2 and the expected photon counts μ_{Obj} at the object distance with $r_{\mathrm{L}} = 10\,\mathrm{MHz}$ is shown in Table 5.14. Apparently, $\overline{C}_{\mathrm{Obj}}$ reflected by close objects (within $(0, 16.77]\,\mathrm{m}$) in D2 is at least twice as much as $\mu_{\mathrm{Obj}}\,|_{r_{\mathrm{L}}=10\,\mathrm{MHz}}$ with the same distance and r_{B}. This results in high SNRs on the extracted features. Figure 5.23 shows a raw TC-Hist from D2 and its extracted features. Since FPC is used, incoming photons after the object distance are blocked, resulting in extremely small count values in Figure 5.23 (a). As shown in Figure 5.23 (b), after the convolution and the normalization, the feature value of the object distance is clearly greater than other features due to high r_{L}. However, as d_{Obj} increases, $\overline{C}_{\mathrm{Obj}}$ decays continuously with an exponential

Table 5.14: Comparison of average counts in D2 and expected photon counts with $r_{\mathrm{L}} = 10\,\mathrm{MHz}$.

$d_{\mathrm{Obj}}\,/\,\mathrm{m}$	$r_{\mathrm{B}}\,/\,\mathrm{MHz}$	$\overline{C}_{\mathrm{Obj}}$	$\mu_{\mathrm{Obj}}\,\vert_{\,r_{\mathrm{L}}=10\,\mathrm{MHz}}$
3.19	8.35	20.24	1.84
6.32	8.58	11.43	1.55
11.38	8.35	3.84	1.17
13.44	8.80	2.62	1.02
16.77	8.53	1.65	0.85
18.78	8.63	1.20	0.76
21.59	7.07	1.00	0.74

trend and gradually approaches $\overline{\mu}_{\mathrm{TOF}}\,|_{\,r_{\mathrm{L}}=10\,\mathrm{MHz}}$. In this case, the characteristics of the TC-Hists in D2 become similar to that in D1.

Overall Accuracy

The prediction results on D2 are shown in Table 5.15. Due to a high r_{L} with $d_{\mathrm{Obj}} \in (0,\ 16.77]\,\mathrm{m}$, the input data is dominated by the object feature, while other features have only minor effects. This greatly reduces the difficulty of distance prediction. Therefore, the accuracies of all methods on D2 almost reach 100 %. As d_{Obj} further increases, the performance differences among different variants starts to emerge. With $d_{\mathrm{Obj}} = \{18.78,\ 21.59\}\,\mathrm{m}$, Acc_{D} of CDP, FNN$_3$-MPA, and NBC*-MPA are clearly reduced, while other MPA variants show a stable performance on all distances. It should be noted that although the performance difference between NBC-MPA and NBC*-MPA is small in D1, the former shows a better generalizability due to the applied binomial estimation and has a clearly higher Acc_{D} with $d_{\mathrm{Obj}} = 21.59\,\mathrm{m}$ over the latter.

Stride Analysis on Dataset D2

The overall accuracies of four variants with different strides on D2 are shown in Table 5.16. With only seven different object distances on D2, there is some degree of fluctuation in Acc_{D}. Similar to Table 5.11, the MPA variants outperform CDP

Table 5.15: Prediction accuracy ($Acc_{\mathrm{D}}\,/\,\%$) of different variants on D2.

	$d_{\mathrm{Obj}}\,/\,\mathrm{m}$						
Variant	3.19	6.32	11.38	13.44	16.77	18.78	21.59
CDP	99.50	100	99.80	99.90	99.00	93.10	88.20
FNN-MPA	99.80	100	99.80	99.90	99.50	97.70	96.80
FNN_1-MPA	99.80	100	99.80	99.90	99.50	97.00	97.60
FNN_2-MPA	99.80	100	99.80	99.90	99.50	97.70	96.10
FNN_3-MPA	99.80	100	99.80	99.90	98.80	87.00	88.50
NBC-MPA	99.80	100	99.70	98.80	99.20	96.80	94.60
NBC^{*}-MPA	99.80	100	99.70	99.80	99.10	96.20	84.60

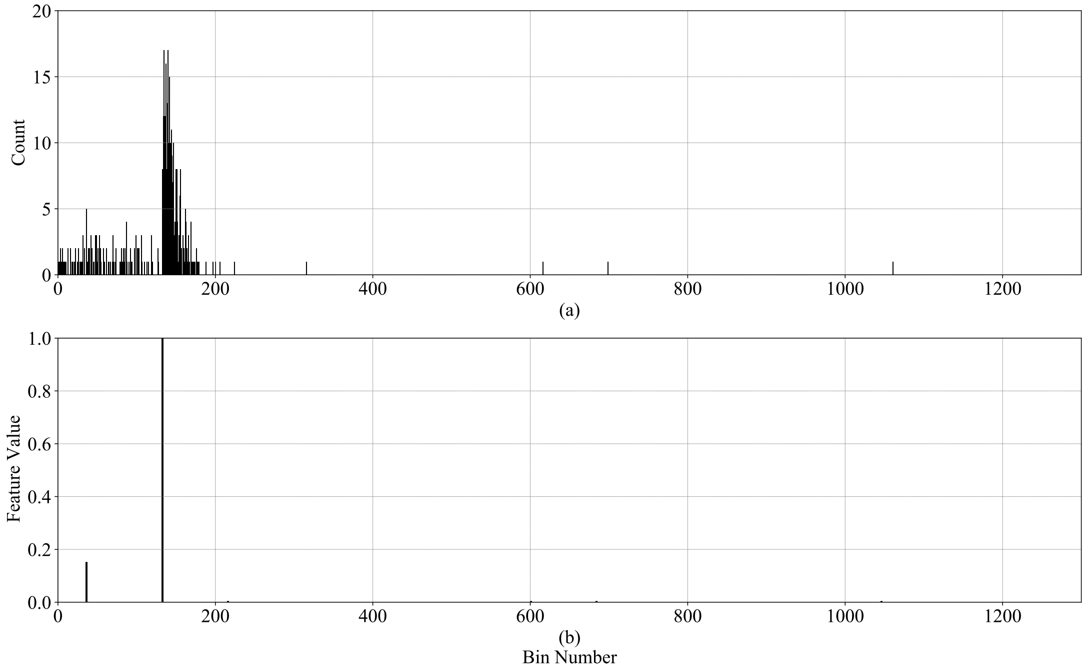


Figure 5.23: An example with $d_{\mathrm{Obj}} = 6.32\,\mathrm{m}$ from D2. (a) raw TC-Hist (b) extracted features.

with all listed strides. Besides, FNN-MPA is slightly better than NBC-MPA, which is observed in Table 5.11 as well.

Table 5.16: Performance of variants with different strides on D2.

	Stride				
Accuracy / %	1	2	3	5	7
Acc_{D}, CDP	97.07	96.11	89.89	92.66	88.97
Acc_{C}, FNN-MPA	99.43	99.51	99.50	99.36	99.33
Acc_{D}, FNN-MPA	98.96	99.13	99.13	99.13	91.14
Acc_{D}, NBC-MPA	98.56	98.61	98.54	98.51	90.07

5.6 Comparative Conclusions

In the previous sections, a number of MPA variants are studied on dataset D1 and D2. Therein, the six most promising variants, including FNN-MPA with stride = $\{1, 3\}$, the FNN$_1$-MPA, CO-FNN-MPA, NBC-MPA, and CO-NBC-MPA, are further listed together with CDP (as a benchmark) and compared jointly regarding their Acc_{D}, timing, and required data amount, as shown in Table 5.17. They are arranged in ascending order of Acc_{D} on D1. It can be seen that the trend of Acc_{D} on D2 is slightly different than Acc_{D} on D1. This is due to the lack of samples with $d_{\mathrm{Obj}} > 30\,\mathrm{m}$ on D2. In this case, Acc_{D} of the NBC-MPA variants on D2 are slightly worse than that of the FNN-MPA variants as their strengths are in the detection scenarios with $30\,\mathrm{m} < d_{\mathrm{Obj}} < 60\,\mathrm{m}$. It should be noted that FNN-MPA with stride = 3 requires only $107\,\mu\mathrm{s}$ for processing and achieves comparable Acc_{D} both on D1 and D2 to FNN-MPA and NBC-MPA with stride = 1. This gives it great potential for applications with high demands on real-time performance. In terms of the variants with the correlation analysis, CO-FNN-MPA with stride = 1 reaches the highest Acc_{D} on both D1 and D2. Besides, compared to CO-NBC-MPA, it has clear advantages in respect of processing speed and required data amount as well.

