University of Genoa

Department of Marine, Electrical, Electronics and
Telecommunication Engineering and Naval Architecture

Ph.D. in Marine Science and Technology
Curriculum in Marine Engineering and Naval Architecture
XXXVIII cycle

Studies on Perceptive Systems for Maritime Autonomous Surface Ships

Development and Experimental Testing of
Multi-Sensor Situational Awareness Algorithms

Candidate:
Filippo Ponzini

Supervisors:
Prof. Michele Martelli, Prof. Raphael Zaccone

After an opinion from:
Prof. Yogang Singh & Prof. María Poo Argüelles

In front of a jury composed of:
Prof. Yogang Singh, Prof. Nikola Lopac & Prof. Silvia Donnarumma

Date:
March 25, 2026

A Franco, Maurizio e Pino

Contents

List of Figures

List of Tables

Preface

This doctoral dissertation is submitted in partial fulfilment of the requirements for the PhD degree in Marine Science and Technology at the University of Genoa.

This work was conducted within the framework of national and institutional research projects and was funded by the National Recovery and Resilience Plan (PNRR) through the MOST – National Centre for Sustainable Mobility project.

The work presented in this thesis is original and has been carried out by the author. Parts of this research have been published in peer-reviewed journals and conference proceedings, and these publications are properly cited throughout the manuscript.

The author would like to thank the supervisors for their guidance and support throughout the doctoral program.

Abstract

The development of Maritime Autonomous Surface Ships (MASS) represents a pivotal advancement in maritime safety and operational efficiency, particularly by mitigating accidents stemming from human error and reduced visibility. While the automotive and aerospace industries have made significant strides in autonomous perception, maritime situational awareness still relies primarily on traditional, non-cooperative sensors such as radar and the Automatic Identification System (AIS), which face substantial limitations in cluttered coastal and high-traffic settings. This dissertation tackles these gaps by adapting and validating advanced sensing technologies—such as LiDAR, thermal imaging, and RGB cameras—to the distinctive challenges of the marine domain. The core aim is to design, implement, and experimentally validate a multi-modal perception and communication framework that delivers high-level situational awareness for autonomous vessels. The study examines sensor fusion techniques, mitigates the shortage of maritime datasets, investigates AI-based target detection and classification, and assesses the transferability of perception algorithms from other domains to maritime applications. A structured experimental approach was adopted, advancing from simulation-based evaluations to controlled field trials and demanding sea tests. A modular sensing architecture was engineered, integrating LiDAR, thermal, and RGB sensors with high-precision INS/GNSS state estimation. To facilitate algorithm validation, two custom datasets were created: a LiDAR dataset comprising 11,680 scans (including 920 manually labelled samples) and a multi-modal "human-in-water" dataset comprising 6,000 samples, augmented with physical and virtual data to replicate environmental perturbations. Quantitative assessments on these datasets confirm the efficacy of the proposed pipelines. System-level validations on the SWAMP autonomous platform in Lake Nemi and the Gulf of Naples affirmed the seamless integration of perception and state estimation, supporting effective autonomous collision avoidance and robust multi-node communication. Multi-modal fusion was crucial in overcoming sensor-specific drawbacks, including LiDAR sparsity and thermal noise. This research contributes validated algorithms, novel datasets, and actionable guidelines for cross-domain adaptation of perception technologies to maritime contexts, laying a solid groundwork for future advancements in autonomous surface vessels.

1. Introduction

This chapter provides an overview of the research context and motivation for this thesis, highlighting the importance of enhanced situational awareness in maritime environments. It identifies key gaps in current technologies and presents the research questions driving this work. Furthermore, the chapter summarises the thesis's original contributions and concludes with an outline of its overall structure.

1.1 Motivation

The increasing complexity of ocean-based economic activities, together with the growing dependence on maritime infrastructure for energy supply and food security, calls for substantial improvements in situational awareness to ensure the safety and security of navigation and the protection of the marine environment.

Over the past decades, the ocean has evolved into a central pillar of the global economy. Since the 1960s, driven by the advent of containerisation and the expansion of offshore oil and gas activities, maritime traffic and seabed resource exploitation have grown exponentially. Today, maritime trade accounts for approximately 80% of global trade by volume and about 70% by value [1, 2]. In addition, the evolution of ships has led to progressively larger vessel dimensions without a corresponding adaptation of port infrastructure, which, in many cases, remains outdated.

This evolution has led to a sharp increase in surface maritime traffic across all sectors—recreational, commercial, and military—thereby heightening the exposure of maritime infrastructure, which is increasingly vulnerable to hybrid threats.

In this context, greater onboard autonomy—particularly enhanced vessels' autonomous perception capabilities—is a key factor in mitigating safety- and security-related risks at multiple levels. The following sections will therefore examine how human error in perception, assessment, and decision-making remains a major contributor to maritime accidents, alongside adverse environmental conditions such as poor visibility. Furthermore, the role of advanced autonomous perception in monitoring and protecting critical maritime infrastructure will be discussed. Finally, it will be shown that enhanced perception capabilities constitute a fundamental enabling technology for autonomous navigation and the development of Autonomous Surface Vehicles (ASVs) and Maritime Autonomous Surface Ships (MASS), thereby reducing or eliminating human involvement in high-risk maritime operations.

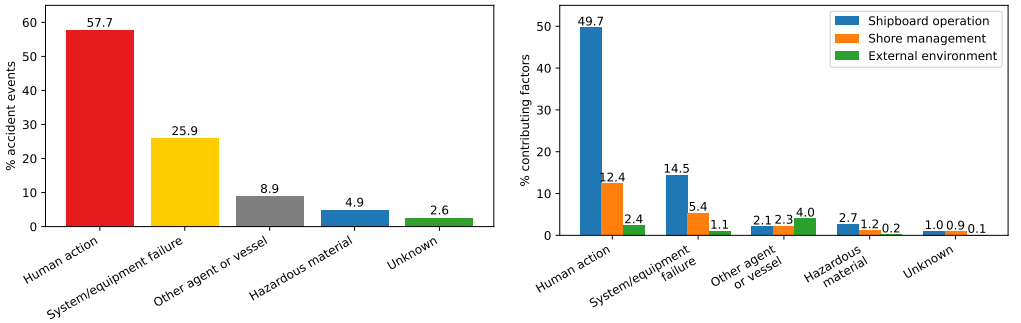
1.1.1 The role of human action and automation

Maritime accidents continue to pose significant risks to human life, marine ecosystems, and global trade. Collisions, groundings, and onboard incidents not only endanger crews and passengers but also may cause severe environmental damage and substantial economic losses. As maritime traffic increases and operational environments become more complex, ensuring navigational safety

remains a critical challenge for both regulatory authorities and technology developers.

The analysis of maritime accidents underscores the critical role of human factors in maritime safety.

The critical role of the human element in maritime safety is underscored by the EMSA Annual Overview of Marine Casualties and Incidents 2025 [3], which reports an annual average of more than 2,660 accidents between 2014 and 2023. The main descriptors are summarised in Figure 1.1; Figure 1.1a shows the main accident event types, while Figure 1.1b shows the main contributing factors broken down by event. Analysis of these occurrences reveals that human error is the primary catalyst for maritime disasters, accounting for 57.7% of incident types and 64.5% of all contributing factors. When accounting for both direct actions and broader behavioural influences, the "human element" is implicated in 78.8% of all contributing factors.



(a) Event type.

(b) Contributing factors.

Figure 1.1: EMSA Annual Overview of Marine Casualties and Incidents 2025 indicators

The last BIMCO/ICS Seafarer Workforce Report [4] warns of a serious potential officer shortage. Given the growing demand for STCW-certified officers, the Report predicts that an additional 89,510 officers will be needed by 2026 to operate the world's merchant fleet. These demands are expected to intensify seafarers' workloads, further aggravating the challenges already highlighted.

Further insight into this phenomenon is provided by DNV's Maritime Safety Trends 2014–2024 report [5], which documents a 22% increase in maritime accidents between 2022 and 2024. This rise is partly attributable to the ageing global fleet, with a substantial proportion of vessels over 25 years old and consequently exhibiting very low levels of autonomy and support for personnel.

Consequently, the automation of onboard operations, the implementation of regulation-compliant collision avoidance frameworks [6], and the integration of advanced decision-support systems are essential for mitigating human error, reducing fatigue and miscommunication, and ultimately improving maritime safety.

1.1.2 Low visibility conditions and their impact

Each year, the US Coast Guard (USCG) publishes a report detailing the dynamics and causes of recreational boating accidents. The 2023 report [7] reveals several key trends. Despite a higher overall number of recreational boating accidents during the summer months (due to the higher recreational traffic), the percentage of fatal accidents (reported in Figure 1.2 is higher in autumn and winter, with adverse weather conditions appearing to be a crucial factor (e.g., 29% fatality rate in December, 22% in October, compared to a 14% average).

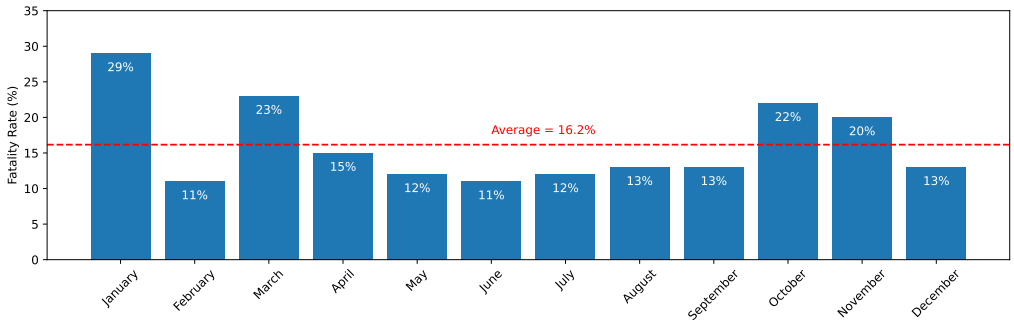


Figure 1.2: Fatality percentage per month according to the US Coast Guard report.

Similarly, while the highest volume of accidents occurs during daylight hours with heavier traffic, the fatality rate is considerably higher at night when visibility is poor (19-23% between 10 PM and 2:30 AM versus a 14% daily average) as reported in Figure 1.3. Directly attributable environmental conditions were the primary cause of 558 of 3844 accidents, resulting in 100 deaths and 257 injuries; an additional 44 accidents were related to restricted vision. While the majority of accidents occur in good visibility due to increased boating activity, accidents in low visibility, though less frequent, have a substantially higher injury and fatality rate (82% versus 68%). Finally, drowning remains the leading cause of death in recreational boating accidents, accounting for 377 of the 564 fatalities in 2023.

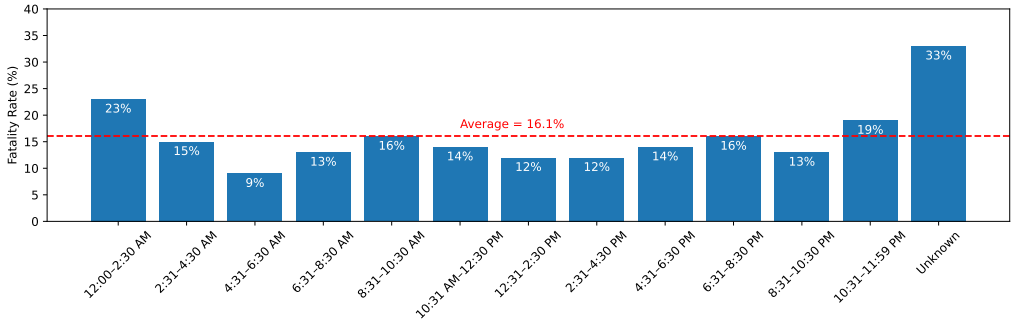


Figure 1.3: Fatality-rate by time according to the US Coast Guard report.

Information on the effects of poor visibility in maritime accidents can be effectively extracted from the scientific literature. [8] provides a systematic analysis of the causes of maritime accidents in Chinese coastal waters using machine learning approaches; the study concludes that adverse weather conditions, such as fog, rain, and rough seas, often lead to catastrophic accidents. Additionally, the authors note that accident probability is higher at night than during the day, and that fog is a critical weather condition for maritime accidents.
In [9] the authors analyze machine learning models for accident risk prediction by integrating weather data obtained by the Norwegian Maritime Authorities (NMA) from 1982 to 2021. The study revealed that the leading weather variables for accident prediction are visibility, wind, sea-level pressure, and moon phase.
In [10] the authors provides a comprehensive review of existing maritime accidents dataset, confirming that adverse weather condition are strongly related to fatality, in particular, wind and fog intensity seems to be proportional to the deaths over injuries ratio.
In [11] the authors confirms the link between adverse weather conditions and low visibility with the severity of the accident and fatality ratio.
Finally, events that support the thesis of poor visibility as a key aspect in the ship incidents scenario can be found in past disasters. In 1955, the Shiun Maru disaster resulted in 168 fatalities due to thick fog. In 1956, SS Andrea Doria and MS Stockholm collided during the night (around 11:00 p.m) in thick fog, resulting in 46 deaths. The Moby Prince disaster, which cost the lives of 140 people off the coast of Livorno (Italy) in 1991, occurred at night (around 10.00 p.m.) and during the legal process [12], the presence of advection fog emerged as a contributory cause of the dramatic collision. In addition, the ship and the surrounding sea were engulfed in flames, creating a thick blanket of smoke and making rescue even more difficult. In the recent North Sea collision between container ship MV Solong and oil tanker MV Stena Immaculate (occurred in March 2025), heavy fog was registered by onboard camera; moreover ships burned, producing a thick blanket of smoke in the following hours [13].

1.1.3 Critical infrastructures monitoring

Emerging sectors such as offshore wind farming and subsea cable laying have further increased the strategic importance of ocean activities. Offshore wind farms are attracting substantial global investment as part of the energy transition effort [14], while more than 95% of the world's data traffic is transmitted through undersea cables, with over 900,000 miles of cables currently lying on the ocean floor and further expansion expected in the coming years [15].

This maritime economic boom, driven in part by the scarcity of land-based resources [16], has made certain maritime infrastructures critical for the functioning of entire nations, leading to the widespread recognition of the concept of Critical Maritime Infrastructure (CMI) [17]. CMIs are exposed to various hazards, both accidental — such as those caused by fishing activities [18] — and intentional, including recent deliberate attacks that have disrupted energy supplies and communications [19, 20]. Such attacks are often carried out by so-called dark ships or shadow vessels, which conceal their activities by disabling or spoofing their Automatic Identification

System (AIS) signals or by sailing under flags of convenience [21, 22].

Protecting CMIs presents unique challenges compared to protecting land-based critical infrastructure, due to operational difficulties at sea and the vast areas involved. Tailored technological solutions are therefore essential, and recent advances in computing and communication technologies enable the effective deployment of Maritime Autonomous Surface Ships (MASS) and Unmanned Underwater Vehicles (UUV) for surveillance and monitoring [23].

1.1.4 Situational awareness as enabling technology

Enhanced situational awareness constitutes a key enabling technology for Maritime Autonomous Surface Ships (MASS), allowing them to operate effectively in high-risk environments while reducing human exposure and increasing mission efficiency. Such capabilities are particularly valuable in applications including seabed mapping, critical infrastructure surveillance, environmental monitoring, and Search and Rescue (SAR) operations.

Although Unmanned Aerial Vehicles (UAVs) have been successfully deployed across multiple domains, including SAR, their effectiveness in maritime rescue scenarios remains limited. Constraints such as restricted operational range, high sensitivity to adverse weather conditions, and the inability to perform direct physical intervention reduce their suitability in the maritime context.

Conversely, Unmanned Surface Vehicles (USVs) and Autonomous Surface Vehicles (ASVs) offer distinct advantages: they are inherently more resilient to atmospheric disturbances, can accommodate larger payloads of essential relief supplies (e.g., food, water, and medical equipment), and are capable of transporting and deploying flotation devices or inflatable life rafts, which can be towed directly by the vehicle.

The increasing adoption of ASVs and USVs for complex and demanding missions is reflected in the growing body of research literature, encompassing applications ranging from the surveillance of sensitive and high-risk maritime areas [24], to environmental data collection [25], spill coverage [26] SAR operations [27, 28], USV swarm deployment [29] and support for military and warfare-related activities [30].

In SAR missions, unmanned surface vessels are highly valued for their ability to rapidly cover extensive search areas, operate safely in hazardous conditions, and coordinate within cooperative swarms to enhance search effectiveness. Furthermore, they can be deployed directly into disaster zones by aerial platforms, enabling rapid response in time-critical scenarios.

1.2 Research gap & questions

The preceding sections have highlighted how improving situational awareness in the marine field is critically important for i) limiting the human factor and compensating for poor visibility in disaster and accident scenarios; ii) monitoring and policing energy and logistics infrastructure in a challenging environment on which a huge slice of the global economy is based; iii) advancing research and development of Maritime Autonomous Surface Ships, which are increasingly in demand for a wide viariety of missions.

Although significant progress has been made in developing technologies for maritime navigation

and safety, situational awareness at sea is still predominantly based on traditional sensing systems, namely radar (Radio Detection And Ranging) and the Automatic Identification System (AIS). These technologies, while well-established and widely deployed, present inherent limitations. Radar systems can be affected by clutter, multi-path reflections, and reduced performance in adverse weather conditions or in the proximity of complex coastal or offshore infrastructures. Similarly, AIS depends on cooperative reporting from vessels and is therefore vulnerable to intentional deactivation, spoofing, or manipulation. As a consequence, relying solely on these technologies is insufficient for ensuring comprehensive and reliable situational awareness, particularly in scenarios involving uncooperative or hostile actors, such as dark ships or shadow vessels.

In parallel, perception systems that integrate advanced sensing modalities, such as LiDAR, visible-light cameras, and thermal imaging, have demonstrated significant capabilities in other domains, most notably in the automotive and aerospace sectors. These systems enable accurate object detection, classification, and tracking under a wide range of operating conditions. However, their adaptation to the maritime environment is far from straightforward.

The marine domain is characterised by unique and challenging features, including dynamic backgrounds created by waves and sea clutter, rapidly changing illumination and weather conditions, and objects with variable signatures and geometries. Furthermore, the highly dynamic nature of both the targets and the viewpoint, subject to ship movements, necessitates multimodal integration of perceptual and proprioceptive sensors, such as the Inertial Navigation (INS) System and Global Navigation Satellite System (GNSS) receivers. This complexity limits the direct transfer of perception solutions developed for land-based applications. Existing research on multimodal integration and machine learning for maritime perception remains limited, with many solutions still at the conceptual or prototype stage.

A further major obstacle to the development of advanced maritime perception systems is the lack of comprehensive and high-quality marine datasets. Unlike in the automotive sector, where large annotated datasets have enabled rapid progress in perception and decision-making algorithms, the maritime field lacks equivalent resources. The available datasets are often narrow in scope, lack sufficient annotations, or fail to capture the full variability of real-world marine environments. This hinders the development, training, and validation of machine learning models, thereby impeding progress in applying artificial intelligence to maritime situational awareness.

In light of these gaps, this research aims to address the following key questions:

RQ1. How can advanced sensing technologies, including LiDAR, optical, and infrared sensors, be effectively integrated and adapted to provide robust perception capabilities in complex and dynamic maritime environments?

RQ2. What strategies can be developed to overcome the current limitations in marine data availability, and how can new datasets be collected and annotated to support the development of reliable perception and decision-support systems onboard ships and yachts?

RQ3. Which sensor fusion approaches and artificial intelligence algorithms can be designed or tailored to enhance object detection, classification, and tracking at sea, particularly in

low-visibility conditions and high-risk scenarios?

RQ4. Which perception algorithms and methodologies, originally developed for other domains such as automotive or aerospace, can be effectively transferred to the maritime domain? What modifications, adaptations, or additional measures are necessary to ensure their robustness and reliability in marine environments, and which approaches are unsuitable without substantial re-engineering?

1.3 Contribution

This thesis makes several original contributions to the field of maritime situational awareness and Maritime Autonomous Surface Ship (MASS) perception, with a particular focus on operations in low-visibility and complex marine environments. The contributions of this research can be summarised as follows:

C1. Design and implementation of a multi-modal maritime perception system: A novel sensor architecture has been developed and validated, integrating LiDAR, thermal and visible light optical sensors, and GNSS/INS for enhanced situational awareness. The design addresses the specific challenges posed by marine environments, including dynamic backgrounds, variable lighting, and atmospheric conditions such as fog and sea spray.

C2. Adaptation of perception algorithms to the maritime domain: The work provides investigations of perception algorithms, originally developed for automotive or aerial applications, and assesses their suitability for maritime use. It identifies which algorithms can be effectively transferred and under what conditions, and proposes the necessary adaptations or enhancements to handle marine-specific issues (e.g., wave clutter, sensor occlusions, dynamic horizon).

C3. Development of hybrid learning multi-modal methods for marine obstacle detection and classification: A set of new algorithms combining supervised and unsupervised learning approaches has been designed to support robust detection, tracking, and classification of marine targets (e.g., vessels, humans in water) under challenging conditions. The methods are tailored to address the lack of large-scale, annotated datasets in the maritime domain.

C4. Creation of novel marine datasets: The thesis contributes to addressing the lack of publicly available marine datasets by generating multi-sensor datasets acquired during controlled and real-environment experiments. These datasets include annotations for object detection, tracking, and classification, and span a range of environmental conditions (e.g., daylight, night, fog, sea clutter).

C5. Experimental validation in real-world scenarios: The proposed methods and systems have been tested extensively both in controlled environments (e.g., indoor facilities, towing tanks) and in real marine scenarios (e.g., lakes, coastal waters, and open sea), providing empirical evidence of their effectiveness and limitations in practical applications.

C6. Guidelines for the adaptation of perception algorithms to the marine context: The thesis formulates a set of recommendations regarding which types of algorithms are suitable for maritime perception tasks, which are not directly transferable, and what modifications or precautions are required. These guidelines aim to support future developments in the field and inform the design of maritime autonomous systems.

Overall, the research bridges the gap between existing perception technologies and their practical deployment in the maritime domain, with direct implications for the development of Maritime Autonomous Surface Ships (MASS).

To clarify the alignment between the research questions and the contributions of this thesis, Table 1.1 summarises how each contribution addresses the identified research questions, together with the resulting publication. Alternatively, the mapping can be described as follows:

- RQ1 (Integration of advanced sensing technologies) is primarily addressed by Contributions 1 and 5, which present the design and validation of a multi-modal maritime perception system.
- RQ2 (Marine data limitations and dataset development) is addressed by Contributions 3 and 4, through the development of hybrid learning approaches tailored for limited data scenarios and the creation of new annotated maritime datasets.
- RQ3 (Sensor fusion and AI algorithms for maritime perception) is addressed by Contributions 1, 3, and 5, which collectively develop, implement, and experimentally validate multi-modal sensor fusion and AI-based detection methods.
- RQ4 (Transferability of perception algorithms from other domains) is addressed by Contributions 2 and 6, which investigate the applicability of existing algorithms and provide structured guidelines for their adaptation to maritime environments.

Research Question	Addressing Contributions	Adressing Publication
RQ1	C1, C3	[31], [32], [33], [34], [24], [35], [36], [37]
RQ2	C3, C4	[34], [38]
RQ3	C1, C3, C5	[34], [36]
RQ4	C2, C6	[32],[39], [37]

Table 1.1: Alignment between the research questions and the contributions of the thesis.

1.4 Outline of the Thesis

The thesis is organised into eight chapters and follows a conventional structure comprising an introduction, materials, methodology, experimental campaigns, results, discussion, and conclusions. All developed technologies and frameworks are presented comprehensively, from their theoretical foundations to their practical implementation and experimental validation.

The research adopts an incremental approach from both theoretical and experimental perspectives. Each technology was initially developed as a standalone component and rigorously validated using datasets collected in relevant environments. Subsequently, as complexity increased, multisensor fusion methods were designed, implemented, and evaluated. Finally, through a series of ad hoc experimental campaigns of progressively increasing complexity, the developed pipelines were integrated into a multidisciplinary framework and deployed on the SWAMP autonomous testing platform.

The structure of the chapters is summarised below:

- **Chapter 1: Introduction** Introduces the study's motivation, the need for enhanced maritime situational awareness, the identified research gaps, the research questions, the thesis's contributions, and the document's outline.
- **Chapter 2: Background** Reviews the state of the art in maritime perception systems, situational awareness technologies, and related sensing and data processing methodologies.
- **Chapter 3: Material and Methods** Describes the experimental facilities, the test platforms (including vehicles and sensors), and the overall setup used for data acquisition and validation.
- **Chapter 4: Methodology** Details the sensing architecture and the perception algorithms developed in this study. It covers detection, classification, tracking, and sensor-fusion strategies applied to LiDAR, optical, and GNSS-INS data.
- **Chapter 5: Experimental Campaign** Presents the experimental activities carried out during the research. It is structured into two main sections: the first describes the field data acquisition activities undertaken to generate the foundational datasets for development and testing. In contrast, the second presents the experimental campaigns aimed at integrating the technology into an autonomous platform with progressively increasing complexity.
- **Chapter 6: Results** Reports the outcomes of the research. It is divided into two main sections, mirroring the structure of the previous chapter. The first section validates the developed state estimation and detection pipelines, including performance metrics for detection, classification, tracking, and sensor fusion under various conditions; the second presents results from testing in a relevant environment within an integrated, multidisciplinary framework.
- **Chapter 7: Conclusions and Future Directions** Synthesises the results, discusses the study's limitations, and explores the implications and potential applications of the proposed methods and suggests directions for future investigation in the field of maritime situational awareness.

2. Background

This chapter establishes the theoretical and methodological background for the research contributions presented in the following chapters. It contextualises the technological and algorithmic components of Maritime Autonomous Surface Ships (MASS) and highlights the key challenges that motivate the approaches developed in this thesis.
It is structured into five main sections: Section 2.1 presents an overview of processing methods for various sensors and sensor fusion techniques employed in the maritime domain; Section 2.2 focuses on sensor fusion procedures for perception sensors; Section 2.3 provides an overview of multi-target tracking algorithms; Section 2.4 presents sensor fusion methods for state estimation purposes; Finally, Section 2.5 focuses on the perception systems currently integrated into operational Maritime Autonomous Surface Ships (MASS) or tested in real-world prototype deployments.
A typical example of MASS architecture is shown in Figure 2.1. Three main layers are highlighted: sensing, situational awareness, and MASS management. Sensors can be divided into two types: perceptual (exteroceptive) sensors that acquire data about the surrounding environment, obstacles, and targets, and navigation sensors (proprioceptive) that provide information useful for estimating the MASS's state. Onboard sensors acquire raw data from the surrounding environment and provide it to the situational awareness layer.
The main task of situational awareness is to process raw sensor data to extract information useful for dynamic MASS management. Finally, the MASS management layer incorporates all aspects of guidance and control. The first two layers are the central focus of this doctoral thesis.

2.1 Sensors and processing

This section examines the primary perceptual sensors, including radar, LiDAR, and optical sensors, which provide information on obstacles and targets in the surrounding environment, as well as the key proprioceptive sensors, such as GNSS and INS, which support the estimation of the vehicle's state, including position, velocity, and attitude.

2.1.1 Radar

Radar (Radio Detection and Ranging) systems are a cornerstone of maritime navigation, thanks to their robust performance in adverse weather and low-visibility conditions such as fog, rain, and darkness. By measuring the time delay between transmitted and received echoes, radar provides range information, while antenna orientation determines bearing; Doppler processing enables estimation of relative velocity.

In maritime platforms, conventional X-band and S-band radars are the dominant solution. Their widespread adoption is driven by regulatory compliance, long detection range, hardware maturity, and moderate computational requirements. Conversely, high-resolution imaging radar and millimetre-wave radar systems have attracted research interest due to their improved angular

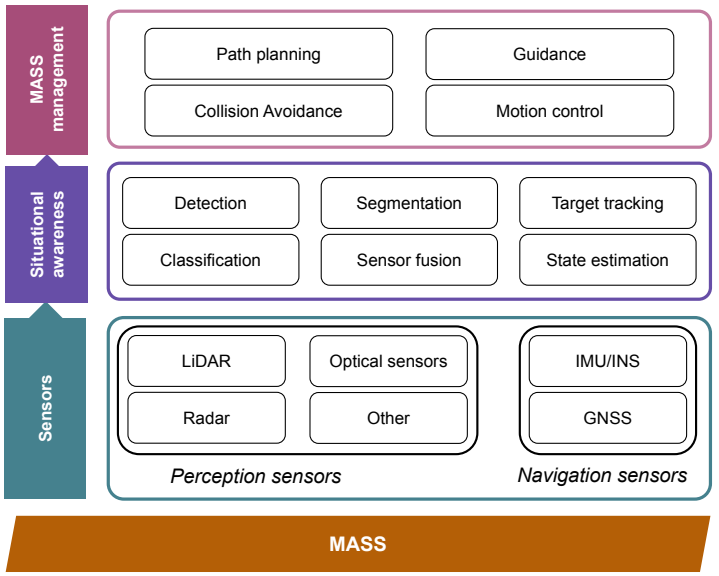


Figure 2.1: Typical MASS architecture with focus on sensors and situational awareness modules.

resolution and capability to detect small objects like floating debris or persons in water [40]. However, their shorter operational range, higher cost, and increased processing demands currently limit their deployment at large scale in operational MASS systems.

A major challenge in maritime radar operation is sea clutter, caused by reflections from dynamic ocean waves. In practice, clutter mitigation relies primarily on Doppler filtering and Moving Target Indication (MTI), which exploit velocity discrimination, and on adaptive Constant False Alarm Rate (CFAR) detectors that adjust detection thresholds based on local clutter statistics. These techniques represent the industrial standard due to their robustness and computational efficiency.

More advanced approaches, such as polarimetric processing, enhance target–clutter discrimination by exploiting differences in scattering properties across polarisation channels. Although promising, their adoption remains limited by hardware complexity and calibration requirements. Similarly, machine-learning-based signal-processing methods are being explored in the literature. However, their integration into safety-critical maritime systems is still constrained by reliability and real-time processing requirements.

At the data level, radar returns are typically represented as 2D or 3D point maps (range, azimuth, and optionally Doppler or elevation). Clustering algorithms are applied to aggregate detections into candidate obstacles without requiring prior knowledge of the number of targets. Density-based methods such as DBSCAN [41], HDBSCAN [42], and OPTICS [43] are commonly used because they can handle irregularly shaped clusters and noise. However, their computational cost scales with the number of detections, which may become critical in high-resolution radars.

In practical maritime autonomy systems, clustering is often followed by lightweight centroid extraction, and each resulting object hypothesis is tracked using a multi-target tracking framework

(see Section 2.3). This pipeline reflects a trade-off between detection robustness and computational feasibility for real-time embedded deployment. Figure 2.2 illustrates representative marine radar systems.

(a) X-band radar. **(b)** S-band radar.

Figure 2.2: Example of typical radar sensors.

Sources: www.maritimeinformed.com, www.furuno.com

2.1.2 LiDAR

Light Detection and Ranging (LiDAR) sensors generate high-resolution three-dimensional representations of the surrounding environment by emitting and receiving laser pulses. Thanks to their geometric accuracy and independence from ambient illumination, LiDAR systems have become a key sensor for perception in autonomous navigation. Compared to radar, LiDAR operates at shorter ranges but provides significantly richer spatial detail, enabling precise reconstruction of object contours and estimation of obstacle shapes.

In maritime environments, LiDAR is primarily used for short- to medium-range perception, complementing radar in scenarios requiring fine geometric resolution (e.g., docking and obstacle avoidance in confined waters). However, its performance can degrade under heavy rain, spray, airborne particles, and strong water-surface reflections. Moreover, the limited detection range and sensitivity to environmental conditions constrain its use as a standalone long-range sensor in MASS platforms.

A typical LiDAR output consists of a four-dimensional point cloud (three spatial coordinates plus reflectivity). Due to its high density and geometric richness, several processing strategies have been proposed. For real-time collision avoidance, occupancy grid mapping remains a widely adopted solution in practical systems, owing to its computational efficiency and robustness. However, this approach does not fully exploit the geometric information available in the point cloud.

More advanced pipelines rely on point clustering and segmentation techniques to extract individual obstacles. Density-based clustering methods, such as DBSCAN and HDBSCAN, are frequently employed. In contrast, Euclidean distance-based clustering is often preferred in real-time implementations due to its lower computational overhead and ease of tuning. Surface-based segmentation methods (e.g., Region Growing) and model-fitting techniques such as RANSAC further exploit local geometric features to identify planar or structured elements. These approaches provide accurate obstacle delineation but may become computationally demanding for high-density point clouds.

In recent years, supervised deep learning models such as PointNet, PointNet++, and PointPillars have demonstrated strong performance in joint detection and classification tasks. These architectures are particularly effective in structured terrestrial environments with abundant annotated datasets. However, their applicability in maritime contexts is limited by the scarcity of labelled marine datasets, the sparsity of long-range returns over water, and the computational requirements of real-time embedded deployment. Consequently, traditional geometric and clustering-based pipelines still dominate operational maritime systems, especially in safety-critical applications where interpretability and robustness are prioritised over semantic richness.

Once obstacles are extracted, their representative positions (e.g., centroids or bounding box centres) are passed to a multi-target tracking framework (see Section 2.3), enabling temporal consistency and motion estimation.

(a) Multi-beam rotating LiDAR.

(b) Full-body rotating LiDAR.

(c) Solid State LiDAR.

(d) Single beam long range LiDAR.

Figure 2.3: Example of typical LiDAR sensors.

Sources: www.businesswire.com, www.aeroexpo.online, www.robotics247.com

2.1.3 Visible light optical sensors

Visible-light optical sensors, including monochrome and RGB cameras, are among the most widely used sensing technologies in maritime and robotic applications.

The popularity of cameras is largely due to several key factors: they are cost-effective, commercially mature, widely available, and benefit from decades of development in computer vision and image processing. Moreover, they can be easily integrated with modern perception pipelines, leveraging well-established deep learning techniques. In particular, convolutional neural networks (CNNs) [44, 45] such as YOLO [46], R-CNN [47], Fast R-CNN [48] and Faster-RCNN [49] have demonstrated strong performance in object detection, classification, and tracking tasks using optical data, enabling robust perception in a variety of robotic and autonomous systems, including maritime platforms.

The typical output of an optical sensor is a two-dimensional image, often represented as a tensor of size $h \times w \times n$, where h and w denote the image resolution and n the number of channels (with n=1 for monochromatic images and n=3 for RGB images). These sensors are particularly effective for tasks involving visual object classification and identification, as they provide rich texture and colour information that facilitates the discrimination of targets against complex backgrounds.

However, visible-light cameras have significant limitations in maritime applications. Their performance is highly dependent on ambient illumination and visibility conditions. As a result, their effectiveness decreases markedly at night, in low-light conditions, or under adverse weather conditions such as fog, rain, or sea spray. This susceptibility severely limits their reliability in many scenarios in which Maritime Autonomous Surface Ships (MASS) are expected to operate.

While optical sensors can be used in stereo configurations to enable depth estimation, their ability to accurately estimate distances diminishes rapidly with increasing range, due to baseline limitations and insufficient disparity at longer ranges. Furthermore, without external dimensional references or calibration targets, monocular optical systems provide no direct means to infer the absolute distance to objects in the scene.

Despite these constraints, optical sensors remain a core component of multimodal maritime perception systems because they complement other sensor modalities, such as radar and LiDAR, particularly in daylight and clear-visibility conditions. While the availability of annotated optical datasets for maritime scenarios is somewhat greater than for LiDAR, the field still lacks comprehensive, large-scale, and diverse datasets that fully capture the variability of real-world marine environments. This aspect will be more clearly addressed in the dedicated section (Section 5.1).

2.1.4 Thermal imaging

Thermal imaging sensors detect infrared radiation emitted by objects, producing images that encode relative temperature differences. This modality is particularly useful for detecting humans or active vessels, especially under low-light or poor-visibility conditions. However, thermal images pose challenges, including lower spatial resolution, reduced texture information, and noise, which can limit performance in complex maritime scenes.

Traditional methods for segmentation and detection in thermal imagery include edge-based operators such as the Sobel [50], and Scharr [51] filters, Laplacian filters, and the Canny algorithm [52], as well as thresholding techniques such as Otsu's method [53]. These approaches are effective when objects exhibit strong thermal contrast with the background, but they may fail in cluttered environments or when temperature differences are subtle.

More recent works adopt deep learning approaches to enhance detection performance. For example, CNN-based methods have been applied on UAV-mounted thermal cameras for human-in-water detection [54], while unsupervised domain adaptation combined with segmentation has been validated on both real and simulated data [55]. Transformer-based models, such as the Segment Anything Model (SAM) [56], offer zero-shot, promptable segmentation, though fine-tuning is typically required due to the domain gap between terrestrial training datasets and maritime environments. A major practical limitation is the scarcity of annotated maritime thermal datasets, often requiring bespoke data collection under regulatory or dual-use constraints.

To overcome individual sensor limitations, multimodal fusion combining thermal imagery with optical cameras or LiDAR is increasingly employed. Such fusion enhances human detection and navigation in low-visibility scenarios by leveraging each modality's strengths to complement the others, thereby improving robustness and reliability in real-world maritime SAR and MOB

operations.

Before moving on to navigation (or proprioceptive) sensors, Table 2.1 summarises the main advantages and disadvantages of the perceptual modalities presented so far.

Table 2.1: Summary of perception sensor modalities typically used in MASS.

Sensor	Typical Range	Advantages	Limitations	Cost
LiDAR	10–200 m	High geometric accuracy; dense 3D point cloud; independent of ambient illumination; omnidirectional coverage	Limited range; sensitive to spray, and water reflections	High
RGB Camera	1–100 m	Cost-effective; rich texture and colour information	Highly dependent on lighting and visibility; affected by glare; No night-time data; Sensitive to fog, rain	Low
Thermal Camera	1–100 m	Low-light and poor visibility; Detects temperature differences; Night-time data availability	Low spatial resolution; limited texture; affected by thermal reflections; single channel; Sensitive to fog and rain	Medium
Radar	50–5000 m (depending on system)	Long-range detection; robust to weather; omnidirectional coverage	Limited spatial resolution; Poor shape reconstruction; Multipath reflections; Sea clutter; Blind zones	High

2.1.5 Global Navigation Satellite System

Global Navigation Satellite Systems (GNSS), including GPS, Galileo, and GLONASS, are the standard solution for global positioning, offering high accuracy and worldwide coverage. In the literature, GNSS is widely recognised as a fundamental component for autonomous navigation and for geo-referencing sensor data. However, GNSS signal reliability can be compromised by attenuation or interference, motivating the integration of complementary navigation technologies.

Processing GNSS data in a marine environment without integrating inertial sensors relies solely on satellite signals to determine the vessel's position and velocity. Raw GNSS measurements, such as pseudorange and carrier phase, are collected and preprocessed to correct errors due to satellite orbit inaccuracies, clock biases, and atmospheric delays, including ionospheric and tropospheric effects. Positioning techniques such as standalone GNSS, Real-Time Kinematic (RTK) [57, 58], and Precise Point Positioning (PPP) [59] are used to improve accuracy, which is crucial for safe navigation and precise vessel tracking.

In the literature, it is often proposed to continuously update the position estimate using filtering methods, such as Kalman filters (see Table 2.3), which help smooth the data, reduce noise, and estimate other kinematic variables, such as speeds. In open marine settings, this approach provides reliable and accurate positioning. However, challenges arise near coastal areas, ports, or regions with signal obstructions from structures or natural features, where GNSS signals may be partially blocked or reflected. Without inertial sensors to assist during signal interruptions, maintaining continuous, stable position estimates becomes more difficult, potentially affecting navigation and

operational decisions.

2.1.6 Inertial Navigation Systems

Inertial Navigation Systems (INS) use accelerometers, gyroscopes, and sometimes magnetometers to estimate a vehicle's position, velocity, and orientation independently of external signals. The literature highlights the extensive use of INS in environments where GNSS signals are weak or unavailable, such as underwater. The primary limitation of INS is the drift error that accumulates over time, necessitating the fusion of INS data with other sensor inputs to improve overall navigation accuracy.

It is well established in the literature that processing the individual sensors embedded in an Inertial Navigation System (INS) separately offers limited possibilities and generally yields unsatisfactory results. Linear accelerations measured by accelerometers or angular velocities from gyroscopes, when treated in isolation, are inherently subject to significant drift over time. This occurs even when advanced filtering techniques, such as extended Kalman filters or particle filters, are applied, as these sensors lack absolute references and their integration accumulates errors in both position and orientation estimates.

Among the individual inertial sensors, the magnetometer is sometimes proposed as a stand-alone solution because it can provide absolute measurements of the Earth's magnetic field and, in certain cases, enable estimation of heading angles in three-degree-of-freedom (3-DOF) configurations. However, magnetometers are highly susceptible to local magnetic disturbances, which limit their reliability when used alone.

For these reasons, it is far more effective to process INS data within a unified system, in which measurements from accelerometers, gyroscopes, and magnetometers (when available) are combined using sensor-fusion algorithms. In this context, simple yet effective solutions, such as the complementary filter, are often proposed, alongside more complex approaches, such as Kalman filters and related methods, that model system dynamics and measurement processes to reduce drift and improve accuracy (see Table 2.3).

In any case, while INS systems are highly useful for providing high-frequency estimates of position, velocity, and orientation, they are almost always integrated with GNSS to obtain an absolute reference for position and to correct integration errors that inevitably accumulate over time. These topics, including the principles and implementation of sensor fusion techniques for combining INS and GNSS data, will be discussed in detail in the following sections dedicated to multi-sensor integration.