In summary, all listed MPA variants have better Acc_{D} on both D1 and D2 and they require less data amount compared to CDP. In particular, the FNN-MPA variants are faster than others regarding the processing time, while the NBC-MPA variants achieve the best Acc_{D} under the harshest conditions on D1. Finally, CO-FNN-MPA shows the best performance in terms of Acc_{D}, timing, and the amount of required data and parameters.

Table 5.17: Comparative performance of CDP and six MPA variants on D1 and D2. They are arranged from top to bottom in ascending order of Acc_{D} on D1.

		Acc_{D} / %			
Variant	Stride	D1	D2	Timing / μs	Data & Parameters
CDP	1	59.34	97.07	320	1311
FNN_1-MPA	1	67.71	99.09	250	192
FNN-MPA	3	68.09	99.13	107	192
FNN-MPA	1	68.56	98.96	260	192
NBC-MPA	1	69.31	98.56	2478	240
CO-NBC-MPA	1	82.46	99.29	2507	312
CO-FNN-MPA	1	82.91	99.34	289	264

Chapter 6

System Implementation and Demonstration

According to section 5.6, FNN-MPA and CO-FNN-MPA are selected and implemented on the LiDAR system “Owl” developed by Fraunhofer Institute for Microelectronic Circuits and Systems (IMS) using the programming environment LabVIEW [27]. First, the hardware specifications of the “Owl” are introduced. Afterwards, a PC-based demonstrator is presented, which includes porting the trained CO-FNN-MPA to LabVIEW. Subsequently, an FPGA-based implementation is described with a step-by-step approximation and implementation process for FNN-MPA. Finally, runtime tests of both implementations are demonstrated.

6.1 Demonstration Platform

The “Owl” [101], shown in Figure 6.1, is a direct TOF flash LiDAR system with a sensor front-end and an embedded processing module. Its sensor front-end includes a 2 × 192 SPAD line detector, two laser diodes with a wavelength of $\lambda = 905\,\mathrm{nm}$, and a TDC with a resolution of 312.5 ps. The laser emitted from each laser diode has 75 W peak power and is expanded into a line via optics and illuminates the scene with 36-degree FOV in the horizontal direction and 1 degree in the vertical direction. Both lasers emit at a repetition frequency of 10 kHz with a pulse width of 17.50 ns, resulting in a mean optical emission power of 11.24 mW. A TC-Hist generated by the “Owl” contains 400 measurement cycles. Each measurement cycle includes two main time windows: 1) a TOF window and 2) a counting window. In the TOF window, the “Owl” detects the distance information using the FPC procedure. In the counting window, the “Owl” operates in a free-running mode and estimates the intensity of background light. The embedded processing module used by the “Owl” is an Enclustra Zynq-7000 Mars ZX3 FPGA module, which is responsible for the control signal generation and data transmission. The module has an Artix-7 28 nm FPGA and an **A**dvanced **R**ISC **M**achines (ARM) dual-core Cortex-A9 microprocessor. The ARM processor establishes control commands for data transmission between the sensor front-end and a PC.

Figure 6.1: Flash LiDAR system “Owl” [101]. On the top is a camera, which is used to capture detection scenes. In the main part of the system, there are two laser sources on the left and right sides and a SPAD detector array in the middle.

In addition to the hardware, a PC-based LabVIEW script for displaying measurement results of the “Owl”, including the formation of TC-Hists and CDP, is programmed at the beginning of this work.

6.2 Implementation Solutions

The MPA variants are implemented both on a PC and on an FPGA module. First, CO-FNN-MPA is implemented in LabVIEW on a PC, including the convolution, the feature extraction, BE&S, a trained FNN, and the correlation analysis. CO-FNN-MPA porting only involves code conversion from Python to the LabVIEW-specific graphical language. No change is made regarding the realization of CO-FNN-MPA.

Afterwards, FNN-MPA is implemented on the FPGA module of the "Owl". Since the used embedded processing module Enclustra Zynq-7000 Mars ZX3 is a low-cost FPGA platform with only 18 Kb dual **B**lock **R**andom **A**ccess **M**emory (BRAM), the correlation analysis is discarded to maintain real-time performance. Hence, optimization and approximation on FNN-MPA must be carried out for the FPGA-based implementation, including (1) determination of a fixed-point arithmetic, (2) data preparation and feature extraction, (3) BES, and (4) prediction algorithm implementation.

6.2.1 Fixed-Point Arithmetic

Implementing fixed-point arithmetic units on FPGAs is much more efficient compared with floating-point ones, because determining the position of decimal points dynamically and dealing with corner cases require significant computational power. Therefore, the floating-point arithmetic utilized in the PC-based implementation is transformed into fixed-point arithmetic by specifying the value range of FNN-MPA. The specification includes 1) the maximum photon counts in a TC-Hist, 2) weights, bias, and intermediate values of FNN.

Photon Counts

According to (4.11), the photon counts in a TC-Hist have a minimum value of 0 and are proportional to $P_{1,b}$. Furthermore, according to (4.10) and (4.8), $P_{1,b}$ is inverse-proportional to t and proportional to r_B and r_L. Based on the parameter settings in D1, the maximum $P_{1,\mathrm{Max}}$ is obtained with $t = 0$, $r_\mathrm{B} = 8\,\mathrm{MHz}$, and $r_\mathrm{L} = 10\,\mathrm{MHz}$. Thus, by considering the pulse width $N_\mathrm{P} = 16$, $T_\mathrm{TDC} = 312.5\,\mathrm{ns}$, and $N_{\mathrm{M}} = 400$, $P_{1,\,\mathrm{Max}}$ is calculated by

$$P_{1,\,\mathrm{Max}} = \int_{b\,T_{\mathrm{TDC}}}^{(b+N_{\mathrm{P}})\,T_{\mathrm{TDC}}} p_1(t)dt \approx 0.08\ . \tag{6.1}$$

Hence, the value range $[0,\, C_{\mathrm{Max}}]$ that covers 99.99 % conditions is determined regarding (4.3) by

$$C_{\mathrm{Max}} = \arg\min_{C_{\mathrm{Max}}}((\sum_{k=0}^{C_{\mathrm{Max}}} P_{\mathrm{B}}(k\,|\,N_{\mathrm{M}},\, P_{1,\,\mathrm{Max}})) > 0.9999) = 56\ . \tag{6.2}$$

Weights, Bias, and Intermediate Values of FNN

Based on the optimal parameters specified for FNN-MPA in subsection 5.3.2, the intermediate value of FNN is determined according to (2.29). However, due to the stacked-layer structure of FNN and the countless number of input combinations, a direct determination of the global v_{Min} and v_{Max} is challenging.

Therefore, instead of the calculation of v_{Min} and v_{Max}, v'_{Min} and v'_{Max} are calculated, where $v'_{\mathrm{Min}} < v_{\mathrm{Min}}$ and $v'_{\mathrm{Max}} > v_{\mathrm{Max}}$, by modifying input x according to the sign of the corresponding weight. By taking the minimum as an example, if the inputs multiplied with negative weights take the maximum value and the inputs multiplied with positive weights take zero, the result v'_{Min} is smaller than or equal to all other possible v. Thus, v'_{Min} is given by

$$v'_{\mathrm{Min}} = \sum_{i=1}^{n} w_i\, x'_i + bias \tag{6.3}$$

with

$$x'_i = \begin{cases} \max_i(x'_i), & w_i \leqslant 0 \\ 0, & w_i > 0 \end{cases}\ . \tag{6.4}$$

Consequently, the global minimum is selected among all v'_{Min}. Similarly, v'_{Max} is calculated by maximizing inputs multiplied with positive weights and making inputs multiplied with negative weights equal to zero.

Finally, the value range of FNN-MPA is summarized in Table 6.1. It can be seen that the overall value range is $[-55.54,\, 56.00]$. Thus, a seven-bit integer part, including a sign bit, is sufficient. Since the overall maximum bits of the DSP block in Enclustra Mars ZX3 is 18, the remaining number of bits for the fractional part is 11. As a result, the fixed-point arithmetic Q7.11 is utilized with the value range of $[-64, 64]$ with the precision of 0.000488.

Table 6.1: Minimum and maximum values of generated weights and bias.

Value	Type	Minimum	Maximum
Photon Counts	Integer	0	56
w	Float	-22.54	14.44
$bias$	Float	-4.96	4.55
v	Float	-55.54 (v'_{Min})	37.57 (v'_{Max})

6.2.2 Data Preparation and Feature Extraction

In contrast to the PC-based implementation, the formation of TC-Hist is carried out on the FPGA module. The steps of accumulating one single measurement in a TC-Hist on FPGAs are: (1) fetching a digital timestamp, (2) calculating its bin number, (3) searching for the corresponding memory, (4) fetching the value in the memory, (5) increasing the value by one, and (6) writing the value back to the same memory. After repeating these steps multiple times until reaching the required N_{M}, a raw TC-Hist is formed.

In the PC-based implementation, the feature extraction includes two separated processing steps, i.e., the convolution and the local maximum extraction. In this case, the required clock cycles are:

$$N_{\mathrm{PC}} = N_{\mathrm{B}}(N_{\mathrm{P}} + 2 + N_{\mathrm{Sum}} + N_{\mathrm{Write}} + N_{\mathrm{Fetch}} + N_{\mathrm{Com}}) \,, \tag{6.5}$$

where N_{B} is the total number of bins in a TC-Hist. N_{Fetch} is the required number of clock cycles for a simple fetching operation, while $(N_{\mathrm{P}} + 2)$ is the number of clock cycles when the fetching operation is repeated N_{P} times continuously. N_{Com} and N_{Write} are the numbers of clock cycles for the comparison and the writing operation. Although the consecutive procedure causes barely negative effects on a PC, it brings a great challenge on FPGA in timing. Therefore, an optimization process is introduced, shown in Figure 6.2. The process combines the convolution and the feature extraction to save redundant operations. In this case, the optimized clock cycles are

$$N_{\mathrm{Opt}} = N_{\mathrm{P}} + 2 + N_{\mathrm{B}}(N_{\mathrm{Fetch}} + N_{\mathrm{Sum}} + N_{\mathrm{Com}}) \,. \tag{6.6}$$

Thus, the number of saved clock cycles ΔN_{CC} by the optimization process is approximated based on (6.5) and (6.6) as

$$\Delta N_{\mathrm{CC}} \approx N_{\mathrm{B}}(N_{\mathrm{Write}} + N_{\mathrm{P}} + 2) = 27237 \, , \tag{6.7}$$

Since the used main clock frequency is 6.25 ns, the proposed optimization saves around 0.17 ms for a TC-Hist from each pixel.

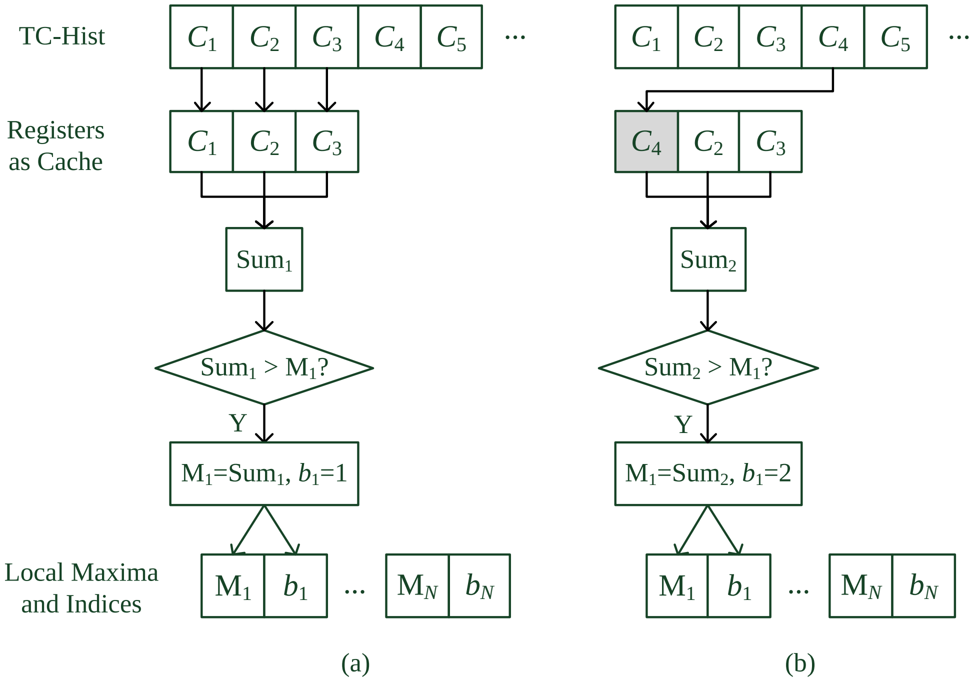


Figure 6.2: The principle of updating LiDAR data registers with the assumption of $N_{\mathrm{P}} = 3$. In (a), the registers as cache are initialized. After that, the sum of them is calculated. Before writing it to the register, it is compared to the previous maximum, and the update is carried out if necessary. In the following steps, as shown in (b), only the least recent data in the register array is updated and the rest processes are identical.

6.2.3 Background Light Estimation

As introduced in subsection 4.2.3, the expected photon counts $\mu_{\mathrm{B},b}$ in bin b is calculated according to (4.17). To further reduce the computations, the integration over $[b\,T_{\mathrm{TDC}}, (b+1)\,T_{\mathrm{TDC}}]$ is approximated by the multiplication of T_{TDC}

and $\mathrm{e}^{-r_{\mathrm{B}}\, b\, T_{\mathrm{TDC}}}$, since T_{TDC} is small. Thus, $\mu_{\mathrm{B},\, b}$ can be written as

$$\mu_{\mathrm{B},\, b} = N_{\mathrm{M}}\, r_{\mathrm{B}}\, T_{\mathrm{TDC}}\, \mathrm{e}^{-r_{\mathrm{B}}\, b\, T_{\mathrm{TDC}}} = r_{\mathrm{B}}\, \mathrm{k}_1\, \mathrm{k}_2^{r_{\mathrm{B}}\, b} \,, \tag{6.8}$$

with

$$\mathrm{k}_1 = N_{\mathrm{M}}\, T_{\mathrm{TDC}} \text{ and } \mathrm{k}_2 = \mathrm{e}^{-T_{\mathrm{TDC}}} \,. \tag{6.9}$$

Obviously, k_1 and k_2 are constants and can be pre-stored in registers. Furthermore, r_{B} is approximated using the quantized value r_{BQ} as

$$r_{\mathrm{BQ}} = \begin{cases} 0 & \text{if } r_{\mathrm{B}} \leqslant 1\,\mathrm{MHz} \\ \mathrm{round}(r_{\mathrm{B}}) & \text{else} \end{cases} \,, \tag{6.10}$$

It should be noticed that r_{B} and r_{BQ} are both in units of [MHz]. Since $b \in (0,\ 1310]$, $r_{\mathrm{BQ}} \in (0,\ 8]\,\mathrm{MHz}$ and both of them are integer, $r_{\mathrm{B}}\, b$ can be represented by an 18-bit integer value. Thus, the calculation of $\mathrm{k}_2^{r_{\mathrm{B}}\, b}$ is further accelerated by loop-operations using a look-up-table, which includes the pre-calculated value of $\{\mathrm{k}_2,\ \mathrm{k}_2^{500},\ \mathrm{k}_2^{1000},\ \mathrm{k}_2^{2000},\ \mathrm{k}_2^{3000},\ ...,\ \mathrm{k}_2^{10000}\}$. In this case, the required loop operations for $\mathrm{k}_2^{r_{\mathrm{B}}\, b}$ can be significantly reduced in cases that $r_{\mathrm{B}}\, b \gg 500$.

6.2.4 Prediction Algorithm Implementation

In terms of FNN, the trained weights and the bias values are stored in BRAM blocks using the Q7.11 fixed-point arithmetic. For fast and reliable arithmetical operations, the **D**igital **S**ignal **P**rocessor (DSP) blocks of the FPGA module are enabled. During the training process on a PC, the softmax is utilized as the activation function of the output layer for backpropagation and assigns a normalized value to each output. A large resource occupation on FPGA is required due to exponential calculations. Since the softmax function only acts as a scaling and shifting function during the prediction process, where only the maximum of the outputs is considered, the selection of the final prediction, i.e., the maximum value, is not changed after the softmax function. Therefore, it is removed in the FPGA-based implementation.