2.2 Sensor-fusion for enhanced perception

The need for sensor fusion in maritime autonomy does not arise merely from a desire to improve performance, but from the intrinsic physical limitations of individual sensing modalities. Optical cameras provide rich semantic information and enable object classification, yet their reliability decreases significantly in low-light, foggy, or high-glare conditions. Thermal cameras mitigate some of these limitations by detecting heat signatures independently of visible illumination;

however, they generally lack precise geometric depth estimation and detailed structural information.

LiDAR sensors offer accurate three-dimensional geometric reconstruction and precise short-range localisation, which is particularly advantageous in manoeuvring and docking scenarios. Nevertheless, their effectiveness degrades at longer ranges and in adverse weather conditions, where point clouds become sparse and noisy. Radar systems, in contrast, provide long-range detection and robustness to weather and lighting variations, but offer limited spatial resolution and weak intrinsic classification capability.

These complementary strengths and weaknesses directly motivate the adoption of multimodal fusion strategies in Maritime Autonomous Surface Ships (MASS). For example, radar robustness can compensate for LiDAR sparsity at range; LiDAR geometric precision can refine radar detections; thermal imaging can preserve detection capability at night, where RGB cameras fail; and vision-based classification can provide semantic interpretation to geometrically detected objects.

The different fusion levels discussed below represent alternative architectural strategies to exploit such complementarities, each balancing computational cost, information richness, and system robustness. The specific architectural choices adopted in this thesis build on this complementary perspective, aiming to ensure reliable perception across diverse maritime operational scenarios.

2.2.1 Low-level fusion: raw data fusion

This approach integrates raw data streams from multiple sensors, such as LiDAR point clouds, Radar returns, RGB and thermal images. Low-level fusion has the potential to yield highly detailed environmental models but requires substantial computational resources and precise temporal synchronisation. In maritime applications, low-level fusion is typically used to build unified 3D maps that combine LiDAR and Radar data for mapping or obstacle detection. Example algorithms include occupancy grid mapping, multimodal SLAM (e.g., LiDAR + radar), and voxel-based point cloud fusion.

Low-level fusion integrates raw data streams from multiple sensors, such as LiDAR point clouds, Radar returns, RGB images, and thermal images. This approach can construct highly detailed and accurate environmental models by leveraging the richness of rawsensor data. However, it comes at the cost of significant computational requirements and the need for precise temporal synchronisation across the different sensors.

In maritime applications, low-level fusion is typically employed to construct unified 3D maps that integrate LiDAR and radar data for tasks such as mapping, obstacle detection, and scene reconstruction; however, it can also be used for low-level image fusion. By fusing raw measurements, systems can achieve fine-grained spatial resolution and better handle partial occlusions or sensor blind spots. Example algorithms and techniques used for low-level fusion include:

- Occupancy grid mapping [60]: a probabilistic method that represents the environment as a grid of cells, each indicating the likelihood of being occupied.
- Multi-modal SLAM (e.g., double LiDAR or Radar, LiDAR/Radar): simultaneous localisation and mapping methods that integrate raw measurements from different sensor modalities [61]

to build consistent maps, possibly considering different points of view.

- Voxel-based point cloud fusion [62]: techniques that aggregate 3D point clouds into voxel structures, enabling efficient storage, processing, and integration of dense spatial data from several sensor modalities and/or points of view.

- Pixel-level data combination [63]: data from multiple sensors, such as visible (RGB) and thermal cameras, can be combined to create a single multisensor representation. Pixel-wise fusion techniques assign different weights to the visible and thermal channels, thereby balancing their contributions as a function of environmental conditions. This approach preserves both visual and thermal details, enhancing perception in low-light, foggy, or nighttime scenarios.

Low-level fusion thus provides the foundation for detailed environment perception, at the expense of increased system complexity and computational load.

2.2.2 Mid-level fusion: feature-level fusion

Mid-level fusion focuses on combining features extracted from sensor data, rather than raw measurements (as in low-level fusion). The features may include geometric or semantic elements such as edges, landmarks, point clusters, keypoints, or thermal blobs. This level of fusion represents a trade-off between data richness and computational complexity: it preserves more structural and spatial information than high-level fusion (where data is already heavily abstracted into decisions), but it is generally more efficient than processing raw data from all sensors directly.

In MASS applications, mid-level fusion enables the combination of complementary feature sets across different modalities to enhance perception capabilities. For example, visual keypoints extracted from RGB or thermal cameras can be fused with spatial clusters or geometric structures identified by radar or LiDAR. This improves object detection, environmental understanding, and robustness in challenging scenarios, such as low visibility or sensor occlusion.

Typical methods applied at this level include:

- Iterative Closest Point (ICP) [64]: a well-known method for aligning point clouds by iteratively minimising distances between corresponding points from different scans or sensors.

- Normal Distributions Transform (NDT) [65]: a probabilistic registration method that models point distributions as normal distributions over a grid, allowing robust alignment especially in sparse or noisy conditions.

- Graph-based fusion of spatial features [66, 67]: techniques where spatial features (e.g., landmarks or keypoints) are represented as nodes in a graph, and their spatial or semantic relationships as edges. Fusion is achieved via graph optimisation, Graph SLAM, or Graph Neural Networks (e.g., GCNs) to produce globally consistent representations.

- Convolutional Neural Networks (CNNs) [45, 44]: deep learning architectures used for extracting and fusing multi-modal features, enabling the network to learn rich representations from combinations of sensor inputs such as images, point clouds, or heatmaps.

Mid-level fusion offers a powerful balance between detail and efficiency, allowing systems to combine the strengths of heterogeneous sensors to achieve more accurate, resilient, and context-aware perception in complex environments.

2.2.3 High-level fusion: decision-level fusion

High-level decision fusion is the process of combining the decisions or classifications generated independently by each sensor or subsystem into a unified, more reliable outcome. This fusion level is especially valuable when integrating sensors with disparate modalities (e.g., camera, radar, LiDAR) or with different update rates, as is often the case in multi-agent and multi-sensor systems (MASS).

In such contexts, high-level fusion improves robustness and confidence in target detection, classification, and tracking by aggregating individual decisions. For example, in MASS applications, high-level fusion might combine object classifications from vision-based neural networks with tracking information from radar or LiDAR to enhance target confirmation or reduce false positives. Common techniques for high-level decision fusion include:

- Majority or weighted voting schemes [68]. In these approaches, each sensor or classifier casts a "vote" for a decision, and the final decision is determined by majority rule or by weighting votes based on each sensor's reliability or confidence. This method is simple and effective, especially when sensors have comparable performance or well-characterised confidence levels.

- Dempster-Shafer theory for evidence combination [69, 70]. The Dempster-Shafer (DS) theory provides a mathematical framework for combining evidence from different sources, allowing the representation of uncertainty and ignorance. It is particularly suitable when sensor outputs provide partial or ambiguous evidence.

- Bayesian network-based decision fusion [71, 72]. Bayesian networks model probabilistic dependencies among variables and can be used to fuse decisions by representing the conditional dependencies between sensor outputs and the underlying state of the world. This approach is powerful for managing uncertainty and incorporating prior knowledge in the fusion process.

2.3 Multi Target Tracking

In perception systems for autonomous platforms, tracking is the process of estimating and continuously updating the dynamic state of one or more targets over time, based on a sequence of noisy, possibly incomplete sensor observations.

Maritime multi-target tracking presents specific challenges that distinguish it from terrestrial or aerial scenarios. Surface vessels typically exhibit slow and smooth dynamics over long time horizons, but may occasionally perform abrupt manoeuvres (e.g., collision avoidance). Moreover, target visibility can be intermittent due to sea clutter, wave occlusions, sensor range limitations, or multipath effects near large structures. Small craft may appear and disappear at long range, while larger vessels generate extended or fluctuating detections depending on sea state.

These characteristics require tracking algorithms that are robust to sparse measurements, moderate clutter, and temporary target loss, while maintaining stable identity management over long observation periods.

Unlike detection, which operates on individual sensor frames to identify the presence of objects, tracking exploits temporal coherence to infer target motion, maintain object identities, and provide robust state estimates even in the presence of noise, clutter, and intermittent observations.

A generic multi-target tracking pipeline can be decomposed into a set of interconnected functional blocks. First, sensor measurements or detections are received at each time step and optionally preprocessed. These detections must then be associated with existing tracks or used to initialise new ones, a step commonly referred to as data association. Data association is a critical component of the tracking process, as incorrect associations may lead to track fragmentation, identity switches, or false tracks. Several standard data association techniques are widely used:

- Nearest Neighbour (NN) [73]: Associates each detection with the nearest predicted track in measurement space, typically based on minimum distance or Mahalanobis distance.
- Gated Nearest Neighbour [74]: Similar to NN, but only considers detections within a validation region (gate) around the predicted track position. The gate is usually an ellipsoid based on track uncertainty.
- Probabilistic Data Association Filter (PDAF) [74]: Considers all validated measurements and computes a weighted average of their contributions, where the weights are the probabilities of each measurement being from the target.
- Joint Probabilistic Data Association Filter (JPDAF) [75]: Extends PDAF to handle multiple targets simultaneously. It computes the joint probabilities of measurements for each target, thereby preventing coalescence when targets are close together.
- Multiple Hypothesis Tracking (MHT) [76]: Maintains multiple hypotheses over time about how measurements might correspond to existing tracks. Rather than making immediate hard decisions, it defers resolution of ambiguity as more data becomes available.

In practical maritime systems, the choice of data association method depends strongly on target density and environmental complexity. In open-sea navigation, where targets are typically sparse and well separated, simple approaches such as Gated Nearest Neighbour are often sufficient and computationally efficient. These methods are widely adopted in real-time embedded maritime platforms due to their simplicity and predictability.

In contrast, port environments or congested waterways may involve multiple closely spaced vessels, small fast-moving craft, and strong clutter. In such cases, probabilistic approaches such as JPDAF provide improved robustness against association ambiguity. Multiple Hypothesis Tracking (MHT), while highly effective in complex scenarios, is generally reserved for high-end or shore-based systems due to its computational cost.

Once detections have been associated with tracks, the state estimation step updates each target's dynamic state—typically position, velocity, and, in some cases, acceleration—by combining the new measurements with prior predictions from a motion model. This fusion process explicitly accounts for uncertainty in both the model and the measurements, enabling robust tracking under realistic operating conditions. Common algorithms include:

- Kalman Filter (KF) [77]: Used for linear dynamic and measurement models with Gaussian noise. Provides the optimal minimum-variance estimate under these conditions.
- Extended Kalman Filter (EKF) [75]: Generalises the Kalman filter for non-linear systems by linearising the dynamics and measurement equations around the current estimate.
- Unscented Kalman Filter (UKF) [78]: Employs a deterministic sampling technique (unscented transform) to better capture the mean and covariance of non-linear transformations, without requiring explicit linearization.
- Particle Filter (PF) [79]: Is a recursive Bayesian estimator that represents the posterior probability density of the target's state using a set of weighted random samples called particles. Each particle represents a possible state of the target, and its weight reflects the likelihood of that state given the measurements.

Beyond the choice of estimation algorithm, the motion model is crucial for maritime tracking. For most vessels navigating at moderate speeds, constant-velocity (CV), constant-turn-rate (CT), or constant-turn-rate-and-velocity (CTRV) models provide a realistic approximation of dynamics over short horizons. These models exploit the relatively smooth kinematics of marine vessels, enabling stable long-term tracking.

However, small high-speed craft or unmanned surface vehicles may exhibit sharper manoeuvres, requiring more adaptive models or higher process noise tuning. Particle filters become advantageous only when target motion or measurement characteristics deviate significantly from Gaussian assumptions, although their computational burden often limits their adoption in embedded maritime systems.

Consequently, KF and EKF-based trackers with constant-velocity or constant-turn models remain the dominant solution in operational MASS platforms, balancing robustness, interpretability, and real-time feasibility.

In addition to data association and state estimation, practical tracking systems often include auxiliary mechanisms such as track initiation, track confirmation, track maintenance, and track termination, which collectively ensure stability and continuity of the estimated trajectories which are essential for smooth tracking during rapid manoeuvres or brief GNSS outages; GNSS, on

Table 2.2: Comparison of common data association techniques in multi-target tracking.

Method	Key idea	Strengths	Weaknesses
NN	Assign to closest track	Fast, simple	Error-prone in clutter or crossing targets
Gated NN	NN with validation region	Reduces false associations	Still fragile in dense clutter
PDAF	Weighted association of validated measurements	Robust to clutter	Struggles with closely spaced targets
JPDAF	Joint association for all tracks	Handles multiple close targets	Computationally expensive
MHT	Maintains multiple association hypotheses	Excellent for complex scenarios	High computational cost

the other hand, delivers globally referenced and Table 2.3, which compares widely used state estimation algorithms.

Table 2.3: Comparison of common state estimation algorithms for multi-target tracking.

Filter	Model type	Strengths	Weaknesses
KF	Linear	Optimal for linear-Gaussian systems	Fails with non-linear dynamics
EKF	Mildly non-linear	Simple extension of LKF	Degrades with strong non-linearities
UKF	Non-linear	Higher accuracy than EKF	More computationally expensive than EKF
PF	Highly non-linear	Flexible, handles complex distributions	Computationally intensive

2.4 Sensor fusion for enhanced state estimation

In marine navigation, the integration of Inertial Navigation Systems (INS) and Global Navigation Satellite Systems (GNSS) is central to modern positioning and attitude determination for Maritime Autonomous Surface Ships (MASS). INS provides high-rate, continuous estimates of position, velocity, and attitude (PVA), ensuring smooth short-term motion tracking, while GNSS supplies globally referenced position fixes with long-term stability.

Both systems have limitations: INS solutions drift over time due to bias and scale-factor errors, whereas GNSS performance may degrade near offshore structures, in narrow channels, or under interference and multipath effects. Their fusion, therefore, is necessary rather than an enhancement in autonomous maritime systems.

2.4.1 Integration Architectures and Operational Relevance

Among the various architectures proposed in the literature, three main configurations are commonly identified: loosely coupled, tightly coupled, and deeply coupled integration. Their adoption in maritime systems depends strongly on operational requirements, cost constraints, and resilience needs.

2.4.1.1 Loosely Coupled Integration

Loosely coupled integration is the simplest and most widely adopted architecture in commercial and civil maritime applications. In this configuration, the INS and GNSS operate independently, and the GNSS-derived PVA solution is periodically fused with the INS solution through an Extended Kalman Filter (EKF), typically implemented as an error-state filter.

This approach offers moderate computational complexity, straightforward implementation, and compatibility with standard GNSS receivers. For these reasons, it is the de facto standard on many operational MASS platforms, particularly in open-sea navigation, where satellite visibility is generally reliable. However, its main limitation is its dependence on a complete GNSS position fix; when fewer than four satellites are visible, the correction mechanism becomes unavailable.

2.4.1.2 Tightly Coupled Integration

Tightly coupled integration directly fuses raw GNSS measurements (e.g., pseudoranges and Doppler) with inertial data within a unified estimation filter, typically an EKF. By exploiting individual satellite measurements rather than relying on a standalone GNSS solution, this architecture maintains partial observability even when satellite visibility is limited.

This configuration significantly improves robustness in cluttered maritime environments, such as ports or near large metallic structures, where multipath and partial signal obstruction are common. Consequently, tightly coupled systems are increasingly adopted in advanced autonomous vessels and high-reliability navigation systems. The trade-off lies in the increased integration complexity and the tighter synchronisation requirements between sensors.

2.4.1.3 Deeply Coupled Integration

Deeply coupled integrationis the most advanced architecture, in which inertial data directly assist the GNSS receiver's signal-tracking loops. By providing high-rate motion predictions, the INS helps the GNSS receiver maintain lock on weak or partially obstructed signals.

This architecture offers superior resilience to interference, jamming, and spoofing, and is particularly relevant for safety-critical or defence-oriented applications. However, it requires specialised hardware, custom firmware, and significant development effort. For this reason, deeply coupled systems remain limited to high-end or mission-critical maritime platforms and are not yet common in commercial MASS deployments.

2.4.2 Fusion algorithm and practical considerations

From an algorithmic perspective, the Extended Kalman Filter (EKF) remains the dominant solution for INS/GNSS integration in maritime systems, balancing computational efficiency with accurate modelling of nonlinear vessel dynamics and sensor error states.

While alternatives such as the Unscented Kalman Filter (UKF) or Particle Filters can better capture strong nonlinearities or non-Gaussian noise, their increased computational load and implementation complexity limit their widespread use in embedded maritime platforms. Consequently, EKF-based error-state filtering remains the industrial standard.

In practical MASS deployments, navigation performance is highly dependent on sensor quality. Low-cost MEMS INS units experience noticeable drift, necessitating frequent GNSS updates, whereas navigation-grade inertial systems provide longer autonomy during GNSS outages at a significantly higher cost. INS–GNSS fusion must also address additional challenges: magnetic disturbances in harbours can perturb heading estimates, MEMS sensors typically drift by 0.5–2°/min in heading and by several meters per minute in position, and GNSS outages due to multipath or signal obstruction may last from seconds to minutes. These factors underscore the need for robust fusion filters and, when required, supplementary aiding sensors such as Doppler velocity logs or radar.

2.5 State of the art

This section reviews recent detection, classification, and tracking pipelines proposed in the literature, as well as those specifically developed for Search and Rescue (SAR) missions, which have unique operational requirements. Finally, an overview of Autonomous Surface Vehicles (ASVs) and Unmanned Surface Vehicles (USVs) currently employed in real-world operations is provided.

2.5.1 Detection, classification, and tracking for navigation purposes

This section provides a comprehensive review of existing literature on detection, classification, and tracking methodologies specifically tailored for marine navigation, exploring various sensing modalities and their integration. Several studies have focused on leveraging RADAR for maritime perception, with [80] introducing a RADAR-only detection framework that constructs an occupancy grid without incorporating semantic information. The feasibility of utilising automotive-grade RADAR for marine target localisation is explored in [81], while [82] investigates the deployment of a lightweight RADAR system to enhance situational awareness by detecting and tracking marine targets and monitoring sea state. For unmanned surface vehicles (USVs), [83] proposes a dynamic obstacle detection system that integrates marine RADAR with Electronic Navigational Charts (ENCs). This method generates two grid maps—one from RADAR and one from ENCs—to classify obstacles as static or dynamic by accumulating temporal data.

LiDAR has also demonstrated significant potential for detection and tracking in maritime environments. [84] proposes a real-time obstacle detection algorithm for USVs that combines LiDAR point-cloud preprocessing with a forward-looking grid-mapping method and improved

versions of RANSAC and DBSCAN. This approach effectively filters out clutter from wake waves and improves clustering performance for both short- and long-range obstacles. In [85], a LiDAR-based ship detection and tracking framework is presented, featuring a modular deep-learning detection network and a Kalman Filter-based tracking approach. Tested on a newly collected real-world maritime LiDAR dataset, this system demonstrates the feasibility of LiDAR-only perception for autonomous surface vehicles (ASVs) in complex environments. Furthermore, [86] proposes a LiDAR-only 3D object detection method for small USVs, utilising the PointPillars network and a custom dataset. This approach employs a late fusion strategy to reduce computational load, making it suitable for energy-constrained maritime platforms.

RGB cameras offer a cost-effective and flexible solution for obstacle detection. [87] introduces WODIS, a lightweight deep semantic segmentation network designed for maritime obstacle detection using RGB images. Its architecture addresses challenges such as small-object detection and sea-sky ambiguity by leveraging an encoder based on depthwise separable convolutions, derived from the Xception network [88], which enables efficient feature extraction at reduced computational cost. Similarly, [89] proposes an obstacle detection approach for USVs based solely on standard RGB cameras, employing the YOLOv4 deep neural network. Trained on publicly available maritime datasets, this method achieves a favourable balance between detection accuracy and real-time processing on low-cost embedded units, presenting a fast and economical solution for autonomous maritime navigation. In [90], the authors propose WaSR, a deep encoder-decoder network tailored for marine semantic segmentation that integrates inertial measurement unit (IMU) data into its decoder. This fusion significantly reduces false positives due to water reflections and wakes, thereby improving obstacle-detection accuracy in maritime environments.

To overcome the inherent limitations of individual sensors and enhance perception robustness, numerous multi-sensor fusion approaches have been developed. [91] proposes a fuzzy logic-based multi-sensor fusion algorithm combining AIS and RADAR data for moving ship detection. This method integrates a two-stage fuzzy association with Kalman filtering to enhance detection reliability under complex maritime conditions, demonstrating effective performance in simulation scenarios with multiple manoeuvring targets. A

A multimodal system integrating a monocular camera, LiDAR, and marine radar is presented in [92], yielding a marine obstacle detection framework in which target classification relies on YOLO-based camera processing and 3D obstacle detection is performed by LiDAR-RADAR clustering. In [93], the authors propose a sensor-fusion system for an autonomous ferry that integrates radar and LiDAR data via spatial clustering. At the same time, RGB camera data is processed using YOLO. All detections are subsequently fused within a probabilistic multi-target tracker. An attempt to achieve target classification from LiDAR data is further described in [94], where the results are fused with optical-sensor-derived classifications.

2.5.2 Perceptive systems for Search And Rescue missions

This section reviews detection and classification approaches specifically developed for human detection in maritime Search and Rescue (SAR) and Man Overboard (MOB) scenarios. While many works address general object detection at sea, the emphasis here is on techniques tailored to

Table 2.4: Summary of marine navigation perception methods.

Reference	Sensors	Detection	Classification	Tracking
[80]	RADAR	Yes	No	No
[81]	RADAR	Yes	No	Yes
[82]	RADAR	Yes	No	Yes
[83]	RADAR ENC	Yes	Yes	No
[84]	LiDAR	Yes	No	No
[85]	LiDAR	Yes	Yes	Yes
[86]	LiDAR	Yes	Yes	No
[87]	RGB Camera	Yes	Yes	No
[89]	RGB Camera	Yes	Yes	No
[90]	RGB camera IMU	Yes	Yes	No
[91]	RADAR AIS	Yes	Yes	Yes
[92]	RADAR LiDAR RGB Camera	Yes	Yes	Yes
[93]	RADAR LiDAR RGB Camera	Yes	Yes	Yes
[94]	LiDAR RGB camera	Yes	Yes	No

SAR applications.

In [40], a feasibility study is conducted on detecting floating life jackets using an FMCW MIMO RADAR. In [28, 95], sonar-equipped ASV platforms are proposed for SAR operations. An RGB-camera-equipped ASV platform for SAR that relies solely on image-based detection using a YOLO convolutional neural network (CNN) is presented in [27].

A more comprehensive ASV platform for SAR missions is proposed in [96], which includes a drowning detection module. Detection in this case is performed by a deep learning neural network trained on synthetic images generated in Unity, though the specific architecture is not disclosed.

Several works [97, 98, 99, 100, 101] deploy UAVs equipped with visible-light optical sensors for detecting humans in water. These approaches typically rely on image-only CNNs and have been tested primarily under calm sea and good-visibility conditions. Similarly, [102] employs a YOLO-based CNN for RGB camera-based human detection in water.

Thermal imaging has also been explored. In [55], a method for thermal-camera-based detection is proposed, utilising unsupervised domain adaptation followed by image segmentation, and validated on both real and simulated data. [54] presents a thermal-camera-equipped UAV that uses Faster R-CNN for human-in-water detection. Table 2.5 summarises the key findings from the reviewed literature.

The analysis reveals a strong reliance on optical sensors for classification, owing to their superior ability to detect targets. RGB cameras, in particular, are commonly used in combination with powerful deep learning detection algorithms. However, several studies acknowledge that these systems become less effective, or even nonfunctional, in nighttime or poor-visibility conditions.

While UAVs are frequently proposed for SAR tasks, they have considerable limitations under the scenarios outlined in Section 1. Thermal imaging systems alleviate some of the shortcomings of visible-light sensors, particularly under low-visibility conditions, but they often fall short in providing accurate localisation.

More robust 3D sensors, such as sonar and radar, which are less affected by environmental disturbances, are also used; however, they typically lack classification capabilities. Despite their widespread use in USV/ASV navigation, LiDAR sensors are not yet commonly employed for human detection in SAR operations. While point clouds can become sparse at long ranges and partially submerged humans may yield weaker returns, LiDAR provides high-resolution geometric information that is valuable for obstacle detection and target shape estimation. When integrated with complementary sensors such as mmWave radar, optical, and thermal cameras, LiDAR can significantly enhance multimodal perception and robustness in SAR scenarios.

2.5.3 Dataset availability

This section examines prevailing maritime datasets, specifically focusing on visual and LiDAR modalities in line with the scope of this thesis. While terrestrial autonomous navigation has reached a high level of maturity thanks to large-scale LiDAR-annotated benchmarks such as KITTI [103] and ApolloScape [104], the maritime domain continues to face a significant data deficit. This scarcity of public, real-world LiDAR data remains a primary obstacle to advancing deep-learning-based perception and multi-sensor fusion for marine applications. Current maritime

Table 2.5: Summary of related work on human-in-water detection systems.

Reference	Platform	Sensor	Classification
[40]	N.D.	Radar	No
[28, 95]	ASV/USV	Sonar	No
[27, 96]	ASV/USV	RGB camera	Yes
[97, 98, 99, 100, 101]	UAV	RGB camera	Yes
[102]	N.D.	RGB camera	Yes
[55]	N.D.	IR camera	No
[54]	UAV	IR camera	Yes

datasets generally fall into two categories, each with specific limitations:

- **Vision-centric datasets:** The Singapore Maritime Dataset (SMD) [105] and the MODD and MODD 2 series [106, 107] provide valuable annotated imagery and GNSS/INS data. However, the absence of LiDAR information limits their utility for precise 3D spatial reasoning.
- **LiDAR-inclusive datasets:** Existing resources that incorporate LiDAR are often constrained by sensor resolution or environmental variety. The MIT Sea Grant dataset [108], while multimodal, employs a 16-channel LiDAR that produces low-density point clouds. Similarly, the work by [109] offers annotated ship detections but is equally limited by a 16-channel sensor. Conversely, the Pohang Canal Dataset [110] utilises high-resolution 32 and 64-channel sensors; however, it lacks object-level annotations and is restricted to inland/canal environments, leaving a clear gap in open-sea data.

To address this gap, a new marine LiDAR dataset [38] is introduced in Chapter 5 and used to validate the pipeline in Chapter 6.

A comparative summary of the analysed datasets is presented in Table 2.6, highlighting the critical need for high-resolution, annotated LiDAR data in diverse maritime scenarios.

Table 2.6: Marine dataset related works.

Name	Reference	LiDAR availability	LiDAR channels	LiDAR labelling	Other sensors	Location
Singapore Maritime Dataset	[105]	No	N.A.	No	RGB	Harbor Blue-water
MODD	[106]	No	N.A.	N.A	RGB camera	Harbor
MODD 2	[107]	No	N.A.	N.A	RGB camera INS/GNSS	Harbor
MIT Sea Grant Marine Perception Dataset	[108]	Yes	16	No	RGB camera IR camera radar	Inland Harbor
Pohang Canal Dataset	[110]	Yes	32-64	No	RGB camera IR camera radar INS/GNSS	Inland Harbor
Maritime LiDAR-based Ship Detection Datasets	[109]	Yes	16	Yes	N.A.	Inland Harbor
Proposed dataset		Yes	32	Yes	N.A.	Harbor Blue-water

3. Material and Methods

This chapter provides a comprehensive overview of the experimental infrastructure employed throughout the research. It describes the test sites and facilities, the primary equipment used, including autonomous vehicles and sensors, and the communication and data acquisition systems. The goal is to outline the practical setup supporting the development and validation of the proposed methodologies.

3.1 COMPASS Lab

Founded in 2019 at the University of Genoa, the COMPASS Laboratory (Control, Manoeuvrability & Propulsion for Autonomous Surface Ship) is a research facility dedicated to supporting experimental activities in autonomous surface navigation. Its primary objective is to enable the development, implementation, and validation of control strategies by integrating computational simulations with physical testing infrastructures.

The laboratory is built on a cyber-physical framework that can replicate a range of maritime scenarios under controlled conditions. Within this setup, simulation models, vehicle dynamics, communication systems, and embedded control units interact in real time, enabling rigorous testing of Guidance, Navigation, and Control (GNC) algorithms. The hybrid nature of the lab supports both standalone software simulations and hardware-in-the-loop (HIL) configurations, enhancing the accuracy, repeatability, and robustness of experimental results.

A central component of the facility is a 5 × 5 meter indoor square tank, which, despite its limited dimensions, provides a highly versatile environment for preliminary trials involving Autonomous and Unmanned Surface Vehicles (ASVs/USVs). Its square geometry facilitates the execution of basic manoeuvres and trajectory tracking in confined spaces, making it especially suitable for early-stage validation.

The tank can be equipped with a roof-mounted overhead camera, providing a top-down view of the water surface for visual tracking and motion analysis. Additionally, a fixed mounting point is available for a LiDAR sensor, enabling high-resolution spatial perception and environment reconstruction. These features, combined with the controlled aquatic environment, enable the synchronised acquisition of inertial, visual, and range-based data, thereby supporting a wide range of experimental scenarios that integrate multiple sensors.

3.2 Real environment test-bed

While controlled laboratory environments are crucial for early-stage testing and development, real-world testbeds are critical for assessing the situational awareness capabilities of autonomous surface vehicles (ASVs) under authentic environmental conditions. Real operational environments expose ASVs to a multitude of unpredictable factors, including dynamic obstacles, varying visibility,

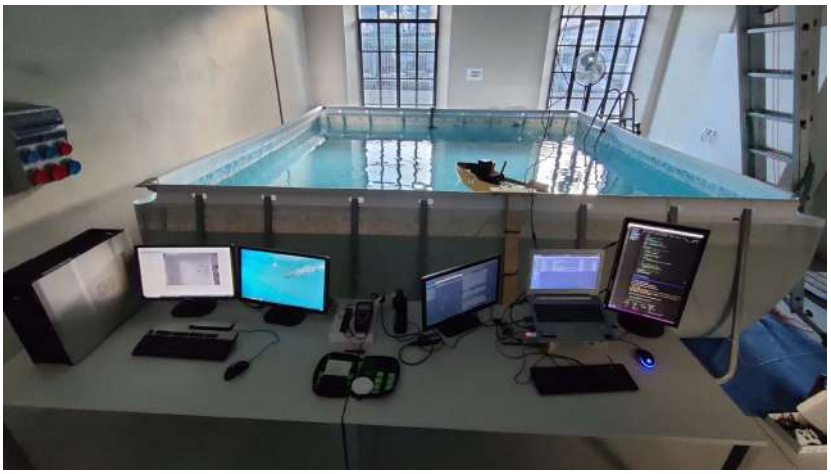


Figure 3.1: COMPASS LAB test tank image.

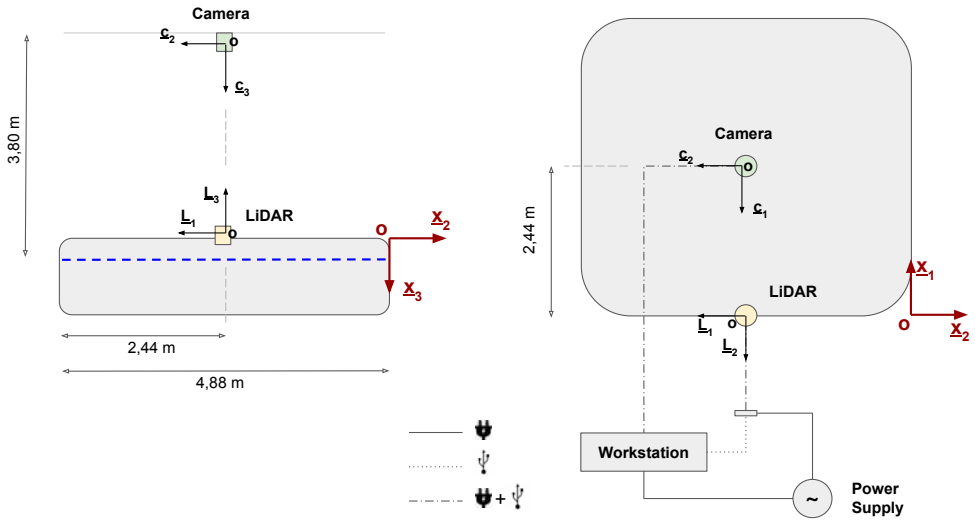


Figure 3.2: COMPASS LAB test tank general scheme.

Figure 3.3: Satellite view of Lake Lagaccio.

environmental disturbances, and sensor noise.

In this study, two lakes of increasing size were selected as test sites for progressively more challenging trials. Specifically, Lake Lagaccio (PC) is presented in Section 3.2.1, and Lake Nemi (RM) is detailed in Section 3.2.2. The choice was guided by several practical considerations: ease of access, the absence of formal authorisations from maritime authorities for autonomous vehicle navigation, safety concerns, and the availability of a reliable power supply. These factors ensured that the testing environments enabled systematic validation of autonomous systems across diverse environmental conditions without compromising logistical feasibility or operational control. Finally, the open sea test location in the Gulf of Naples is presented in Section 3.2.3

3.2.1 Lake Lagaccio

Lago Lagaccio is an artificial lake located at 44.92682° N, 9.57081° E, in the province of Piacenza (PC), within the Emilia-Romagna region of Italy. The lake has an approximately rectangular shape with maximum dimensions of about 100 meters by 65 meters, as shown in Figure 3.3. The site provides ample space for conducting experimental tests in a controlled yet realistic environment. Permission to conduct research at this location was granted by the property owners, who also ensured a reliable power supply.

3.2.2 Lake Nemi

The Lake of Nemi is a volcanic lake located in 41.712222° N, 12.702500° E, near the town of Genzano di Roma in the province of Roma (RM), within the Italian region of Lazio. The lake has a rounded shape with maximum dimensions of about 1.8 km by 1.3 km (see Figure 3.4). The lake hosts an important experimental facility managed by the National Research Council – Institute of Marine Engineering (CNR-INM). This facility is authorised to conduct autonomous and manned navigation trials within the lake, providing a unique natural environment for testing marine technologies.

Figure 3.4: Satellite view of Lake Nemi.

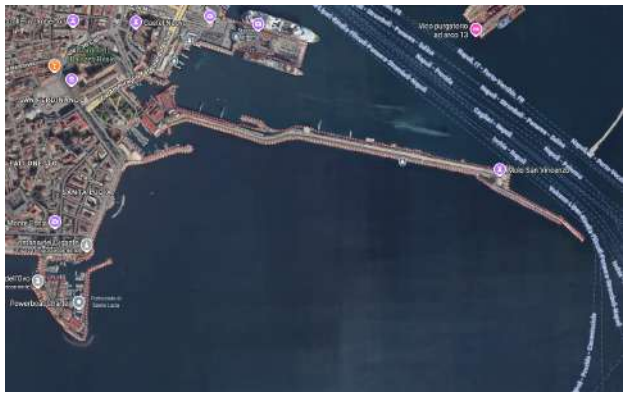


Figure 3.5: Satellite view of the experimental site in the Gulf of Naples.

3.2.3 Gulf of Napes

The Gulf of Naples was chosen as the experimental marine site in the Italian region of Campania. In particular, thanks to the project partners, the guarantee was secured at the Lega Navale and the Circolo Cannottieri of Naples, located at 40.834430° N, 14.253689° E (see Figure 3.5). The open sea test area is limited only by the range of the communication systems.

3.3 Vehicle

To validate the situational-awareness logics under realistic conditions, the SWAMP autonomous surface vehicle [111] was selected as the testbed. Developed by CNR-INM, SWAMP is designed for shallow and confined waters such as rivers, lakes, harbours, and coastal zones (Figure 3.6). Its compact, modular design allows single-operator transport and payload flexibility, making it ideal for testing navigation, perception, and autonomy algorithms in real-world conditions.

Table 3.1: SWAMP vehicle key specifications

Parameter	Value / Description
Hull type	Flat-bottom, double-ended Wigley
Total length	1.25 m
Beam	1.1 m (adjustable 0.7–1.25 m)
Hull height	0.4 m
Full system height	1.1 m
Unloaded weight	38 kg
Payload capacity	Up to 100 kg
Draft	0.1 m
Maximum cruise speed	1.6 m/s (deep water)
Shallow water operation	0.2 m depth, up to 1.0 m/s
Propulsion	Twin pump-jet modules [112]
Thruster control	360° azimuth, Maxon EC-4pole motors, Faulhaber encoder
Control interface	Wi-Fi (Huzzah ESP8266), RS-232 motor, SPI azimuth
Steering accuracy	0.1°; full rotation < 1 s

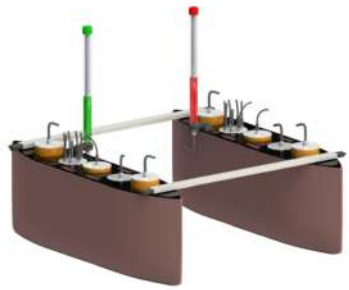


Figure 3.6: SWAMP vehicle image.

SWAMP's robust design and high manoeuvrability stem from several key engineering choices. Its flat-bottom, twin-hull layout provides excellent stability and reserve buoyancy, allowing safe operation under full payloads up to 100 kg. The pump-jet propulsion system minimises the risk of grounding or collisions with debris, while each module's 360° azimuth control enables precise trajectory tracking, even in confined or shallow waters.

The compact footprint and modular architecture allow rapid deployment by a single operator and ease of transport between test sites.

The system has been validated through static and dynamic tests, including bollard pull and towing tank trials at DITEN, University of Genoa (Figures 3.7 and 3.8).

Figure 3.8 shows two images of the SWAMP autonomous platform on the shore, ready to be launched in the water.

3.4 Sensors used

This section provides an overview of the sensors used for the research, together with their technical characteristics.

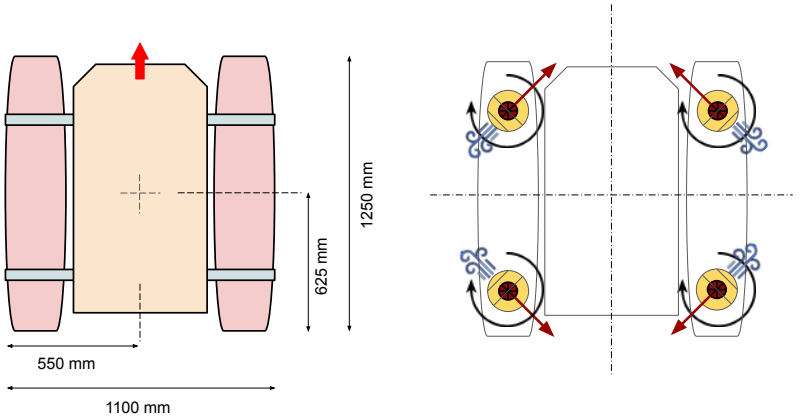


Figure 3.7: SWAMP vehicle general scheme. (left) vehicle's measures in top-view; (right) vehicle's thruster configuration.

Figure 3.8: SWAMP vehicle ready to be launched in the water.

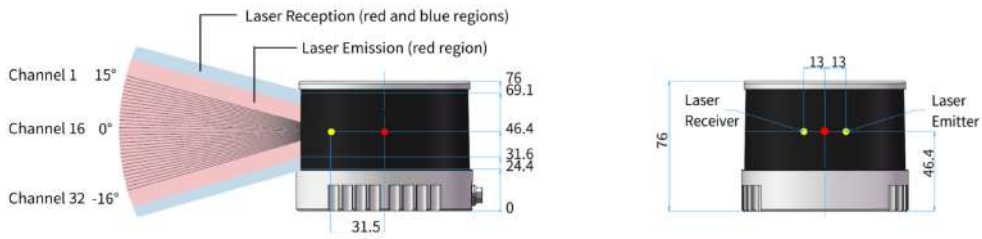


Figure 3.9: HESAI PandarXT-32 channel distribution and main dimension (mm).

3.4.1 LiDAR

The LiDAR sensors employed in this research are mechanical spinning LiDARs, composed of a laser emitter and receiver mounted on a rotating head. This architecture provides continuous 360° horizontal coverage of the surrounding environment and is widely adopted in autonomous systems for high-resolution 3D perception. Both sensors are manufactured by HESAI and differ in resolution, vertical field of view, and angular channel arrangement.

The first model is the PandarXT-32, a 32-channel sensor with equally spaced vertical channels spanning a 31° vertical field of view, yielding a constant vertical angular resolution of 1°. The horizontal resolution ranges from 0.09° to 0.36°, depending on the rotation frequency, which is configurable between 5 Hz and 20 Hz. The maximum measurement range is 120 m, with an accuracy of 1 cm. Its physical dimensions and the vertical arrangement of the channels are illustrated in Figure 3.9. Main specifications are reported in Table 3.2.

Table 3.2: HESAI Pandar XT-32 main specifications.

Specification	value
Channels	32
Vertical FOV	31°
Vertical resolution	1°
Horizontal FOV	360°
Horizontal resolution	0.09 - 0.18 - 0.36°
Operating Frequency	5 - 10 - 20 Hz
Range	120 m
Accuracy	0.01 m

The second model is the Pandar64, equipped with 64 vertical channels non-uniformly distributed over a 40° vertical field of view. The vertical resolution varies across the field of view, reaching a minimum of 0.167° in the central region. This configuration yields higher point-cloud density near the sensor horizon. The horizontal resolution varies from 0.2° to 0.4°, with a configurable rotation frequency from 10 Hz to 20 Hz. The sensor provides a maximum range of 200 m with a measurement accuracy of 2 cm. Figure 3.10 shows its physical structure and vertical channel distribution, while Table 3.3 summarises the main specifications.