6.2.5 Software Simulation

A simulation for the aforementioned implementation steps is carried out on the Xilinx Vivado Design Suite. The hardware resource utilization from a simulation result is shown in Table 6.2. Firstly, an implementation with 384 pixels is deployed. It is seen that FNN-MPA has small occupations on most resources. However, a high BRAM occupation (72.50 %) is observed, causing more than 2800 timing violations. It should be mentioned that the resource utilization shown in Table 6.2 includes not only FNN-MPA, but also other necessary processes, e.g., measurement accumulation and TC-Hists formation. The high BRAM occupation is mainly caused by the formation and storage of TC-Hists. After gradually reducing the number of pixels (it reduces the number of TC-Hists per frame), an implementation with 96 pixels is selected for stability purposes.

Table 6.2: FNN-MPA resource occupation on FPGA.

	Memory Occupation / %	
Resources	384 Pixels	96 Pixels
LUT as Logic	12.39	11.98
LUT as Memory	2.85	2.85
Slice Registers as Flip-Flop	8.97	8.85
Slice Registers as Latch	0	0
BRAM	72.50	21.07
DSP Blocks	6.82	6.82

6.3 Demonstration Results

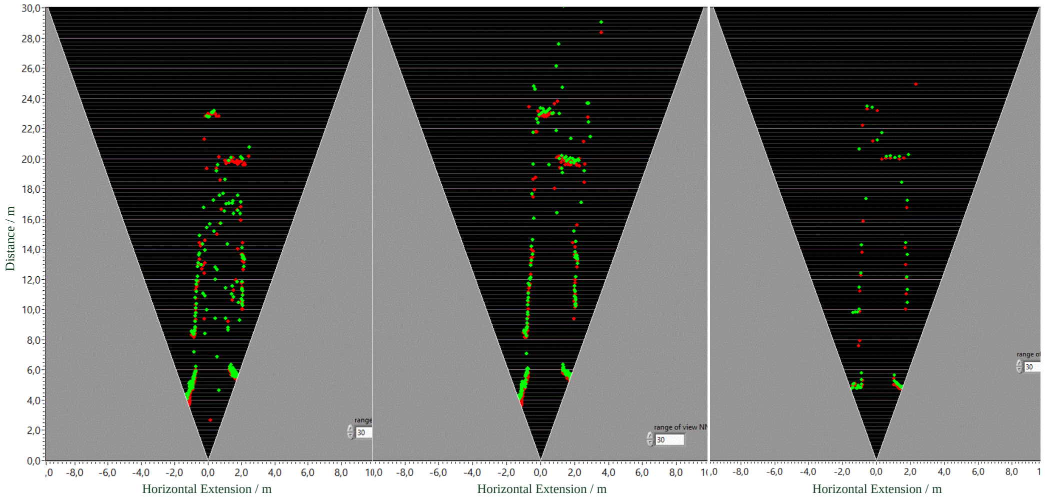
In order to demonstrate runtime performance, an indoor detection scene and an outdoor detection scene were constructed.

The indoor scene is set up in the same environment used for D2 in Figure 5.1. However, in contrast to the setup in Figure 5.1, the halogen floodlight is aimed directly at the detector array to create a higher coverage of high background light (from 5 to 10 MHz) over the entire detector array. Figure 6.3 shows the specified indoor scene with an object at 20 m. The runtime records of detection results are shown in Figure 6.4. Due to high r_{B} generated by the halogen floodlight, a large number of noise points are observed in Figure 6.4 (a). Besides, since the background light is incident from slanted above, the green detection points are

Figure 6.3: Front view of indoor detection scene. The scene includes an object at 20 m, two side walls, and two background walls at 23 m and 27.5 m. A halogen floodlight is aimed directly at the "Owl". It creates high r_{B} (from 5 to 10 MHz) over the entire detector array.

more affected than the red detection points. In Figure 6.4 (b), the noise points are clearly reduced and the detection points from the object at 20 m and the background wall at 23 m can be well identified. There are some noise detection points at the edges of objects, since there are barely any laser echoes due to the large incidence angle of the laser on the object edges. In terms of Figure 6.4 (c), the detection points of the object at 20 m are clear and stable as well. Since the pixel number in the FPGA-based implementation is reduced to 96, there are only a few detection points recorded from the background wall. Therefore, it cannot be observed if the FPGA-based implementation detects the background wall correctly. However, it should be noticed that similar to Figure 6.4 (b), Figure

6.4 (c) has barely any noise points in the area of short distances.

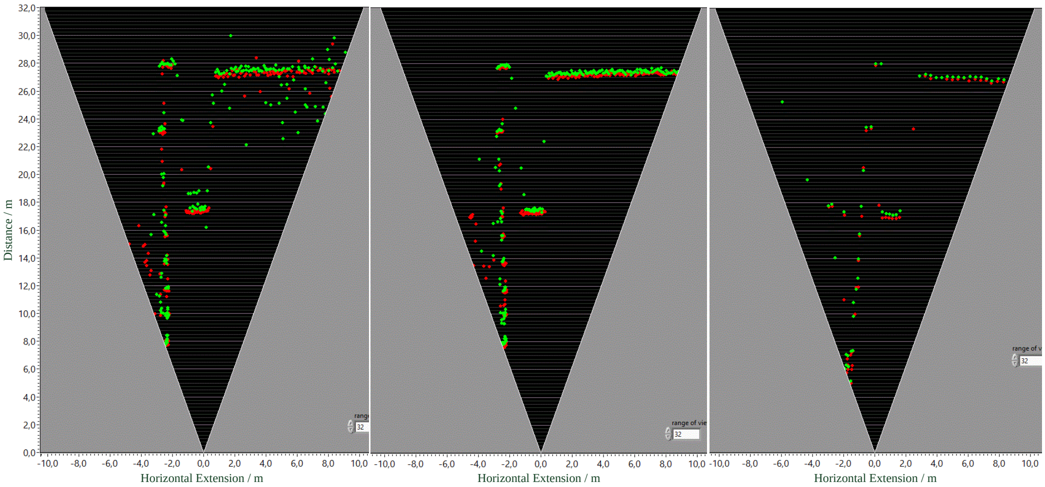


(a) (b) (c)

Figure 6.4: Bird's-eye view of indoor runtime records with experimental setup in Figure 5.1. The green points represent the detection results from the upper detector array and the red points represent the detection results from the lower detector array of the "Owl" system. (a) CDP. (b) FNN-MPA on PC. (c) FNN-MPA on FPGA.

The outdoor scene is shown in Figure 6.5. Since $r_{\mathrm{B}} < 3\,\mathrm{MHz}$, objects and background walls can be clearly recognized in the detection results of all three implementations. However, the numbers of noise points in Figure 6.5 (b) and Figure 6.5 (c) are less than that in Figure 6.5 (a).

Figure 6.5: Front view of outdoor detection scene. The scene includes an object at 17.50 m, a side wall, and two background walls at 27.50 m and 28 m. r_B is within (1, 3] MHz.



(a) (b) (c)

Figure 6.6: Bird's-eye view of outdoor runtime records. The green points represent the detection results from the upper detector array and the red points represent the detection results from the lower detector array of the "Owl" system. (a) CDP. (b) FNN-MPA on PC. (c) FNN-MPA on FPGA.

Chapter 7

Summary

The overall objective of this work is to design a machine learning-based method on low-level LiDAR data with focus on improving robustness against background light and real-time performance of distance determination. In order to develop the most suitable method, solutions available in the literature were first reviewed with respect to LiDAR hardware, data workflow, and machine learning in the field of distance measurements. On the basis of the review, multiple attempts to combine machine learning and low-level LiDAR data were carried out with theory and simulation for desired improvements. Ultimately, a novel approach called MPA, including a physics-guided feature extraction, two machine learning-based distance prediction algorithms, a correlation algorithm, and three optimization possibilities, is proposed. MPA includes two main variants, i.e., FNN-MPA and NBC-MPA. They are evaluated on a dataset with 96,000 synthetic TC-Hists and a dataset with 7000 TC-Hists generated by the LiDAR system “Owl”. The parameter determination of MPA is specified on the basis of comparative analysis. As a benchmark for comparison, one widely-used CDP method is utilized, including an average filter, a background subtraction process, and a peak detection. Finally, two runtime demonstrations, including a PC-based implementation and an FPGA-based implementation, are provided.

The entirety of the work leads to the conclusion that by using the concept of machine learning, the integrity and efficiency of low-level LiDAR data processing, i.e., TC-Hist, can be improved. The resulting distance determination with a $\pm 5\,\%$ error bound is more reliable under high background light and faster than that of CDP. In particular, outcomes based on the datasets used in this work are made as: 1) MPA outperforms CDP on both the synthetic and the real data, especially when the object distance is large and the background light is high. Moreover, FNN-MPA is at least 18.75 % faster than CDP. NBC-MPA achieves a better overall performance over FNN-MPA, while requiring considerably more time for data processing. 2) With the specified parameters, a distance prediction is challenging with background photon detection rates greater than 5 MHz and object distances from 40 m to 60 m. A TC-Hist accumulated under such conditions has a low SNR. Since only a few laser signals are collected in a TC-Hist under these conditions, we argue that instead of further development in processing algorithms, a promising prediction result can only be obtained with an improvement before or during the formation of a TC-Hist, e.g., increasing effective laser photon rate or increasing the number of measurements per TC-Hist. 3) In terms of FNN-MPA, the background suppression could be saved to improve the feasibility and timing performance on hardware, when the possible background light intensities in practice are well-covered in the training dataset. Nevertheless, a degradation of generalizability is observed using both FNN-MPA and NBC-MPA. 4) Using the correlation analysis proposed in this work, which utilizes the time and space information at the level of TC-Hists, the reliability of distance prediction is visibly improved. 5) MPA has better stability than CDP as the convolution stride increases. This enables implementation possibilities in applications requiring extremely high frame rate.

The contributions of the work can be divided into six aspects: 1) A joint study is carried out between low-level LiDAR data features and machine learning algorithms and a possible solution of combining them is proposed. 2) A physics-guided feature extraction is designed, which reduces a complete TC-Hist (with more than 1000 values) to multiple features (with dozens of values). By configuring subsequent algorithms, e.g., FNN and NBC, the extracted features are proved to be sufficient for object distance determination. In the datasets used in this work, the proposed algorithms on these features even show a superior performance over CDP, which relies on complete TC-Hist for distance determination. It shows that the distance information could be determined by focusing most computations on representative features instead of on the complete histograms, which may significantly reduce the required data to be stored and transmitted for processing. 3) The point cloud formed from the output of MPA has more information dimensions than traditional point clouds. For example, MPA provides not only the distance information, but also the corresponding certainty of each measured point. This enables a distinguishability between object and noise. Besides, MPA can provide multiple potential object distances for one TC-Hist. This property could be beneficial for the application of multi-reflection analysis. 4) The correlation analysis proposed in this work is carried out at the TC-Hist level. In contrast to a correlation at the point cloud level, the correlation analysis involves multiple potential distances and photon statistics in time and space domain. This results in a high suppression effect on noise and positive impacts even when both the original data and the reference data are noisy. The working principle could inspire future research in the field of low-level sensor data fusion. 5) In the field of high-level LiDAR data processing, such as depth completion and object detection/tracking, the additional information provided by MPA (multiple potential distances and prediction certainties) brings advantages in minimizing the negative effects of noise points. 6) In ADAS, MPA could provide a reliable and low latency solution for object distance determination in a fast-changing environment due to its high robustness to the background light and short processing time. In addition to the distance prediction, MPA has a high performance in determining the region where the object locates. This could be useful in the application relying on the region-of-interest.

In the scope of this work, the objectives proposed in section 1.2 were fulfilled. In addition, the research gaps in section 3.6 were addressed. In summary, the work has been successfully conducted and promising results have been obtained. Potential works could be carried out in the future:

1) **New data form**: A large memory is required for storing TC-Hists of all pixels. This results in high demands on memory space and the transmission rate of LiDAR systems. In this work, it is observed that the distance prediction performs well using a small number of key features from a complete TC-Hist. Therefore, instead of TC-Hist, a new data form and a corresponding generation process could be carried out, which directly outputs required key features using a series of LiDAR timestamps. In this case, the volume of LiDAR data could be reduced and the processing efficiency could be improved as well.

2) **Further detection conditions**: This work focuses on the system robustness and distance prediction reliability under changing background light. Other

conditions are of great interest as well, such as rain, fog, and multi-reflections. MPA could be applied to extract and to process key features of these conditions individually. Finally, a decision algorithm could be designed by considering MPAs specified for different conditions aiming at making optimal predictions under various of complex detection scenarios.

3) **Pixel-level object tracking**: The correlation analysis in this work studies the information correlation in time and space domain at the level of TC-Hist. It could be extended to estimate the momentum vector of each pixel. Different sensors, for example radars, could be used to provide velocity information of the object.

4) **Smart scanning**: In addition to software-level extensions, a combination with scanning mechanisms in time and space is attractive as well. With the potential distances and prediction certainties provided by MPA, the LiDAR system equipped with the scanning technology could have solid arguments for dynamically adjusting the detection area in time and in space. In this case, the focus and resources of a LiDAR system could be more rationally deployed and better system performance could be achieved.

List of Publications

Peer Reviewed Journal Articles

Gongbo Chen, Christian Wiede, and Rainer Kokozinski. Data processing approaches on SPAD-based d-TOF LiDAR systems: A review. *IEEE Sensors Journal*, 21(5):5656-5667, 2021. doi: 10.1109/JSEN.2020.3038487.

Gongbo Chen, Felix Landmeyer, Christian Wiede, and Rainer Kokozinski. Feature extraction and neural network-based multi-peak analysis on time-correlated LiDAR histograms. *Journal of Optics*, 24(3):034008, feb 2022. doi: 10.1088/2040-8986/ac486d.

Conference Proceedings

Gongbo Chen, Pierre Gembaczka, Christian Wiede, and Rainer Kokozinski. Feature extraction and neural network-based analysis on time-correlated LiDAR histograms. In *PHOTOPTICS*, 2021.

Gongbo Chen, Giray Atabey Kirtiz, Christian Wiede, and Rainer Kokozinski. Implementation and evaluation of a neural network-based LiDAR histogram processing method on FPGA. In *IEEE International System-on-Chip Conference*, 2021.

Appendix A

Experimental Datasheets

Table A.1: Sub-region division and corresponding ranges in MPA with $N_{\mathrm{F}} = 12$.

Sub-Region	Range / m	Sub-Region	Range / m
1	[0, 5.06)	7	[30.35, 35.41)
2	[5.06, 10.12)	8	[35.41, 40.47)
3	[10.12, 15.18)	9	[40.47, 45.53)
4	[15.18, 20.24)	10	[45.53, 50.59)
5	[20.24, 25.29)	11	[50.59, 55.65)
6	[25.29, 30.35)	12	[55.65, 60.71)

Table A.2: Accuracy of feature extraction on D1.

	1	2	3	4	5	6	7	8
(0, 10]	93.50	93.40	91.25	89.70	87.50	83.50	82.60	76.40
(10, 20]	100	99.80	99.35	98.60	96.45	92.00	87.55	83.50
(20, 30]	100	99.85	98.95	97.45	93.75	90.50	84.10	80.55
(30, 40]	100	99.75	98.35	97.20	92.90	90.05	84.15	81.40
(40, 50]	100	99.70	99.00	96.35	94.55	91.35	88.70	85.20
(50, 60]	99.95	99.70	98.55	96.60	94.65	91.90	90.00	89.15

Table A.3: FNN prediction performance with $N_{\mathrm{F}} = 8$ on $\mathrm{D}_{\mathrm{Valid}}$.

AF	N_{HN}	Acc_{C}/%	PPV/%	DPR/%	F1-Score/%	AUC/%
None	0	71.75	71.35	71.75	71.29	94.01
None	8	71.67	71.39	71.67	71.19	94.01
None	18	71.7	71.34	71.7	71.26	94.01
None	28	71.63	71.38	71.63	71.19	93.98
ReLU	8	71.68	71.36	71.68	71.16	94.02
ReLU	18	71.71	71.39	71.71	71.26	94.02
ReLU	28	71.73	71.32	71.73	71.36	94.06
Sigmoid	8	71.57	71.26	71.57	71.12	93.96
Sigmoid	18	71.68	71.36	71.68	71.3	94.19
Sigmoid	28	71.7	71.35	71.7	71.3	94.2
Average	0	71.68	71.35	71.68	71.24	94.05
STD	0	0.05	0.03	0.05	0.07	0.07

Table A.4: FNN prediction performance with $N_\mathrm{F} = 12$ on $\mathrm{D}_\mathrm{Valid}$.