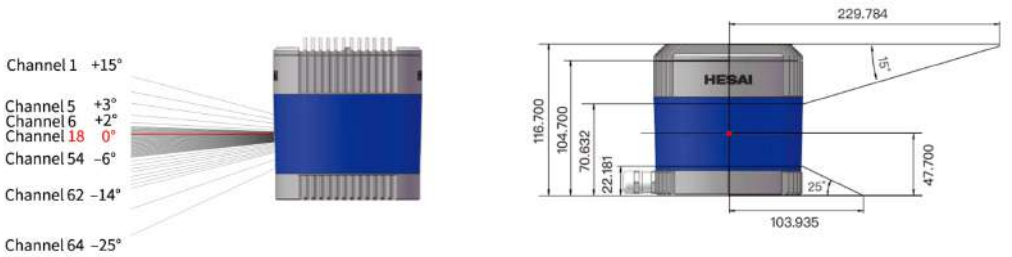


Figure 3.10: HESAI Pandar64 channel distribution and main dimension (mm).

Table 3.3: HESAI Pandar 64 main specifications.

Specification	value
Channels	64
Vertical FOV	40°
Vertical resolution	up to 0.167°
Horizontal FOV	360°
Horizontal resolution	0.2 - 0.4°
Operating Frequency	10- 20Hz
Range	200 m
Accuracy	0.02 m

3.4.2 Optical sensors

Four primary optical sensors are employed in this research: two RGB cameras, one monochrome camera, and one thermal camera. These sensors are integrated into different configurations depending on the specific application requirements.

The first RGB camera is a Canon EOS 650D (see Figure 3.11a), which offers very high-resolution imaging with a maximum pixel resolution of 5184 × 3456. Equipped with an 18–55 mm lens and a CMOS APS-C sensor, it captures rich, high-quality colour images at up to 30 frames per second. Although optimised for photographic use and not ideal for deployment on autonomous

(a) **(b)** **(c)** **(d)**

Figure 3.11: Optical sensor used.

Sources: www.canon.it, www.kayetoncctv.com, www.baslerweb.com, www.flir.com

vehicles due to its size and weight, the Canon EOS 650D was utilised extensively in various ground-based data acquisition campaigns. The main specifications are detailed in Table 3.4.

Table 3.4: Main specifications of the Canon EOS 650D camera

Spec.	value
Sensor model	APS-C CMOS
Color type	RGB
Lens	18–55 mm
Resolution (HxV pixels)	up to 5184 × 3456
Maximum Frame Rate	30 fps
Interface	USB 2.0, HDMI
Housing (L×W×H)	133 × 100 × 79 mm
Weight	530 g (body only)

The second RGB sensor is the compact Kayeton KYT-U200 (see Figure 3.11b). Featuring an OmniVision CMOS sensor, a 6–60 mm varifocal lens, and a resolution of 1280 × 720 pixels, it delivers high frame rates up to 120 fps via a USB 2.0 interface. Its small form factor (50 × 40 × 55 mm) and lightweight design (270 g) make it particularly suitable for onboard installation, although it lacks formal IP certification. The main specifications are reported in Table 3.5.

Table 3.5: Main specifications of the KYT-U200 camera

Specification	Value
Sensor model	OmniVision CMOS
Color type	RGB
Lens	6-60 mm varifocal
Resolution (HxV pixels)	1280 × 720
Max frame rate	120fps
Interface	USB 2.0
Housing (L×W×H)	50 × 40 × 55mm
Weight	270 g

The third optical sensor operating in the visible spectrum is a monochrome Basler acA1920-40gm camera (see Figure 3.11c). It employs a Sony IMX249 CMOS sensor delivering a resolution of 1920 × 1200 pixels at frame rates up to 42 fps. Designed with a fixed 8 mm lens and powered through GigE Vision with Power over Ethernet (PoE), this camera provides reliable, high-accuracy imaging despite the limited feature richness of monochrome data, making it well-suited for controlled laboratory measurements and fixed-point monitoring applications. The key specifications are summarised in Table 3.6.

Finally, a FLIR A65 thermal camera (see Figure 3.11d) was used for thermal imaging. With a wide 90° × 69° field of view and an IR resolution of 640 × 512 pixels, it captures detailed thermal data at 30 frames per second. Its compact size (108 × 50 × 47 mm) and light weight (210 g) enable flexible deployment in various experimental scenarios. Main technical data are listed in Table 3.7.

Table 3.6: Main specifications of the Basler acA1920-40gm camera

Specification	Value
Sensor model	Sony IMX249 CMOS
Color type	Monochrome
Lens	8 mm fixed
Resolution (HxV pixels)	1920 × 1200
Frame rate (max)	42 fps
Interface	GigE Vision (PoE supported)
Housing (L×W×H)	42 × 29 × 29 mm
Weight	90 g

Table 3.7: Main specifications of the FLIR A65 Thermal camera.

Specification	Value
FOV	90°x 69°
IR Resolution	640 × 512 pixels
Frame rate	30 fps
Lens	7.5 mm fixed
f-number	1.25
Detector Pitch	17 μm
Spectral Range	7.5–13 μm
Thermal Sensitivity	<0.05°C @ 30°C
NETD	50mK
Accuracy	± 5°C
Housing (L x W x H)	108 × 50 × 47 mm
Weight	210 g

3.4.3 INS-GNSS

To acquire accurate data on the vessel's position and orientation, a high-performance GNSS/INS (Global Navigation Satellite System / Inertial Navigation System) unit was utilised. Specifically, the Xsens MTi-G-710 system was selected for its reliable integration of inertial and satellite navigation capabilities at high frequency.

The MTi-G-710 integrates accelerometers, gyroscopes, magnetometers, and a GNSS receiver into a single device, enabling robust proprioceptive sensing of translational and rotational dynamics. An image of the GNSS/INS is provided in Figure 3.12, while the detailed technical specifications are summarised in Table 3.8.

3.5 Computing and communication

This section provides an overview of the computing platforms and the communication infrastructure that enable data exchange across system components.

Sharing sensor measurements and control commands among multiple entities is a fundamental requirement of the proposed system. These entities operate as nodes (or clients) within an IoT-inspired network, where each node may require full or partial access to sensor streams for logging, filtering, processing, or control.

To support this distributed architecture, the communication middleware must be scalable,

Figure 3.12: Xsens MTi-G-710 image.

Source: www.xsens.com

Table 3.8: GNSS/INS technical specifications.

Sensor	Spec.	Value
Gyroscope	Standard full range	450 deg/s
	In-run bias stability	10 deg/h
	Noise density	0.01 °/s/$\sqrt{Hz}$
Accelerometer	Standard full range	20 g
	In-run bias stability	15 μg
	Noise density	0.01 °/s/$\sqrt{Hz}$
Magnetometer	onboard	Yes
Barometer	onboard	Yes
GNSS receiver	Brand	u-blox
	Model	MAX-M8

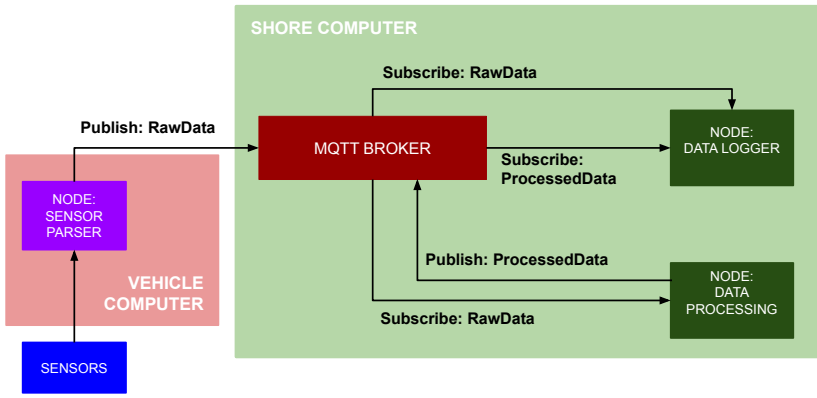


Figure 3.13: MQTT broker-based communication architecture,

modular, and platform-agnostic, enabling interoperability among heterogeneous hardware and software components. For this reason, a publish–subscribe paradigm, widely adopted in IoT networks, has been selected. In this model, publishers generate data streams, while subscribers receive only the information of interest, without direct knowledge of each other.

The implemented architecture is based on MQTT (Message Queuing Telemetry Transport), a lightweight, broker-based messaging protocol widely used in IoT applications. MQTT is designed to minimise bandwidth usage and computational overhead, making it well-suited for low-power devices and for the experimental setup considered in this work. Its maturity and broad adoption further support its reliability in distributed systems.

In MQTT, publishers send messages to string-identified topics, and subscribers register their interest in specific topics. A central broker receives all published messages and forwards them to their corresponding subscribers, enabling asynchronous, decoupled communication among nodes. This mechanism ensures flexible data exchange without requiring direct connections between communicating entities.

The implementation relies on Mosquitto [113], an open-source MQTT broker selected for its stability, simplicity, and widespread adoption.

Figure 3.13 shows a simple MQTT setup. The onboard computer runs a parsing node that reads raw sensor data and publishes it to the *RawData* topic. Onshore, a processing node subscribes to *RawData*, performs tasks such as obstacle detection, tracking, or state filtering, and publishes results to *ProcessedData*. A logging node subscribes to both *RawData* and *ProcessedData* and stores all messages for post-mission analysis.

The physical infrastructure comprises an onshore Local Area Network (LAN) connected to a central switch via a high-gain Wi-Fi antenna. This antenna forms a point-to-point link with a paired antenna on board the SWAMP vessel. All onboard modules, including propulsion controllers and the main computer, connect to this same Wi-Fi network. The link provides reliable coverage

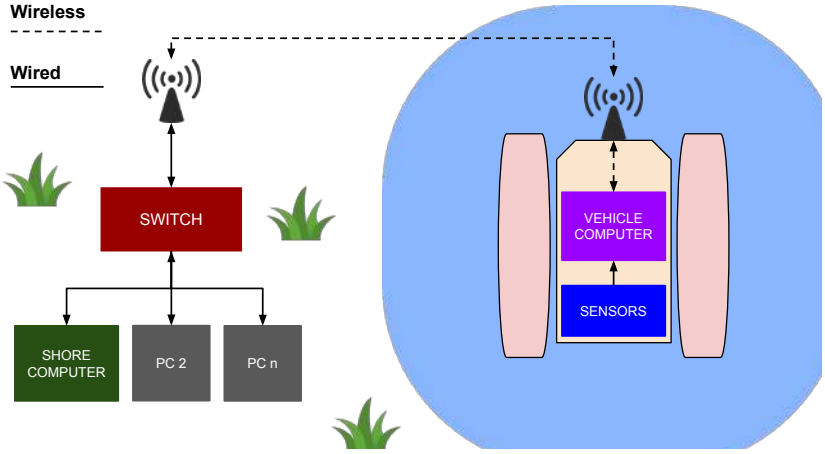


Figure 3.14: Physical communication infrastructure.

beyond 500 m under line-of-sight. Sensors interface with the onboard computer via dedicated wired connections. Figure 3.14 summarises the full physical architecture.

The computing environment relies on two main machines: an on-shore Asus ROG Zephyrus G14 for simulation, development, and data processing, and an onboard semi-rugged Acer Enduro N3 for real-time sensor handling during field trials. Table 3.9 reports their technical specifications.

Table 3.9: Computer's main specifications.

Specification	Shore PC	Vehicle PC
Model	ASUS ROG G14	ACER ENDURO N3
CPU	AMD Ryzen 9 6900HS 4.9 GHz	Intel Core i3-1115G4 4.10 GHz
RAM	16 GB DDR5	8 GB DDR4
GPU	AMD Radeon RX 6700S	UHD Graphics
VRAM	8 GB GDDR6	N.A.
OS	Fedora Linux 40-42 Ubuntu 22.04 LTS	Fedora Linux 41

4. Methodology

4.1 Sensing architecture

This chapter presents the methodological framework for designing, implementing, and validating the proposed situational awareness system. The objective is to describe how the individual sensing, perception, fusion, and state-estimation modules are constructed and integrated into a complete operational architecture.

The chapter is organised as follows. First, the overall functional architecture of the system is introduced and explained (Figure 4.1). Then, each sensing and processing pipeline is described in detail, including LiDAR, Radar, and optical sensors, along with their fusion strategies. Subsequently, the state-estimation module based on INS–GNSS fusion is presented. Finally, the indoor positioning system used for experimental validation is described. A dedicated subsection discusses the main limitations and sources of uncertainty affecting the proposed architecture.

Figure 4.1 illustrates the overall functional architecture of the proposed system. The diagram is organised from left to right and represents the data flow from sensing to high-level modules.

On the left side, the six sensor categories are shown: RGB camera, thermal camera, LiDAR, radar, GNSS receiver, and Inertial Navigation System (INS). Each sensor feeds one or more dedicated processing pipelines. The colour-coding indicates the data's origin and helps track how information propagates through the architecture. Note that the double colour indicates a merge module that inherits the colours of the modules that converge on it.

Some sensors operate independently (e.g., standalone LiDAR or radar pipelines), while others contribute exclusively within fusion frameworks. Multiple fusion strategies are implemented, including early fusion and late fusion configurations.

On the right, the outputs of the detection and fusion pipelines converge on the multi-target tracking module, which subsequently provides information to the Guidance and Control layer. Although Guidance and Control are shown for completeness, they are not within the scope of this dissertation.

Note that the figure shows the entire architecture in its fully expanded, overlapping configuration, showing all its components and possible interconnections simultaneously. In practical operation, however, the system is intended to be used modularly, activating only the specific branches and processing pipelines required to address the target application or mission objective.
To better clarify this concept, two possible system configurations, depending on the operational requirements, are proposed.
Figure 4.2 shows a configuration suitable for daytime transfer navigation. In this case, obstacle perception is achieved through two parallel multimodal pipelines: one oriented toward target classification using short-range LiDAR and RGB camera data, and another focused on target tracking through Radar–LiDAR fusion to cover a wider detection range. For standard navigation

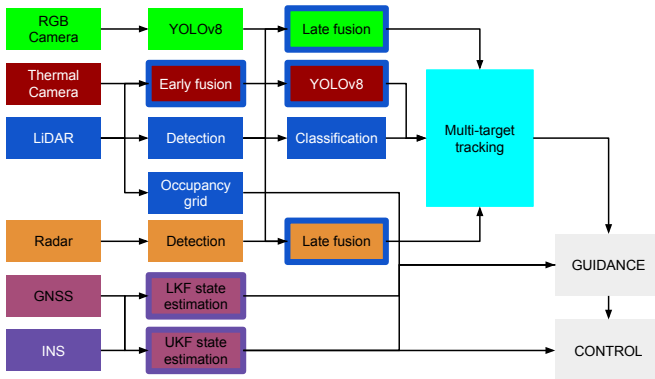


Figure 4.1: Functional sensing architecture.

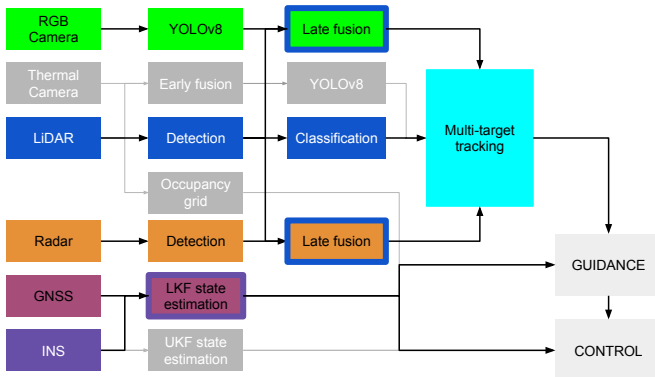


Figure 4.2: Examples of modular activation of the sensing architecture: daylight navigation configuration.

tasks, state estimation can rely on a conventional Linear Kalman Filter, thereby avoiding unnecessary computational costs.

Figure 4.3 illustrates a different configuration tailored to nighttime Search and Rescue (SAR) operations. In this scenario, the perception layer relies on a LiDAR–thermal imaging fusion pipeline to detect survivors and obstacles under low-visibility conditions. State estimation is entrusted to an Unscented Kalman Filter, due to the strong nonlinearities associated with low-speed SAR manoeuvres, abrupt trajectory changes, and the requirement for accurate positioning in complex disaster-response environments.

The main processing pipelines described in the following sections include:

- LiDAR processing pipeline (Section 4.2), operating either in detection/classification mode or via a fast occupancy grid approach;
- Radar processing pipeline (Section 4.4);

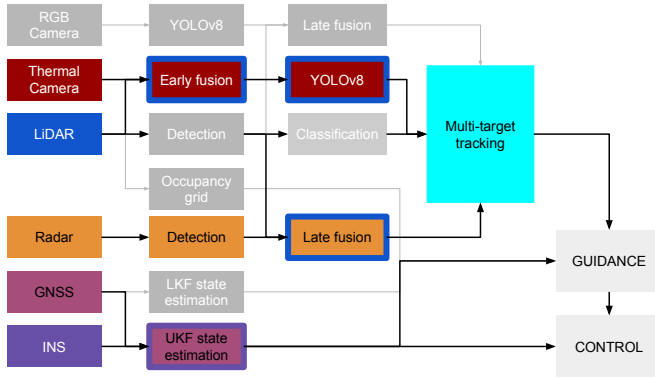


Figure 4.3: Examples of modular activation of the sensing architecture: night SAR configuration.

- LiDAR–Radar late fusion pipeline (Section 4.5);
- LiDAR–RGB late fusion pipeline (Section 4.6.1);
- LiDAR–Thermal early fusion pipeline (Section 4.6.2);
- GNSS–INS state estimation pipeline, implemented using either a standard Kalman Filter (Section 4.8.1) or an Unscented Kalman Filter (Section 4.8.2).

4.2 LiDAR

This section presents the LiDAR processing pipeline developed within the sensing architecture. The objective of this branch is to extract reliable geometric and kinematic information about surrounding obstacles to support navigation and collision-avoidance tasks.

For safe navigation, the system must provide consistent estimates of each detected target's position, velocity, and course angle. These quantities are fundamental inputs to higher-level modules, such as multi-target tracking, risk assessment, and path planning, thereby enabling both strategic trajectory generation and reactive manoeuvres.

In addition to geometric detection, preliminary object classification based solely on LiDAR data is considered beneficial. In coastal and near-shore environments, not all vessels are equipped with cooperative systems such as AIS (Automatic Identification System). Therefore, the ability to infer basic object categories directly from point cloud data enhances situational awareness in non-cooperative scenarios and improves the robustness of the overall sensing layer.

Figure 4.4 summarises the full pipeline. LiDAR data is first filtered to remove noise. DBSCAN clustering segments individual objects, each treated as a potential target. PCA is then applied to compute bounding boxes and estimate object orientation and size. These steps form the detection stage. Detected objects are then passed to the multi-target tracking module as point targets.

In parallel, a classification branch extracts features from each clustered point cloud. These features are fed into a pre-trained Random Forest Classifier, which outputs an object class without

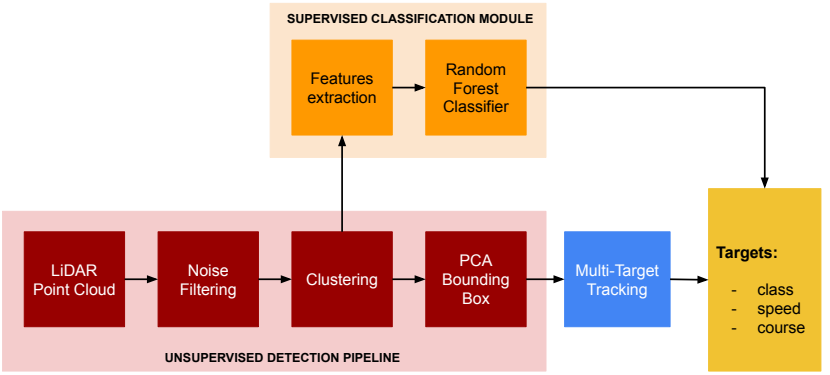


Figure 4.4: LiDAR detection, classification, and tracking pipeline.

affecting the core detection–tracking flow. The system thus provides both kinematic and semantic information.

The final result is a complete pipeline for detection, classification, and tracking that combines unsupervised (DBSCAN) and supervised (Random Forest) learning. This framework was developed over a three-year research activity and reported in [32, 35, 31, 33].

4.2.1 Detection pipeline

The detection process in this framework is structured into three main phases: (i) noise filtering, (ii) clustering, and (iii) bounding box construction. Each stage plays a critical role in preparing the LiDAR point clouds for subsequent classification.

The pipeline was intentionally designed around unsupervised detection methods. Unlike deep learning-based alternatives such as PointPillars or PointNet++, which require large annotated training datasets and often yield models that are difficult to interpret and whose performance may not generalise beyond the dataset context, the proposed methodology provides a more explainable and broadly applicable detection framework. This makes it particularly suitable for maritime scenarios, where operational conditions are highly variable and annotated datasets are scarce.

Additionally, the pipeline was constructed to meet specific operational requirements. It is modular, allowing individual components (e.g., the clustering algorithm or bounding box construction) to be easily replaced or updated if more effective methods become available. It is also general-purpose and functional in marine environments, enabling the detection framework to handle the specific challenges of coastal navigation, such as sparse and dynamic point clouds, while remaining flexible for future extensions or adaptations to different scenarios.

4.2.1.1 Noise filtering

Noise filtering is the first step in the LiDAR pipeline and removes spurious points that could compromise clustering and feature extraction. In marine environments, these points often originate from wake trails, surface reflections, ripples, breaking waves, and other unstable returns, as observed from experimental data.

LiDAR provides 3D geometry plus reflectivity, which helps distinguish noise: low-intensity points usually correspond to weak or scattered reflections.

A simple approach discards points below a fixed intensity threshold, but this is scenario-dependent. Experimental analysis suggests relative thresholds (e.g., removing points below 10% of maximum scene intensity) suppress some noise while preserving most valid measurements. Higher thresholds improve noise rejection but risk discarding low-reflectivity objects.

The method developed in this work combines intensity-based and spatial filtering, tuned based on experimental observations, for effective first-stage denoising. A relative intensity threshold higher than the conventional 10% (typically 15–25%) is applied in conjunction with spatial constraints. Specifically, spatial filtering targets points near the sea surface, with a wide vertical tolerance, based on the known sensor position, as shown in (4.1). Let $\mathbf{p}_i = (x_i, y_i, z_i, I_i)$ be the i-th LiDAR point, where $(x_i, y_i, z_i) \in \mathbf{R}^3$ is the spatial position and $I_i \in \mathbf{R}$ is the intensity (reflectivity) value. Are defined:

- $I_{\max} = \max\{I_j\}$: the maximum intensity in the point cloud;
- $\tau_I \in [0, 1]$: the relative intensity threshold (e.g., $\tau_I = 0.15$ for 15%);
- z_{sea}: the estimated sea surface level in the sensor reference frame;
- δ_z: the vertical tolerance around z_{sea}.

The filtering condition is then defined as:

$$\mathbf{p}_i \text{ is discarded if } (I_i < \tau_I \cdot I_{\max}) \wedge (|z_i - z_{\text{sea}}| < \delta_z) \tag{4.1}$$

This approach allows selective removal of low-intensity points near the sea surface, while preserving meaningful points that belong to actual objects.

4.2.1.2 Clustering

The clustering phase groups points that are likely from the same object, enabling the extraction of meaningful geometric and statistical features.

Clustering is a key step when processing unstructured point clouds, as it organises spatially and semantically related points into coherent groups. Various clustering methods differ in how they define a “cluster”, and each offers benefits depending on the data and application. In this work, clustering is applied to LiDAR point clouds to isolate individual objects based on spatial proximity.

To reduce computational cost, the 3D LiDAR cloud is first projected onto a 2D plane aligned with the sea surface (Bird's-Eye View). This is justified by the fact that vertical overlap between independent objects is rare in marine navigation. The 2D representation, therefore, preserves essential information while significantly simplifying and accelerating the clustering process.

Clustering is then performed using DBSCAN [41], a density-based method that identifies clusters as dense regions separated by sparsely populated areas. DBSCAN relies on two parameters:

- ε (epsilon): the maximum distance between two points for them to be considered part of the same neighbourhood;
- `minPts`: the minimum number of points required to form a dense region.

The core idea of DBSCAN is to classify points into three categories:

1. **Core points**: points that have at least `minPts` neighbors within a radius ε;
2. **Border points**: points that have fewer than `minPts` neighbors within ε, but are within the ε-neighborhood of a core point;
3. **Noise points (outliers)**: points that are neither core nor border points.

The algorithm iteratively visits each point in the dataset and, if it is a core point, it expands a cluster by recursively collecting all density-connected points. This recursive expansion stops when no more points can be added to the cluster under the defined criteria.

The selection of DBSCAN is motivated by several properties that are particularly relevant in maritime LiDAR processing. First, unlike methods that require specifying the number of clusters a priori (e.g., k-means and Gaussian mixture models), DBSCAN does not require a priori knowledge of the number of objects in the scene. This is essential in maritime navigation, where the number and position of nearby vessels or obstacles can vary unpredictably.

Second, the explicit notion of noise is fundamental in this application. Marine point clouds often contain sparse returns due to wave crests, spray, reflections, or multipath effects. Unlike simple Euclidean-distance grouping methods, DBSCAN naturally labels isolated or low-density points as outliers, preventing them from contaminating object clusters.

Third, DBSCAN mitigates the so-called "bridging" effect, which is typical of purely distance-based clustering approaches (e.g., region growing with a fixed radius). In maritime environments, thin structures such as ropes, fishing lines, or partial wake patterns may connect two otherwise distinct objects through a small number of intermediate points. Algorithms that rely solely on pairwise Euclidean distances can incorrectly merge such objects into a single cluster. Because DBSCAN requires a minimum local density to expand a cluster, sparse connections are less likely to lead to artificial merges.

Alternative density-based methods, such as OPTICS or HDBSCAN, could also be considered. While these approaches offer advantages in handling clusters with varying densities and may reduce sensitivity to parameter tuning, they introduce increased computational complexity and additional hyperparameters.

After clustering, each identified group of points is treated as a candidate object. The clustering stage thus produces spatially coherent entities that are forwarded to the subsequent detection, classification, and multi-target tracking modules.

In this work, DBSCAN is applied after projecting the 3D point cloud onto a 2D plane to approximate the sea surface. Each resulting cluster is treated as a candidate object. The clustering stage thus produces spatially coherent groups of points, which are then forwarded to the subsequent detection, classification, and multi-target tracking modules.

4.2.1.3 Oriented bounding box

Bounding box construction provides a compact spatial representation of each detected cluster, which forms the basis for feature extraction and ensures consistency across the classification and tracking pipeline.

Each cluster corresponds to a LiDAR-derived point cloud representing a detected object. For tracking, a stable reference point is needed. Using the cluster centroid is straightforward but often inaccurate, as LiDAR measurements bias points toward the first surfaces hit by the laser, leading to systematic shifts.

To mitigate this, the geometric centre of a bounding box fitted to the cluster is used as a reference, thereby providing a more robust position estimate.

Bounding box construction relies on Principal Component Analysis (PCA), which identifies the directions of greatest variance, allowing the box to align with the object's true orientation. As in clustering, the box is computed on the 2D horizontal plane (bird's-eye view), assuming two orthogonal principal axes.

For efficiency, PCA is implemented via Singular Value Decomposition (SVD) (Eq. (4.2)). A cluster with n points is expressed by LiDAR coordinates $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$, stacked into the matrix $\boldsymbol{X}$. The centered cloud is computed as $\boldsymbol{X_c} = \boldsymbol{X} - \bar{\boldsymbol{X}}$, where $\bar{\boldsymbol{X}}$ is the mean vector. The right singular vectors matrix $\boldsymbol{V}$ contains the principal directions and is used to express the point cloud in the new reference system defined by the principal directions, according to Eq. (4.3), yielding $\boldsymbol{Z}$, the cluster point cloud expressed in principal coordinates.

$$\boldsymbol{X_c} = \boldsymbol{USV}^T \tag{4.2}$$

$$\boldsymbol{Z} = \boldsymbol{X_c V} = \{z_1, z_2\} \tag{4.3}$$

From the $\boldsymbol{Z}$ cloud, the maximum and minimum values for each principal direction are extracted and re-expressed in the original reference system following Eq. (4.3). The vertices of the oriented bounding box are thus calculated by vector sum. The process is summarised in Figure 4.5.

The geometric centre of the resulting rectangular bounding box serves as a more accurate and stable reference point for the object, which is subsequently used in tracking stages.

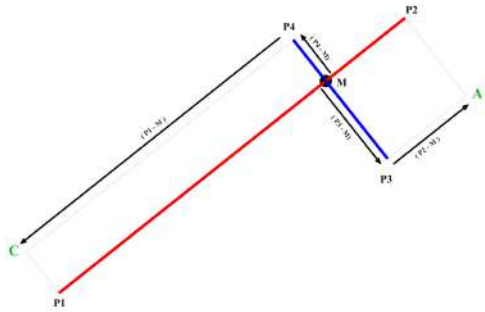


Figure 4.5: PCA-based bounding box construction method.

4.2.2 Classification Module

This section follows the logical thread adopted during the research. The aim is to develop a classification module that predicts the class of a cluster of LiDAR points by recalling the learned model from a suitable training dataset. In marine environments, LiDAR point clouds are inherently sparse due to the large distances involved, which limits the applicability of direct classification methods operating on raw point-level data. To address this challenge, the pipeline relies on the extraction of features hypothesised to carry predictive information (discussed in Section 4.2.2.1). The discriminative capacity of these features is subsequently analysed to identify the most informative subset for classification through Exploratory Data Analysis (EDA), both qualitatively and quantitatively (Section 4.2.2.2). On this basis, a supervised learning model is selected and systematically optimised through training and hyperparameter tuning, as detailed in Section 4.2.2.3 and 4.2.2.4. It is important to emphasise that this framework necessarily relies on a labelled dataset, in which each point cloud is associated with a specific class. In line with the purpose of this chapter, only the methodological and mathematical aspects are presented here; the approach remains valid regardless of the particular dataset employed. For details on the dataset used in this work, see Chapter 5.1.

4.2.2.1 Feature selection

The framework is designed to produce, for each cluster, a compact yet information-rich feature set. These features are derived from labelled point clouds, with particular attention given to statistical descriptors and geometric quantities that characterise the target. Geometric information is obtained by applying Principal Component Analysis (PCA) to the point cloud. Specifically, the eigenvalues (explained variances) and eigenvectors (principal directions) of the covariance matrix are extracted. Importantly, PCA represents a fundamental step in this framework because expressing all points in the new reference frame defined by the cluster's principal directions effectively removes any dependence on the object's orientation. The procedure follows the steps

outlined in Section 4.2.1.3. Let $\boldsymbol{S}$ denote the diagonal matrix of singular values s_i obtained from the Singular Value Decomposition; the eigenvalues (λ_i) of the covariance matrix, corresponding to the explained variance along each principal direction, are then computed as:

$$\lambda_i = \frac{s_i^2}{n-1} \tag{4.4}$$

For each cluster, the three eigenvalues $\lambda_1, \lambda_2, \lambda_3$ are obtained, which capture the variance of the point distribution along the principal axes. Their absolute values and relative ratios highlight essential aspects of the cluster's geometry, such as the predominance of one direction over the others. Elongated or surface-like objects exhibit characteristic eigenvalue ratios that enable their identification. To further quantify these differences, the ellipticities $e_{1,2}$, $e_{1,3}$, and $e_{2,3}$ are computed in the planes defined by pairs of principal directions, as:

$$e_{i,j} = 1 - \frac{\lambda_j}{\lambda_i} \tag{4.5}$$

While the eigenvalues indicate the relative importance of each principal direction, the ellipticity measures provide additional information about the shape of the distribution and its deviation from a spherical or isotropic structure. Transforming the target point clouds from the sensor reference frame to the PCA coordinate system is therefore crucial, as it standardises their representation and facilitates the extraction of orientation-independent statistical and geometric descriptors.

Finally, higher-order statistical moments such as skewness γ and kurtosis k can be computed along each principal direction, as expressed in Eq.(4.6) and Eq.(4.7).

$$\gamma = \frac{\sum_{i=1}^{n} (\boldsymbol{Z}_i - \bar{\boldsymbol{Z}})^3}{(n-1)\,\boldsymbol{\sigma}^3} \tag{4.6}$$

$$k = \frac{\frac{1}{n}\sum_{i=1}^{n} (\boldsymbol{Z_i} - \bar{\boldsymbol{Z}})^4}{\boldsymbol{\sigma}^4} \tag{4.7}$$

In summary, skewness captures the asymmetry of the point distribution along each principal direction, while kurtosis describes the peakedness of the distribution relative to a normal distribution. Together, these higher-order moments provide additional insight into the structural and geometric properties of the point cloud, complementing the information obtained from the eigenvalues and ellipticities.

The LiDAR point cloud provides reflectivity values for each point, represented as integers in the range [0, 255]. These values can be further analysed to extract additional descriptive features. A natural approach is to compute statistics based on central moments, including the mean μ_ι, standard deviation σ_ι, skewness γ_ι, and kurtosis k_ι of the reflectivity distribution. Moreover, the Shannon entropy is calculated to capture the overall variability of reflectivity within the cluster. The entropy H is defined as:

$$H = -\sum_{i=1}^{n} p_i \log_2(p_i), \tag{4.8}$$

where p_i represents the probability of observing a specific reflectivity value within the cluster.

In addition to these statistical descriptors, spatial autocorrelation measures are computed to quantify the spatial organisation of reflectivity values. Specifically, Moran's I [114] (Eq.(4.9)) and Geary's C [115] (Eq.(4.10)) are calculated. The spatial weights matrix $\boldsymbol{W}$ is derived from the point cloud coordinates $\boldsymbol{X}$, with zeros along the diagonal, and each element w_{ij} is based on the normalised Euclidean distance between points. The reflectivity values $\boldsymbol{\iota}$ are used as the variable of interest in these spatial correlation analyses, providing insight into the clustering tendencies of reflectivity within each cluster.

$$I = \frac{n}{S_0} \cdot \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \cdot (\iota_i - \bar{\iota})(\iota_j - \bar{\iota})}{\sum_{i=1}^{n} (\iota_i - \bar{\iota})^2} \tag{4.9}$$

$$C = \frac{(n-1)}{2S_0} \cdot \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (\iota_i - \iota_j)^2}{\sum_{i=1}^{n} (\iota_i - \bar{\iota})^2} \tag{4.10}$$

where S_0 is the sum of all the weight w_{ij}, i.e. $S_0 = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}$.

These 19 features encompass all the proposed characteristics hypothesised to be predictive. Naturally, any dataset used must be preprocessed to include these features alongside the corresponding class labels.

4.2.2.2 Feature analysis

The next step assesses which extracted features contribute most to predicting an object's class by quantifying their importance. As part of the Exploratory Data Analysis (EDA), feature distributions are visualised using violin plots, which provide a comprehensive representation of the data by showing means, variability, and distribution shapes through kernel density estimation (KDE) [116].

To evaluate quantitatively whether a feature can discriminate between object classes, a one-way analysis of variance (ANOVA) [117] is commonly used. However, when the assumptions of a standard one-factor ANOVA are not satisfied, the Kruskal–Wallis H-test [118]—also referred to as a one-way ANOVA by ranks—is employed instead. This nonparametric test tests the null hypothesis of no significant difference in feature values across groups. The results are typically expressed as p-values. Features yielding p-values below a chosen significance threshold can be considered likely to have discriminative power and, therefore, may be selected as candidates for inclusion in the set of features used for classification.

4.2.2.3 Model training

After extracting the features, it is necessary to select a classification algorithm that effectively utilises these variables to predict the target class with high accuracy. To this end, three well-established

classifiers from the literature are considered: a Random Forest [119], a Multi-Layer Perceptron (MLP) [120], and a Support Vector Machine (SVM) [121]. The labelled dataset is partitioned into 70% for training and 30% for testing. A grid search over hyperparameters is performed, and the optimal combination is determined based on the average accuracy obtained from 5-fold cross-validation. Results are provided in Chapter 6.

The Random Forest classifier is ultimately selected for its consistently high accuracy and the added advantage of providing a direct assessment of feature importance. Once the model is trained on all relevant features, the contribution of each feature to the classification decisions can be quantified. In particular, the Mean Decrease of Impurity (MDI) is computed across the ensemble of decision trees, providing insight into which features are most influential in the model's predictions.

4.2.2.4 Model tuning

Given the need to deploy the model in an operational environment, real-time performance represents a critical constraint. Consequently, features whose extraction time is incompatible with real-time requirements are excluded from the feature set. To identify such features, the computational cost is measured on a real LiDAR scan containing 40 targets of varying nature and size, and averaged over 100 repetitions.

To further investigate feature relevance, a Recursive Feature Elimination (RFE) procedure is applied. Starting from the MDI-based feature-importance distribution and excluding features deemed incompatible with real-time constraints, features are removed one at a time in descending order of importance. At each iteration, the model is retrained on the reduced feature set and evaluated using 5-fold cross-validation to compute the mean accuracy. Moreover, given that the prediction time scales approximately linearly with the number of estimators, the number of Random Forest trees is systematically varied between 64 and 128 for each feature-set iteration. The minimum number of features is restricted to 2. This configuration generates a test grid of size 15×65, corresponding to the number of feature sets and estimator variations, resulting in 975 combinations. All other hyperparameters are kept fixed at their optimised values obtained from prior tuning.

For each combination in the test grid, the following key performance indicators (KPIs) are computed:

- $\boldsymbol{T}1$: the feature extraction time matrix, where each entry $T1, ij$ represents the average time required to extract the corresponding feature set in the ij^{th} combination of the grid. Measurements are performed on the 40-cluster test scenario and averaged over 100 runs. While a simple vector could suffice, the matrix $\boldsymbol{T}1$ is constructed by tiling the computational time vector 65 times to match the dimensions of the test grid.
- $\boldsymbol{T}2$: the classification time matrix, where each element $T2, ij$ indicates the mean time required to classify the test set for the ij^{th} combination, averaged over the number of targets.
- $\boldsymbol{A}$: the mean accuracy matrix, with entries Aij representing the average 5-fold cross-validation accuracy for the ij^{th} combination.

The generalised computational cost matrix $\boldsymbol{T}$ is obtained by summing the feature extraction and classification time matrices, $\boldsymbol{T}_1 + \boldsymbol{T}_2$. Once a specific combination of features and hyperparameters is selected, the Random Forest classifier is retrained on the entire dataset and evaluated across two dynamic operational scenarios. During this process, the overall computational cost—from sensor data parsing to detection and classification—is continuously monitored as an additional KPI.

4.2.3 A simple but effective approach: the occupancy grid

A very simple yet effective method for processing LiDAR data solely to map obstacles for navigation is the *occupancy grid* approach. The LiDAR point cloud can be represented in either three-dimensional or planar coordinates and can be subsampled to match the voxel or grid size of arbitrary dimensions. This grid is then forwarded to guidance modules, which mark regions of the navigable space that may contain potential obstacles.

The limitations of this approach are evident: there is no information about the nature or motion of obstacles, contiguous objects cannot be distinguished as separate entities, and it is not possible to provide data suitable for tracking algorithms or predicting object motion. Nevertheless, when the basic presence of obstacles is sufficient, the method proves extremely effective for simple collision-free navigation, treating obstacles as static within each time step, particularly when the update frequency is high.

This method performs best when expressed in planar 2D coordinates and strongly subsampled, using the maximum grid size appropriate for the application. In this way, a lightweight set of obstacle points is obtained relative to the original point cloud, containing all the information required for collision-free navigation.

The extreme efficiency of this simple method arises from its ability to be interposed as a parallel processing step alongside more sophisticated pipelines. For example, a detection, classification, and tracking pipeline may operate asynchronously at around 1 Hz, while the occupancy grid pipeline can run at the sensor frequency of 5–20 Hz, starting from the same point cloud, providing a fast and reliable representation of static obstacles in the environment.

4.3 Optical Sensors

In this work, the RGB camera is primarily employed for target detection in the two-dimensional image plane. Specifically, the objective of this branch is to identify the pixel region corresponding to a target of interest, without directly estimating its three-dimensional position. The output of the camera-based pipeline is therefore a 2D detection (e.g., bounding box or segmented region) expressed in image coordinates.

It is important to clarify that, in maritime environments, optical sensors are not intended to operate as standalone localisation systems. While in terrestrial applications, cameras can often provide sufficient information for both detection and position estimation, this assumption does not generally hold at sea. Maritime scenarios are characterised by large observation distances, limited visual texture, dynamic backgrounds (e.g., waves, reflections), and moving targets. These factors make accurate depth inference from monocular imagery particularly challenging and often

unreliable.

For this reason, the role of the camera in the proposed architecture is primarily semantic rather than geometric: it contributes to target recognition and classification, while metric localisation is delegated to three-dimensional sensors such as LiDAR. The integration of visual and geometric sensing enables the system to exploit the complementary strengths of both modalities—rich appearance information from the camera and accurate spatial measurements from LiDAR—thereby improving situational awareness compared to single-sensor approaches.

Nevertheless, although the research conducted during the doctoral program has primarily focused on integrating cameras into sensor fusion frameworks, relevant references and procedures on image processing are also included in this section. These are either directly employed in the proposed work or are necessary to ensure a complete understanding of the methodologies adopted.

4.3.1 Standard Procedures

In this work, some standard procedures widely adopted in computer vision are employed to ensure a correct representation and use of optical sensors. These procedures form the basis for most image-based perception systems and are included here to provide the necessary background for subsequent sections. In particular, reference is made to the pinhole camera model, which provides a mathematical formulation of image formation, and to the calibration procedure, which enables estimation of the camera's intrinsic parameters and compensation for lens distortion.

4.3.1.1 Pin-hole Camera Model

The pinhole camera model is the most commonly used approximation for describing the projection of three-dimensional points onto the two-dimensional image plane. According to this model, a 3D point $\mathbf{P} = [X, Y, Z]^\top$ in the camera reference frame is projected to a 2D image point $\mathbf{p} = [u, v]^\top$ through a perspective projection, which can be expressed in homogeneous coordinates as:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4.11}$$

where s is a scaling factor, $\mathbf{K}$ is the intrinsic calibration matrix, and $\mathbf{R}, \mathbf{t}$ represent the extrinsic parameters, i.e., the orientation and position of the camera in the world reference frame. This formulation allows a compact description of the imaging geometry and provides the foundation for camera calibration.

4.3.1.2 Calibration and Distortion Model

Camera calibration is the process of estimating the intrinsic parameters of the camera, such as focal length, principal point, and skew coefficient, as well as the extrinsic parameters relating the camera reference frame to the world frame. In this work, calibration is performed using the widely

adopted method proposed by Zhang [122], which is based on observing a planar calibration pattern (typically a chessboard) from multiple viewpoints. This method provides an effective, practical solution for obtaining accurate estimates of camera parameters.

In addition to intrinsic and extrinsic parameters, calibration also accounts for lens distortions, which can significantly affect measurement accuracy if not properly corrected. A common choice is the five-parameter distortion model, which includes three radial distortion coefficients (k_1, k_2, k_3) and two tangential distortion coefficients (p_1, p_2). Radial distortion accounts for the "barrel" or "pincushion" deformation of images, while tangential distortion arises from lens misalignment. Correcting for these distortions is essential to ensure the reliability of image-based measurements, especially in applications requiring high geometric accuracy. The following equations give the polynomial five-parameter distortion model:

$$\begin{aligned} u_{\text{dist}} &= u\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) + 2p_1 uv + p_2(r^2 + 2u^2) \\ v_{\text{dist}} &= v\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) + p_1(r^2 + 2v^2) + 2p_2 uv \\ r^2 &= u^2 + v^2. \end{aligned} \tag{4.12}$$

where (k_1, k_2, k_3) and (p_1, p_2) are the radial and tangential distortion parameters, respectively; u, v are the undistorted image coordinates and u_{dist}, v_{dist} are the distorted image coordinates.

4.3.2 YOLO-based Detection

The rationale for selecting YOLOv8 as the object detection backbone is primarily driven by operational requirements and the need for a fast, modular, and reliable detection method suitable for maritime scenarios. YOLO networks are single-stage detectors that provide real-time performance with competitive accuracy, making them ideal for embedded systems with limited computational resources, such as autonomous surface vessels.

Compared to more complex or two-stage detectors, YOLO offers a favourable trade-off between speed and precision, allowing the perception system to maintain high update rates for downstream tasks such as multi-target tracking, path planning, and collision avoidance. The use of smaller models ("n" and "s" variants) minimises computational load during routine operation, whereas the larger "l" variant is reserved for scenarios where higher detection accuracy justifies the additional cost.

Another key motivation is modularity. The YOLO module is implemented as a self-contained component, which can be replaced or upgraded independently without affecting the rest of the perception pipeline. This aligns with the system design philosophy of flexibility, allowing future integration of alternative detection approaches or retraining on new datasets specific to the operational environment.

Finally, YOLO's architecture is particularly well-suited to applications with limited annotated datasets. Its end-to-end training approach, combined with transfer learning from general-purpose datasets, allows effective adaptation to maritime contexts without requiring enormous volumes of

custom-labelled data. This makes it a practical and scalable solution for robust obstacle detection in complex, dynamic, and often unstructured maritime environments.

The YOLO module is treated as a self-contained component within the broader perception pipeline. Its primary role is to process a single input image and produce a set of detected obstacles, each associated with a bounding box and class label. This functionality provides the system with a fast, reliable method for extracting relevant objects from raw images, supporting downstream tasks such as tracking, path planning, and collision avoidance.

For completeness, note that YOLO formulates object detection as a single regression problem, mapping image pixels directly to bounding boxes and class probabilities in a single forward pass. The input is typically a resized image, for example 416×416 or 640×640 pixels, which is divided into a grid. For each cell, the network predicts a fixed number of bounding boxes, along with associated confidence scores and class probabilities. The resulting output is a set of candidate detections that are refined through post-processing steps, such as non-maximum suppression.

The choice of training dataset depends on the specific objectives of the detection pipeline, and may consist of open-source collections or custom-built datasets tailored to the operational environment. By efficiently converting raw images into a set of classified obstacles, the YOLO module is a key enabler of real-time perception within the overall system, while remaining modular and interchangeable with alternative detection approaches if required.

4.4 Radar

The radar sensor was not initially considered in this research in favour of other sensors such as LiDAR, optical, and inertial sensors. This decision, beyond the limitations related to sensor availability and the time required to conduct the studies, has broader justification. A close collaboration was established with members of the Institute for Electromagnetic Sensing of the Environment (IREA) of the National Research Council of Italy (CNR), based in Naples, who specialise specifically in radar data processing. As a result, to support this collaboration, radar data have been integrated with LiDAR via sensor fusion, using the processing pipelines developed by the project partner and those presented in this thesis.

4.5 3D to 3D fusion: LiDAR - Radar

This section presents the methodology for fusing data from two ranging sensors. The approach is general and can be applied to multiple LiDARs, multiple radars, or, as in the example discussed here, a combination of a LiDAR and a radar.

The developed methodology treats obstacles as individual points. It is therefore applicable to sensors whose outputs can be processed to provide a list of points or centroids representing obstacles, such as the LiDAR processing pipeline described in Section 4.2.

Centroid-based measurements can be directly fed into a tracking module as separate, asynchronous inputs. The tracking algorithm is then responsible for associating and updating tracks, as will be detailed in Section 4.7. However, this often requires operating the tracker and its

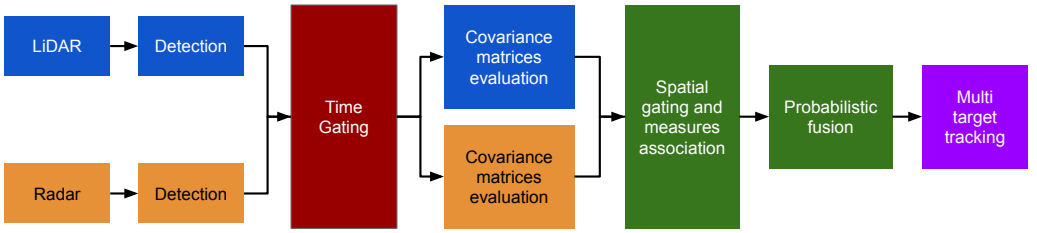


Figure 4.6: LiDAR-Radar probabilistic fusion pipeline.

corresponding Kalman filter with non-uniform time steps.

In this section, a sensor fusion module is proposed that operates upstream of the tracker, providing it with a set of pre-fused obstacle sets at a consistent time step. This ensures that the tracking module receives a harmonised and temporally regularised representation of the environment, simplifying subsequent processing and improving robustness.

Figure 4.6 illustrates the overall processing pipeline. The approach follows a late-fusion strategy, in which data from individual sensors are preprocessed independently to obtain a list of obstacles, a step assumed to be performed by dedicated pipelines. In contrast, an early fusion approach would generate input data of significantly larger size, making real-time processing challenging.

The first block in the pipeline is a temporal gating module, which discretises continuous time into discrete bands using a buffer. Detections are then assigned to a band based on their timestamps, and at each time step, detections within the same buffer are sent to the association or fusion module. The output is a list of fused obstacles, which can then be forwarded to an external multi-target tracking module.

The core of the system is the detection-association module, which is discussed in detail in the following section.

4.5.1 Probabilistic fusion module

The fusion module operates on detections already grouped in time, aiming to cluster them in the spatial domain. The problem is formulated on the navigation plane, considering a set of detections ($\boldsymbol{D}$) obtained as the union of LiDAR and radar measurements:

$$\boldsymbol{D} = (x_i, y_i) = \boldsymbol{D}\text{LiDAR} \cup \boldsymbol{D}\text{RADAR} \tag{4.13}$$

Detections are first grouped into fusion candidates by clustering spatially close measurements, either through standard clustering algorithms or distance-based gating. This step produces a set of obstacles composed of one or more detections. Single-detection obstacles originate from a single sensor and may be filtered or forwarded to the tracker depending on the adopted strategy. Obstacles composed of multiple detections are passed to the fusion stage.

Each sensor measurement is modelled as a multivariate normal distribution with mean $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$. Since measurement uncertainty depends on the relative target-sensor

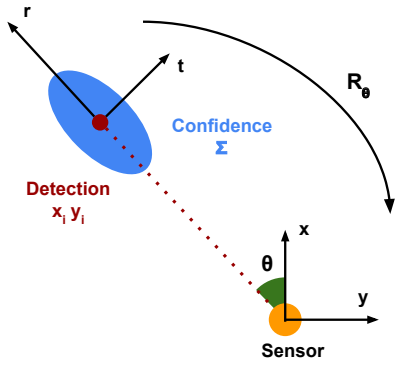


Figure 4.7: Detection confidence evaluation process.

geometry, the covariance is expressed in a local reference frame defined for each detection, with radial (r) and tangential (t) directions. In this frame, the covariance matrix is diagonal:

$$\mathbf{\Sigma}L = \begin{bmatrix} \sigma r^2 & 0\,0 & \sigma_t^2 \end{bmatrix}. \tag{4.14}$$

This formulation adapts the measurement uncertainty to the detection location. The covariance is then expressed in the sensor reference frame through a rotation matrix $\boldsymbol{R}(\theta)$, defined by the sensor-to-object azimuth angle θ:

$$\mathbf{\Sigma}S = \boldsymbol{R}(\theta), \mathbf{\Sigma}L, \boldsymbol{R}(\theta)^{\mathsf{T}}. \tag{4.15}$$

The resulting covariance matrix in the planar sensor frame is:

$$\mathbf{\Sigma}S = \begin{bmatrix} \sigma x^2 & \sigma_{xy}\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}. \tag{4.16}$$

This process is illustrated in Figure 4.7.

Before fusion, all detections are transformed into a common global reference frame that accounts for sensor installation offsets and vehicle pose. The transformation consists of a rotation $\boldsymbol{R}*$ and a translation $\boldsymbol{t}*$ applied to each detection $\boldsymbol{d}_i = \boldsymbol{\mu}i, \mathbf{\Sigma}S_i$:

$$\boldsymbol{\mu}G_i = \boldsymbol{R}*, \boldsymbol{\mu}i + \boldsymbol{t}* \ \ \mathbf{\Sigma}G_i \qquad\qquad = \boldsymbol{R}*, \mathbf{\Sigma}S_i, \boldsymbol{R}*^{\mathsf{T}}. \tag{4.17}$$

A spatial gating procedure is applied to identify LiDAR and RADAR measurements eligible for fusion. Gating is based either on the Mahalanobis distance, which accounts for measurement uncertainty, or on a Euclidean distance threshold. Only measurements within the selected threshold are considered compatible and fused.

Compatible RADAR and LiDAR detections are fused by combining their multivariate normal distributions, resulting in a fused mean $\boldsymbol{\mu}f_i$ and covariance $\mathbf{\Sigma}f_i$:

$$\boldsymbol{\mu} f_i = \boldsymbol{\Sigma} f_i \left(\boldsymbol{\Sigma} {G_a}^{-1} \boldsymbol{\mu} G_a + \boldsymbol{\Sigma} {G_b}^{-1} \boldsymbol{\mu} G_b \right) \boldsymbol{\Sigma} f_i \qquad = \left[\boldsymbol{\Sigma} {G_a}^{-1} + \boldsymbol{\Sigma}_{G_b}^{-1} \right]^{-1} \tag{4.18}$$

where a and b denote two generic detections in the global reference frame.

The output is a consolidated set of detections that includes both single-sensor and fused measurements. This formulation enables robust handling of heterogeneous sensor uncertainties and partial observability, providing improved localisation accuracy for subsequent tracking stages.

4.5.2 Computation of covariance matrices for individual measurements

Each sensor measurement is associated with an uncertainty modelled by the covariance matrix $\boldsymbol{\Sigma}_{S_i}$, which represents the measurement's variability in the planar sensor frame.

Measurements are characterised by radial (r) and tangential (t) components relative to the sensor, corresponding to the depth and lateral directions, respectively. Accordingly, the standard deviations $\sigma_{r,i}$ and $\sigma_{t,i}$ are defined for each detection and vary as a function of the azimuth angle and relative target position.

A basic approach assigns fixed uncertainty values based on sensor resolution, optionally scaled to reflect conservative assumptions. While simple, this method neglects important dependencies of the measurement process. A more effective strategy accounts for variations induced by target geometry and sensor configuration.

For instance, in mechanical LiDAR systems, measurements correspond to the surface of first impact and do not capture internal structure. As a consequence, the tangential position of an object centroid is typically more reliable than the radial one, which depends on object shape, size, and orientation. This effect can be accounted for by scaling the covariance with object dimensions, such as the bounding box area.

In addition to directional anisotropy, measurement uncertainty strongly depends on target distance. In mechanical LiDARs (see Section 4.8.1), beam divergence causes the spacing between points to increase with distance, as determined by the angular resolution. These effects can be characterised through simulation or experimental analysis and used to derive suitable covariance models.

A practical formulation for the radial and tangential standard deviations is:

$$\sigma_{r,i} = \alpha_r, d_i + \beta_r, A_i, \quad \sigma_{t,i} = \alpha_t, d_i + \beta_t, A_i, \tag{4.19}$$

where d_i denotes the detection distance, A_i the bounding box area, and $\alpha_r, \alpha_t, \beta_r, \beta_t$ scaling coefficients. This model captures the dominant sources of uncertainty while remaining computationally efficient.

For other sensors, such as planar RADAR, uncertainty may strongly depend on the target's angular position due to non-uniform resolution. In these cases, experimental characterisation is required. A common solution is the use of interpolation matrices or look-up tables derived from experimental data or Monte Carlo simulations. In this work, RADAR uncertainty models provided

by the project partner CNR-IREA are integrated through look-up tables to assign $\mathbf{\Sigma}_{S_i}$ to each measurement.

4.6 3D to 2D fusion: LiDAR - Optical sensors

This section presents the LiDAR–optical sensor fusion modules developed in this dissertation. The overarching goal is to leverage the complementary strengths of LiDAR and optical sensors to enhance both the accuracy and robustness of perception outputs. While optical sensors provide rich semantic and texture information that supports advanced detection and classification, their ability to estimate object position and distance is inherently limited. In contrast, LiDAR delivers precise and reliable spatial measurements but provides minimal semantic context.

Two fusion methodologies, both experimentally validated, are described. The first follows a late-fusion strategy that combines RGB camera detections with LiDAR point clouds and then performs multi-target tracking to support moving-obstacle detection in navigation tasks. The second approach implements an early-fusion strategy that integrates thermal camera data with LiDAR for human detection in disaster management and search-and-rescue scenarios.

Although each pipeline is tailored to specific sensors and operational contexts, the proposed frameworks are designed to be general and modular, allowing adaptation to alternative sensor combinations or future system upgrades without fundamental redesign.

4.6.1 LiDAR - RGB camera decision-level fusion

This approach combines RGB imagery and LiDAR measurements to achieve robust object detection and classification in dynamic marine environments. The processing pipeline, shown in Fig. 4.8, consists of two independent sensing branches integrated through a late fusion strategy.

The rationale for adopting a late-fusion strategy between LiDAR and RGB cameras stems from the need for a robust, modular, and adaptable perception system in dynamic maritime environments. One primary advantage is the resilience to single-sensor failures: since the two sensing branches operate asynchronously, spatial estimates can still be updated reliably using LiDAR alone, even when water spray, reflections, or poor visibility temporarily obstruct the camera.

Additionally, late fusion enhances the system's modularity and flexibility. Each pipeline can be developed, updated, or replaced independently without affecting the others. For instance, improvements in the camera-based classification model or modifications to the LiDAR clustering algorithm can be implemented without requiring extensive changes to the overall architecture.

Furthermore, this design leverages the complementary strengths of the sensors while mitigating their weaknesses. LiDAR provides precise spatial measurements and robustness to lighting and environmental variations, whereas the RGB camera contributes rich semantic information for object classification. Late fusion allows the system to exploit these advantages independently before combining the results, ensuring both accurate localisation and reliable classification.

Finally, the architecture supports scalability and the integration of additional sensor modalities, such as thermal cameras or radar. New sensing branches can be incorporated in parallel pipelines without compromising the existing system's decoupling and modularity, which is particularly

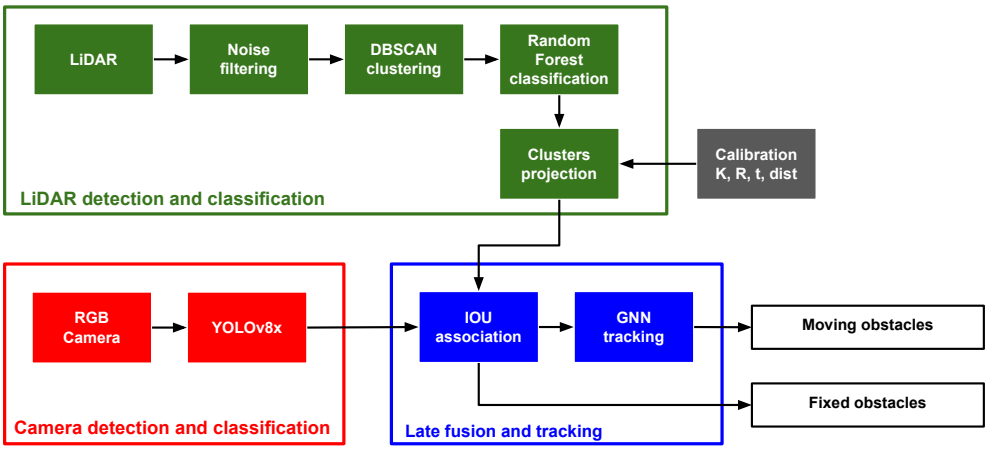


Figure 4.8: General pipeline for late fusion of LiDAR and RGB camera data.

important in research and experimental contexts where sensor configurations and algorithms must be frequently updated or tested.

The LiDAR branch follows the pipeline described in Section 4.2. After filtering spurious points, clustering is applied to generate object hypotheses, from which centroids and spatial extents are extracted using PCA-based bounding boxes. A Random Forest classifier assigns each cluster to one of four classes: Quay, Motor Boat, Sailing Boat, and Ship. This branch provides accurate spatial localisation and range estimation.

In parallel, the image branch performs one-shot object detection on RGB frames using a YOLOv8s detector [123] (see Section 4.3.1), selected for its balance between accuracy and inference speed. The network is pre-trained on the COCO dataset [124], enabling reliable detection of marine obstacles without domain-specific retraining.

Geometric alignment between sensors is achieved by projecting three-dimensional LiDAR points onto the image plane using the pinhole camera model. Extrinsic calibration parameters are derived from the sensor mounting configuration. Since the LiDAR provides a full 360° horizontal field of view, the point cloud is clipped to the camera's field of view before projection.

Sensor integration is performed via late fusion, allowing each branch to operate independently and asynchronously. This design preserves modality-specific optimisations while improving robustness to occlusions and temporary sensor failures.

Fusion is carried out in two stages. First, LiDAR- and image-based detections are represented as two-dimensional bounding boxes on the image plane, each with an associated class label and confidence score. Bounding boxes are matched using the Intersection over Union (IoU) metric, or Jaccard index J, defined as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{4.20}$$

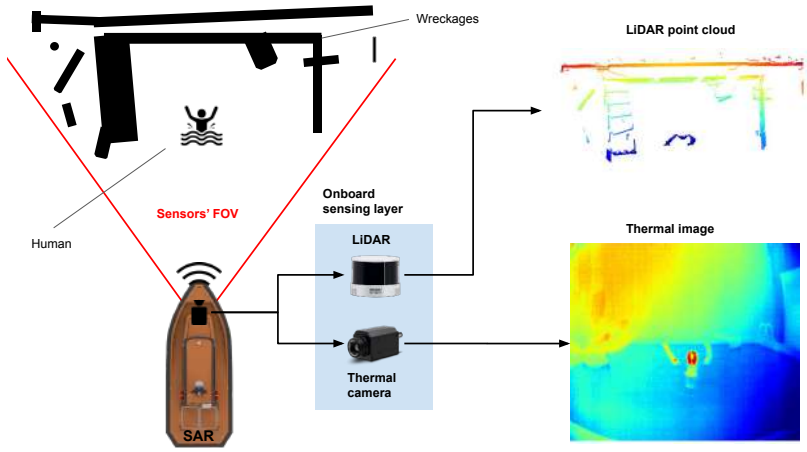


Figure 4.9: Operative Search And Rescue Scenario with sensing layer outcome.

Only matches exceeding a predefined IoU threshold are retained as candidate moving obstacles.

Each candidate obstacle is associated with two independent semantic labels, produced by the YOLO and Random Forest classifiers. The final decision can be obtained through different strategies, such as agreement-based selection, logical OR, or probabilistic fusion with weighted confidences. This flexibility allows the system to be tuned to application-specific requirements.

Confirmed obstacles are forwarded to the multi-target tracking module, which maintains temporal consistency and estimates object trajectories over time (see Section 4.7).

4.6.2 LiDAR - thermal camera data-level fusion

This pipeline implements an early fusion strategy for thermal camera and LiDAR data, developed to support autonomous search-and-rescue (SAR) operations for human detection at sea. The design is motivated by operational constraints identified during research activities at the University of Ghent.

LiDAR sensors, commonly employed on autonomous surface vehicles (ASVs) for collision avoidance, provide accurate spatial localisation but exhibit limited vertical resolution, which restricts their ability to detect and classify floating persons at long range. Thermal cameras complement LiDAR by enabling long-range, one-shot detection under low-visibility conditions, although they do not directly provide range information. Early fusion combines the detection capability of thermal imaging with the precise three-dimensional localisation provided by LiDAR.

Unlike late fusion, which associates independent detections, early fusion enables LiDAR geometric information to directly support thermal-based detection. The LiDAR point cloud is cropped to the camera field of view and projected onto the image plane using calibrated extrinsic parameters. Two LiDAR-derived images encoding reflectivity and range are generated and stacked with the thermal image to form a three-channel input for a YOLO-based detector. After detection, three-dimensional target coordinates are recovered from the original point cloud.

Figure 4.9 illustrates a representative SAR scenario.

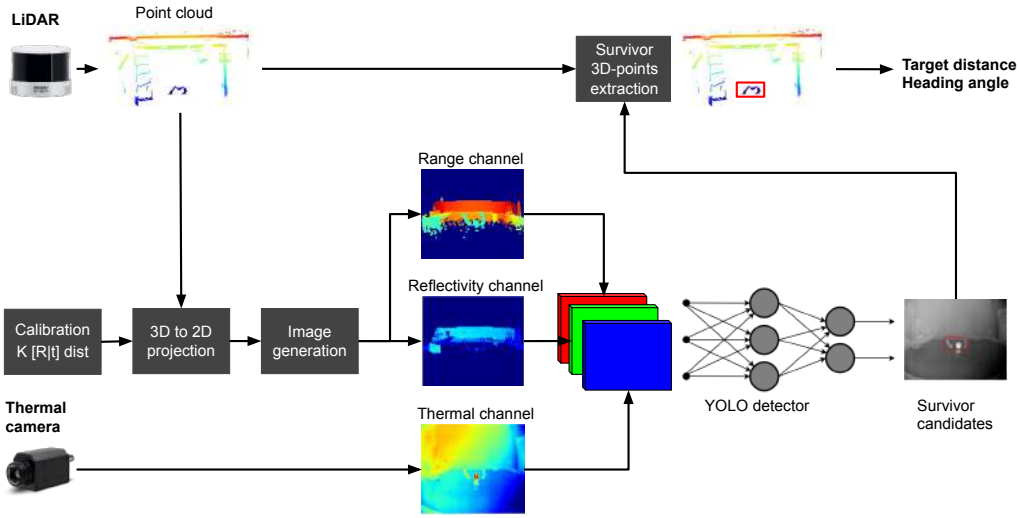


Figure 4.10: Overview of the early fusion pipeline integrating thermal and LiDAR data.

In the early fusion framework, thermal and LiDAR measurements are combined at the raw-data level prior to feature extraction, thereby preserving the spatial relationships critical to convolutional neural networks. Figure 4.10 provides an overview of the processing pipeline.

The early fusion strategy was chosen to leverage the complementary strengths of LiDAR and thermal sensors at the raw-data level. By projecting LiDAR points onto the thermal image plane and incorporating range and reflectivity information, the network benefits from precise geometric priors while exploiting the thermal contrast of targets. This enhances the detection of partially submerged humans or small objects, preserves spatial relationships for the neural network, and improves robustness in dynamic maritime conditions where a single modality alone may be ambiguous or partially occluded.

4.6.2.1 Calibration and Projection

Accurate calibration is required to align the two sensing modalities. Intrinsic and extrinsic parameters were obtained as an integral part of the experimental campaign, and results in this regard are presented in Section 5.1.

The LiDAR point cloud is cropped to the thermal camera field of view, transformed into the camera frame, and projected onto the image plane using the pinhole camera model, establishing a correspondence between three-dimensional points and thermal image pixels.

(a) Projected point cloud on the thermal image plane.

(b) Range map before interpolation.

(c) Range map after interpolation.

Figure 4.11: LiDAR field-map generation from sparse projection to dense images.

4.6.2.2 Generation of the Multi-Source Image

Two LiDAR-derived images encoding reflectivity and range are generated at the same resolution as the thermal image and normalised to the 0–255 range. Due to the sparsity of the projection, nearest-neighbour interpolation is applied within a bounded pixel radius to obtain dense maps. These images are stacked with the thermal frame to form a three-channel representation encoding temperature, material response, and distance. Figure 4.11 illustrates this process.

4.6.2.3 Detection and Classification

Human detection is performed directly on the three-channel image using a YOLOv8 convolutional neural network. The network supports three-channel input without architectural modifications and is retrained from scratch on a dedicated thermal–LiDAR dataset augmented with diverse environmental conditions (see Section 5.1).

4.6.2.4 Rescue Point Extraction

The detector outputs image-plane bounding boxes for each detected survivor. Each bounding box is associated with its corresponding subset of LiDAR points via the established projection, eliminating the need for explicit 3D clustering. When background structures interfere, range variance analysis and adaptive thresholding are applied to isolate the human cluster. From the final cluster, the three-dimensional position, relative distance, and bearing of the survivor are estimated and forwarded to navigation and path-planning modules, enabling timely and accurate rescue actions [125].

4.7 Multi-target tracking

Tracking estimates and maintains the state of objects over time from noisy measurements. In the maritime domain, it enables autonomous surface vessels to predict surrounding traffic for collision avoidance and trajectory planning. In the proposed framework, each vessel is represented by its centroid as a point target on the sea plane. The state is limited to planar position and heading, ensuring computational efficiency while remaining adequate for surface navigation.

The multi-target tracking (MTT) module developed in this work is modular and sensor-agnostic. It is implemented within the StoneSoup Python framework [126], allowing flexible configuration of the tracking components and seamless integration with heterogeneous perception pipelines. The tracker operates on generic detection lists, independently of the sensing modality.

The tracker follows a standard MTT architecture comprising detection preprocessing, prediction, data association, state update, and track management. Each component is described below.

Gating Gating limits the data-association space by rejecting detections inconsistent with predicted track states. Elliptical gates are defined using the Mahalanobis distance:

$$d_M^2(\mathbf{z}, \hat{\mathbf{z}}) = (\mathbf{z} - \hat{\mathbf{z}})^T \mathbf{S}^{-1} (\mathbf{z} - \hat{\mathbf{z}}), \tag{4.21}$$

where $\mathbf{z}$ is the measurement, $\hat{\mathbf{z}}$ the predicted measurement, and $\mathbf{S}$ the innovation covariance. This step reduces computational load and improves robustness in cluttered environments.

Prediction Track states are propagated using a Kalman Filter (KF) with a kinematic motion model, yielding predicted states and covariances at the next time step. This prediction supports anticipatory decision-making in dynamic maritime scenarios.

Update When detections fall within the gate of a predicted track, the Kalman correction step fuses predictions and measurements, refining the state estimate and reducing uncertainty.

Data Association Detection-to-track association is performed using the Global Nearest Neighbour (GNN) algorithm. Association costs are defined using the squared Mahalanobis distance, and the global assignment minimising the total cost is computed as:

$$\min_{\mathcal{A}} \sum_{(i,j)\in\mathcal{A}} C_{ij}, \tag{4.22}$$

subject to one-to-one assignment constraints. The resulting linear assignment problem is solved via the Hungarian algorithm [127]. Gating is applied before assignment to discard implausible hypotheses.

The selection of the Global Nearest Neighbour (GNN) algorithm for data association reflects a deliberate trade-off between computational efficiency and tracking performance. In the proposed modular framework, each tracking component—including gating, prediction, and association—can

be easily swapped or reconfigured without altering the overall architecture, enabling flexible adaptation to different sensors or operational requirements. While alternative association methods (e.g., Joint Probabilistic Data Association or Multiple Hypothesis Tracking) are available and can be tuned for specific scenarios, GNN provides a robust and interpretable default. This choice ensures a favourable balance between accuracy and computational cost, while maintaining the modularity necessary to adapt the pipeline to future algorithmic improvements or scenario-specific requirements.

Track Management Track management controls track initiation, confirmation, and deletion. New tracks are initialised from unassociated detections and confirmed after a minimum number of consistent updates. Tracks are deleted when the trace of the KF covariance matrix exceeds a predefined threshold. Confirmation and deletion parameters can be tuned to balance sensitivity and robustness.

When multimodal classification is available, redundant semantic information can be exploited in the confirmation logic. Permissive or agreement-based strategies enable the tracker to adapt to varying operational requirements and sensor configurations.

4.8 INS-GNSS based state estimation feedback

To define the vessel state, two reference frames are introduced:

- Earth-fixed frame **b**-frame (O_n, $\underline{n}_1$, $\underline{n}_2$, $\underline{n}_3$): A local reference frame compliant with NED convention centered at a convenience reference point O_n; $\underline{n}_1$ and $\underline{n}_2$ aligned with North and East respectively, while $\underline{n}_3$ points downward.
- Body-fixed frame **b**-frame (O_b, $\underline{b}_1$, $\underline{b}_2$, $\underline{b}_3$): A vessel-fixed reference frame, compliant with the NED convention, ideally located at midship ($l_{pp}/2$), in which onboard sensor measurements are expressed. The axis $\underline{b}_1$ is aligned with the aft direction, $\underline{b}_2$ with starboard, and $\underline{b}_3$ points downward.

For computational efficiency, the vessel is modelled with three degrees of freedom (3-DOF) in the Earth-fixed frame. The generalised position consists of planar coordinates (x, y) and heading angle ψ, defined as the angle between $\underline{n}_1$ and $\underline{b}_1$. Vertical motion is neglected, and $\underline{b}_3$ is assumed parallel to $\underline{n}_3$. The rotation matrix relating the body-fixed and Earth-fixed frames therefore, depends only on ψ:

$$R_\psi = \begin{bmatrix} \cos(\psi) & -\sin\psi & 0 \\ \sin(\psi) & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.23}$$

Figure 4.12 shows a visual reference of this concept.

The state is augmented with the corresponding generalized velocities (v_x, v_y, r) and, when required, generalized accelerations $(a_x, a_y, \dot{r})$.

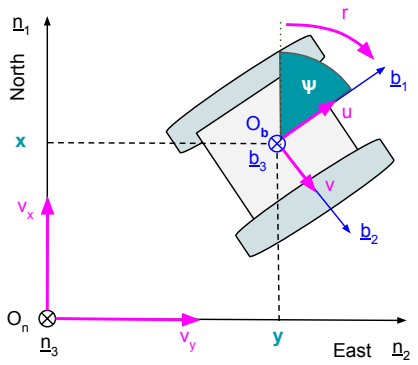


Figure 4.12: 3-DOF reference systems.

Considering that the sensors used are the GNSS receiver and an inertial measurement system, the raw measurements available at any given moment are summarised in Table 4.1.

Table 4.1: Useful measures extracted from the INS-GNSS sensors to feed the state estimation module among the available ones.

Sensor	**Measure**	**Symbol**
GNSS	Latitude	Φ
	Longitude	λ
Magnetometer	Magnetic intensity along b_1	B_1
	Magnetic intensity along b_2	B_2
Gyro	Angular speed around b_3 direction	r
Accelerometer	Linear acceleration along b_1	$\dot{u}$
	Linear acceleration along b_2	$\dot{v}$

Magnetic field and linear acceleration measurements are expressed in the body-fixed reference frame, while the yaw rate provided by the gyroscope is frame-invariant under the adopted 3-DOF representation.

GNSS latitude and Longitude are projected into planar metric coordinates using the Universal Transverse Mercator (UTM) system [128]. The resulting coordinates are consistent with the NED convention and are translated to a local Earth-fixed frame centred at a reference point $\boldsymbol{O}$:

$$x(*),\ y(*) = \text{UTM}(\ \Phi(*),\ \lambda(*)\ ) - \text{UTM}(\ \Phi(\boldsymbol{O}),\ \lambda(\boldsymbol{O})\ ). \tag{4.24}$$

The vessel heading ψ is obtained from magnetometer measurements by computing the arctangent of the planar magnetic field components:

$$\psi = \arctan 2(B_2, B_1), \tag{4.25}$$

Linear accelerations measured in the body frame are transformed into the Earth-fixed frame as:

$$\begin{bmatrix} a_x \\ a_y \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\sin\psi \\ \sin(\psi) & \cos\psi \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} \tag{4.26}$$

Although raw measurements can be forwarded directly to downstream modules, this approach is suboptimal: measurements remain unfiltered and sensitive to outliers, unobserved states (e.g., velocities and angular acceleration) cannot be reliably estimated, and multisensor integration is impractical. For these reasons, advanced filtering techniques are employed to fuse proprioceptive measurements and provide a consistent, robust state estimate, as detailed in the following sections.

4.8.1 Fast and effective: Kalman Filter

As an initial filtering solution for the feedback system, a Kalman Filter (KF) was implemented [129], providing a good compromise between simplicity and estimation accuracy.

4.8.1.1 Working principles

The filter follows the standard predict–update cycle: during prediction, the state and its covariance are propagated using the process model, while the update step corrects the prediction using incoming measurements and their uncertainties. The filter is uncontrolled, i.e., no external control input is considered. The state and measurement models are defined as:

$$\boldsymbol{x}_k = \boldsymbol{F}, \boldsymbol{x}_{k-1} + \boldsymbol{w}_{k-1}, \tag{4.27}$$

$$\boldsymbol{z}_k = \boldsymbol{H}, \boldsymbol{x}_k + \boldsymbol{v}_k, \tag{4.28}$$

where $\boldsymbol{w}k-1 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$ and $\boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$ denote process and measurement noise, respectively.

The prediction step computes the prior state estimate and covariance:

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{F}, \hat{\boldsymbol{x}}_{k-1|k-1}, \tag{4.29}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}, \boldsymbol{P}_{k-1|k-1}, \boldsymbol{F}^\top + \boldsymbol{Q}. \tag{4.30}$$

When a new measurement is available, the Kalman gain is computed, and the estimate is updated:

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\, \boldsymbol{H}^\top \big(\boldsymbol{H}\, \boldsymbol{P}_{k|k-1}\, \boldsymbol{H}^\top + \boldsymbol{R}\big)^{-1}, \tag{4.31}$$

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \big(\boldsymbol{z}_k - \boldsymbol{H}\, \hat{\boldsymbol{x}}_{k|k-1}\big), \tag{4.32}$$

$$\boldsymbol{P}_{k|k} = \big(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}\big) \boldsymbol{P}_{k|k-1}. \tag{4.33}$$

The Kalman gain balances model prediction and measurement confidence: larger measurement noise reduces the influence of the measurement, while smaller noise increases it.

4.8.1.2 Implementation

A purely kinematic constant-velocity model is adopted. The state and measurement vectors are defined as:

$$\boldsymbol{x}_k = \left[x, y, \psi, v_x, v_y, r\right]^T \tag{4.34}$$

$$z_k = \left[x, y, \psi, r\right]^T \tag{4.35}$$

Here, (x, y) are the planar GNSS position, ψ is the heading angle obtained from the magnetometer (Eq. (4.25)), and r is the yaw rate measured by the gyroscope. This formulation enables the KF to estimate the full vessel state $(x, y, \psi, v_x, v_y, r)$ at a high rate, providing suitable feedback for guidance and control.

Although the onboard INS provides linear acceleration measurements (Table 4.1), these are expressed in the body-fixed frame and would introduce nonlinearities if transformed into the Earth-fixed frame (Eq. (4.26)), violating the assumptions of the Kalman Filter. Moreover, for small vessels, such measurements are strongly affected by environmental disturbances and sensor noise, often degrading estimation performance. For these reasons, accelerations are not included in the filter.

The constant-velocity KF therefore represents an effective and computationally efficient solution for experimental validation with perception-driven systems. A known limitation concerns the periodic nature of the heading angle ψ, which introduces a discontinuity at the wrap-around boundary. To avoid spurious innovations, the innovation term is computed using the minimal angular difference. A more general and elegant treatment, applicable when the state includes periodic variables, is presented in the following section.

4.8.2 Nonlinear approach: Unscented Kalman Filter

The second approach adopted for state estimation was the use of an Unscented Kalman Filter (UKF) to better handle the nonlinearities in the system model. In many cases, an Extended Kalman Filter (EKF) is used, which linearises the model via a first-order Taylor expansion and requires the computation of Jacobian matrices for the state transition and measurement functions.

While the EKF approximates the nonlinear transformation locally and may introduce errors when nonlinearities are significant, or the state covariance is large, the UKF propagates a set of carefully chosen *sigma points* through the nonlinear functions to compute the mean and covariance. This provides a second-order accurate approximation without the need for Jacobians.

For these reasons, the UKF is particularly suitable for systems with compact vehicles and complex dynamics, such as perception-based experimental setups, where the model includes nonlinear combinations of angles, velocities, and accelerations. It delivers more accurate and robust state estimates than the EKF, with a comparable computational cost and reduced risk of

instability.

4.8.2.1 Working principles

The Unscented Kalman Filter (UKF) relies on the Unscented Transform to propagate mean and covariance through nonlinear functions. The UKF uses a set of carefully chosen sample points, called *sigma points*, which capture the first two moments (mean and covariance) of the state distribution.

Given a state vector $\boldsymbol{x}_{k-1}$ of dimension n with mean $\hat{\boldsymbol{x}}_{k-1|k-1}$ and covariance $\boldsymbol{P}_{k-1|k-1}$, $2n+1$ sigma points $\boldsymbol{\chi}_{k-1}^{(i)}$ are generated as:

$$\boldsymbol{\chi}_{k-1}^{(0)} = \hat{\boldsymbol{x}}_{k-1|k-1}, \tag{4.36}$$

$$\boldsymbol{\chi}_{k-1}^{(i)} = \hat{\boldsymbol{x}}_{k-1|k-1} + (\sqrt{(n+\lambda)\boldsymbol{P}_{k-1|k-1}})_i, \quad i = 1, \ldots, n, \tag{4.37}$$

$$\boldsymbol{\chi}_{k-1}^{(i)} = \hat{\boldsymbol{x}}_{k-1|k-1} - (\sqrt{(n+\lambda)\boldsymbol{P}_{k-1|k-1}})_{i-n}, \quad i = n+1, \ldots, 2n, \tag{4.38}$$

where λ is a scaling parameter and $\sqrt{\cdot}$ denotes a matrix square root, typically computed via Cholesky decomposition.

These sigma points are then propagated through the nonlinear process function $\boldsymbol{f}(\cdot)$ to obtain the predicted sigma points:

$$\boldsymbol{\chi}_{k|k-1}^{(i)} = \boldsymbol{f}(\boldsymbol{\chi}_{k-1}^{(i)}), \quad i = 0, \ldots, 2n. \tag{4.39}$$

The predicted state mean and covariance are reconstructed as weighted sums of the propagated sigma points:

$$\hat{\boldsymbol{x}}_{k|k-1} = \sum_{i=0}^{2n} W_m^{(i)} \boldsymbol{\chi}_{k|k-1}^{(i)}, \tag{4.40}$$

$$\boldsymbol{P}_{k|k-1} = \sum_{i=0}^{2n} W_c^{(i)} \left(\boldsymbol{\chi}_{k|k-1}^{(i)} - \hat{\boldsymbol{x}}_{k|k-1}\right)\left(\boldsymbol{\chi}_{k|k-1}^{(i)} - \hat{\boldsymbol{x}}_{k|k-1}\right)^\top + \boldsymbol{Q}, \tag{4.41}$$

where $W_m^{(i)}$ and $W_c^{(i)}$ are the weights for the mean and covariance, respectively, and $\boldsymbol{Q}$ is the process noise covariance.