AF	N_HN	Acc_C/%	PPV/%	DPR/%	F1-Score/%	AUC/%
None	0	70.44	70.01	70.44	69.87	94.24
None	12	70.28	70.05	70.28	69.64	94.23
None	22	70.33	69.91	70.33	69.78	94.23
None	32	70.3	69.91	70.3	69.79	94.22
ReLU	12	70.4	69.98	70.4	69.83	94.24
ReLU	22	70.4	69.94	70.4	69.82	94.28
ReLU	32	70.3	69.89	70.3	69.71	94.26
Sigmoid	12	70.19	69.76	70.19	69.65	94.23
Sigmoid	22	70.21	69.75	70.21	69.71	94.22
Sigmoid	32	70.4	70.0	70.4	69.9	94.29
Average	0	70.32	69.92	70.32	69.77	94.24
STD	0	0.08	0.09	0.08	0.08	0.02

Table A.5: FNN prediction performance with $N_\mathrm{F} = 16$ on $\mathrm{D}_\mathrm{Valid}$.

AF	N_HN	Acc_C/%	PPV/%	DPR/%	F1-Score/%	AUC/%
None	0	69.86	69.36	69.86	69.25	95.01
None	16	69.73	69.34	69.73	69.13	95.01
None	26	69.79	69.38	69.79	69.19	95.01
None	36	69.7	69.38	69.7	69.09	95.02
ReLU	16	69.71	69.32	69.71	69.18	94.96
ReLU	26	69.7	69.23	69.7	69.16	95.04
ReLU	36	69.65	69.28	69.65	69.04	95.05
Sigmoid	16	69.62	69.15	69.62	69.01	94.98
Sigmoid	26	69.55	69.08	69.55	68.94	94.99
Sigmoid	36	69.7	69.32	69.7	69.17	95.1
Average	0	69.7	69.28	69.7	69.12	95.02
STD	0	0.08	0.09	0.08	0.09	0.04

Table A.6: FNN prediction performance with $N_\mathrm{F} = 24$ on $\mathrm{D}_\mathrm{Valid}$.

AF	N_HN	Acc_C/%	PPV/%	DPR/%	F1-Score/%	AUC/%
None	0	67.59	67.3	67.59	66.96	95.29
None	24	67.49	67.35	67.49	66.85	95.36
None	34	67.57	67.31	67.57	67.0	95.35
None	44	67.51	67.19	67.51	66.91	95.37
ReLU	24	67.48	67.28	67.48	66.91	95.35
ReLU	34	67.46	67.38	67.46	66.83	95.35
ReLU	44	67.53	67.22	67.53	66.89	95.41
Sigmoid	24	67.18	66.95	67.18	66.6	95.3
Sigmoid	34	67.26	67.11	67.26	66.7	95.32
Sigmoid	44	67.35	67.31	67.35	66.81	95.35
Average	0	67.44	67.24	67.44	66.85	95.34
STD	0	0.12	0.12	0.12	0.11	0.03

Table A.7: Saved values of $({k_2}^{b\,r_\mathrm{B}})$ as LUT.

	$(\mathrm{k}_2)^{b\,r_\mathrm{B}}$	
$b\,r_\mathrm{B}$	Floating-Point	Q7.11
1	0.99968754882	0.99951171875
500	0.85534532730	0.85498046875
1000	0.73161562895	0.73144531250
2000	0.53526142852	0.53466796875
3000	0.39160562668	0.39111328125
4000	0.28650479686	0.28613281250
5000	0.20961138715	0.20947265625
6000	0.15335496685	0.15332031250
7000	0.11219689052	0.11181640625
8000	0.08208499862	0.08203125000
9000	0.06005466789	0.05957031250
10000	0.04393693362	0.04345703125

Bibliography

[1] European Road Safety Observatory. Annual accident report 2018. URL `https://ec.europa.eu/transport/road_safety/sites/roadsafety/files/pdf/statistics/dacota/asr2018.pdf`. Accessed: 05-05-2021.

[2] Amir Mukhtar, Likun Xia, and Tong Boon Tang. Vehicle detection techniques for collision avoidance systems: A review. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2318–2338, 2015. doi: 10.1109/TITS.2015.2409109.

[3] Radu Horaud, Miles Hansard, Georgios Evangelidis, and Clément Ménier. An overview of depth cameras and range scanners based on time-of-flight technologies. *Machine Vision and Applications*, 27(7):1432–1769, 2016. doi: 10.1007/s00138-016-0784-4.

[4] François Blais. Review of 20 years of range sensor development. *J. Electronic Imaging*, 13(1):231–243, 2004. doi: 10.1117/1.1631921.

[5] Stephen Alland, Wayne Stark, Murtaza Ali, and Manju Hegde. Interference in automotive radar systems: Characteristics, mitigation techniques, and current and future research. *IEEE Signal Processing Magazine*, 36(5):45–59, 2019. doi: 10.1109/MSP.2019.2908214.

[6] Behnam Behroozpour, Phillip A. M. Sandborn, Ming C. Wu, and Bernhard E. Boser. Lidar system architectures and circuits. *IEEE Communications Magazine*, 55(10):135–142, 2017. doi: 10.1109/MCOM.2017.1700030.

[7] Mubariz Zaffar, Shoaib Ehsan, Rustam Stolkin, and Klaus McDonald Maier. Sensors, slam and long-term autonomy: A review. In *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, pages 285–290, 2018. doi: 10.1109/AHS.2018.8541483.

[8] Cristiano Premebida, Goncalo Monteiro, Urbano Nunes, and Paulo Peixoto. A LiDAR and vision-based approach for pedestrian and vehicle detection and tracking. In *2007 IEEE Intelligent Transportation Systems Conference*, pages 1044–1049, 2007. doi: 10.1109/ITSC.2007.4357637.

[9] François Piron, Daniel Morrison, Mehmet Rasit Yuce, and Jean-Michel Redouté. A review of single-photon avalanche diode time-of-flight imaging sensor arrays. *IEEE Sensors Journal*, 21(11):12654–12666, 2021. doi: 10.1109/JSEN.2020.3039362.

[10] Velodyne LiDAR Inc. products main page. URL `https://velodynelidar.com`. Accessed: 01-07-2020.

[11] Ibeo automotive systems GmbH. URL `https://www.ibeo-as.com/de/produkte`. Accessed: 01-07-2020.

[12] Quanergy systems. URL `https://quanergy.com`. Accessed: 01-07-2020.

[13] Ouster. URL `https://ouster.com`. Accessed: 01-07-2020.

[14] Maik Beer, Jan F. Haase, Jennifer Ruskowski, and Rainer Kokozinski. Background light rejection in SPAD-based LiDAR sensors by adaptive photon coincidence detection. *Sensors*, 18(12), 2018. ISSN 1424-8220. doi: 10.3390/s18124338. URL `https://www.mdpi.com/1424-8220/18/12/4338`.

[15] Danilo Bronzi, Yu Zou, Federica Villa, Simone Tisa, Alberto Tosi, and Franco Zappa. Automotive three-dimensional vision through a single-photon counting SPAD camera. *IEEE Transactions on Intelligent Transportation Systems*, 17(3):782–795, 2016. doi: 10.1109/TITS.2015.2482601.

[16] Aude Martin, Delphin Dodane, Luc Leviandier, Daniel Dolfi, Alan Naughton, Peter O'Brien, Thijs Spuessens, Roel Baets, Guy Lepage, Peter Verheyen, Peter De Heyn, Philippe Absil, Patrick Feneyrou, and Jérôme Bourderionnet. Photonic integrated circuit-based FMCW coherent LiDAR. *Journal of Lightwave Technology*, 36(19):4640–4645, 2018. doi: 10.1109/JLT.2018.2840223.

[17] Cristiano Niclass, Mineki Soga, Hiroyuki Matsubara, Masaru Ogawa, and Manabu Kagami. A 0.18-μ m CMOS SoC for a 100-m-range 10-frame/s

200 × 96-pixel time-of-flight depth sensor. *IEEE Journal of Solid-State Circuits*, 49(1):315–330, 2014. doi: 10.1109/JSSC.2013.2284352.

[18] J. Nissinen, I. Nissinen, S. Jahromi, T. Talala, and J. Kostamovaara. Time-gated CMOS SPAD and a quantum well laser diode with a CMOS driver for time-resolved diffuse optics imaging. In *2018 IEEE Nordic Circuits and Systems Conference (NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC)*, pages 1–4, 2018. doi: 10.1109/NORCHIP.2018.8573525.

[19] Chao Zhang, Scott Lindner, Ivan Michel Antolovic, Juan Mata Pavia, Martin Wolf, and Edoardo Charbon. A 30-frames/s, 252 × 144 SPAD flash LiDAR with 1728 dual-clock 48.8-ps TDCs, and pixel-wise integrated histogramming. *IEEE Journal of Solid-State Circuits*, 54(4):1137–1151, 2019. doi: 10.1109/JSSC.2018.2883720.