The measurement update follows a similar procedure, where the predicted sigma points are propagated through the measurement function $\boldsymbol{h}(\cdot)$ to compute the predicted measurement mean, covariance, and cross-covariance. The resulting Kalman gain is then used to update the state estimate and covariance, just as in the standard Kalman filter equations.

4.8.2.2 Heading angle

One use of UKF in the development of the feedback system was to manage heading-angle discontinuities. In fact, instead of managing the heading angle directly, it is possible to filter its components in terms of sine and cosine, which removes the discontinuity at the $[-180°, 180°]$ boundary and allows for a continuous representation suitable for nonlinear state estimation.

$$c = \cos(\psi) = \frac{B_x}{|\boldsymbol{B}|} \tag{4.42}$$

$$s = \sin(\psi) = \frac{B_y}{|\boldsymbol{B}|} \tag{4.43}$$

$$\boldsymbol{B} = \left[B_x, B_y\right] \tag{4.44}$$

This representation is particularly compatible with the magnetometer measurements as shown in Eq. (4.25). The planar components of the magnetic field measured by the sensor can be directly associated with the sine and cosine of the heading angle, effectively embedding the magnetic field into the vehicle state and measuring vectors:

$$\boldsymbol{x}_\mathrm{k} = \left[x, y, c, s, v_x, v_y, r\right] \tag{4.45}$$

$$\boldsymbol{z}_\mathrm{k} = \left[x, y, c, s, r\right] \tag{4.46}$$

$$\tag{4.47}$$

In this case, the state transition function for the sine and cosine components includes the nonlinearity, while the rest of the model remains linear:

$$c_{k+1} = c_k \cos(r_k \Delta t) - s_k \sin(r_k \Delta t) \tag{4.48}$$

$$s_{k+1} = s_k \cos(r_k \Delta t) + c_k \sin(r_k \Delta t) \tag{4.49}$$

$$\tag{4.50}$$

The heading angle ψ can then be recovered a posteriori from the estimated sine and cosine values, or its component can be used directly by the other modules.

By employing the UKF with this representation, the state-transition model can naturally incorporate nonlinear relationships involving ψ, while maintaining a consistent, continuous estimate of the heading.

4.8.2.3 Accelerations

A second application of the UKF during the development phase was for handling accelerations. In particular, the accelerations measured by the accelerometer are expressed in the body frame (b-frame), while the system state is represented in the earth-fixed frame. This requires rotating the accelerations from the body frame to the earth frame using Eq. 4.26, thereby introducing a nonlinearity in the measurement model.

In this context, the system kinematics are modelled using a constant-acceleration model, in which both the state and the measurements are described by position, velocity, and acceleration in the earth-fixed frame.

$$\boldsymbol{x}_\mathrm{k} = \left[x, y, \psi, v_x, v_y, r, a_x, a_y, \dot{r}\right] \tag{4.51}$$

$$\boldsymbol{z}_\mathrm{k} = \left[x, y, \psi, r, \dot{u}, \dot{v}\right] \tag{4.52}$$

$$\tag{4.53}$$

The evolution of the state from step k to step $k+1$ is linear, except when variables such as angles derived directly from the magnetic field are explicitly included. The main nonlinearity instead resides in the measurement function $\boldsymbol{h}(\cdot)$ of the UKF, which maps the state to the measurement space, which embeds:

$$\begin{bmatrix}\dot{u}\\ \dot{v}\end{bmatrix} = \boldsymbol{R}_\psi^{-1}\begin{bmatrix}a_x\\ a_y\end{bmatrix} \tag{4.54}$$

4.8.2.4 Alternative Nonlinear Motion Models

While the UKF has been presented so far with constant-velocity or constant-acceleration state transitions, it is fully compatible with more sophisticated nonlinear motion models, such as the Constant Turn rate (CT) model. These models explicitly account for the vehicle's rotational dynamics, allowing the state to evolve along curved trajectories and capturing effects such as coordinated turns and rapid heading changes, which are common in maritime navigation and SAR manoeuvres. By embedding such models within the UKF framework, the sigma points propagate through the nonlinear motion equations, naturally accommodating the vehicle's kinematics without linear approximations. This flexibility enhances prediction accuracy in scenarios involving sharp manoeuvres or complex trajectories, while maintaining the same UKF-based update mechanism for integrating measurements from GNSS, IMU, and other sensors.

4.9 Fiducial Marker Detection

In this thesis, optical sensors were primarily used as a complementary modality or to enhance other detection pipelines by extracting semantic features. However, a distinct contribution of this work is the development of a complete indoor positioning system leveraging a single optical sensor in combination with fiducial markers.

Until early 2023, the COMPASS Lab test tank lacked any reliable positioning system. As a result, although various experimental trials could be performed, it was not possible to fully close the loop for guidance and control algorithms due to the absence of accurate position and velocity feedback. Additionally, the lack of satellite visibility precludes the use of GNSS receivers in indoor facilities. To address these limitations, and as part of the doctoral research, a complete indoor positioning system was designed and implemented from the ground up.

As an initial step, the test tank (Figure 4.13) was equipped with a monochrome camera mounted

on the roof (see Section 3.4.2 for further details). The IPS pipeline is summarised in Figure 4.14.

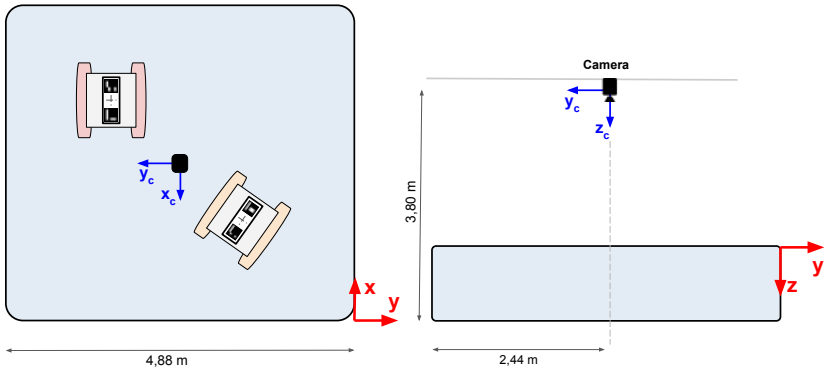


Figure 4.13: COMPASS lab test tank scheme.

The proposed system relies on ArUco-type fiducial markers [130], which can be reliably and

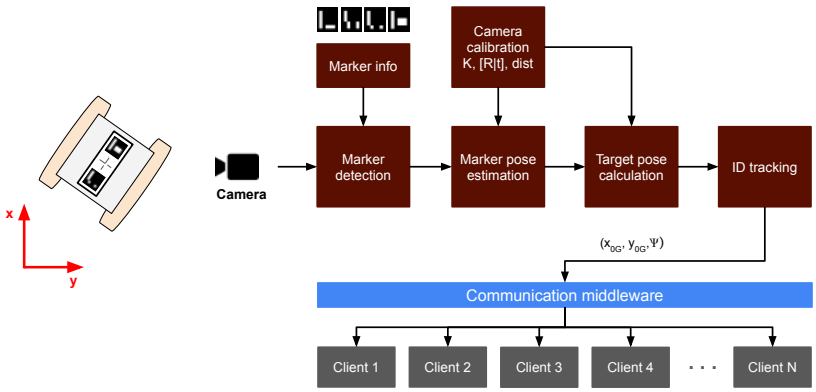
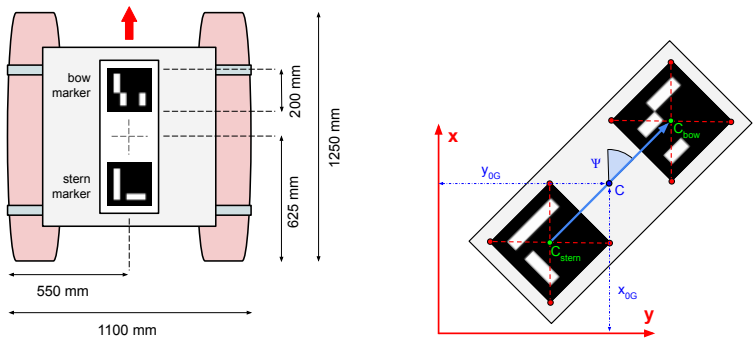


Figure 4.14: Indoor Positioning System pipeline.

unambiguously identified in images through dedicated detection algorithms. The overall estimation process is formulated within the pinhole camera model, with calibration parameters determined using Zhang's method.

Each vehicle is equipped with a dedicated marker board containing two fiducial markers of known size, position, and identifier, as illustrated in Figure 4.15a. One marker is placed at the bow and the other at the stern, thereby increasing the number of detectable points and improving the accuracy of the heading estimation.



(a) Fiducial marker installation on SWAMP vehicle. **(b)** Marker-board detection and pose estimation.

Figure 4.15: Fiducial marker on target vehicle.

A monochrome overhead camera, mounted on the roof of the facility, acquires images of the test area (see Figure 4.13). In each frame, the four corners of the two fiducial markers are detected, yielding eight reference points per vehicle: four associated with the bow marker and four with the stern marker. These image points are then projected into the sensor reference frame by solving the Perspective-n-Point (PnP) problem [131], and subsequently transformed into the Earth-fixed (laboratory) frame using the known camera pose.

To further refine the estimates, the centroid of each marker is computed by averaging the coordinates of its four corners, resulting in the positions C_{stern} and C_{bow}. Although this step is not strictly required, it helps reduce the influence of small detection errors. The vehicle position C is then defined as the midpoint of the segment joining C_{stern} and C_{bow}, while the heading angle Ψ is obtained from the relative displacement between the bow and stern centres.

Figure 4.15b provides a schematic representation of this process: the marker corners are highlighted in red, the centroids of the bow and stern markers in green, and the estimated vehicle centre in blue, together with its orientation in the Earth-fixed reference frame.

Since the identifiers of the markers on each board are predefined, the bow and stern can be reliably distinguished, and the vehicle itself can be uniquely recognised in multi-target scenarios. This significantly simplifies the multi-target tracking process. At this point, the data can be used as is or filtered according to the application and requirements.

5. Experimental campaign

This chapter describes the organisation and execution of the experimental campaigns from a methodological perspective. The chapter is structured into two main sections: Section 5.1 presents the acquisition of the main dataset used to validate the proposed pipeline on real-world data, as described in the methodology chapter. Section 5.2 describes the experimental trials conducted during the research, following a progressive increase in complexity. The chapter starts with indoor experimental trials, proceeds to lake tests, and up to sea trials. In this context, the focus is on deploying the proposed method on an autonomous vessel platform as part of a multidisciplinary enabling-technology framework.

5.1 Dataset acquisition campaigns

5.1.1 LiDAR

This section describes the acquisition and preprocessing of the primary LiDAR dataset used in this research. The dataset comprises a series of LiDAR acquisitions collected during several field experiments conducted between late 2021 and early 2024.

The acquisition mode has been streamlined to capture a wide range of fixed, floating, and dynamic objects from diverse fixed and dynamic viewpoints. Captured targets include ships, yachts, sailboats, multihulls, powerboats, dinghies, workboats, inflatable boats, buoys, docks, and quays. Depending on the acquisition set, the Lidar was mounted on a fixed tripod on a dock, on a small motorboat, or on a medium-sized motor vessel.

The acquisition spots are selected in specific locations along the Italian coastline. In particular, Several port acquisitions are performed in the Gulf of La Spezia (SP), Liguria, from a fixed location on the quay; in this case, the LiDAR is mounted on a fixed tripod. Additional acquisitions in the same gulf are performed using the same setup mounted on a small open motorboat, moored in a marina. Blue water acquisitions are performed in the Gulf of Augusta (SR), Sicily, by mounting the LiDAR sensor on the antenna mast of a 20-m-long motor vessel.

Table 5.1 shows further details of the acquisition spots. Please note that the provided approximate height is the sensor's height relative to the average waterline.

The acquisitions were carried out in daylight under generally good weather conditions. These conditions are summarised in Table 5.2 for each location, reporting significant wave height and Visibility levels categorised on a [0-9] scale according to the World Meteorological Organisation (WMO) [132].

The spots covered during the acquisition campaign are shown in the maps provided in Figure 5.1 and Figure 5.2. Locations can be referenced using their identifiers in Table 5.1.

The dataset is divided into two sub-repositories: the *Raw* repository and the *Annotated* repository. The *Raw* repository contains raw data useful for obstacle detection, tracking, and

Table 5.1: Acquisition spots information.

ID	Location Name	City	Location	Sensor mounting	Approx. height
1	Molo Italia lighthouse	La Spezia (SP)	44.104367 N, 9.833815 E	Fixed,tripod	2 m
2	Thaon di Revel quay	La Spezia (SP).	44.103966 N, 9.826763 E	Fixed,tripod	2 m
3	Porto Mirabello	La Spezia (SP)	from 44.099176 N, 9.830839 E	Moving vessel, tripod	1.5 m
4	Porto Lotti	La Spezia (SP)	44.096126 N, 9.860641 E	Fixed, tripod	2 m
5	Porto Lotti	La Spezia (SP)	44.094823 N, 9.857200 E up to 44.100641 N, 9.833017 E	Fixed, tripod	2 m
6	Augusta quay	Augusta (SR)	37.220951 N, 15.193242 E	Moving vessel, mast	6 m
7	Augusta deep waters	Augusta (SR)	37.195500 N, 15.221402 E	Moving vessel, mast	6 m
8	Augusta deep waters	Augusta (SR)	37.209011 N, 15.236037 E	Moving vessel, mast	6 m

Table 5.2: Acquisition weather information.

ID	Significant wave height [m]	Visibility
1	<0.1	8
2	<0.1	8
3	<0.1	8
4	0.1 - 0.5	8
5	<0.1	8
6	<0.1	7
7	0.1 - 0.5	7
8	0.1 - 0.5	7

testing. It consists of several recordings divided into 4 categories:

- *quay_static*: fixed obstacles, such as docks and moored boats, captured from a stationary position. It contains scans acquired from spots 1, 2 and 5.
- *quay_dynamic*: dynamic obstacles, such as vessels sailing in near-coastal areas, captured from a stationary position. It contains a scan acquired from spot 4.
- *marina*: fixed obstacles, such as docks and moored boats, captured from a vessel sailing inside a marina. It contains scans acquired from spots 3 and 6.
- *blue_water*: static and dynamic obstacles acquired from a vessel sailing in blue water. It contains scans acquired from spots 7 and 8, as well as some from spot 6.

The *Annotated* folder contains a sample series of LiDAR targets labelled in 4 classes. This part of the dataset aims to support the development and testing of object detectors and classifiers.

5.1.1.1 Data presentation

Each category of *raw* data is populated by several acquisition series, and each acquisition consists of multiple scans. The raw LiDAR scans are stored as floating-point binary files to improve efficiency. Each point is stored with its (x, y, z) coordinate and an additional reflectance value (ref), producing an n by 4 matrix where n is the number of points. The datasets report 31 different scenarios, totalling 11680 scans recorded at 5 Hz. The scenario and scans are divided into the four

(a) Spots in the West side of the gulf.

(b) Spots in the Eest side of the gulf.

Figure 5.1: Acquisition spots in the La Spezia gulf.

Figure 5.2: Acquisition spots in Augusta bay.

batches as reported in Table 5.3 The characteristics of the scans, within each batch, are presented as histograms, as shown in Figure 5.3. The data offers valuable insights for interpreting and utilising the scans. For instance, acquisitions in *quay_static* and *marina* scenarios tend to be more densely populated. Scans from *marina* scenarios contain objects at greater distances compared to those in other scenarios, whereas acquisitions in *blue_water* scenarios are characterised by lower average reflectivity values.

Four representative sample scans are shown to provide an overview of the point clouds contained in the four batches. The scans were parsed using the provided dataset scripts, projected into Bird's Eye View (BEV), and coloured by reflectivity. Specifically, Figure 5.4a shows a scan from the *quay_static* batch (*scenario_1, 75.bin*), depicting a marina with static objects acquired from a fixed position on a quay. Figure 5.4b illustrates a dynamic scene from the *quay_dynamic* batch (*scenario_12, 100.bin*), showing two moving boats acquired from a stationary sensor mounted on a quay. Figure 5.4c presents a scene from the *marina* batch (*scenario_1, 10.bin*), representing multiple moored boats acquired from a boat navigating within a harbour area. Figure 5.4d displays a scan from the *blue_water* batch (*scenario_3, 75.bin*), showing the point-cloud of a ship captured

Table 5.3: Raw data scenarios.

Batch	Scenarios	Total scan	Total time [s]
quay_static	4	260	52
quay_dynamic	15	3070	614
marina	5	5700	1140
blue_water	7	2650	530

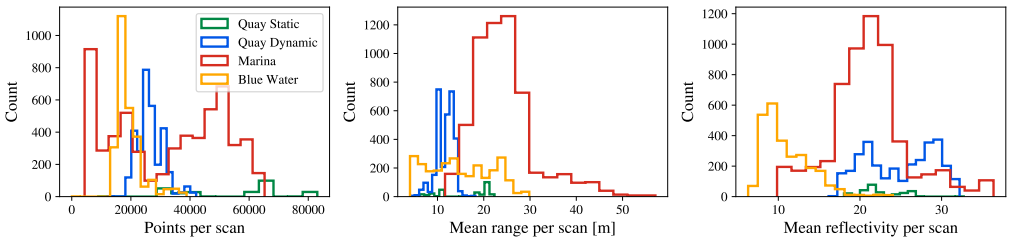


Figure 5.3: Distribution of the general features across the raw scans.

from a vessel navigating in open sea.

A subset of 300 frames is selected from the *raw* dataset, covering a wide variety of marine obstacles. Individual targets are extracted through a semi-automatic labelling procedure combining manual inspection and clustering. Candidate objects are first identified by visualising the point cloud in four views: the three orthogonal planes (XY, XZ, YZ) and a 3D perspective. For each target, a Region of Interest (ROI) is defined to spatially isolate the object.

When necessary due to point density or scene complexity, DBSCAN clustering [41] is applied within the ROI to separate the target from surrounding noise or nearby structures. The selected cluster is then extracted and manually refined by removing spurious points, retaining only the object of interest. The resulting clean point cloud is saved to a separate file.

This procedure yields a curated dataset of 926 labelled samples. Each sample is assigned to a semantic class, as summarised in Table 5.4. Annotations are performed by a designated annotator and independently verified by a second team member.

The dataset is organised into four classes:

- *Quay*: point clouds representing docks, piers, concrete structures, and harbour infrastructure.
- *Motor Boat*: point clouds of small motorised vessels (typically < 24 m), including motorboats, dinghies, Rigid-hull Inflatable Boats (RIBs), and small pilot boats.
- *Sailing Boat*: point clouds of vessels equipped with one or more masts for sail deployment, whether sails are raised or lowered.
- *Ship*: point clouds of large vessels, generally exceeding 24 m in length.

The *annotated* folder contains point clouds of individual objects stored in JSON format. Each file includes the x, y, and z coordinates and the reflectivity value, represented as lists of n

Table 5.4: Class division of the annotated dataset.

Class	Members
Quay	238
Motor Boat	313
Sailing Boat	221
Ship	148

floating-point values, where *n* denotes the number of points. A `class` field specifies the object category.

This single-object, labelled representation is adopted for generality and flexibility. From these point clouds, bounding boxes, pillars, or statistical descriptors can be readily derived. Moreover, multiple objects can be imported, translated, and rotated to synthesise virtual scenes. The dataset structure also facilitates data augmentation, enabling scaling, density variations, aspect-ratio modifications, and other object-level transformations.

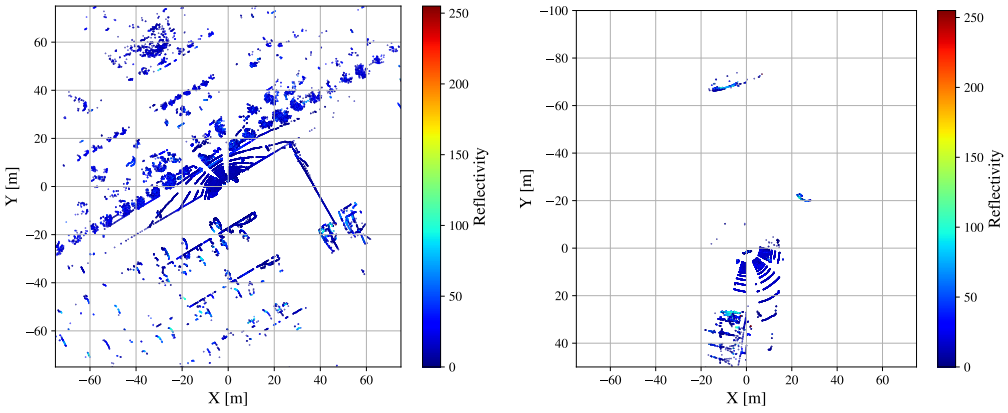
Figure 5.5a–5.5d show four representative samples from the annotated dataset: a quay (*120.json*), a motor boat (*839.json*), a sailing boat (*704.json*), and a ship (*3.json*). The displayed point clouds are fully isolated, with no background, and points are colored according to reflectivity.

An overview of the main target characteristics is shown in Figure 5.6 using histograms to highlight intra-class trends. The *Ship* class exhibits the highest average point count and acquisition distance, mainly due to larger object size and increased safety distances during data collection.

In terms of reflectivity, *Sailing Boat* and *Motor Boat* show similar average values, consistent with comparable hull geometries and the widespread use of composite materials. *Ships* generally present higher reflectivity, reflecting their larger, flatter, and predominantly metallic surfaces. Conversely, *Quays* display lower average reflectivity, largely attributable to the prevalence of raw concrete structures in the dataset.

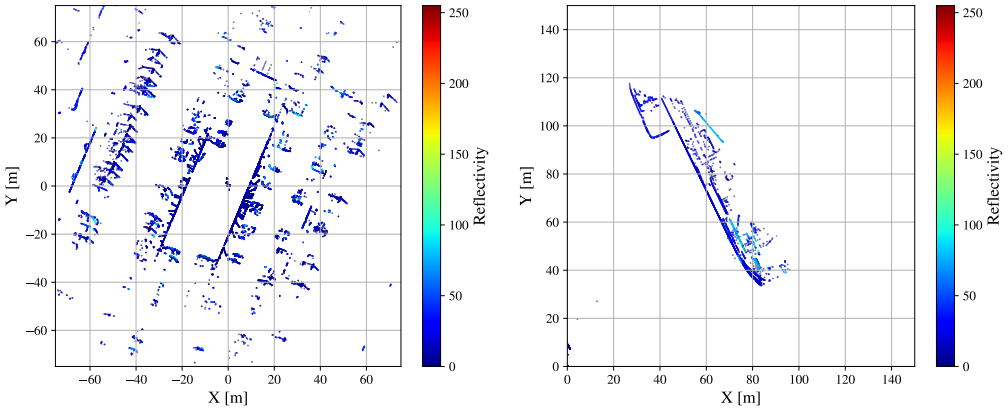
5.1.1.2 Data Availability

The dataset is publicly available online via *Zenodo* under the title ”ARNOLD – Annotated Repository of Navigational Obstacles from LiDAR Data” at `https://doi.org/10.5281/zenodo.15270275`, together with the parsing codes used in this study. The dataset and accompanying code are distributed under the Creative Commons Attribution Non-Commercial No Derivatives 4.0 International (CC BY-NC-ND 4.0) license, allowing reuse and redistribution with proper attribution.



(a) *quay_static* batch.

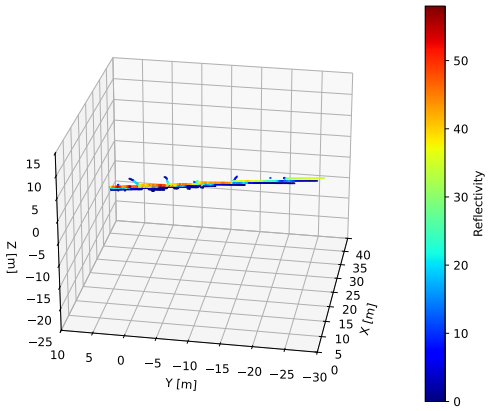
(b) *quay_dynamic* batch.



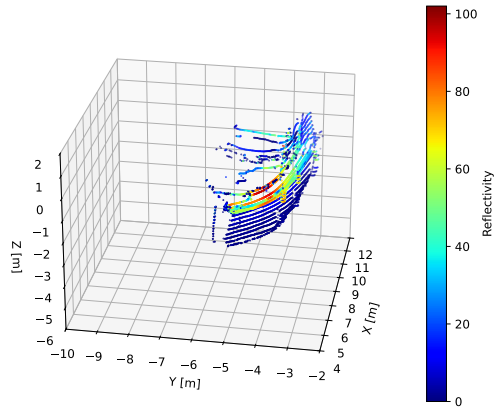
(c) *marina* batch.

(d) *blue_water* batch.

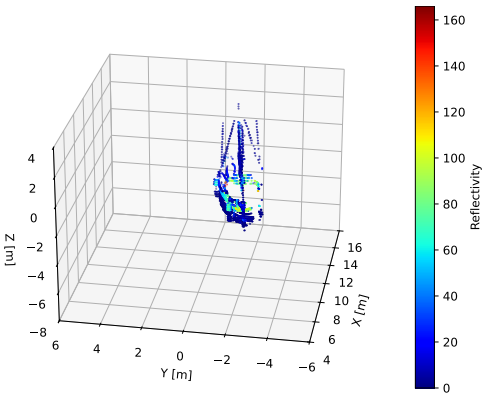
Figure 5.4: Bird's Eye View of scan from the dataset's batches.



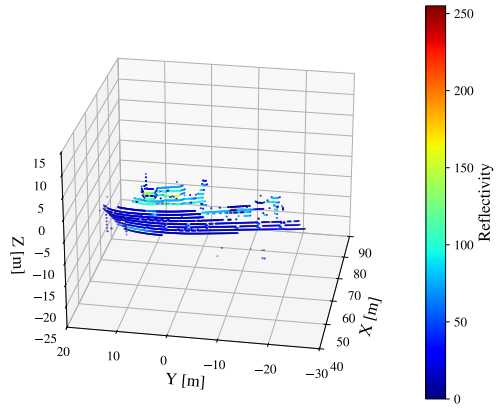
(a) Quay sample.



(b) Motor boat sample.



(c) Sailing boat sample.



(d) Ship sample

Figure 5.5: 3d point-cloud of the class members colored by reflectivity

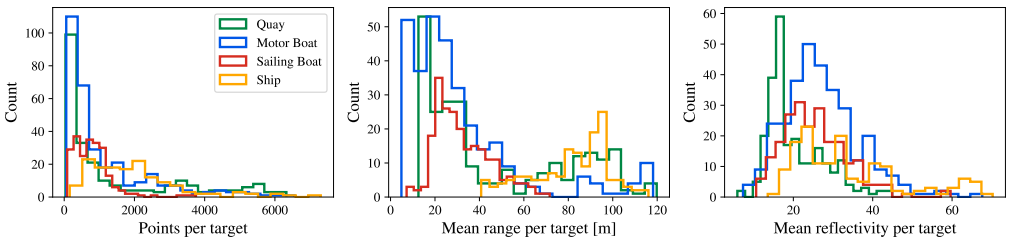


Figure 5.6: Distribution of the general features across the classes.

5.1.2 Human in water multi-modal dataset

A major limitation of maritime computer vision is the scarcity of open-access data, particularly for specialised applications such as the one considered here. To mitigate this issue, several experimental acquisition sessions were conducted using the setup described in Section 5.1.2.1. Data were collected in a controlled indoor environment at the COMPASS Lab (University of Genoa, La Spezia campus).

The laboratory includes a test tank for validating Guidance, Navigation, and Control (GNC) architectures for ASV/USV platforms (Section 3.1), which was repurposed to reproduce human-in-water scenarios. A subject was made to float in the tank and imaged from multiple viewpoints. To increase scenario variability, the subject was recorded with and without a life jacket, in the presence of floating debris or flotation aids, with tank boundaries masked to emulate a wreck, and while performing different activity levels.

Actions included head submersion for thermal regulation, passive floating, clinging to debris or flotation aids, gripping the tank edge or simulated wreck, and raising arms to mimic distress signals. These conditions were randomly combined to generate a diverse and generalised dataset. A complete summary of the performed actions is reported in Table 5.5. Overall, the dataset comprises 3531 paired thermal images and point clouds.

Table 5.5: Dataset physical augmentation.

Category	Item
Floating objects	Generic debris
	Flotation aid
	Life-jacket
Human actions	Cling to tank-side
	Cling to debris
	Cling to flotation aid
	Raise arms
	Submerge head
	Float passively
	Float actively
Human wear	Wet-suit
	Life-jacket

During acquisition, thermal images and LiDAR point clouds were recorded with synchronised timestamps using custom software. To maximise horizontal resolution, the LiDAR operated at 5 Hz; image–point cloud pairs were therefore synchronised at the same rate by associating each point cloud with the temporally closest image. Images were manually annotated using a custom OpenCV-based tool in YOLO format. Bounding boxes were drawn around all visible human body parts emerging from the water to ensure accurate localisation. Each synchronised pair was stored in a JSON file, accompanied by a TXT file containing the YOLO annotations. Range and reflectivity LiDAR images were generated on the fly during testing via perspective projection and nearest-neighbour interpolation, enabling realistic replication of real-time operating conditions.

To increase dataset size and variability, including adverse environmental conditions, several

weather-related augmentations were applied, as summarised in Table 5.6. Although synthetic, these augmentations were designed to preserve physical plausibility. Fog and smoke were modelled through reductions in Normalised Global Contrast, following [133]. Spray and green sea effects were simulated by extracting water-surface intensities from reference images and reintroducing them with random variations up to ±50% to account for thermal fluctuations (water temperature during testing was 19°C).

Bright spot artifacts were modeled by considering their observed sources: (i) recurring patterns reported in the literature [55, 134], (ii) reflections on the water surface observed in the acquired data (Figure 5.7), and (iii) localized heating of droplets on the lens due to camera operating temperatures 5–10°C above ambient. Standard augmentations, including flipping and distance scaling, were also applied. To achieve realistic distance scaling of LiDAR data, point density was reduced in addition to scaling the range values. Typical LiDAR noise induced by water-surface ripples was naturally present due to actor motion in the tank.

Table 5.6: Thermal image meteorological augmentation.

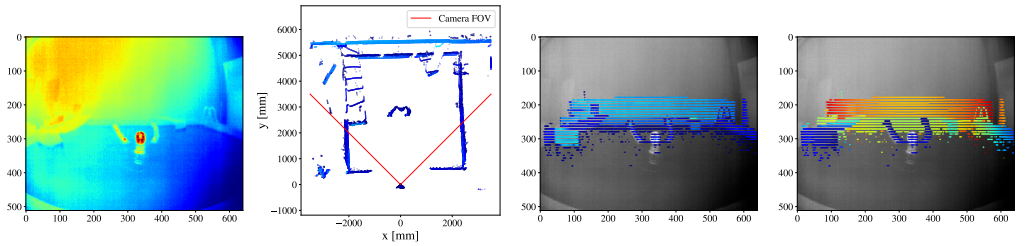
Phenomenon	Effect	Augmentation
Fog	Hazed image	Weighted mask and/or blur
Thick smoke	Heavily hazed image	Strong gradient
Green seas/spray	Water drops	Small-sized random cold spot
Horizon lights	False positive bright spots	Medium-sized random hot spots

Figure 5.7: Bright spot reflection on the water surface observed during the tests.

Synchronised thermal images and LiDAR point clouds are processed to project the point cloud onto the image plane and to generate LiDAR-derived range and reflectivity images, following the procedure described in Section 4.6.2. Each resulting multi-source three-channel image is then annotated in YOLO format using a custom-developed tool.

Figure 5.8 illustrates a representative sample after preprocessing. Figure 5.8a shows the thermal image rendered with a rainbow colourmap to emphasise temperature variations. Figure 5.8b presents the Bird's Eye View (BEV) of the point cloud, depicting a survivor with raised arms, the tank boundary, and artificial debris. Figures 5.8c and 5.8d show the point cloud projected onto

the thermal image plane and colored by reflectivity and range, respectively, demonstrating the alignment between modalities and the basis for generating the LiDAR-derived images.



(a) Reference thermal image. **(b)** BEV LiDAR point cloud **(c)** Projected point cloud colored by reflectivity. **(d)** Projected point cloud colored by range.

Figure 5.8: Preprocessing step on the LiDAR point cloud.

Figure 5.9 shows three random examples from the labelled dataset of non-augmented 3-channel images. Rows correspond to different examples, while columns represent the individual channels (thermal, reflectivity, and range). The images are displayed in grayscale format, and the bounding box enclosing the human-in-water is shown in red. The greater visual impact of the range field is due to the fact that in this channel, the differences between objects are much more pronounced than on the intensity channel. Examples of augmented thermal images are shown in Figures 5.10 and 5.11. Figure 5.10 shows progressive fog augmentation (and thus a meteorological optical range decay), which, according to [133], is reflected in a decrease of the target-to-background contrast.

Figure 5.11 shows a random example of the combined effect of the other possible augmentations. From left to right, distance augmentation plus light haze, tin smoke with some water drips, sea-blast, heavy rain with wet lens. This procedure enables the creation of a dataset augmented as desired to include environmental disturbance phenomena of varying aggressiveness, thereby increasing the number of samples useful for training the YOLO network.

5.1.2.1 Acquisition set-up

To support the experimental acquisition campaign and the validation of the proposed pipeline on real data, a dedicated setup was developed, comprising a LiDAR sensor, a thermal camera, and a custom aluminium-alloy mounting structure. The support can be installed either on a tripod via a dedicated insert or on a vehicle using a bolted bracket. The system integrates a 32-channel HESAI Pandar XT-32 LiDAR and a Teledyne FLIR A65 thermal camera; the main sensor specifications are reported in Section 3.4.

An overview of the mounting structure is shown in Figure 5.12. Figure 5.12a details the sensor installation dimensions, while Figure 5.12b presents a photograph of the complete acquisition system.

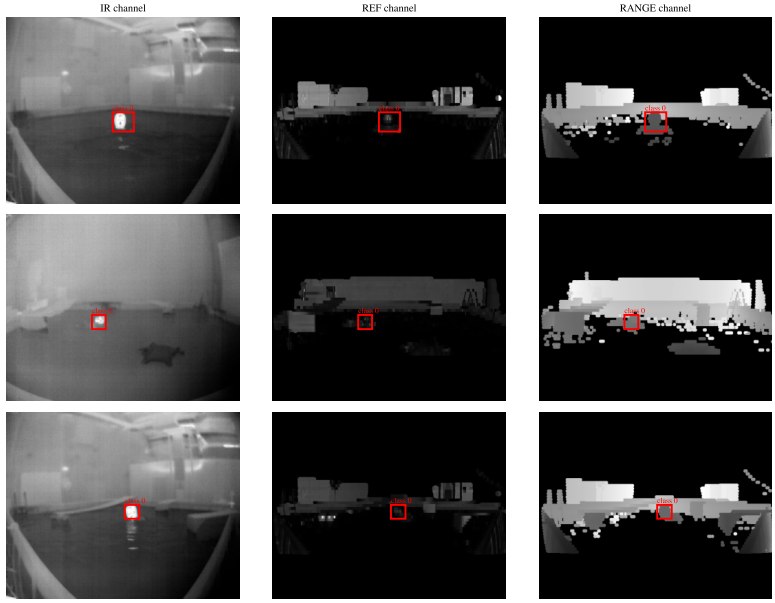


Figure 5.9: Shuffle samples of the labelled 3-channel images.

Figure 5.10: Fog augmentation with progressive Meteorological Optical Range decay.

5.1.2.2 Calibration

To achieve LiDAR–IR camera projection, it is essential to estimate the intrinsic and extrinsic calibration parameters together with an appropriate distortion model. Several approaches to thermal camera calibration have been proposed in the literature, employing different patterns and strategies to improve infrared visibility. In this study, a static checkerboard pattern was heated using a halogen lamp to provide sufficient thermal contrast, as shown in Figure 5.13. The intrinsic parameters of the thermal camera were estimated using a pinhole camera model with a five-parameter distortion formulation (three radial and two tangential coefficients). The procedure was implemented using *OpenCV* framework.

The calibration achieved a global root-mean-square (RMS) reprojection error of 0.325 pixels, with a mean reprojection error of 0.045 pixels and a standard deviation of 0.013. The global mean error over all corners was 0.27 pixels, with a standard deviation of 0.18 pixels, and 95% of points had reprojection errors below 0.61 pixels. Reprojection action during calibration is shown in

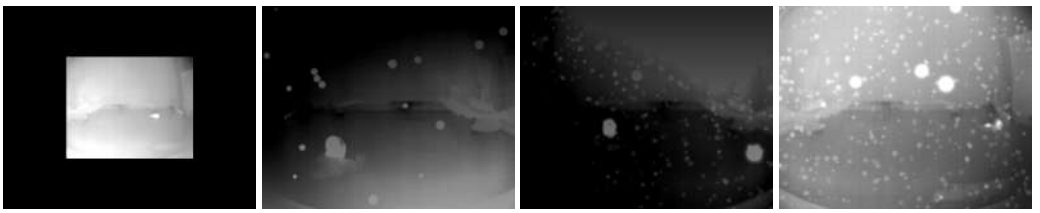
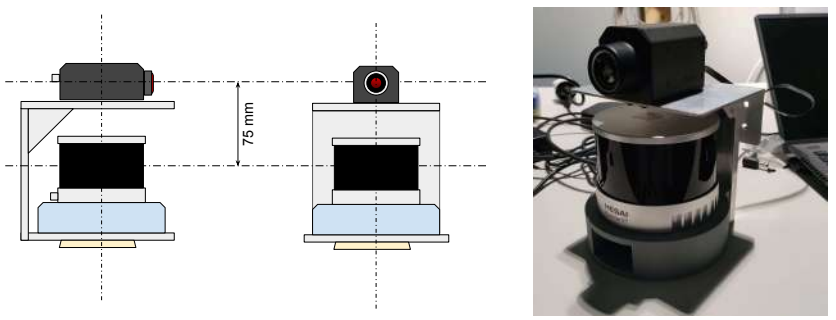


Figure 5.11: Random augmentations.



(a) Sensors set-up, technical drawing. **(b)** Sensor set-up, real image.

Figure 5.12: LiDAR - thermal camera set-up

Figure 5.14.

The extrinsic parameters were initially derived from the technical drawings of the sensor mounting system and subsequently refined manually by adjusting the rotation angles to obtain satisfactory alignment between object boundaries in the thermal images and the corresponding structures in the LiDAR point cloud.

5.2 Experimental trials

5.2.1 Indoor controlled environment

The main tests performed in a controlled environment were conducted at COMPASS LAB (see Section 3.1) in collaboration with the project partner CNR-INM. These trials represent the first integration test of the developed systems with other modules, specifically guidance and control, by providing position feedback and vehicle heading within the test tank through the Indoor Positioning

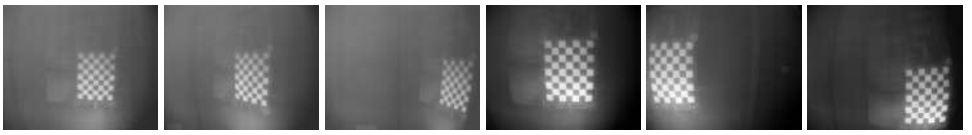


Figure 5.13: Thermal source calibration images slideshow.

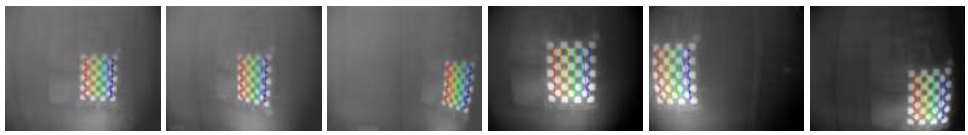


Figure 5.14: Slideshow of re-projection on thermal images during calibration procedure.

System (Section 4.9). This effectively closes the vessel GNC system loop and enables the initial autonomous navigation tasks.

Within the scope of this thesis, the campaign is treated primarily as an application of the Indoor Positioning System, which leverages it as an enabling technology. Discussions of the other integrated modules are presented only descriptively, as they constitute the intellectual property of other researchers and are included solely to illustrate their application.

5.2.1.1 Experimental setup

The campaign employed the SWAMP vehicles without onboard sensors. Position and heading feedback were provided by the IPS based on ArUco-type fiducial markers. Pose information was transmitted via UDP to both the vehicles and ground-based computers acting as route managers. Table 5.7 summarises the setup.

Table 5.7: Summary of the experimental setup and tests.

Experimental setup	
Vehicle	SWAMP
On-board sensors	none
External systems	Indoor Positioning System
Additional equipment	ArUco-type fiducial markers
	Shore computer

5.2.1.2 Experiment description

The campaign included incremental tests to evaluate single-vehicle and cooperative manoeuvres. The first test performed was the single-vehicle four-corner test, aimed at tuning the controller and the path manager developed by the project partner CNR-INM. The 4-corner experiment consists of the following six motions:

1. Forward motion, while keeping the vehicle heading constant.
2. Turning manoeuvre, making SWAMP turn 90° port.
3. Forward motion, similar to the first segment, preparing the vehicle for the next motion request.
4. Lateral motion (Crabbing), moving straight with respect to the vehicle's port side.
5. Backward motion, inverting motor thrust and keeping vehicle heading constant.

6. Turning manoeuvre, making the vehicle turn 90° starboard to return to the initial orientation.

Once a satisfactory level of parameter tuning was achieved, the experimental campaign progressed to cooperative control tests. Specifically, the project partners required the execution of the same four-corner test in the presence of a second vehicle, simulated solely via position feedback for simplicity. In this scenario, the primary vessel performing the four-corner manoeuvre is required to manage its state not only based on its own pose feedback, but also considering the pose feedback of the second vehicle, with the objective of avoiding collisions. In this case, the cooperative-mode four-corner test was repeated multiple times in order to obtain a sufficiently large dataset for subsequent processing and analysis.

Finally, a second vehicle was deployed in the water to test the project partners' GNC system's ability to handle bidirectional communication. The experimental campaign concluded at this stage. Figure 5.15 illustrates the vehicles during the last communication trial.

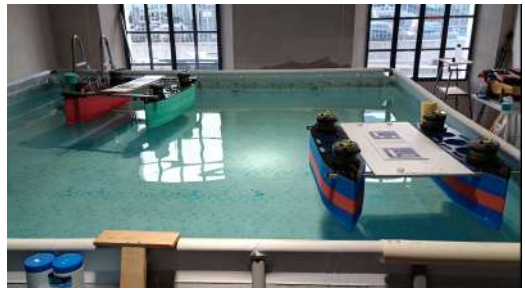


Figure 5.15: SWAMP vehicles performing cooperative four-corner test during the trials in the COMPASS lab.

5.2.2 Lake Lagaccio

The Lake Lagaccio trials represented a step forward in the interdisciplinary integration of technologies in a relevant environment. Although only the state estimation system was integrated at this stage, the trials enabled verification of the SWAMP platform's advanced operational capabilities in a relevant environment. In this context, the state estimation system served as an enabling technology for autonomous surveillance of an area of interest.

Within the scope of this thesis, the campaign is treated primarily as an application of the state estimation feedback module, which leverages it as an enabling technology for autonomous navigation. Discussions of the other integrated modules are presented only descriptively, as they constitute the intellectual property of other researchers.

5.2.2.1 Experimental setup

The experimental setup is summarised in Table 5.8. The trials were conducted using the SWAMP autonomous surface vehicle, equipped with GNSS/INS for navigation and state estimation. Position and velocity feedback were provided by the state estimation system based on a Kalman filter. Computation was performed on an onboard semi-rugged PC powered by its own battery pack. For

communication, a Wi-Fi range extender and a shore-based processing PC were employed. During the trials, a NED-compliant local reference system was placed at 44.92682° N, 9.57081° E.

Table 5.8: Summary of the experimental setup.

Experimental setup	
Vehicle	SWAMP
On-board sensor	GNSS/INS
Additional on-board equipment	On-board semi-rugged PC
External equipment	Wi-fi antenna range extender
	Shore processing PC

5.2.2.2 Experiment description

The experimental activities focused on autonomous surface-vehicle navigation for area surveillance. The SWAMP platform was tasked with autonomously following a sequence of waypoints generated by a coverage algorithm, using position and velocity feedback to patrol a hypothetical area of interest. In this context, the research presented in this thesis contributed the vehicle state-estimation module, which provided the navigation feedback required by the guidance and control layers. In a preliminary phase, the vehicle was operated remotely to establish the communication system's operating range while simultaneously collecting GNSS/INS data. The subsequent validation phase was structured through two incremental tests:

- **Line Of Sight navigation experiment**: this test was aimed at assessing guidance accuracy and tracking performance between individual waypoints. Specifically, a set of seven waypoints was defined on the map in the local reference frame, and the SWAMP vessel was autonomously navigated among them.
- **Surveillance experiment**: This was conducted to evaluate navigation robustness over extended trajectories and the effectiveness of area coverage. Specifically, the waypoints were no longer provided in advance, but were generated one by one by the path generation and area coverage system, creating a new waypoint as soon as the previous one was reached. This approach ensured both area coverage and path randomness. In this case, the total number of waypoints was not predefined, and the system operated fully autonomously in patrolling mode for approximately 30 minutes.

Some images captured during testing at this site are presented in Figure 5.16 as a slideshow.

5.2.3 Lake Nemi

The most extensive experimental activity in a representative real-world environment was a five-day campaign conducted at the CNR-INM research facility on Lake Nemi.

This campaign represented the culmination of integrating the systems developed in this research with modules created by other researchers, achieving full integration into a system capable of autonomously performing tasks in a relevant environment.

Figure 5.16: Lake Lagaccio slideshow during experimental trials.

Specifically, thanks to the verifications and validations conducted during previous campaigns, it was possible to add a perceptual layer to the SWAMP platform, comprising a LiDAR sensor in addition to the GNSS/INS sensors for state estimation. This enabled the platform not only to navigate autonomously between waypoints but also to perform reactive, collision-free path planning by detecting obstacles.

5.2.3.1 Experimental setup

The experimental setup is summarised in Table 5.9. The trials were conducted using the SWAMP autonomous surface vehicle, equipped with GNSS/INS for navigation and state estimation, a LiDAR sensor for perception, and an RGB camera providing a visual reference of onboard operations. Computation was handled by the onboard semi-rugged PC, powered by a Bluetti AC70 power station. For communication and monitoring, a Wi-Fi range extender and a shore-based processing PC were employed. During the trials, a NED-compliant local reference system was placed at 41.71983°N, 12.70038°E.

Table 5.9: Summary of the experimental setup.

Experimental setup	
Vehicle	SWAMP
On-board sensor	GNSS/INS
	LiDAR
	RGB camera
Additional on-board equipment	On-board semi-rugged PC
	Bluetti AC70 power station
External equipment	Wi-Fi antenna range extender
	Shore processing PC

Figure 5.17 shows the SWAMP vehicle ready to be launched in the water with all the sensors highlighted. Note that, to support project partners in acquiring the SWAMP vehicle with a shore radar, it was additionally equipped with a corner reflector and a thermal imaging camera provided by other partners.

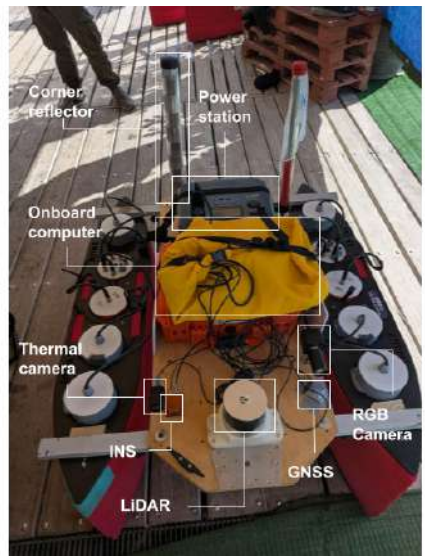


Figure 5.17: SWAMP vehicle ready to be launched in the water of Lake Nemi.

5.2.3.2 Experiment description

The main activity of the Lake Nemi campaign focused on an integration test of the collision-avoidance system.

The situational-awareness and state-estimation algorithms developed during the doctoral research were integrated with control, path-planning, and collision-avoidance modules developed by other team members. In particular, a LiDAR-based perception module operating in obstacle-map mode and a Kalman-filter–based state-estimation module were integrated into a modular end-to-end architecture.

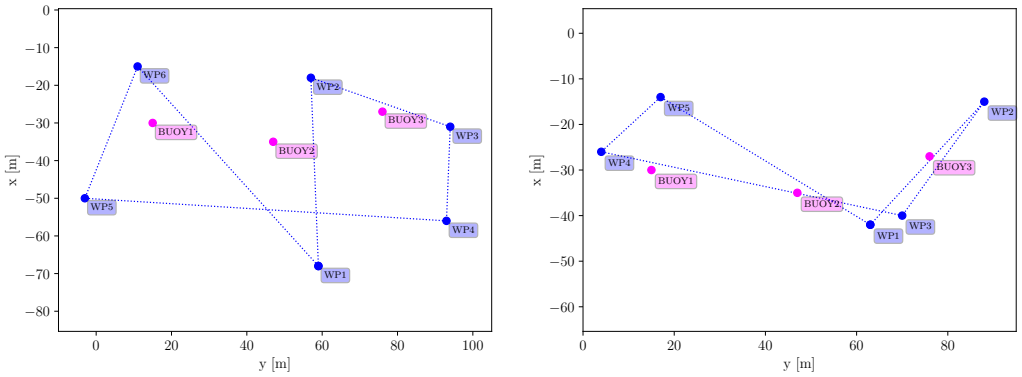
The RGB camera was not directly used by the control stack but provided a synchronised video stream for qualitative assessment.

The primary objective was to validate the full system integration under realistic operating conditions. The test was organised into two phases:

- **Line Of Sight navigation experiment**: the vehicle autonomously navigated a predefined polygonal route of six waypoints across the lake, as illustrated in Figure 5.18a, relying exclusively on state-estimation feedback. The assigned polygon comprised six waypoints, with a total route length of approximately 320 meters. The coordinates of the waypoints in the local reference system, as well as the lengths of the individual legs, are reported in Table 5.10. This experiment mirrors the one conducted during the first phase of the Lake Lagaccio trials, but introduces the added complexity of significantly greater distances and more substantial environmental disturbances.

- **Collision avoidance experiment**: The second experiment was designed as a fully autonomous

navigation scenario, requiring the coordinated operation of all system modules, with particular emphasis on testing the CAV module. The ASV was assigned to follow a polygonal route, illustrated in Figure 5.18b and with waypoints detailed in Table 5.11, covering a total distance of approximately 206 meters. Along the route, both fixed and moving obstacles were introduced. The vehicle was required to detect these hazards, dynamically replan its trajectory to avoid collisions, and then continue the mission along the original path.



(a) Line Of Sight navigation experiment.

(b) Collision avoidance experiment.

Figure 5.18: Assigned paths.

Table 5.10: Coordinates of the planned path for the Line Of Sight navigation experiment.

Leg	Coordinates [m]	Dist.
WP 1-2:	(-68.0, 59.0) → (-18.0, 57.0)	50.0 m
WP 2-3:	(-18.0, 57.0) → (-31.0, 94.0)	39.2 m
WP 3-4:	(-31.0, 94.0) → (-56.0, 93.0)	25.0 m
WP 4-5:	(-56.0, 93.0) → (-50.0, -3.0)	96.2 m
WP 5-6:	(-50.0, -3.0) → (-15.0, 11.0)	37.7 m
WP 6-1:	(-15.0, 11.0) → (-68.0, 59.0)	71.5 m
	Total dist.:	319.7 m

Table 5.11: Coordinates of the planned path for the CAV experiment.

Leg	Coordinates [m]	Dist.
WP 1-2:	(-42.0, 63.0) → (-15.0, 88.0)	36.8 m
WP 2-3:	(-15.0, 88.0) → (-40.0, 70.0)	30.8 m
WP 3-4:	(-40.0, 70.0) → (-26.0, 4.0)	67.5 m
WP 4-5:	(-26.0, 4.0) → (-14.0, 17.0)	17.7 m
WP 5-1:	(-14.0, 17.0) → (-42.0, 63.0)	53.9 m
	Total dist.:	206.6 m

During navigation, the ASV encounters buoys obstructing the planned path along several legs of the polygon. In other words, the polygon cannot be followed in the same manner as the first

experiment, as this would result in multiple collisions. In contrast, once the perception system detects obstacles, the CAV module computes and updates alternative collision-free trajectories as needed, maintaining a safety distance of 5 meters from the obstacles while maintaining progress toward the assigned sequence of waypoints within the polygon.

Figure 5.19 shows the launch of the SWAMP vessel from the floating quay, while Figure 5.20 depicts an autonomous collision-avoidance manoeuvre during the approach to a buoy detected by the onboard LiDAR.

(a) SWAMP launch. **(b)** SWAMP navigating on the lake.

Figure 5.19: Launching operation of the SWAMP vessel during the Lake Nemi campaign.

Figure 5.20: SWAMP collision-avoidance test during a buoy approach.

5.2.4 Gulf of Naples

The final experimental test conducted during the doctoral period took place in the Gulf of Naples.

On the one hand, the area provided significant benefits for testing the hardware and communication devices in a relevant environment characterised by considerable environmental disturbances. However, under these conditions and given the vehicle's small size, the system experienced strong motion across all degrees of freedom, thereby limiting the effectiveness of the onboard perceptual sensors.

For this reason, and supported by the positive results obtained in previous experimental campaigns, the trials were primarily used to evaluate the communication system and to test the deployment of the computing framework across a geographically distributed architecture.

5.2.4.1 Experimental setup

The experimental setup is summarised in Table 5.12. The trials were conducted using the SWAMP autonomous surface vehicle, equipped with GNSS/INS for navigation and state estimation, and with LiDAR and an RGB camera providing visual references for onboard operations. Computation was handled by the onboard semi-rugged PC, powered by a Bluetti AC70 power station. For communication and monitoring, a Wi-Fi range extender and a shore-based processing PC were employed. During the trials, a NED compliant local reference system was placed in 40.833287°N, 14.267719°E In addition, thanks to the Tailscale service, two remote computers were connected to the system. Compared to the Lake Nemi tests, the SWAMP vehicle's computing and communication architecture was expanded to include two geographically distributed nodes—one in La Spezia at the UniGE COMPASS Lab and the other in Genoa at the DITEN laboratories in Villa Cambiaso. These nodes were integrated with the on-board and shore-based computers connected via the local Wi-Fi network. The remote nodes had internet access, the shore station had both internet access and connection to the local vehicle control network, while the on-board computer was connected to the control network. Using the *Tailscale* service, the remote computers could access the shore station via SSH and, through it, connect to the SWAMP vehicle over the local Wi-Fi network.
The whole communication infrastructure is depicted in Figure 5.21. The purple blocks are network interfaces, the blue blocks are computing nodes, the red blocks are the sensors, and the green block is the energy power system. The LiDAR is connected to the onboard PC via Ethernet using UDP, the GNSS/INS system via USB with a serial protocol, and the RGB camera via USB using the UVC standard.

Table 5.12: Summary of the experimental setup.

Experimental setup	
Vehicle	SWAMP
On-board sensor	GNSS/INS
	LiDAR
	RGB camera
Additional on-board equipment	On-board semi-rugged PC
	Bluetti AC70 power station
External equipment	Wi-Fi antenna range extender
	Shore processing PC
	Remote PC

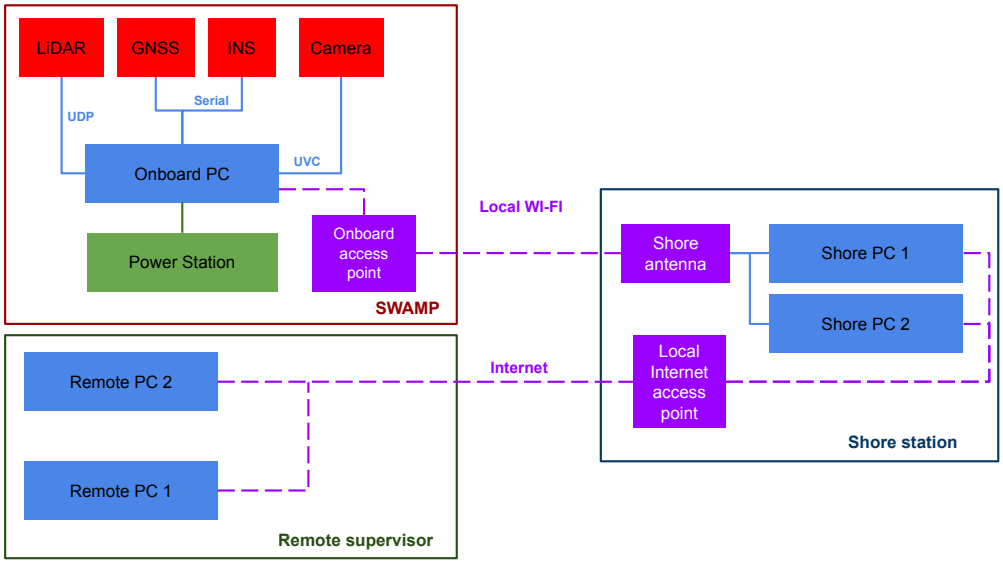


Figure 5.21: Distributed communication infrastructure used during the Gulf of Naples trials.

5.2.4.2 Experiment description

The experimental campaign was conducted in two phases: an initial phase in which the vehicle was operated entirely on-site via remote control, followed by a second phase that also relied on remote control but enabled system access from geographically distributed nodes.

- **Remote control testing**: the vehicle was navigated within the area of interest under remote control, enabling other team members to collect data related to forces and manoeuvrability under adverse conditions. During this phase, the state estimation system remained active to evaluate data reception from the various system nodes, distribute the data across different machines, and assess their time intervals to ensure time consistency.
- **Integration of remote nodes**: a test analogous to the previous one was conducted, with the addition of remote nodes connected to the network (La Spezia and Genoa). This phase focused on monitoring system accessibility, signal reception, and sensor management across the distributed architecture.

Some images captured during testing at this site are presented in Figure 5.22 as a slideshow.

Figure 5.22: Gulf of Naples slideshow during experimental trials.

6. Results

In this chapter, the results of the testing and validation activities conducted on the algorithms developed during the doctoral research, as detailed in Chapter 4, are presented. The chapter is divided into two main sections, following the structure of the experimental campaigns. Section 6.1 presents the validation of the pipelines developed during the research, using real data acquired throughout the data collection campaigns. The aim is to evaluate the individual pipelines, their constituent modules, and the adopted sensor-fusion strategies. These experiments aim to assess the performance and functionality, with particular emphasis on real-time compliance.

Section 6.2 presents the results obtained from the experimental activities aimed at integrating the developed technologies into the SWAMP autonomous platform within a broader multidisciplinary framework. This section focuses on the system-level role of the developed pipelines as an enabling technology for enhanced autonomy in marine navigation, rather than on their standalone validation, thereby demonstrating their versatility and integration with other system modules to accomplish specific tasks such as monitoring, autonomous surveillance, and collision avoidance.

6.1 Pipeline verification

6.1.1 LiDAR detection and classification

This section summarises the key findings from the LiDAR detection and classification study, including model training, tuning, and field testing. Both quantitative and qualitative analyses are presented to evaluate the effectiveness of the proposed LiDAR detection and classification methodology. The detection results and related discussion are reported in [32, 35, 31], while classification-specific outcomes are discussed in [33].

6.1.1.1 Clustering analysis

As a preliminary analysis, this section compares the computational costs of different clustering algorithms. Considering the potential use of processing pipelines in real-time and embedded systems, computational efficiency represents a critical constraint.

Three widely used clustering algorithms for point-cloud-based obstacle detection were selected: (i) DBSCAN (Density-Based Spatial Clustering of Applications with Noise); (ii) HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise); (iii) OPTICS (Ordering points to identify the clustering structure). These algorithms were chosen because they share fundamental characteristics: (i) they do not require the number of clusters as a priori data; (ii) they are not strictly based on the concept of distance alone, thus mitigating bridging phenomena. This analysis provides quantitative justification for selecting the DBSCAN algorithm within the pipeline.

For the preliminary test, 25 random scenarios were selected from batch 0 of the ‘marina’ section of the raw ARNOLD dataset. The clustering algorithms listed above were applied to these

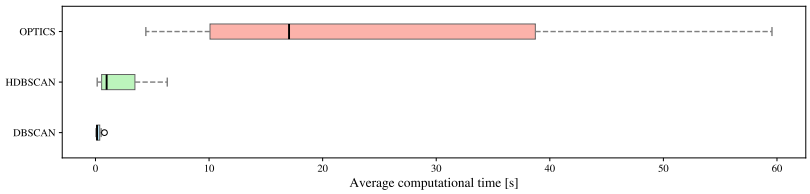


Figure 6.1: Clustering algorithms computational time analysis.

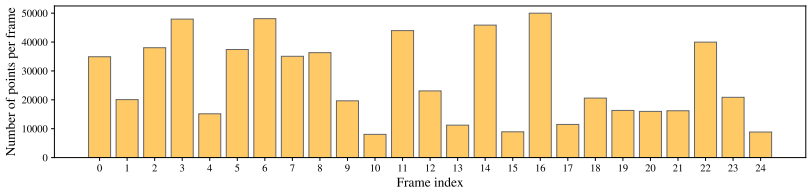


Figure 6.2: Number of points per scan.

scenarios. All algorithms were executed using the `scikit-learn` implementation with default hyperparameters.

Figure 6.1 reports the average computational cost for each clustering algorithm. The red line indicates the LiDAR acquisition time of 0.2 s, corresponding to a 5 Hz sampling rate. Figure 6.2 shows the number of points per scenario.

In light of this, DBSCAN is the only algorithm among those tested to meet the requirements for real-time implementation, with an average computational cost of 0.052 s across the tested scenarios.

This analysis alone shows that DBSCAN has a computational cost more suitable for real-time deployment in this domain. In particular, it is evident that the OPTICS algorithm has a computational cost that is not comparable to that of the other two and, in any case, always exceeds the real-time limits imposed by LiDAR acquisition (5Hz).
However, a performance analysis including the three algorithms has been carried out following the steps described below:

1. Tuning of the main hyperparameters for the three algorithms on a test scenario to set them to a comparable value.

2. Creation of 50 random test scenarios with ground truth. Each scenario was composed by including a random number n (chosen between 5 and 9) of samples from the labelled section of the ARNOLD dataset.

3. Clustering analysis on the scenarios and extraction of standard KPIs.

The standard KPIs chosen are:

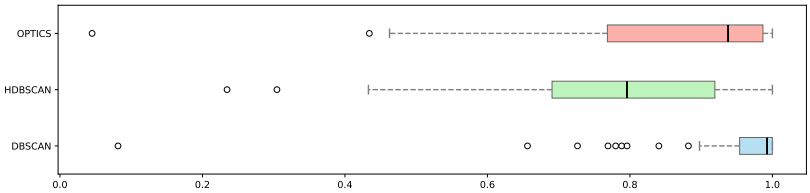


Figure 6.3: Clustering algorithms Adjusted Rand Index analysis.

- **ARI (Adjusted Rand Index)**: measures the similarity between the clustering results and the ground truth, taking into account chance agreements. A higher value indicates better agreement with the true labels.
- **Silhouette Score**: evaluates how well each point fits within its assigned cluster compared to other clusters. Values range from -1 to 1, with higher values indicating more compact, well-separated clusters. Points labelled as noise were excluded from the calculation.
- **Noise Ratio**: represents the fraction of points identified as noise by the clustering algorithm (i.e., points not assigned to any cluster). Lower values indicate that fewer points were considered outliers.

Table 6.1 summarises the hyperparameters used. Hyperparameters not mentioned have been left at their default values.

Algorithm	Hyperparameter	Meaning	Chosen value
DBSCAN	eps	Maximum distance between two points to be considered neighbours	2
	min_samples	Minimum number of points required to form a dense region	5
HDBSCAN	min_cluster_size	Minimum number of points to form a cluster	25
	min_samples	Minimum number of samples considered for core points (density threshold)	None
OPTICS	min_samples	Minimum number of points required to form a core point	10
	max_eps	Maximum distance between points to consider for clustering	2
	xi	Steepness threshold for extracting clusters from reachability plot	0.5

Table 6.1: Summary of clustering algorithms and their main hyperparameters with chosen values.

Table 6.2 shows the average values for the selected KPIs.
The analysis reveals that the DBSCAN algorithm not only guarantees lower computational costs but also performs better in terms of ARI and Noise Ratio. OPTICS, on the other hand, is the algorithm that guarantees the best Silhouette Score value. Given the trade-off among all KPIs, DBSCAN remains the best choice for this application.

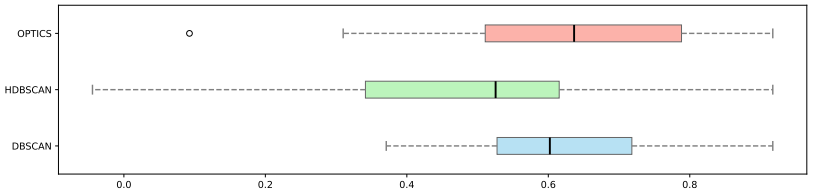


Figure 6.4: Clustering algorithms Silhouette Score analysis.

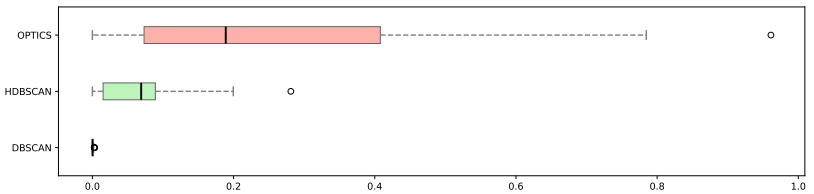


Figure 6.5: Clustering algorithms Noise Ratio analysis.

6.1.1.2 Features analysis

Feature distributions for each class are illustrated through violin plots. The geometric information of the point cloud is shown through the feature distributions in Figure 6.6 and Figure 6.7. Hence, Figure 6.6a shows the 3 PCA eigenvalues, i.e., the data variance along the 3 main directions, using a logarithmic scale. In Figure 6.6b, the ellipticities are presented as the ratio of eigenvalues.

These two graphs show the variance along the principal directions and their ratios, reflecting the geometric characteristics of the targets. Bulky targets such as *Motor Boat* and *Ship* exhibit similar mean values of ellipticities involving the second eigenvalue ($e_{1,2}$, $e_{2,3}$), but can be distinguished by the eigenvalue magnitudes, reflecting their size differences.

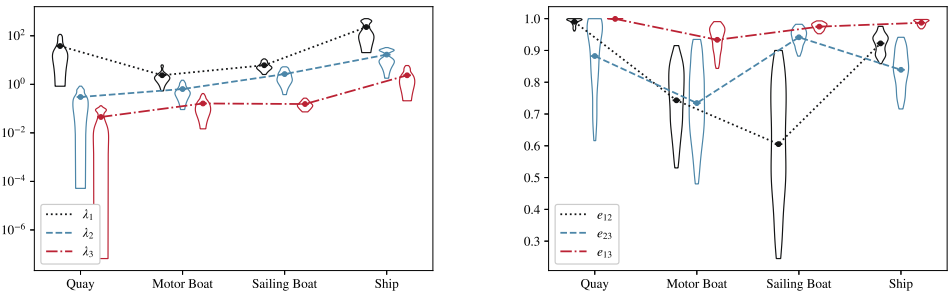
Linear targets in the *Quay* class, such as dock edges, have one dominant direction; the corresponding eigenvalue (λ_1) exceeds the others, and the ellipticities $e_{1,2}$ and $e_{1,3}$ involving this eigenvalue are higher.

Planar targets, like a *Sailing Boat*, show similar means for the first two eigenvalues (λ_1, λ_2), yielding a lower $e_{1,2}$ compared to the other ellipticities. This arises from the mast and sails, which create two principal directions with comparable variance.

Figures 6.7a and 6.7b report the distributions of point-cloud Skewness and Kurtosis along the principal directions. The *Quay* class exhibits positive Skewness in the second and third directions, corresponding to the width and height of the quay. This is explained by the predominance of

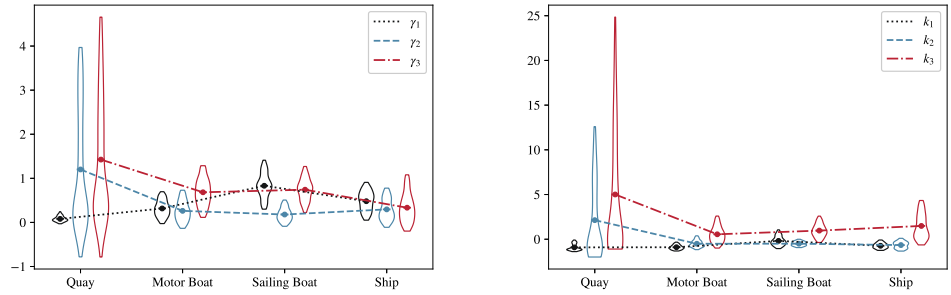
Table 6.2: Average clustering metrics for DBSCAN, HDBSCAN, and OPTICS.

Metric	DBSCAN	HDBSCAN	OPTICS
Average ARI	0.9349	0.7770	0.8488
Average Silhouette	0.6237	0.4969	0.6364
Average Noise Ratio	0.0006	0.0710	0.2583



(a) Log-scale point-cloud eigenvalues obtained from the PCA.**(b)** Point cloud ellipticities calculated in the principal directions.

Figure 6.6: Point cloud violin plots: first set.



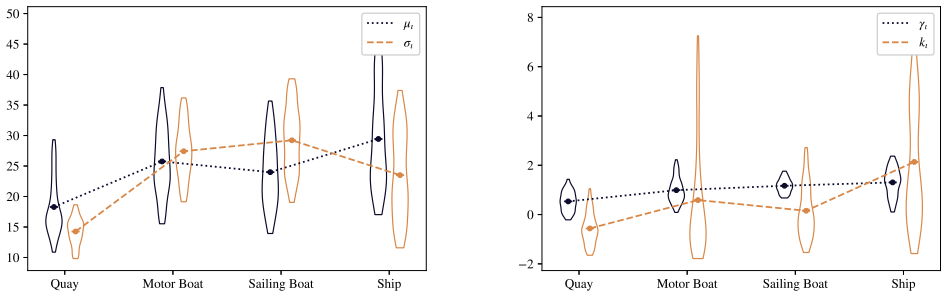
(a) Point cloud Skewness calculated in the principal directions.**(b)** Point cloud Kurtosis calculated in the principal directions.

Figure 6.7: Point cloud violin plots: second set.

points on the sensor's first-impact surfaces, which are further amplified by their flatness [31]. For *Motor Boat* and *Ship*, Skewness variations are minimal and mainly affected by distance and POV. In *Sailing Boat*, the second-direction Skewness (γ_2) is near zero, due to the mast and rigging producing a nearly symmetrical distribution along that axis.

Kurtosis distributions across the three principal directions are generally platykurtic with no big inter-class differences. An exception is the *Quay* class, where k_2 and especially k_3 indicate leptokurtic distributions, confirming the concentration of points on first-impact surfaces as observed in Skewness.

Intensity-related features are reported in Figure 6.8. Specifically, Figure 6.8a shows the mean and standard deviation of intensity for each class. The mean values are largely similar across classes, except for the *Quay* class, which exhibits lower values due to the lower reflectivity of raw, unpainted concrete.



(a) Intensity mean and standard deviation.

(b) Intensity Skewness and Kurtosis.

Figure 6.8: Intensity related features violin.

Figure 6.8b reports intensity Skewness and Kurtosis, showing no distinctive patterns. Skewness values are generally similar, and intensity distributions are mostly platykurtic. The *Ship* class exhibits a slightly higher mean Kurtosis, but this is not a prominent distinguishing feature.

The distributions of additional features are reported in Figure 6.9. Shannon Entropy, shown in Figure 6.9a, distinguishes *Quay* and *Ship* classes: the former exhibits low entropy, reflecting uniform intensity values, while the latter shows higher entropy due to greater variability. This difference arises from both size and material: quays present flat, homogeneous surfaces with narrow cross-sections, whereas ships have large, heterogeneous surfaces with varying materials and incidence angles, resulting in greater intensity variability.

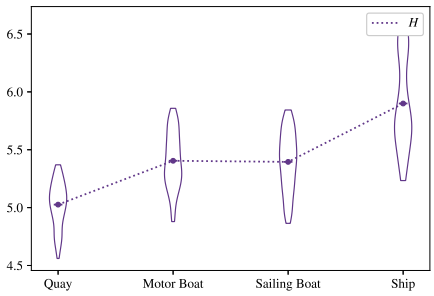
Spatial auto-correlation indices, Moran's I and Geary's C, are reported in Figure 6.9b. No big inter-class differences are observed, although the *Sailing Boat* class shows slightly lower Moran's I (and correspondingly higher Geary's C), indicating that neighbouring points have less similar intensity values. This effect is attributed to the mast and rigging, which produce a more heterogeneous intensity distribution in the LiDAR point cloud.

Figure 6.10 summarises the mean values of the extracted features for each class, normalised by the maximum absolute value of each feature across the dataset. The *Quay* class is shown as a green solid line with circular markers, *Motor Boat* as a blue dashed line with cross markers, *Sailing Boat* as a red dotted line with square markers, and *Ship* as an orange dash-dot line with diamond markers.

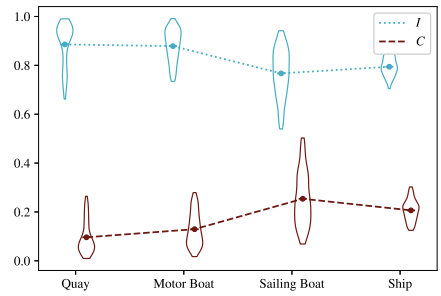
Analysis of the proposed features, particularly the mean and spread of the first quartiles, indicates that a combination of these 19 parameters could effectively discriminate between classes. This motivates the development of predictive models leveraging these features to classify LiDAR-detected objects.

Additionally, a quantitative analysis is conducted to complement the preceding qualitative observations. Figure 6.11 shows the correlation matrix of the selected features, highlighting that eigenvalues and ellipticities are highly correlated.

Figure 6.12 shows the p-values of each feature obtained from the Kruskal-Wallis test, plotted on a logarithmic scale due to their small magnitude. The standard significance threshold of 0.05 is



(a) Intensity Shannon Entropy.



(b) Spatial auto-correlation indices.

Figure 6.9: Additional features: violin plots.

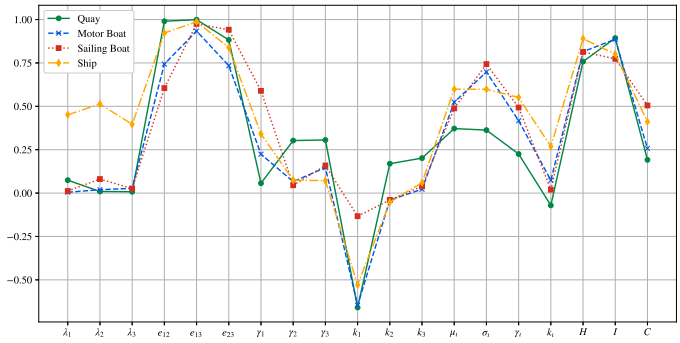


Figure 6.10: Global normalised feature distribution.

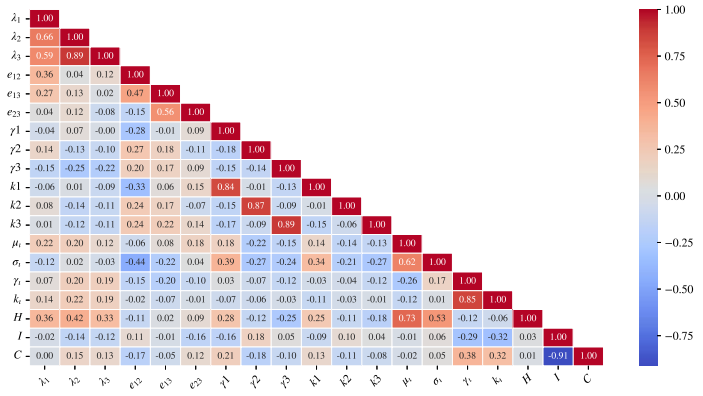


Figure 6.11: Correlation matrix within the proposed features.

indicated by a red dashed line, and a random variable is included for comparison. All selected features strongly reject the null hypothesis. Notably, the third-direction point-cloud kurtosis (k_3), the least important feature, has a p-value of 4.77×10^{-12}, compared to an average p-value of 0.65 for the random feature over 100 runs.

The quantitative analysis confirms the possibility of using these features, in whole or in part, for predictive purposes.

6.1.1.3 Supervised model training

This section discusses the training of the supervised model and the tuning of hyperparameters, as described in Chapter 4.

To identify a suitable classification model for the feature dataset, three algorithms were evaluated: Random Forest (RF), Multi-Layer Perceptron (MLP), and Support Vector Machine (SVM). Hyperparameter tuning was performed for all models, with the search grids reported in Tables 6.3, 6.4, and 6.5 for RF, MLP, and SVM, respectively. Each hyperparameter combination was evaluated using 5-fold cross-validation, and the best-performing configurations are reported in the tables. The classifiers were trained and tested on the labelled part of the ARNOLD dataset (see Section 5.1) on a total of 926 samples.

While the computational cost was comparable across models, the RF classifier achieved the highest average cross-validation accuracy of 94%, compared to 90% for MLP and 91% for SVM. Additionally, RF provides feature importance rankings, which are critical for the proposed framework. For these reasons, the Random Forest Classifier was selected as the classification model.

The Random Forest classification report on the entire test set is shown in Table 6.6, providing an assessment of the model's predictive performance across classes and in a global view. Moreover, the 5-fold cross-validation result is reported in Table 6.7.

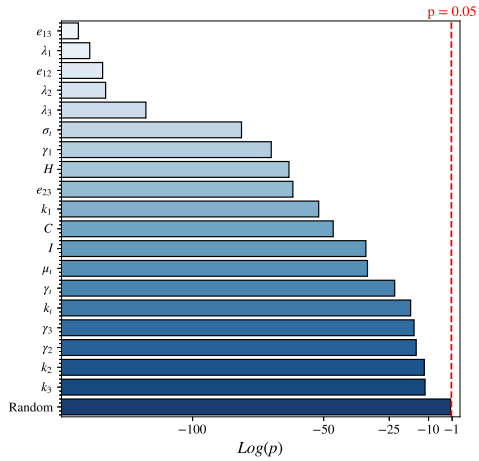


Figure 6.12: Kruskal-Wallis p-values bar plot.

Table 6.3: RF hyperparameters Tuning Grid

Hyperparameter	Search range	Best
N° estimators	[64-256]	128
Max Depth	[2-50, None]	10
Max Features	$[\sqrt{n}, \log_2 n$, None]	$\sqrt{n}$
Min Samples Leaf	[1-5]	2
Min Samples Split	[2-10]	5

Table 6.4: MLP hyperparameters Tuning Grid

Hyperparameter	Search range	Best
Hydden layer size	[(10-200,)(10-200,10-200)]	(100,20)
Activation	[relu, tanh]	tanh
Solver	[adam, lbfgs]	adam
Alpha	[0.001-0.1]	0.005
Learning rate	[constant, adaptive]	constant
Batch size	[auto, 10-100]	auto

Table 6.5: SVM hyperparameters Tuning Grid

Hyperparameter	Search range	Best
C_{svm}	[0.1-100]	10
Kernel	[linear, rbf, poly, sigmoid]	rbf
Gamma	[scale, auto]	scale

Table 6.6: Classification Report

Class	Precision	Recall	F1-score	Support
Motor Boat	0.94	0.96	0.95	85
Quay	0.99	0.96	0.97	72
Sailing Boat	0.99	0.99	0.99	76
Ship	0.95	0.95	0.95	43
Macro average	0.97	0.97	0.97	276
Weighted average	0.97	0.97	0.97	276
Accuracy			0.97	276

Table 6.7: 5-folds cross-validation result

Fold	Accuracy
1	0.90
2	0.98
3	0.99
4	0.97
5	0.87
mean	0.94

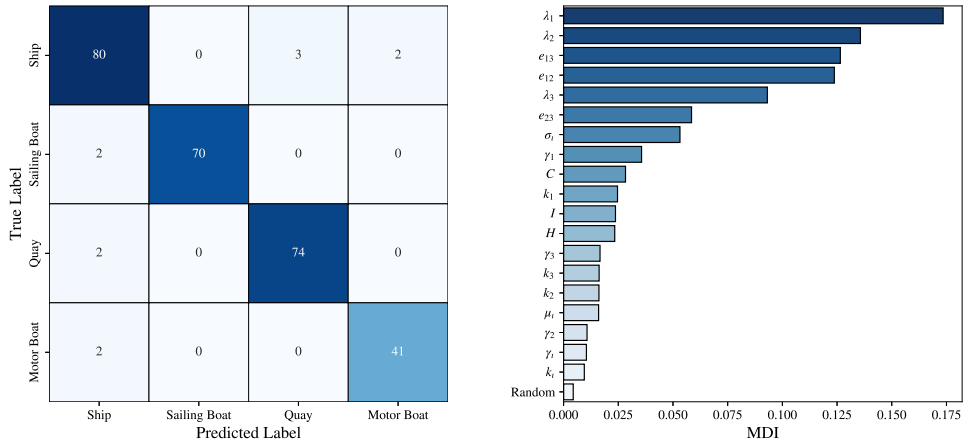
The training results are shown in Figure 6.13, and the confusion matrix on the test set is reported in Figure 6.13a. Figure 6.13b presents the MDI feature importance, including a random variable added to the training set to verify that all selected features exceed the importance of a random feature. Consistent with the Exploratory Data Analysis, the features contribute to classification with an importance distribution similar to that observed in the Kruskal-Wallis test.

6.1.1.4 Model optimisation and tuning

The model achieves sufficient accuracy for the intended application; however, optimising the feature set could improve the trade-off between computational efficiency and prediction performance.

In real-time applications, the computational cost of feature extraction from LiDAR point clouds is critical. The two spatial auto-correlation indices (Moran's I and Geary's C) are the most computationally expensive. To assess their impact, feature extraction times were measured on a real LiDAR scan containing 40 targets of varying sizes, averaged over 100 runs, using an off-the-shelf laptop (see Section 3.5 for specifications). The full feature set requires, on average, 0.12 s per target (SD = 0.28 s), whereas excluding the two spatial indices reduces the time to 0.0035 s (SD = 0.0005 s).