[20] Guoqing Zhou, Xiang Zhou, Jiazhi Yang, Yue Tao, Xueqin Nong, and Oktay Baysal. Flash lidar sensor using fiber-coupled APDs. *IEEE Sensors Journal*, 15(9):4758–4768, 2015. doi: 10.1109/JSEN.2015.2425414.

[21] Atul Ingle, Andreas Velten, and Mohit Gupta. High flux passive imaging with single-photon sensors, 2019.

[22] Waymo main page. URL `https://waymo.com/tech`. Accessed: 01-07-2020.

[23] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. doi: 10.1109/CVPR.2012.6248074.

[24] Andrea Gallivanoni, Ivan Rech, and Massimo Ghioni. Progress in quenching circuits for single photon avalanche diodes. *IEEE Transactions on Nuclear Science*, 57(6):3815–3826, 2010. doi: 10.1109/TNS.2010.2074213.

[25] Walther Bothe. Coincidence method. URL `https://www.nobelprize.org/prizes/physics/1954/bothe/lecture/`. Accessed: 25-08-2021.

[26] M.D. Adams. Lidar design, use, and calibration concepts for correct environmental detection. *IEEE Transactions on Robotics and Automation*, 16 (6):753–761, 2000. doi: 10.1109/70.897786.

[27] LabVIEW. URL `https://www.ni.com/de-de/shop/labview.html`. Accessed: 06-12-2021.

[28] Anna Vil, Anna Arbat, Eva Vilella, and Angel Dieguez. *Geiger-Mode Avalanche Photodiodes in Standard CMOS Technologies*. 03 2012. ISBN 978-953-51-0358-5. doi: 10.5772/37162.

[29] S. Cova, M. Ghioni, A. Lacaita, C. Samori, and F. Zappa. Avalanche photodiodes and quenching circuits for single-photon detection. *Appl. Opt.*, 35(12):1956–1976, Apr 1996. doi: 10.1364/AO.35.001956.

[30] Klaus Pasquinelli, Rudi Lussana, Simone Tisa, Federica Villa, and Franco Zappa. Single-photon detectors modeling and selection criteria for high-background LiDAR. *IEEE Sensors Journal*, 20(13):7021–7032, 2020. doi: 10.1109/JSEN.2020.2977775.

[31] Han Woong Yoo, Norbert Druml, David Brunner, Christian Schwarzl, Thomas Thurner, Marcus Hennecke, and Georg Schitter. MEMS-based lidar for autonomous driving. *Elektrotechnik und Informationstechnik*, 135 (6):408–415, 2018. doi: 10.1007/s00502-018-0635-2.

[32] P.F. McManamon, T.A. Dorschner, D.L. Corkum, L.J. Friedman, D.S. Hobbs, M. Holz, S. Liberman, H.Q. Nguyen, D.P. Resler, R.C. Sharp, and E.A. Watson. Optical phased array technology. *Proceedings of the IEEE*, 84(2):268–298, 1996. doi: 10.1109/5.482231.

[33] Ching-Pai Hsu, Boda Li, Braulio Solano-Rivas, Amar R. Gohil, Pak Hung Chan, Andrew D. Moore, and Valentina Donzella. A review and perspective on optical phased array for automotive LiDAR. *IEEE Journal of Selected Topics in Quantum Electronics*, 27(1):1–16, 2021. doi: 10.1109/JSTQE.2020.3022948.

[34] Santiago Royo and Maria Ballesta-Garcia. An overview of LiDAR imaging systems for autonomous vehicles. *Applied Sciences*, 9(19), 2019. ISSN

2076-3417. doi: 10.3390/app9194093. URL `https://www.mdpi.com/2076-3417/9/19/4093`.

[35] Kurt E. Petersen. Silicon torsional scanning mirror. *IBM Journal of Research and Development*, 24(5):631–637, 1980. doi: 10.1147/rd.245.0631.

[36] Sven T. S. Holmström, Utku Baran, and Hakan Urey. MEMS laser scanners: A review. *Journal of Microelectromechanical Systems*, 23(2):259–275, 2014. doi: 10.1109/JMEMS.2013.2295470.

[37] Standard solar spectra. URL `https://www.pveducation.org/pvcdrom/appendices/standard-solar-spectra`. Accessed: 19-08-2021.

[38] Keith. Knight. *Mathematical Statistics*. Chapman and Hall/CRC, 2000. ISBN 9781584881780.

[39] James Joyce. Bayes' theorem. *The Stanford Encyclopedia of Philosophy (Fall 2021 Edition)*, 2021. URL `https://plato.stanford.edu/archives/fall2021/entries/bayes-theorem/`.

[40] Gernot A. Fink. *Markov Models for Pattern Recognition*. Springer-Verlag London, 2nd edition edition, 2014.

[41] T. Kohonen. An introduction to neural computing. *Neural Networks*, 1: 3–16, 1988. doi: 10.1016/0893-6080(88)90020-2.

[42] R. M. Hristev. *The ANN Book*. 1998.

[43] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5: 115–133, 1943. doi: 10.1007/BF02478259.

[44] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[45] Gongbo Chen, Christian Wiede, and Rainer Kokozinski. Data processing approaches on SPAD-based d-TOF LiDAR systems: A review. *IEEE Sensors Journal*, 21(5):5656–5667, 2021. doi: 10.1109/JSEN.2020.3038487.

[46] T. Hanke, N. Hirsenkorn, B. Dehlink, A. Rauch, R. Rasshofer, and E. Biebl. Generic architecture for simulation of ADAS sensors. In *2015 16th International Radar Symposium (IRS)*, pages 125–130, 2015. doi: 10.1109/IRS.2015.7226306.

[47] Philipp Rosenberger, Martin Holder, Sebastian Huch, Hermann Winner, Tobias Fleck, Marc René Zofka, J. Marius Zöllner, Thomas D'hondt, and Benjamin Wassermann. Benchmarking and functional decomposition of automotive lidar sensor models. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 632–639, 2019. doi: 10.1109/IVS.2019.8814081.

[48] Thomas Fersch, Robert Weigel, and Alexander Koelpin. Challenges in miniaturized automotive long-range lidar system design. In Bahram Javidi, Jung-Young Son, and Osamu Matoba, editors, *Three-Dimensional Imaging, Visualization, and Display 2017*, volume 10219, pages 160–171. International Society for Optics and Photonics, SPIE, 2017. doi: 10.1117/12.2260894.

[49] Juha Kostamovaara, Jaakko Huikari, Lauri Hallman, Ilkka Nissinen, Jan Nissinen, Harri Rapakko, Eugene Avrutin, and Boris Ryvkin. On laser ranging based on high-speed/energy laser diode pulses and single-photon detection techniques. *IEEE Photonics Journal*, 7(2):1–15, 2015. doi: 10.1109/JPHOT.2015.2402129.

[50] Garner Rob. Solar irradiance. URL `https://www.nasa.gov/mission_pages/sdo/science/solar-irradiance.html`. Accessed: 25-08-2021.

[51] Andreas Süss, Veronique Rochus, Maarten Rosmeulen, and Xavier Rottenberg. Benchmarking time-of-flight based depth measurement techniques. In Sailing He, El-Hang Lee, and Louay A. Eldada, editors, *Smart Photonic and Optoelectronic Integrated Circuits XVIII*, volume 9751, pages 199–217. International Society for Optics and Photonics, SPIE, 2016. doi: 10.1117/12.2212478.

[52] Matteo Perenzoni, Daniele Perenzoni, and David Stoppa. 6.5 a 64 × 64-pixel digital silicon photomultiplier direct ToF sensor with 100mphotons/s/pixel background rejection and imaging/altimeter mode with 0.14 % precision up

to 6 km for spacecraft navigation and landing. In *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 118–119, 2016. doi: 10.1109/ISSCC.2016.7417935.

[53] Majeed M. Hayat, Sergio N. Torres, and Leno M. Pedrotti. Theory of photon coincidence statistics in photon-correlated beams. *Optics Communications*, 169(1):275–287, 1999. ISSN 0030-4018. doi: 10.1016/S0030-4018(99)00384-3.

[54] Davide Portaluppi, Enrico Conca, and Federica Villa. 32 × 32 CMOS SPAD imager for gated imaging, photon timing, and photon coincidence. *IEEE Journal of Selected Topics in Quantum Electronics*, 24(2):1–6, 2018. doi: 10.1109/JSTQE.2017.2754587.

[55] Safety of laser products, 2014. URL `https://www.vdeverlag.de/iec-normen/220821/iec-60825-1-2014.html`. Accessed: 30-08-2021.