Given the LiDAR operating frequency of 5–20 Hz, including Moran's I and Geary's C is impractical for real-time use. Moreover, their Mean Decrease Impurity (MDI) values are low. Removing these indices reduces the number of features from 19 to 17, significantly lowering computational load with minimal impact on accuracy. Retraining the Random Forest Classifier without these indices and performing 5-fold cross-validation yields an average accuracy of 93% (Table 6.8).



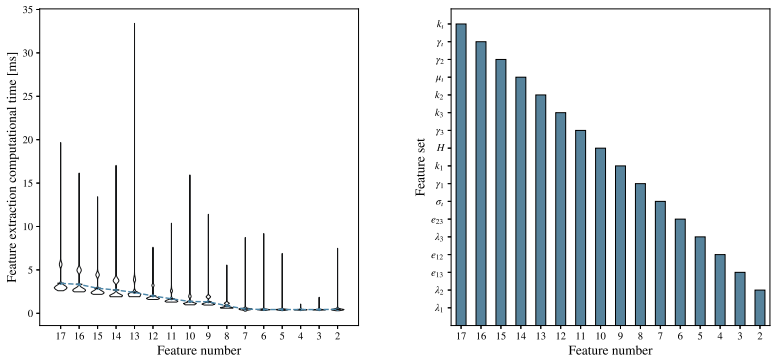
(a) Random Forest confusion matrix.

(b) Random Forest feature importance.

Figure 6.13: Random Forest training results.

Table 6.8: 5-fold cross-validation result without the spatial auto-correlation indices

Fold	Accuracy
1	0.875
2	0.95
3	0.99
4	0.96
5	0.85
mean	0.93



(a) Average computational cost for each feature set.

(b) Selected feature set for each feature number.

Figure 6.14: Computational cost calculation according to Recursive Feature Elimination.

A Recursive Feature Elimination (RFE) procedure was performed to investigate feature dependencies. Features were removed sequentially by importance, with the model retrained and evaluated via 5-fold cross-validation at each step to compute the mean accuracy. The number of Random Forest estimators was varied from 64 to 128 for each feature set, resulting in a 15×65 grid (975 combinations). All other hyperparameters were fixed as in Table 6.3.

Figure 6.14a shows the mean feature extraction time T_1 (in ms) as a function of the number of features, while Figure 6.14b reports the specific feature sets corresponding to each step in the RFE procedure.

The global computational time T is shown as a heat map in Figure 6.15. Feature extraction time T_1 dominates the overall computational cost, while T_2 has minimal influence.

Figure 6.16 presents the average accuracy A from 5-fold cross-validation for each combination in the test grid. Accuracy is strongly influenced by the number of features and only marginally by the number of estimators. Figure 6.17 shows the optimal number of decision trees (green) for each feature set, alongside achievable accuracy (red) and computational cost (blue). A minimum of 8 features yields near-90% accuracy, and all combinations require less than 5 ms of computational time.

For the case study, a computational budget below 3 ms is imposed while maximising accuracy. Accordingly, 15 features are selected (see Figure 6.14b), excluding intensity skewness (γ_ι) and kurtosis (k_ι), and the number of decision trees is set to 72. The Random Forest Classifier is then trained on the entire dataset using these parameters.

6.1.1.5 Field verification

This section reports the results of the classification module embedded in the complete LiDAR detection pipeline. Marine LiDAR scans are read, parsed, and processed in real-time using the

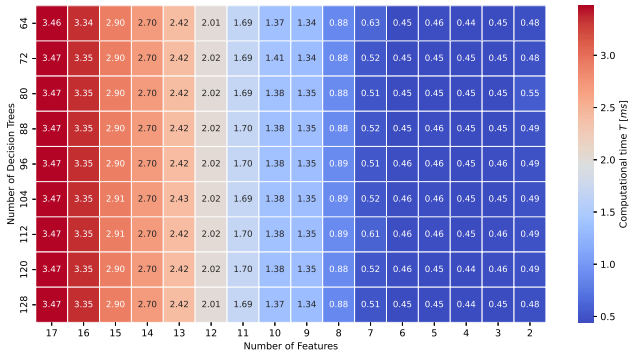


Figure 6.15: Global computational time heat map.

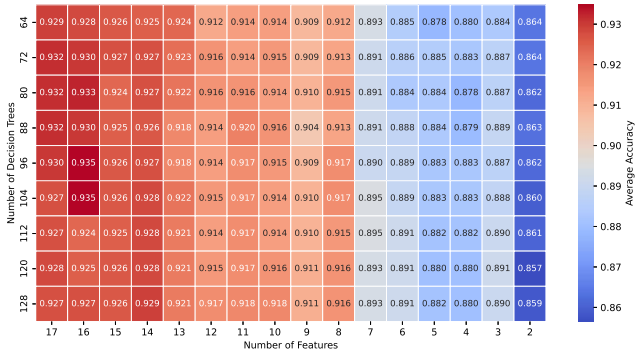


Figure 6.16: Average accuracy heat map.

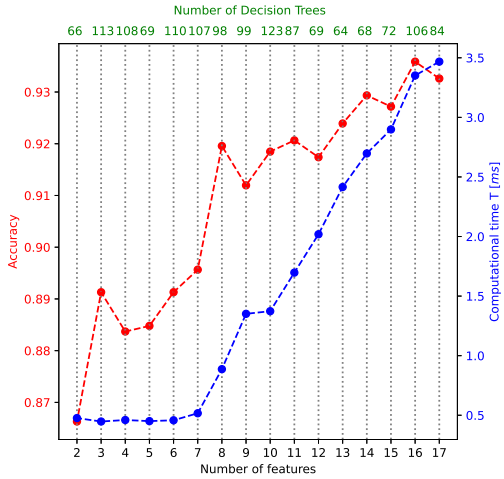


Figure 6.17: Best number of Decision Trees to maximise the accuracy for each feature-set.

pipeline described in Chapter 4. The detection module employs DBSCAN to cluster the filtered point cloud. Simultaneously, features are extracted from each detected cluster and fed into the Random Forest Classifier, with classification results and overall computational cost recorded as KPIs.

Two real-time test scenarios were considered. In Scenario 0, 150 frames were acquired at 5 Hz over 30 s of a small motor yacht (*Motor Boat*) sailing from right to left near the acquisition point. In Scenario 1, 150 frames at 5 Hz over 30 s were acquired of a sailing vessel (*Sailing Boat*) with lowered sails, sailing from left to right at a greater distance. Key characteristics of the two tests are summarised in Table 6.9.

Table 6.9: Test characteristics.

Test	Scan	Frequency	Duration	Distance	Vessel length	Ref. class
Scenario 0	150	5 Hz	30 s	49 m	16 m	*Motor Boat*
Scenario 1	150	5 Hz	30 s	67 m	10 m	*Sailing Boat*

Figure 6.18 shows three RGB reference images for Scenario 0 at the beginning, middle, and end of the acquisition, while Figure 6.19 displays the corresponding 2D-projected LiDAR scans. Out of 150 frames with ground truth class *Motor Boat*, the classifier correctly predicted 140, misclassifying 10 as *Sailing Boat* and none as the other classes, yielding an accuracy of 93.33%. The average computational time for detection, feature extraction, and classification is approximately 12 ms per frame.

Figure 6.18 shows three RGB reference images for Test 1 at the beginning, middle, and end of the acquisition, while Figure 6.21 displays the corresponding 2D-projected LiDAR scans.

Figure 6.18: Reference RGB images of the Scenario 0 acquisition time series.

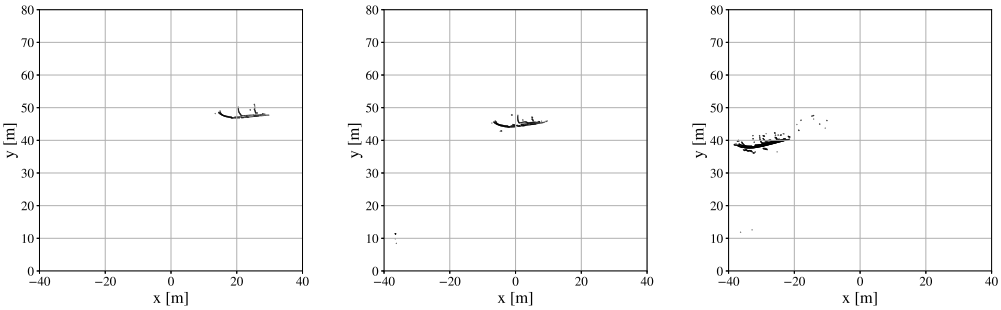


Figure 6.19: LiDAR 2D point cloud of the Scenario 0 acquisition time series.

Out of 150 frames with ground truth class *Sailing Boat*, the classifier correctly predicted 148, misclassifying 2 as *Motor Boat* and none as the other classes, yielding an accuracy of 98.6%. The average computational time for detection, feature extraction, and classification is approximately 9 ms per frame.

To evaluate the combined performance of the detection and classification pipeline, three more complex scenarios (Scenarios 2–4) were selected. Unlike the open-water tests (Scenarios 0 and 1), these scenarios simulate quay-approach operations. Each scenario comprises a single static frame containing multiple boats, either sailing or moored, of varying types, along with portions of the quay. Boat clusters vary in size from fewer than 100 points to over 1,000, with some well-defined and others less distinct, enabling analysis of partial occlusions, point density, and object distance.

Water noise was filtered by removing points with intensity below 10 and heights near sea level.

Figure 6.20: Reference RGB images of Scenario 1 acquisition time series.

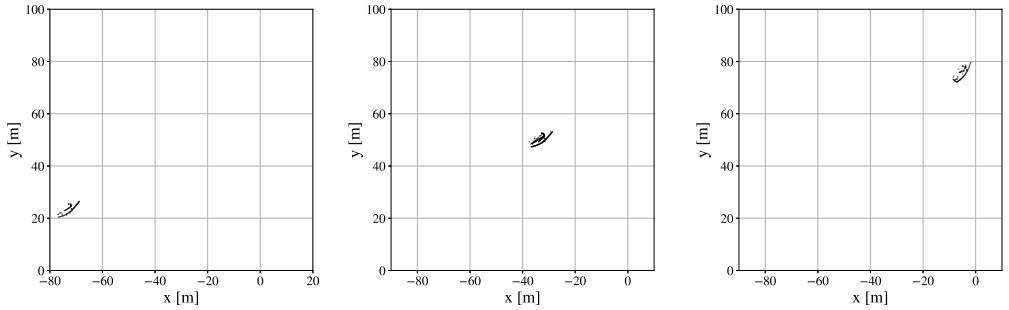
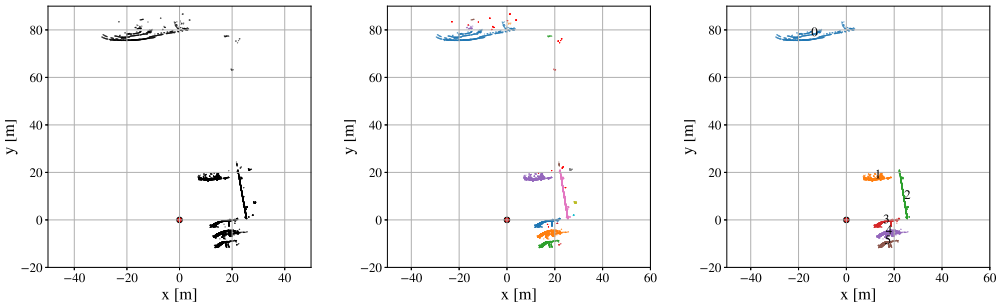


Figure 6.21: LiDAR 2D point cloud of the Scenario 1 acquisition time series.



(a) Scenario 2. **(b)** Clustering on Scenario 2. **(c)** Cluster filtering on Scenario 2.

Figure 6.22: Detection and Classification pipeline tested on Scenario 2.

DBSCAN was applied with $\epsilon = 1$ m and *minpts* = 5. Noise points were excluded, and clusters with fewer than 50 points were discarded. The remaining clusters were classified using the trained Random Forest Classifier.

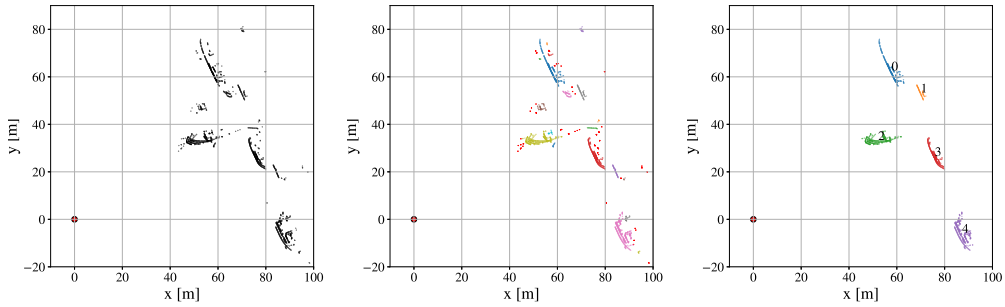
Scenario difficulty increases from 2 to 4: Scenario 2 contains few, well-defined targets with minimal noise; Scenario 3 has some partially occluded or fragmented targets and disturbances such as mooring lines; Scenario 4 features many fragmented or occluded targets with substantial noise.

Figures 6.22–6.24 illustrate the processing workflow for Scenarios 2–4. Figures 6.22a–6.24a show the raw 2D-projected point clouds, while Figures 6.22b–6.24b display DBSCAN-identified clusters (points marked by red X's indicate noise). Figures 6.22c–6.24c show the numbered valid clusters ready for classification. The sensor POV is marked with a red circle with a black border in all figures. Results and cluster information are summarised in Tables 6.10–6.12 for Scenarios 2–4, respectively.

Scenario 2 results demonstrate that the algorithm performs effectively on well-defined scenes with properly segmented targets. Noise caused by hull-induced waves, such as in cluster 0, is

Table 6.10: Scenario 2 clusters' information

Cluster ID	n° of points	Distance [m]	Ref. label	Predicted Label
0	722	78.78	Ship	Ship
1	919	21.14	Motor Boat	Motor Boat
2	1430	25.72	Quay	Quay
3	933	15.42	Motor Boat	Motor Boat
4	2091	17.5	Sailing Boat	Sailing Boat
5	1059	18.96	Motor Boat	Motor Boat

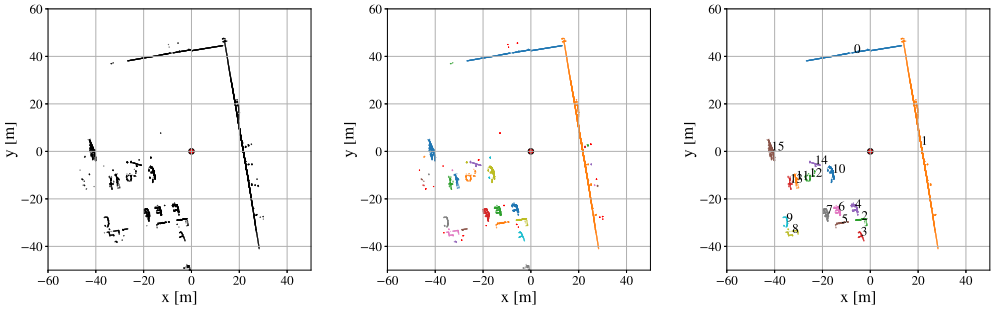


(a) Scenario 3. **(b)** Clustering on Scenario 3. **(c)** Cluster filtering on Scenario 3.

Figure 6.23: Detection and Classification pipeline tested on Scenario 3.

Table 6.11: Scenario 3 clusters' information.

Cluster ID	n° of points	Distance [m]	Ref. label	Predicted Label
0	305	85.28	Ship	Sailing Boat
1	58	88.08	Quay	Quay
2	322	61.98	Motor Boat	Motor Boat
3	259	80.47	Motor Boat	Motor Boat
4	353	87.5	Motor Boat	Motor Boat



(a) Scenario 4 point cloud. **(b)** Clustering on Scenario 4. **(c)** Cluster filtering on Scenario 4.

Figure 6.24: Detection and Classification pipeline tested on Scenario 4.

Table 6.12: Scenario 4 clusters' information

Cluster ID	n° of points	Distance [m]	Ref. label	Predicted Label
0	564	42.02	Quay	Quay
1	3076	21.45	Quay	Quay
2	188	29.07	Quay	Motor Boat
3	74	35.69	Motor Boat	Motor Boat
4	383	24.74	Motor Boat	Motor Boat
5	126	32.46	Quay	Motor Boat
6	306	28.21	Motor Boat	Motor Boat
7	328	32.0	Sailing Boat	Sailing Boat
8	81	47.32	Quay	Motor Boat
9	75	45.91	Motor Boat	Motor Boat
10	690	18.25	Motor Boat	Motor Boat
11	169	33.23	Sailing Boat	Sailing Boat
12	101	27.65	Motor Boat	Motor Boat
13	122	36.09	Sailing Boat	Sailing Boat
14	87	23.79	Quay	Quay
15	499	41.37	Sailing Boat	Sailing Boat

successfully filtered.

Scenario 3 highlights robustness in noisier environments. For instance, cluster 2 shows successful filtering of mooring lines and correct classification. However, accurate segmentation remains critical: in cluster 0, a segmentation error due to shadow cones detaches the stern, which is classified as noise. Small occluded objects are also excluded as valid clusters due to low point counts.

Scenario 4 illustrates the impact of occlusion and partial acquisition on classification accuracy. Classes are generally correctly assigned, except for partially shielded or poorly segmented clusters (e.g., 2, 5, 8). Objects acquired without obstruction are correctly segmented and classified, largely independent of distance or point count. Overall, low point counts or greater distances do not compromise classification if the object's main proportions and features are captured; conversely, nearby objects with medium point counts can be misclassified if occluded or incompletely captured.

Additional tests examined the sensitivity of classification to target truncation. Point clouds were progressively truncated along the first principal direction, and performance was evaluated at each truncation level, recording the percentage at which the first misclassification occurred and the confused class. This simulates partial acquisition due to occlusion or clustering failures. Table 6.13 summarises the results, showing that classification becomes unreliable when truncation exceeds 40%, with the most frequent confusion class reported for each target type.

Table 6.13: Progressive truncation test results.

Class	Truncation [%]	Confusion class
Quay	55%	Motor Boat
Motor Boat	47%	Quay
Sailing Boat	40%	Motor Boat
Ship	56%	Motor Boat

6.1.1.6 Limitations and Sources of Uncertainty

Despite the overall effectiveness of the proposed LiDAR detection and classification pipeline, several limitations and sources of uncertainty remain. The performance is sensitive to point cloud density and occlusions: low-density clusters or partially occluded targets, as observed in Scenario 4, can lead to misclassification or even failure to detect the object. Segmentation errors can further exacerbate this issue, as clustering algorithms may split elongated or irregularly shaped targets into multiple clusters or merge nearby objects, increasing the likelihood of incorrect classification.

Sensor noise and environmental conditions also affect performance. Factors such as surface reflectivity, water reflections, weather conditions, and sensor motion introduce noise and outliers into the point cloud, thereby affecting both clustering and feature extraction. Additionally, certain features, including skewness, kurtosis, and intensity distributions, may vary depending on the sensor viewpoint.

The Random Forest classifier's performance depends on the representativeness of the training data. Unseen object types, extreme shapes, or environmental conditions not present in the training

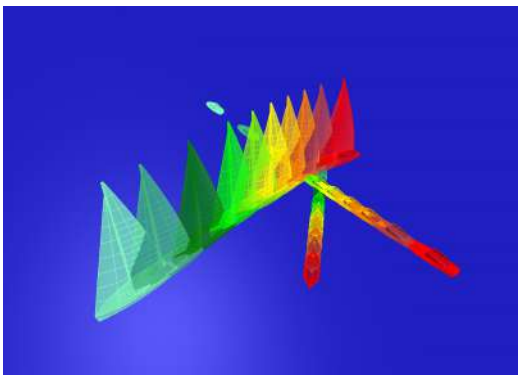


Figure 6.25: 3D virtual scenario.

dataset may reduce predictive accuracy. Finally, the ground-truth annotations used for scenario evaluation are subject to human error and may not perfectly align with LiDAR frames, thereby introducing additional uncertainty into KPI measurements.

Overall, while the proposed pipeline demonstrates strong performance under controlled and moderately complex scenarios, caution should be exercised when generalising to highly cluttered, occluded, or dynamic maritime environments.

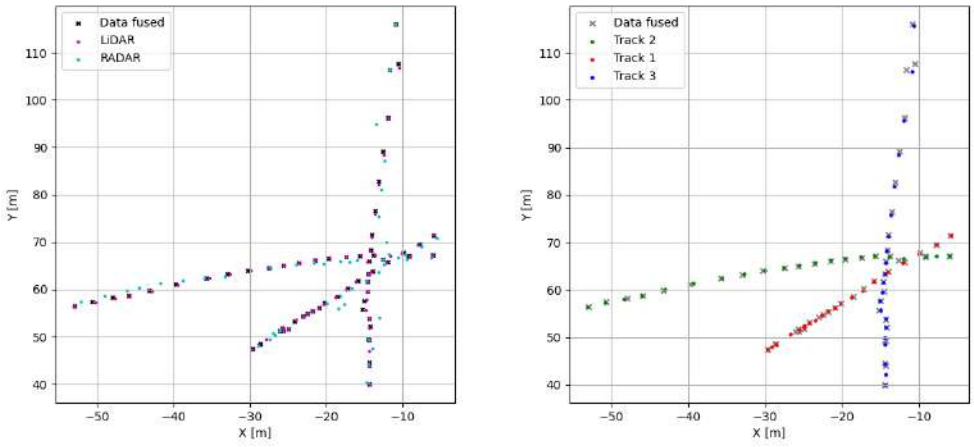
6.1.2 LiDAR - radar decision-level fusion

Preliminary results regarding the decision-level fusion pipeline for LiDAR and Radar detections are reported in this section. The outcomes of this research were documented in technical reports produced within the MOST project.

This work integrates the LiDAR detection pipeline, previously discussed in Section 4.2, with a Radar detection pipeline based on CFAR analysis, developed by the project partners at CNR-IREA. Due to the absence of synchronised acquisitions from the two sensors, an asynchronously managed simulated scenario was employed, which is described in the following subsection.

6.1.2.1 Scenario Description

For pipeline testing, a virtual scenario was generated using the 3D CAD software Rhinoceros. Three vessels of different types, moving at different velocities along intersecting trajectories, were modelled to create a challenging situation for the tracker. LiDAR point clouds were extracted from the 3D models in accordance with the sensor's datasheet specifications, with intensity values ignored. Radar target points were simulated at the centroids of the 3D models, considering the error model provided by the project partners via Monte Carlo simulation. Exact Radar data are withheld for confidentiality. Deliberate misdetections, including missing detections and outliers, were introduced to assess the resilience of the sensor fusion and tracking system. The resulting scenario is illustrated in Figure 6.25, where the colour-map illustrates the temporal evolution.



(a) Sensors' detections and fusion results.

(b) Track assignment and filtering results.

Figure 6.26: Tracking results.

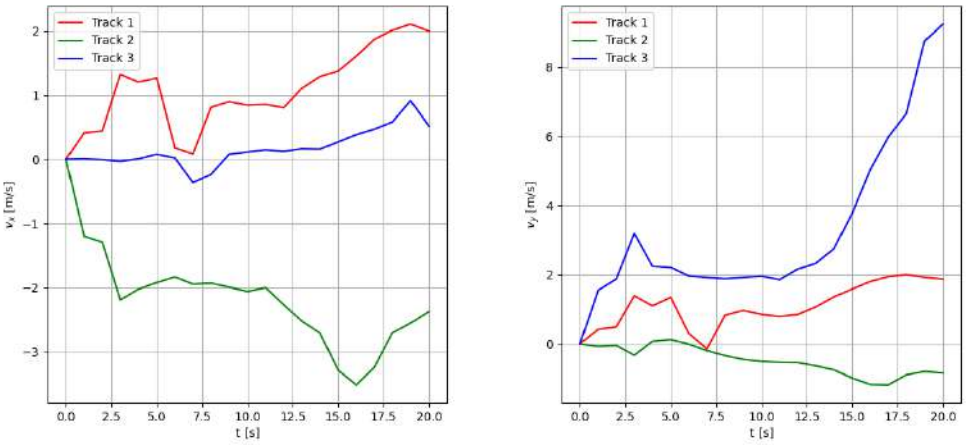
6.1.2.2 Fusion and tracking Results

The tracking performance on this scenario is illustrated in Figures 6.26. Figure 6.26a shows the two sensor acquisitions (LiDAR in violet, Radar in cyan) alongside the fused data (black crosses). It can be noted that fused data remain available even when only one sensor provides a detection, either directly from the single sensor or through the fusion procedure when both are present. In Figure 6.26b, fused data (grey crosses) are overlaid with tracker outputs, including Kalman-filtered estimates and their assignment to individual tracks (green, blue, red). The tracker is shown to smooth the oscillations inherent to point detections and consistently assign points to the correct track.

Finally, Figure 6.27 presents the estimated velocities along the x and y components of the simulation space (Figures 6.27a and 6.27b, respectively).

Since the test is based on a simulation, the ground truth is available, enabling the extraction of KPIs. Table 6.14 reports the main Key Performance Indicators (KPIs) adopted to evaluate the tracking performance for the three targets. In particular, position and velocity accuracy have been assessed through the Root Mean Square Error (RMSE), Mean Absolute Error (MAE), maximum absolute error, and standard deviation of the estimation error.

From the results, Track 1 and Track 2 exhibit comparable performance, with Track 1 showing slightly lower position and velocity errors overall. Conversely, Track 3 exhibits significantly higher RMSE and maximum errors, particularly in position, indicating degraded estimation accuracy and reduced tracking stability for that target. This behaviour is associated with a greater injection of artificial misacquisitions into the radar data and with the vessel's increased speed.



(a) Estimated trajectories of the tracked targets.

(b) Estimated speeds of the tracked targets.

Figure 6.27: Tracking results.

Table 6.14: Tracking performance metrics.

Metric	Track 1	Track 2	Track 3	Unit
RMSE Position	1.600	2.251	4.275	m
MAE Position	1.486	2.146	3.092	m
Max Position error	3.118	3.524	10.868	m
Std Position error	0.593	0.677	2.952	m
RMSE Velocity	0.941	1.030	1.837	m/s
MAE Velocity	0.654	0.797	1.363	m/s
Max Velocity error	2.798	2.075	3.789	m/s
Std Velocity error	0.677	0.651	1.231	m/s

6.1.2.3 Limitations and Sources of Uncertainty

Since the testing was conducted in a simulated environment and did not include the sensor data acquisition stage, a computational cost analysis was not performed, as it would not have been representative of the actual system. Moreover, the evaluation was based on a simplified simulation due to the limited availability of real-world data. Although the simulation included ground truth information and the results are encouraging, the generalizability of the findings remains moderate. A comprehensive real-world testing campaign is therefore necessary to fully validate the approach and is planned for future development.

6.1.3 LiDAR - RGB camera decision-level fusion

This section presents the results of testing the decision-level fusion framework for LiDAR and RGB cameras, as described in [36].

6.1.4 Calibration

The 3D LiDAR points are mapped onto the 2D image plane using a pinhole camera projection model (see Chapter 4.

The camera intrinsic parameters are derived from the camera producer's data. The extrinsic transformation between the LiDAR and the camera is initially determined from the sensor mounting and then refined based on the acquired data.

Given that LiDAR provides full 360° horizontal coverage, the point cloud is cropped to match the camera's horizontal Field Of View (FOV). Vertical cropping is unnecessary because the LiDAR's vertical coverage is already contained within the camera's vertical FOV (31° against approx. 40°).

6.1.4.1 Scenarios and testing methods

Based on the data from the experimental campaign, two test cases were defined to assess the proposed processing chain.

Scenario 1 represents the baseline condition. It includes 90 time-synchronised LiDAR–camera scan pairs acquired at 5 Hz, for a total duration of 18 s. The scene contains a single 18 m white motor yacht moving along a straight path at nearly constant speed over the entire sequence.

Scenario 2 is more complex and includes 143 synchronised LiDAR–camera scan pairs acquired at 5 Hz, corresponding to about 29 s. The scene involves three targets of different sizes and trajectories. A 24 m motor yacht, characterised by a black hull and white superstructure, moves along a straight background path throughout the sequence. From frame 44 to frame 139, a 6-m outboard-powered dinghy follows an oblique path and leaves the scene in the opposite direction. From frame 116 to the end of the sequence, a 5.5 m outboard-powered RHIB travels in planing conditions along an oblique trajectory towards the sensor.

For the DBSCAN clustering stage, the neighbourhood radius was set to 5 and the minimum number of points to 15. Track initiation required at least four consecutive detections.

Both scenarios were processed by starting from raw sensor measurements acquired at the native sampling rate and then passing them through the proposed pipeline. At each time step, the following key performance indicators (KPIs) were computed:

- *LiDAR candidates*: number of detections extracted from the LiDAR point cloud;
- *Camera candidates*: number of detections produced in the image plane by the YOLO network;
- *Confirmed candidates*: number of targets validated by the decision-level fusion module;
- *Active tracks*: number of tracks maintained by the GNN-based tracker;

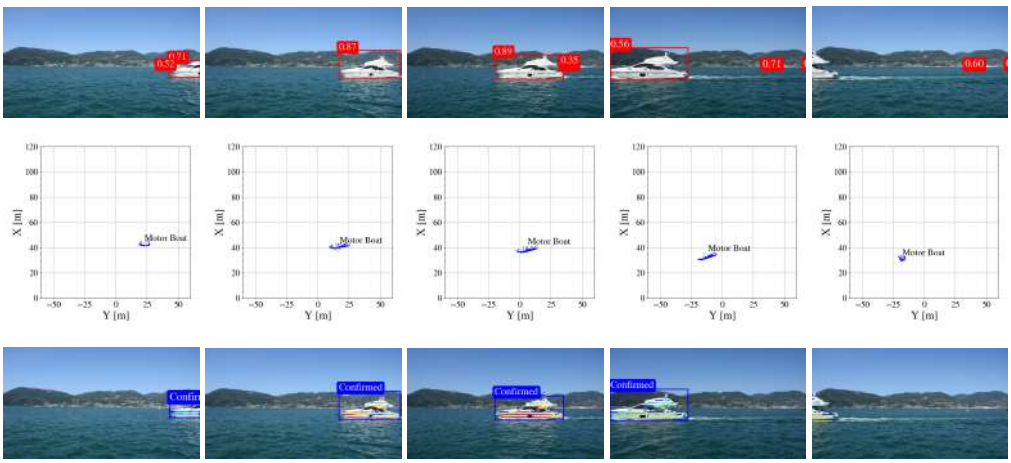


Figure 6.28: Multi-modal detection results on scenario 1.

- *Track state*: estimated kinematic state (position and velocity) of each active track expressed in the LiDAR reference frame;
- *Processing time*: total computation time from raw sensor data to the final set of track states.

For qualitative analysis of the LiDAR data, a point-cloud heatmap was generated for each scenario. The scene was divided into a two-dimensional grid of square cells with a side length of 25 cm. For each cell, the number of LiDAR points accumulated over time was computed. This representation highlights the spatial distribution of returns, their density, and the presence of noise.

6.1.4.2 Scenario 1 results

Fig. 6.28 shows a set of representative frames (frames number 10, 24, 41, 70, 86) for *Scenario 1*, highlighting the detection results at different processing stages.

The first row reports the RGB images with the bounding boxes generated by the YOLO network.

The second row presents the filtered LiDAR point cloud after DBSCAN clustering, displayed in Bird's-Eye View (BEV). Each cluster is labelled according to the class assigned by the Random Forest classifier. The LiDAR sensor is placed at the origin of the reference frame, i.e., at coordinates $(0, 0)$.

The last row depicts the projection of the LiDAR point cloud onto the image plane, with points coloured by their reflectivity values. The bounding boxes refer only to obstacles identified as moving by the decision-level fusion module.

Fig. 6.29 summarises the temporal evolution of detections and tracks.

The upper plot shows the number of camera candidates (red dashed line), LiDAR candidates (purple dash–dot line), and confirmed candidates after fusion (thick solid green line). The black vertical dashed lines mark the frames displayed in Fig. 6.28.

The lower plot reports the number of active tracks at each time step (blue line).

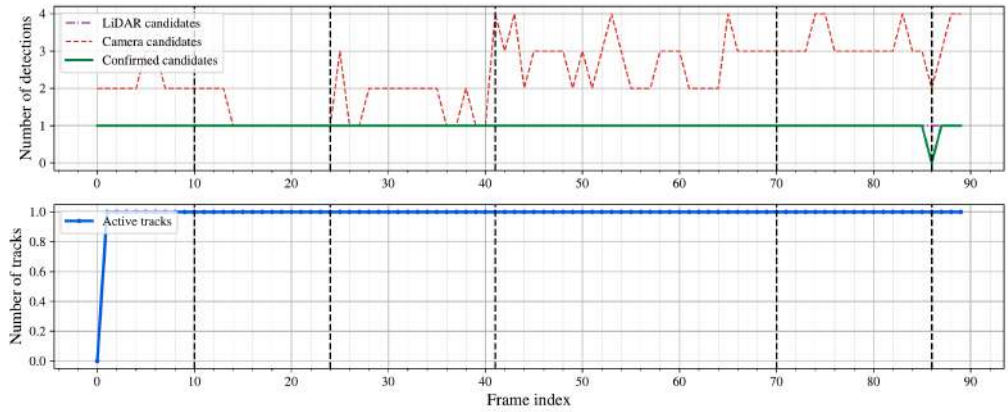


Figure 6.29: Detection and tracking overview in scenario 1

The overall detection, fusion, and tracking chain correctly identified the correct number of active tracks in 96% of the frames compared to the ground truth. The agreement increases to 100% when compared with the expected system behaviour, excluding the four frames required for track initialisation.

Fig. 6.30a shows the spatial distribution of the cluster centroids (dark grey) together with the estimated track positions provided by the tracker (blue).

Fig. 6.30b reports the estimated velocity components along the x and y axes and the corresponding speed magnitude. All quantities are expressed in the LiDAR reference frame.

Assuming a reference speed between 5 and 7.5 kn (including small variations), the corresponding Froude number ranges approximately from 0.19 to 0.29. This interval is consistent with a fully displacement regime and matches the observed navigation behaviour.

6.1.4.3 Scenario 2 results

Fig. 6.31 reports a set of representative frames (frames number 40, 70, 94, 112, 133)for *Scenario 2*, highlighting the outputs of the detection stages.

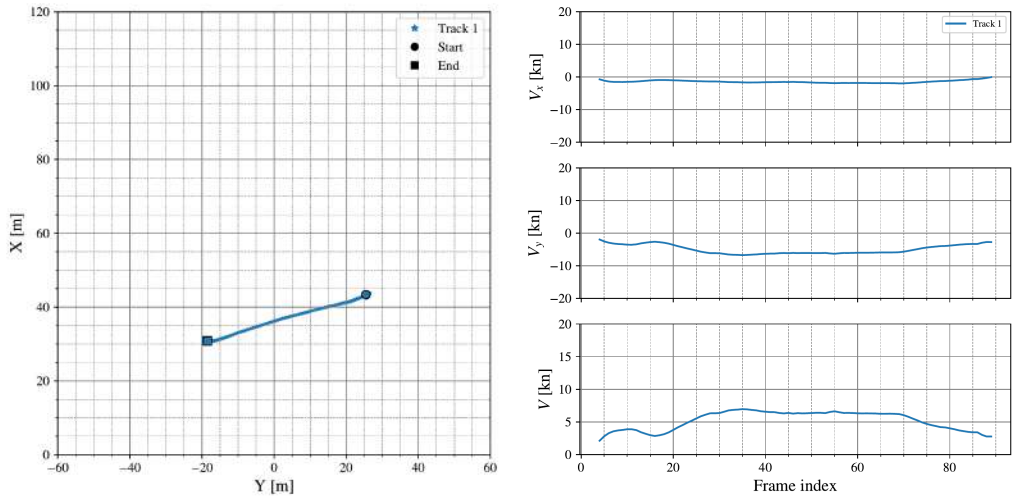
The first row displays the RGB images with the bounding boxes generated by the YOLO network.

The second row shows the filtered and clustered LiDAR point cloud in Bird's-Eye View (BEV). Each cluster is labelled according to the class assigned by the Random Forest classifier. The LiDAR sensor is positioned at the origin of the reference frame, i.e., at coordinates $(0, 0)$.

The last row shows the projection of the LiDAR point cloud onto the image plane, with points coloured by reflectivity. The bounding boxes refer only to targets classified as moving by the decision-level fusion module.

Fig. 6.32 summarises the KPIs related to detections and tracking performance over time.

The upper plot shows the number of camera candidates (red dashed line), LiDAR candidates (purple dash–dot line), and confirmed candidates after fusion (thick solid green line). The lower



(a) Trajectory estimation .

(b) Speed estimation.

Figure 6.30: Tracking results in scenario 1

plot reports the number of active tracks at each frame (blue line). The black vertical dashed lines indicate the frames displayed in Fig. 6.31.

Overall, the detection, fusion, and tracking chain produced the expected number of active tracks in 84% of frames relative to the ground truth. This percentage increases to 92% when compared with the expected system behaviour, excluding the four frames required for track initialisation.

Fig. 6.33a shows the spatial distribution of the cluster centroids and the corresponding estimated track positions. Confirmed LiDAR centroids are plotted in dark grey, while the tracker estimates are displayed using different colours for each target: Track 1 in blue, Track 2 in orange, and Track 3 in green.

Using the same colour convention, Fig. 6.33b reports the estimated velocity components along the x and y axes, together with the resulting speed magnitude. All quantities are expressed in the LiDAR reference frame.

The motor yacht (Track 1) reaches speeds between 5 and 10 kn, considering moderate fluctuations. This corresponds to a Froude number in the range 0.17–0.34. The outboard-powered dinghy (Track 2) travels at approximately 5–7 kn, with a Froude number of 0.33–0.47. Both vessels operate in a displacement regime.

The RHIB (Track 3) moves at about 15–17.5 kn, corresponding to a Froude number of approximately 1.05–1.23, consistent with pre-planing or planing conditions.

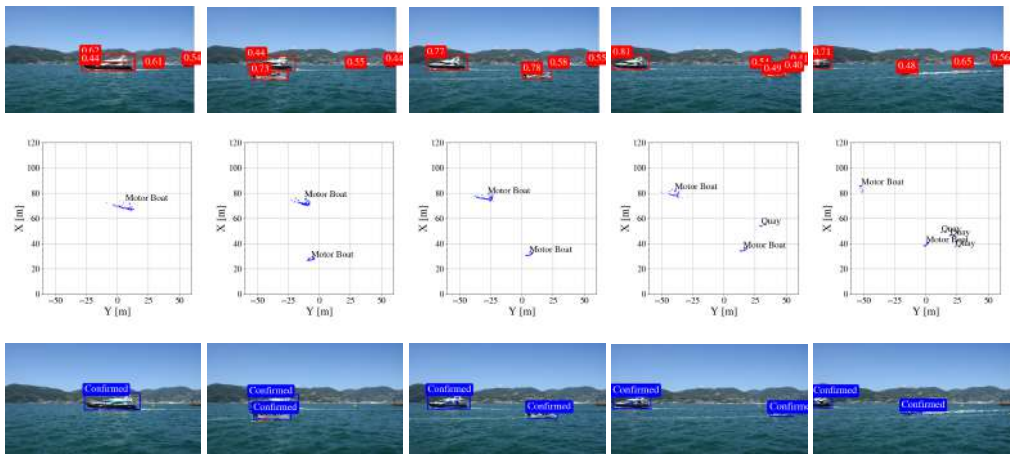


Figure 6.31: Multi-modal detection results on scenario 2.

6.1.4.4 Discussion

The results in Fig. 6.28 show the behaviour of the proposed multimodal framework in a single-target configuration. In some frames (10, 41, 70 and 86), the YOLO detector generates false positives, mainly background buoys misclassified as boats due to the limited specificity of the general-purpose training dataset. Occasional duplicate detections are also observed. These effects are removed by the decision-level double-confirmation logic, which validates only targets that are consistently detected. As a result, the set of confirmed objects remains stable.

This behaviour is reflected in Fig. 6.29. The number of camera candidates is consistently higher than the ground truth (equal to 1), indicating that a COCO-trained YOLO model tends to misclassify static maritime structures. In contrast, the number of LiDAR candidates closely follows the true target count. The fusion strategy suppresses all false positives. In frame 86, the yacht is partially outside the camera's field of view and is not detected by YOLO; therefore, no confirmation is issued. However, the tracker propagates the existing track and prevents track loss. After the four frames required for initialisation, the number of active tracks remains correct, showing that a strict confirmation policy does not compromise tracking continuity.

Figs. 6.30a and 6.30b report coherent position and velocity estimates. The speed profile—initial acceleration up to frame 30, quasi-steady motion until frame 70, and subsequent deceleration—is consistent with the spatial trajectory and with the yacht manoeuvre, including speed reduction to about 2.5 kn before port entry.

A more demanding evaluation is provided by *Scenario 2*, which includes multiple targets, partial occlusions and high-dynamic motion.