[56] Anup K. Sharma, Arnaud LaflaquiÈre, Gennadiy A. Agranov, Gershon Rosenblum, and Shingo Mandai. SPAD array with gated histogram construction, U.S. Patent 1 234 567, Nov. 1969.

[57] Desmond V. O'Connor. *Time-Correlated Single Photon Counting*. Elsevier Inc., 1st edition edition, 1984. ISBN 978-0-12-524140-3.

[58] Wolfgang Becker. *Advanced Time-Correlated Single Photon Counting Techniques*. Springer-Verlag Berlin Heidelberg, 2005. ISBN 978-3-540-28882-4.

[59] Ivan Michel Antolovic, Samuel Burri, Claudio Bruschini, Ron Hoebe, and Edoardo Charbon. Nonuniformity analysis of a 65-kpixel CMOS SPAD imager. *IEEE Transactions on Electron Devices*, 63(1):57–64, 2016. doi: 10.1109/TED.2015.2458295.

[60] G. Turin. An introduction to matched filters. *IRE Transactions on Information Theory*, 6(3):311–329, 1960. doi: 10.1109/TIT.1960.1057571.

[61] Sung-You Tsai, Yu-Cheng Chang, and Tzu-Hsien Sang. SPAD LiDARs: Modeling and algorithms. In *2018 14th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, pages 1–4, 2018. doi: 10.1109/ICSICT.2018.8565812.

[62] Hugh G. Gauch and Gene B. Chase. Fitting the Gaussian curve to ecological data. *Ecology*, 55(6):1377–1381, 1974. ISSN 00129658, 19399170.

[63] Dongeek Shin, Ahmed Kirmani, Vivek K Goyal, and Jeffrey H. Shapiro. Photon-efficient computational 3-D and reflectivity imaging with single-photon detectors. *IEEE Transactions on Computational Imaging*, 1(2): 112–125, 2015. doi: 10.1109/TCI.2015.2453093.

[64] Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. University of Rhode Island, 1993. ISBN 978-0133457117.

[65] Adithya K. Pediredla, Aswin C. Sankaranarayanan, Mauro Buttafava, Alberto Tosi, and Ashok Veeraraghavan. Signal processing based pile-up compensation for gated single-photon avalanche diodes, 2018.

[66] Joshua Rapp, Yanting Ma, Robin M. A. Dawson, and Vivek K Goyal. Dead time compensation for high-flux depth imaging. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7805–7809, 2019. doi: 10.1109/ICASSP.2019.8683805.

[67] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34:189–206, 2013. doi: 10.1007/s10514-012-9321-0.

[68] Tim Golla and Reinhard Klein. Real-time point cloud compression. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5087–5092, 2015. doi: 10.1109/IROS.2015.7354093.

[69] Xuelian Cheng, Yiran Zhong, Yuchao Dai, Pan Ji, and Hongdong Li. Noise-aware unsupervised deep lidar-stereo fusion. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6332–6341, 2019. doi: 10.1109/CVPR.2019.00650.

[70] Yoann Altmann, Ximing Ren, Aongus McCarthy, Gerald S. Buller, and Steve McLaughlin. Lidar waveform-based analysis of depth images constructed using sparse single-photon data. *IEEE Transactions on Image Processing*, 25(5):1935–1946, 2016. doi: 10.1109/TIP.2016.2526784.

[71] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. doi: 10.1109/TIP.2007.901238.

[72] Ahmed Kirmani, Dheera Venkatraman, Dongeek Shin, Andrea Colaço, Franco N. C. Wong, Jeffrey H. Shapiro, and Vivek K Goyal. First-photon imaging. *Science*, 343(6166):58–61, 2014. ISSN 0036-8075. doi: 10.1126/science.1246775. URL `https://science.sciencemag.org/content/343/6166/58`.

[73] Xian-Feng Han, Xin-Yu Yan, and Shi-Jie Sun. Novel methods for noisy 3D point cloud based object recognition. *Multimedia Tools and Applications*, 80:26121–26143, July 2021. doi: 10.1007/s11042-021-10794-3. URL `https://doi.org/10.1088/2040-8986/ac486d`.

[74] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant CNNs, 2017.

[75] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image, 2018.

[76] David Ferstl, Christian Reinbacher, Rene Ranftl, Matthias Ruether, and Horst Bischof. Image guided depth upsampling using anisotropic total generalized variation. In *2013 IEEE International Conference on Computer Vision*, pages 993–1000, 2013. doi: 10.1109/ICCV.2013.127.

[77] Yevhen Kuznietsov, Jörg Stückler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction, 2017.

[78] Suliman Gargoum and Karim El-Basyouny. Automated extraction of road features using LiDAR data: A review of LiDAR applications in transportation. In *2017 4th International Conference on Transportation Information and Safety (ICTIS)*, pages 563–574, 2017. doi: 10.1109/ICTIS.2017.8047822.

[79] Juan Borrego-Carazo, David Castells-Rufas, Ernesto Biempica, and Jordi Carrabina. Resource-constrained machine learning for ADAS: A systematic

review. *IEEE Access*, 8:40573–40598, 2020. doi: 10.1109/ACCESS.2020.2976513.

[80] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

[81] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3D lidar using fully convolutional network, 2016.

[82] Bo Li. 3D fully convolutional network for vehicle detection in point cloud, 2017.

[83] Hongbo Gao, Bo Cheng, Jianqiang Wang, Keqiang Li, Jianhui Zhao, and Deyi Li. Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment. *IEEE Transactions on Industrial Informatics*, 14(9):4224–4231, 2018. doi: 10.1109/TII.2018.2822828.

[84] Piotr DollÀr, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, 2014. doi: 10.1109/TPAMI.2014.2300479.

[85] Xiangmo Zhao, Pengpeng Sun, Zhigang Xu, Haigen Min, and Hongkai Yu. Fusion of 3D LIDAR and camera data for object detection in autonomous vehicle applications. *IEEE Sensors Journal*, 20(9):4901–4913, 2020. doi: 10.1109/JSEN.2020.2966034.

[86] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3D object detection network for autonomous driving, 2017.

[87] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation, 2017.

[88] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space, 2017.

[89] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum PointNets for 3D object detection from RGB-D data, 2018.

[90] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition, 2013.

[91] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks?, 2014.

[92] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun. MultiNet: Real-time joint semantic reasoning for autonomous driving, 2018.

[93] Mengye Ren, Andrei Pokrovsky, Bin Yang, and Raquel Urtasun. SBNet: Sparse blocks network for fast inference, 2018.

[94] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 21–37, Cham, 2016. Springer International Publishing.

[95] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[96] Maschinelles lernen, 2018. URL `https://www.bigdata-ai.fraunhofer.de/content/dam/bigdata/de/documents/Publikationen/Fraunhofer_Studie_ML_201809.pdf`. Accessed: 07-09-2021.

[97] Gongbo Chen., Pierre Gembaczka., Christian Wiede., and Rainer Kokozinski. Feature extraction and neural network-based analysis on time-correlated LiDAR histograms. In *Proceedings of the 9th International Conference on Photonics, Optics and Laser Technology - PHOTOPTICS,*, pages 17–22. INSTICC, SciTePress, 2021. ISBN 978-989-758-492-3. doi: 10.5220/0010185600170022.

[98] Gongbo Chen, Felix Landmeyer, Christian Wiede, and Rainer Kokozinski. Feature extraction and neural network-based multi-peak analysis on

time-correlated LiDAR histograms. *Journal of Optics*, 24(3):034008, feb 2022. doi: 10.1088/2040-8986/ac486d. URL `https://doi.org/10.1088/2040-8986/ac486d`.

[99] Philip Bevington and D.Keith Robinson. *Data Reduction and Error Analysis for the Physical Sciences*. McGraw-Hill Education Ltd, revised edition edition, 2002.

[100] Daniel J. Schroeder. *Astronomical optics*. Academic Press, San Diego, second edition edition, 2000. ISBN 978-0-12-629810-9. doi: 10.1016/B978-012629810-9/50002-5. URL `https://www.sciencedirect.com/science/article/pii/B9780126298109500025`.

[101] Flash LiDAR system Owl. URL `https://www.ims.fraunhofer.de/de/Geschaeftsfelder/CMOS-Image-Sensors/Technologien/LiDAR.html`. Accessed: 07-12-2021.

DOI: 10.17185/duepublico/76556
URN: urn:nbn:de:hbz:465-20220923-120009-6