As shown in Fig. 6.31, the vision branch again produces several false positives and duplicate detections. In this scenario, LiDAR processing is also affected by environmental disturbances and target interactions. In frame 112, the RHIB is partially occluded by the outboard boat, generating a LiDAR shadow region and a sparse point cloud; the target is detected but misclassified. In

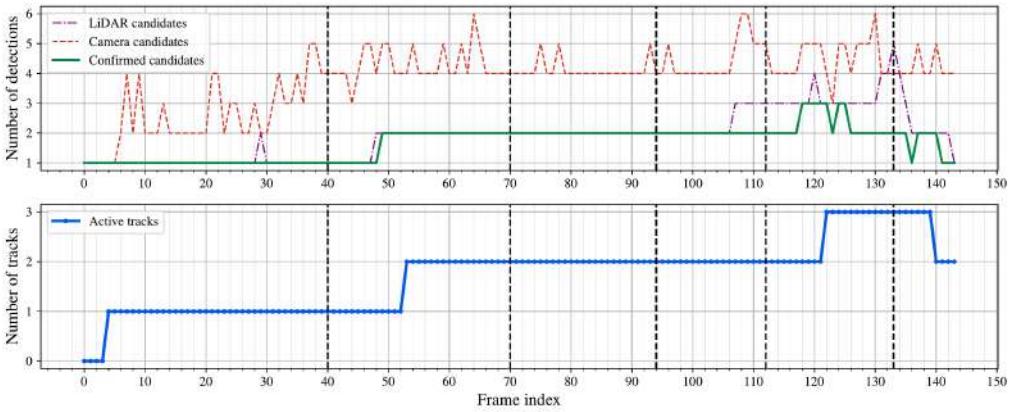


Figure 6.32: Detection and tracking overview in scenario 2

frame 133, the wake produced by the planing RHIB survives filtering and appears as fragmented clusters, sometimes misclassified as quays. Moreover, the black yacht generates fewer returns than expected because the dark hull and long range reduce reflectivity.

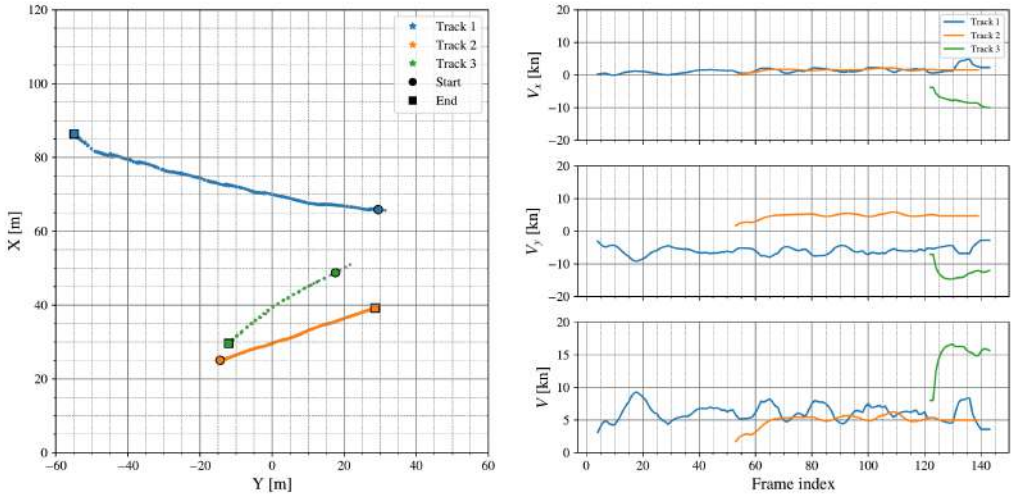
These effects appear in Fig. 6.32. The number of camera candidates remains high, and increases as new targets enter the scene. LiDAR detections are reliable until the RHIB appears. From frame 105 onward, the third target is acquired at long range and high speed under partial occlusion, yielding sparse measurements and delayed confirmation. From frame 120, once the RHIB is closer and fully visible, detection and classification stabilise. However, its strong wake occasionally increases the number of LiDAR candidates beyond the ground truth.

The double-confirmation logic correctly validates the first two targets up to frame 100. The third target experiences a confirmation delay of about 15 frames (3 s), including 4 frames (0.8 s) required for track initialisation. Despite these effects, the number of active tracks remains stable, showing that the tracker compensates for detection-level fluctuations.

As shown in Fig. 6.33a, no track switching or fragmentation occurs. The estimated velocities in Fig. 6.33b are consistent with the expected vessel dynamics. The black yacht exhibits speed oscillations, primarily due to long-range LiDAR limitations that introduce variability in centroid estimation and, consequently, in velocity estimates. Since the vessels are non-cooperative, independent speed measurements are not available for further validation.

6.1.4.5 Computational time

A histogram of the computational cost for each scenario scan is presented in Fig. 6.34 to demonstrate real-time compliance; the computational cost includes the entire pipeline, from raw data parsing to the end. The average computational cost is approximately 104 ms and always lies between 80 and 120 ms. The 200 ms threshold, corresponding to the time constraint imposed by the 5 Hz LiDAR frequency, is indicated in red.



(a) Trajectory estimation .

(b) Speed estimation.

Figure 6.33: Tracking results in scenario 2

6.1.4.6 Limitations and Sources of Uncertainty

Despite the encouraging results, several limitations must be acknowledged. System performance remains sensitive to the reliability of detection and classification, especially in the presence of occlusions, limited visibility, or partial sensor coverage.

Moreover, the vision module relies on a general-purpose detector not specifically trained for maritime environments. The adoption of a domain-specific detection model would likely improve robustness and reduce false positives. The development and validation of such dedicated solutions, therefore, remain open research directions.

An additional source of uncertainty concerns the ground truth. Since the targets are non-cooperative, no reference measurements of position or velocity are available. Ground-truth

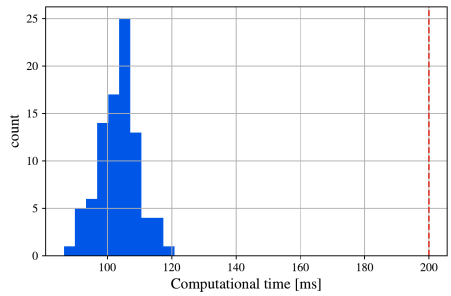


Figure 6.34: Histogram of the computational time.

information is limited to the number of active tracks, as assessed by expert supervision. As a result, a quantitative validation of the estimated kinematic states is not possible.

Finally, the analysis is based on only two experimental scenarios. Although the results are consistent and promising, the limited dataset limits the methodology's generalisability. A broader experimental campaign would be required to assess the scalability and robustness of the proposed framework under more diverse operational conditions.

6.1.5 LiDAR thermal imaging data-level fusion

This section reports the results of testing the LiDAR–thermal imaging fusion pipeline reported in [34].

Section 6.1.5.1 provides a preliminary analysis supporting sensor fusion, along with the YOLO network's training and validation metrics. Section 6.1.5.2 presents results from the complete pipeline, from multi-sensor acquisition to the generation of the target point for the rescue path.

The results are also compared to single-modality detection or late-fusion strategies. Late fusion can be implemented in two ways: (i) image-based detection with LiDAR spatial characterisation, where detection and classification rely solely on image data (e.g., thermal) and LiDAR provides spatial context via perspective projection. In this case, detection performance matches the single-modality branch, with only spatial characterisation differing (see Section 6.1.5.1); (ii) independent detection and association, where thermal data are processed by a detector and LiDAR data by a separate clustering pipeline, with the results subsequently associated. This approach is highly sensitive to parameter tuning, association strategy, and candidate confirmation, making objective performance comparison challenging. For this reason, only the computational cost of this approach is monitored in this work (see Section 6.1.5.3).

6.1.5.1 Rough weather influence

An analysis was conducted to demonstrate the necessity and benefits of a multi-modal approach for human-in-water detection. Specifically, a YOLOv8 network was trained using thermal images only, and another on 3-channel multi-source (thermal–LiDAR) images.

The base training dataset, referred to as *Original*, consists of 6,000 samples with standard augmentations (vertical/horizontal flips, scaling), excluding environmental disturbance augmentations. Two levels of adverse-condition augmentation, *Augmented* and *Augmented+*, were then applied. Both thermal-only and multi-modal YOLOv8 detectors were trained on 80% of the Original dataset and tested on the unseen 20% of the Original, Augmented, and Augmented+ datasets, using a 5-fold cross-validation scheme. Figure 6.35 reports average precision, recall, mAP50, and F1-score across folds.

The two models perform similarly on the *Original* dataset; however, as adverse condition severity increases, the multi-modal model degrades less than the thermal-only model. This indicates that the multi-modal detector generalises better to challenging conditions, likely by leveraging redundant and complementary features from both modalities, thereby maintaining useful performance even under harsh augmentations.

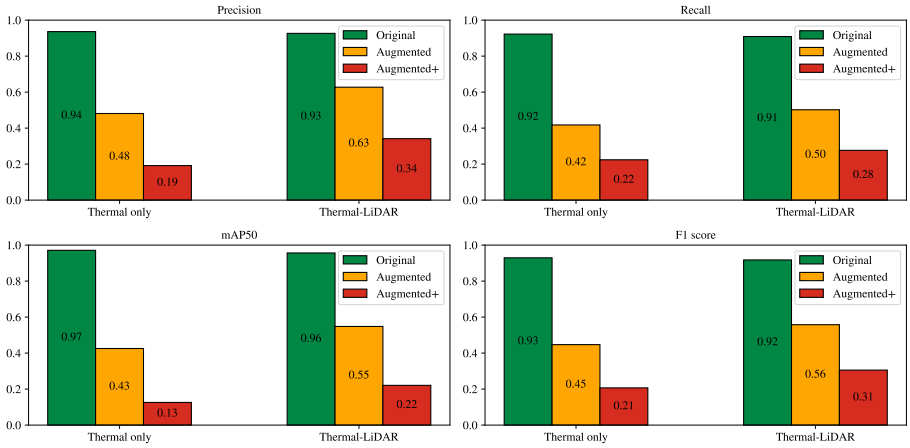


Figure 6.35: Comparison of the metrics of the YOLO detector trained on thermal-only images and thermal-LiDAR combined images with varying augmentation. Note that both detectors are only trained on the *Original* samples.

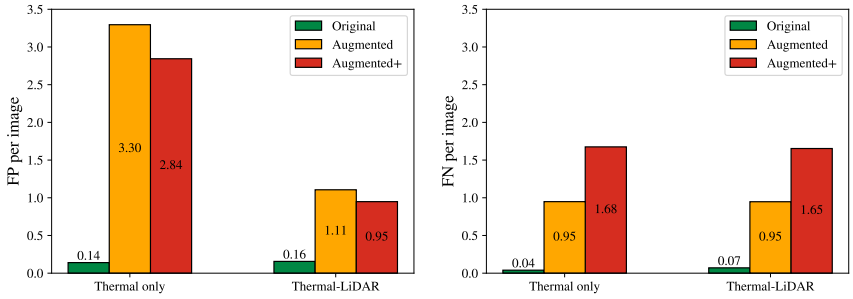


Figure 6.36: Average number of false positives (FP) and false negatives (FN) per image on the validation set.

Following the same evaluation, Figure 6.36 shows the average number of false positives and false negatives per image as a function of augmentation level. In Search and Rescue scenarios, minimising false positives and negatives is critical. The multimodal model, aided by LiDAR input, more effectively distinguishes true targets from false ones. The apparent reversal in the false-positive trend for the *Augmented+* set is attributed to severe disturbances that reduce detectable features, thereby limiting both correct detections and false positives.

Overall, these results highlight that the early-fusion multimodal approach is more robust to adverse conditions than a single-modality or late-fusion strategy.

To evaluate the YOLO network's detection and classification performance on multi-source images, a k-fold cross-validation was performed using the *Augmented* dataset, which contains a balanced mix of non-augmented, mildly augmented, and strongly augmented images to enhance

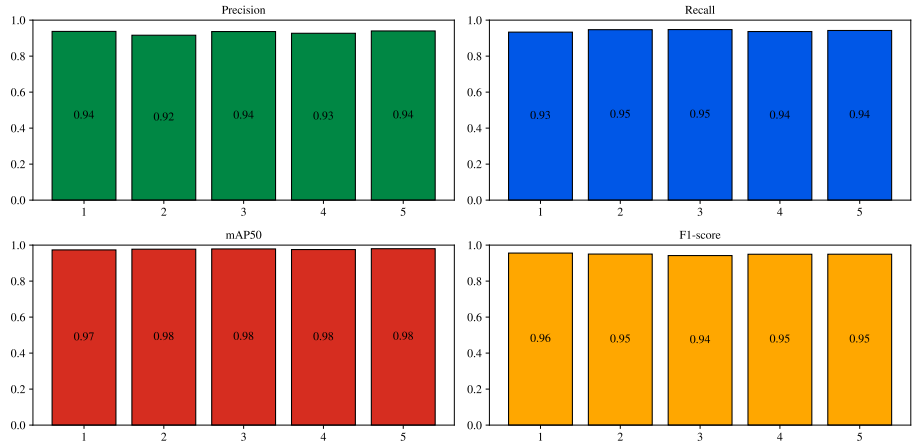


Figure 6.37: 5-fold cross-validation metrics results.

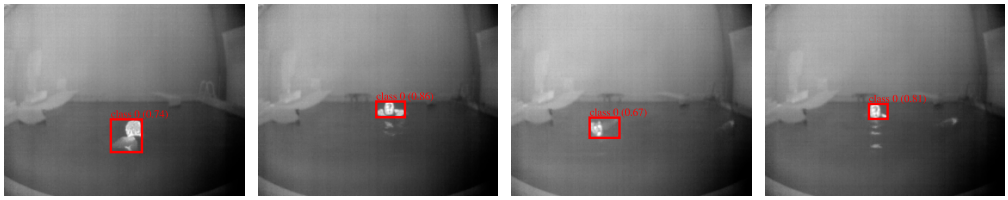


Figure 6.38: Yolo detection on the multi-source image.

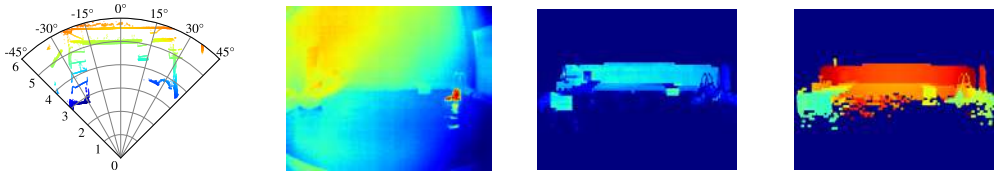
generalisation.

Figure 6.37 reports the results in terms of precision, recall, mAP50, and F1-score, with each column representing a k-fold permutation.

To provide a visual representation of the YOLO detector's output, Figure 6.38 shows the bounding boxes obtained on the three channels for three samples from the dataset. Rows correspond to the three samples, while columns represent the channels of the multi-source image, namely thermal, reflectivity, and range.

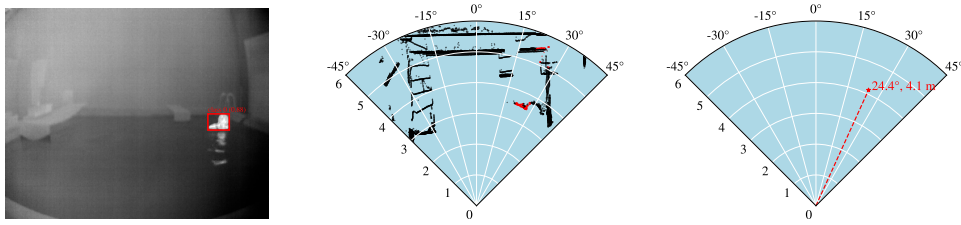
6.1.5.2 Sensor-to-rescue results

The trained YOLO model is embedded throughout the pipeline, which begins with the sensor and culminates in estimating the human's position in water. The input data is shown in Figure 6.39 within the field of view (FOV) of a hypothetical ASV/USV. Figure 6.39a displays the point cloud in BEV, clipped to the camera's FOV, using a colourmap proportional to the range; Figure 6.39b



(a) LiDAR point cloud. **(b)** Thermal image **(c)** Reflectivity image. **(d)** Range image.

Figure 6.39: Input data of the pipeline.



(a) Detection in the thermal image. **(b)** Survivor point candidates in the navigation plane. **(c)** Survivor candidate position in the navigation plane.

Figure 6.40: Survivor detection on thermal image and LiDAR point cloud.

shows the corresponding thermal image. Figures 6.39c and 6.39d show the LiDAR-derived images of the intensity and range fields, respectively.

Figure 6.40 illustrates the pipeline output. Specifically, Figure 6.40a shows YOLO network detections overlaid on the thermal image for visual clarity. Figure 6.40b presents the ASV navigation plane with LiDAR points in BEV: black points represent the full scan, while red points correspond to the region identified by YOLO as the human-in-water. Figure 6.40c shows the extracted target point in range and relative heading, indicating the survivor's location. This example represents a worst-case scenario in which the image-plane region includes LiDAR points that do not belong to the survivor, making target-point estimation challenging.

Figure 6.41 shows histograms for the three channels. In the range channel, adaptive thresholding distinguishes two spatial regions within the bounding box (human and debris). The positions of these pixels are associated with thermal intensity and LiDAR reflectivity, and using the median of these values allows effective separation of the survivor from surrounding objects. Figure 6.42 shows the final separation, Figure 6.42a highlights the points identified as the human in water, and Figure 6.42b presents the resulting Target Point.

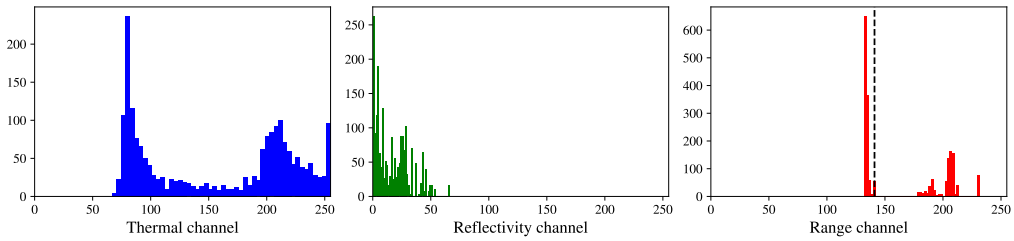
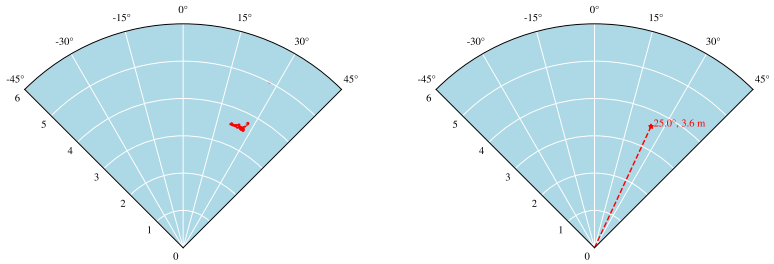


Figure 6.41: Histogram of the Region Of Interest for the 3 channels.



(a) Survivor's LiDAR points. **(b)** Survivor's position.

Figure 6.42: Survivor detection in the navigation plane.

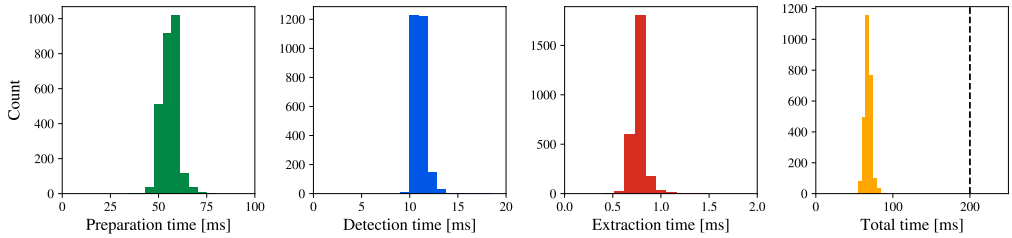


Figure 6.43: Pipeline computational times distribution.

6.1.5.3 Computational cost analysis

To assess real-time compliance, the system's computational cost was analysed over 2,500 scans on the hardware described in Section 3.5, with YOLOv8 detection executed on the GPU and all other processing on the CPU.

The total processing time is divided into three components: *Preprocessing Time*, for sensor data acquisition and generation of the three-channel multi-source image; *Detection Time*, for YOLO-based detection; and *Extraction Time*, for deriving the survivor's target point in the 3D reference frame from the detected bounding box. The sum of these components constitutes the *Total Time*.

Figure 6.43 summarises the results graphically, with numerical values provided in Table 6.15.

Table 6.15: Complete pipeline computational times.

Time	**Mean value** [ms]	**Standard deviation** [ms]
Preprocessing	55.86	4.19
Detection	11.82	0.61
Extraction	0.76	0.07
Total	68.46	4.57

The real-time constraint is 200 ms, corresponding to the interval between consecutive LiDAR scans at 5 Hz, as shown by the dashed black line in Figure 6.34. The average computational cost remains below 100 ms, allowing operation at higher LiDAR frequencies, although this may reduce spatial resolution and affect performance. Preprocessing, which dominates the runtime, was not optimised for parallel execution, leaving potential for implementation on less powerful embedded platforms.

For comparison, a late-fusion pipeline combining a thermal-only YOLO detector with LiDAR DBSCAN clustering [32] was tested, yielding an overall computational cost of 188.57 ms (standard deviation 13.68 ms). In contrast, the early-fusion strategy using multi-modal images demonstrated greater robustness to environmental disturbances, improved precision and recall, lower false-positive rates (see Section 6.1.5.1), and required only 36% of the computational cost.

6.1.5.4 Limitations and Sources of Uncertainty

Despite promising results, several limitations should be noted. The dataset relies on artificially augmented images that cannot fully replicate real-world disturbances, potentially leading to an overestimation of detector robustness. Only a single subject was used, limiting exposure to natural variations in body shape and thermal signature. Moreover, the experiments were conducted in a carefully controlled laboratory environment that does not fully replicate open-water conditions. Finally, the influence of sensor-to-target distance on detection performance was not considered, which may affect LiDAR density, thermal resolution, and feature extraction. While early fusion shows advantages under controlled conditions, caution is needed when generalising to real-world scenarios.

6.1.6 State estimation

6.1.6.1 Kalman Filter state estimation

The state estimate from Kalman filtering served as the basis for feedback in several experimental campaigns involving integration with other modules. Therefore, the method is evaluated on real data to assess its capabilities and performance. Data obtained from the GNSS/INS system are subsampled at 10 Hz and then processed according to Section 4. To validate the state estimation feedback, a recording obtained during early-stage remote-control testing campaigns (see Chapter 5) is used.

- Timestep Accuracy: The mean and standard deviation of the time interval between samples are calculated to ensure compliance with the expected signal sampling frequency of 10 Hz. It is used to assess the capability of the state estimation system.
- Residual Analysis: The statistical properties of the residuals are analysed. The k^{th} iteration residual is defined as the difference between the predicted state vector projected in the measure space and the measure vector itself. This allows evaluation of whether the KF accurately tracks the object's true dynamics given the measured quantities. It is used to assess the capability of the state estimation system. In addition, the distribution of residuals is checked for normality.
- Evaluation of Estimated Quantities: A focused analysis is performed on the quantities estimated (rather than directly measured) by the system, specifically the velocity components V_x and V_y. In a trajectory leg characterised by pure surge at a constant speed, the velocities estimated by the filter are compared with those computed from finite differences of position data. The KPI is expressed in terms of the Root Mean Square Error (RMSE) relative to a reference average velocity. It is used to assess the capability of the state estimation system.

The first KPI reported concerns the verification of timestep consistency. Log files were analysed to obtain the Unix timestamps of each message sent by the state estimation module during a two-day experimental campaign. The average timestep obtained from this analysis was 0.100067 s, with a

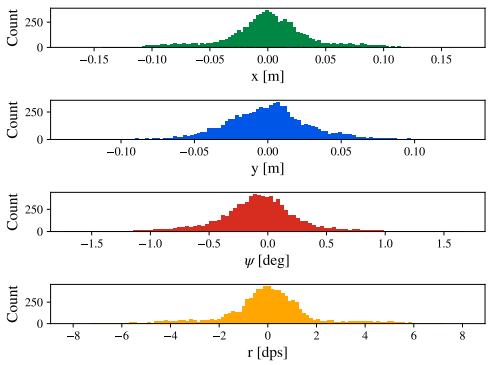
standard deviation of 0.00028 s, confirming both the system's ability to operate at the desired 10 Hz frequency and its capability to maintain signal frequency stability at levels sufficient for the application.

Using the state estimation data relative to the path shown in Figure 6.52b, the residuals were evaluated for each iteration. In particular, Table 6.16 summarises the major statistics of interest. The residual values are generally close to zero.

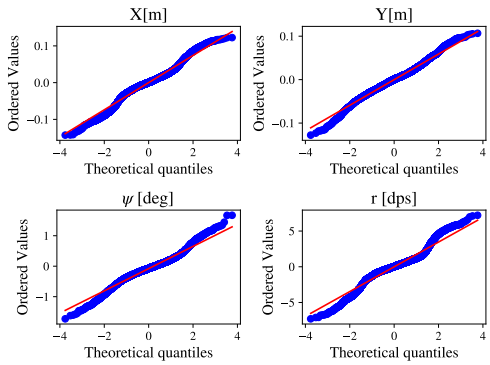
Table 6.16: Statistical KPI indicators of residuals for the estimated states.

Stat.	$x[m]$	$y[m]$	$\psi[deg]$	$r[dps]$
Mean	0.00019	-0.00015	-0.01392	0.02079
σ	0.03752	0.02960	0.37001	1.78554

Furthermore, to assess the normality of the residuals, histograms illustrating the residual distribution for each measured variable (see Figure 6.44a) are provided, along with the corresponding Q–Q plot (see Figure 6.44b).



(a) Residuals distribution histograms.



(b) Q–Q plot.

Figure 6.44: Residual analysis for each measure variable.

Finally, an analysis was conducted on the quantities estimated but not directly measured by the filter, specifically the two velocities V_x and V_y. Specifically, a pure surge motion test was performed in remote-control mode by sending a constant-velocity command to the vehicle. A reference leg was extracted after the acceleration phase, and both the velocities estimated by the filter and those computed through direct differentiation of position were considered. The velocity signals were then compared against the reference average speed of the leg; statistics of the comparison are reported in Table 6.17.

In conclusion, the Kalman-filter–based feedback system can effectively estimate velocities while simultaneously smoothing the measured quantities, yielding normal residuals and reliable estimates relative to the measured values.

Table 6.17: Speed components RMSE on a reference value.

Component	Ref value [m/s]	RMSE_{KF}	RMSE_{Raw}
V_N	0.42	0.035	0.25
V_E	-0.32	0.043	0.15

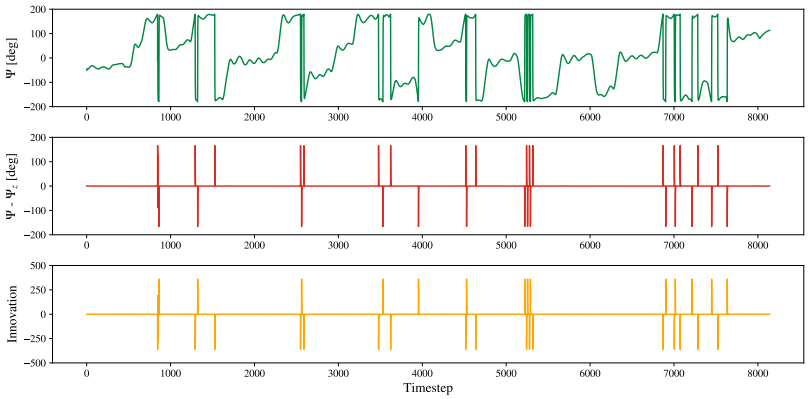


Figure 6.45: Angle estimation using the Kalman Filter.

6.1.6.2 Unscented Kalman Filter

As discussed in the methodology, the Unscented Kalman Filter (UKF) better handles nonlinear system dynamics and measurements compared to the standard Kalman Filter (KF), which is simpler and computationally cheaper. In addition to the use of more sophisticated state transition models, a key issue arises with angular states defined in $[-\pi, \pi]$: the linear KF operates directly on the angle, leading to discontinuities at wrap-around points, while the UKF uses a non-singular sine–cosine representation.

Analysis of a real-world dataset shows that the KF produces large errors and innovations near angular singularities, whereas the UKF provides continuous, robust angle estimation by correctly propagating the nonlinear dynamics of the sine–cosine components. Figure 6.45 illustrates these effects, highlighting the KF's limitations at wrap-around points.

Figure 6.46 shows the estimation results obtained using the Unscented Kalman Filter (UKF). From top to bottom, the estimated sine and cosine components, their corresponding innovations, the heading angle obtained a posteriori via the arctangent of the estimated components, and the difference between the estimated and the ground-truth angle are reported.

The UKF provides a more stable angular estimate, with smooth innovations, by using the sine–cosine representation that naturally handles the circular nature of heading and avoids the discontinuities of a linear KF in $[-\pi, \pi]$. Finally, the UKF and KF were evaluated in terms of computational cost, as reported in Figure 6.47. The results show that the UKF's mean computational cost is approximately 7 times that of the standard KF. Although this difference is not critical for typical operations, it may become significant in onboard or edge computing scenarios with limited

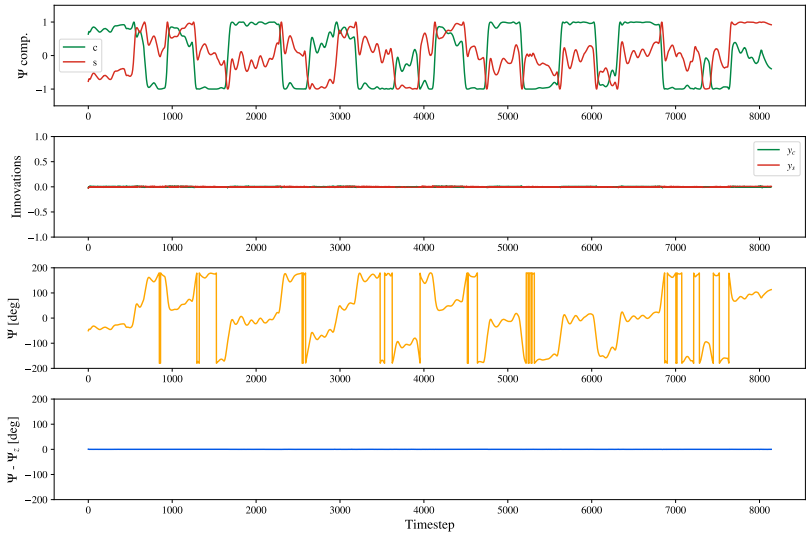


Figure 6.46: Angle estimation using the Unscented Kalman Filter.

processing resources.

Despite the KF producing abrupt but short-lived errors, these transients can be absorbed by downstream modules, preserving the filter's simplicity and computational efficiency, crucial for real-time, low-resource platforms. For this reason, the standard KF was used in the experimental campaigns, while UKF deployment in field applications will be explored in future work.

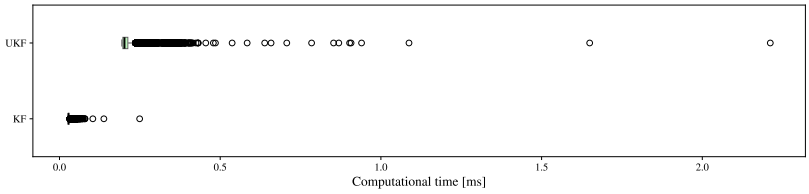


Figure 6.47: Computational time comparison between UKF and standard KF.

6.1.6.3 Limitations and Sources of Uncertainty

The main limitation of this study is the lack of ground truth for the USV trajectory. Consequently, it is not possible to perform a precise quantitative comparison between the estimation methods presented in this work. The evaluation in this section illustrates the module's functionality and behaviour, which is then applied in the experimental campaigns.

Figure 6.48: ChAruco board pose estimation slideshow.

6.2 Experimental campaigns results

6.2.1 Indoor Positioning System calibration

The intrinsic calibration of the monochrome camera was carried out using a ChArUco board and the standard pinhole camera model with a five-parameter distortion model. A total of 107 high-quality calibration frames were retained for the final optimisation (see Figure 6.48), yielding a final RMS reprojection error of approximately 0.984 pixels. The numerical results are summarised in Table 6.18, which presents the mean, standard deviation, minimum, maximum, and percentile statistics of the reprojection errors. These results confirm the spatial consistency achieved by the calibration process.

Table 6.18: Final calibration error statistics for the IPS monochrome camera.

Metric	Mean	Std	Min	Max
Per-image reprojection error [pixels]	0.109	0.014	0.083	0.137
Global point-to-point error [pixels]	0.912	0.368	0.015	2.866
95^{th} percentile [pixels]	1.545	-	-	-
99^{th} percentile [pixels]	1.858	-	-	-
Final RMS reprojection error [pixels]	0.984	-	-	-

6.2.2 Indoor experimental campaign results

A brief overview of the results achieved using the indoor positioning system installed in the COMPASS Lab test tank is provided in this section. Guidance and control aspects are not discussed, as they are considered out of scope and fall under the intellectual property of the researchers involved. Nevertheless, the positioning system acquisitions are presented to highlight its capabilities and effectiveness as an enabling technology for interdisciplinary integration.

The first result shown is the trajectory acquired by the IPS (Figure 6.49) and provided as feedback during a four-corner test navigation. The blue line represents the trajectory of the tag centre, with fiducial markers shown. The green line indicates the starting point, the red line the final point, and the black lines correspond to intermediate positions and orientations.

During the cooperative control experimental campaigns conducted in collaboration with the project partner CNR–INM, the IPS pipeline was further tested. In particular, a repeated rectangular four-corner test was executed under consistent conditions to calibrate the controller. An overlay of the feedback provided by the positioning system during these trials is shown (Figure 6.50).

Specifically, Figure 6.50a illustrates the superposition of the trajectories within the tank

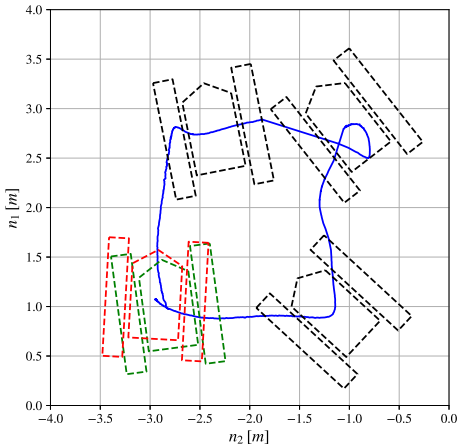


Figure 6.49: 4 corner test with the IPS feedback.

reference frame, while Figure 6.50b shows the corresponding overlay of the heading angles. Each colour represents a different run.

Finally, the overall computational cost was analysed by monitoring the timestamps of the feedback messages. The results are presented as a histogram in Figure 6.51, demonstrating the publication of the target pose with an average latency of 0.0238 seconds and a standard deviation of 0.0024 seconds.

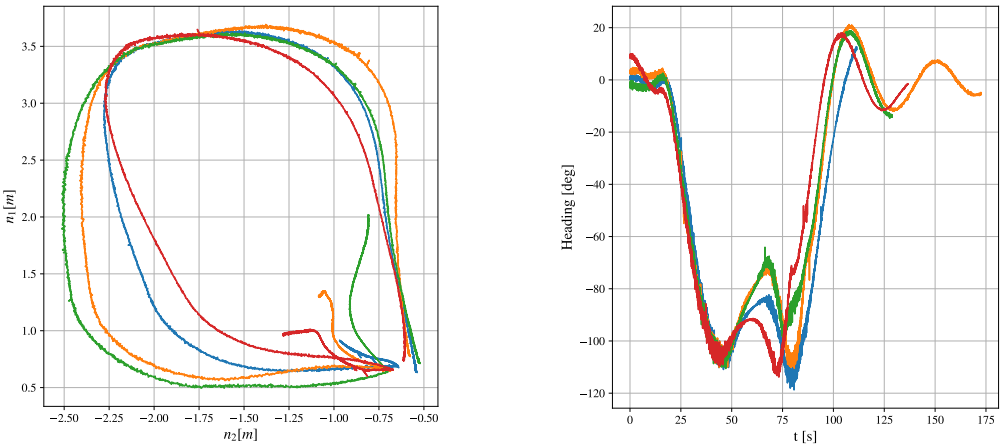
6.2.3 Lake Lagaccio results

This section reports selected results from the experimental campaigns conducted at Lake Lagaccio and published in the conference proceedings [24].

Given a set of waypoints generated by a coverage algorithm, the vehicle was required to autonomously navigate between them by estimating heading and course errors using the feedback provided by the state-estimation system described in Section 4.8. The reported results focus on the trajectory reconstructed from state-estimation feedback, obtained via a Kalman filter applied to onboard INS/GNSS data.

Figure 6.52 shows the paths obtained during the two main tests. In the first test (Figure 6.52a), seven waypoints were assigned on the map, and the vehicle was commanded to navigate in autonomous mode between the assigned waypoints, using the error between the desired values and the feedback provided by the state estimation system.

Figure 6.52b shows the result of the second test, namely the autonomous patrolling test. In this case, the waypoints were autonomously generated by the guidance and area coverage systems, with a new waypoint being generated each time the vessel reached the previous one. The vessel was therefore commanded to navigate in autonomous mode, managing both its state and its route, for approximately 30 minutes.



(a) Trajectories overlayed.

(b) Headings overlayed.

Figure 6.50: Repeated 4 corner test result with the IPS feedback.

In both plots, the executed path is shown in blue. The vehicle orientation is visually represented by the waterlines: green at the initial time instant, red at the final time instant, and black at intermediate instants. The waypoints are represented by red star markers, while the desired path, when present, is shown as a cyan dashed line.

Results related to waypoint generation, guidance, and control modules are not discussed, as they are developed and analysed by other researchers within a multidisciplinary framework. Nevertheless, the state-estimation system developed in this research played a central role in the experimental testing, enabling closed-loop feedback and autonomous navigation. For additional information, see [24].

6.2.4 Lake Nemi results

This section presents the results of the full-integration experimental campaign conducted at Lake Nemi. This work, together with contributions from the other researchers involved, was published in a journal article dedicated to this experimental campaign [37].

In accordance with Section 5.2.3, the campaign consisted of two main tests: the Line-of-Sight navigation test and the Collision Avoidance test, which are presented in the following sections.

Also, in this case, the reported results are limited to, or derived from, data obtained through the sensor processing pipelines (LiDAR and state estimation) presented in this work. The integrated modules for path planning, guidance, and control are not discussed in detail, as they constitute the intellectual property of other researchers. For a comprehensive description of these components, see [37].

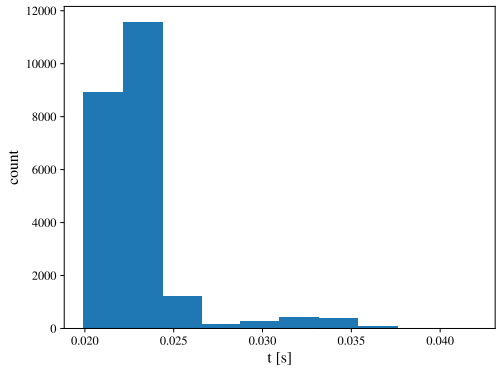


Figure 6.51: Processing time for the IPS.

Table 6.19: Main features of the LOS experiment.

Distance	668.5	m
Total time	884.6	s
Avg. Speed	0.76	m/s

6.2.4.1 Line Of Sight Experiment

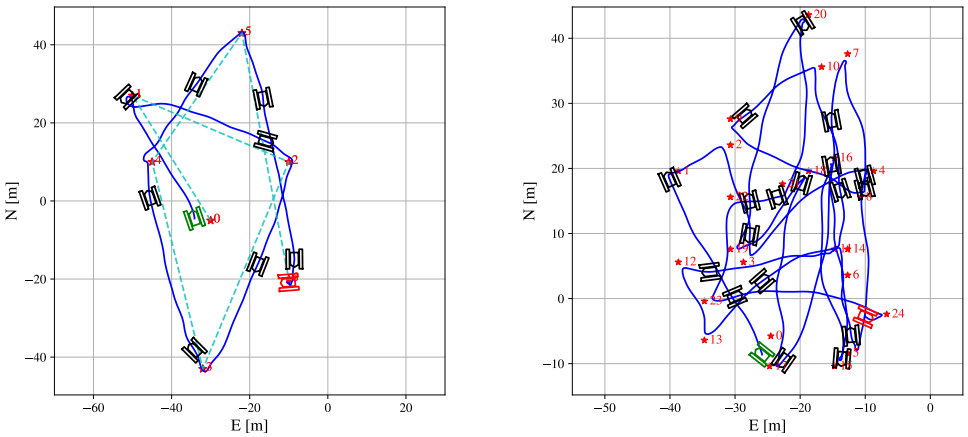
The SWAMP vessel was commanded to navigate along a predefined polygonal path composed of six waypoints, spaced at significantly greater distances than those used in previous experimental campaigns, owing to the large size of the experimental site.

Figure 6.53 illustrates the ASV trajectory during the LOS navigation experiment. The planned path is depicted as a blue dotted line connecting the waypoints in sequential order, while the solid line represents the trajectory actually executed by the vehicle. The corresponding measured data are reported in Figure 6.54, including the heading evolution (Figure 6.54a), the surge speed V, the body–frame velocity components u and v (Figure 6.54b), and the distance to both the assigned waypoints and the active LOS waypoint (Figure 6.54c).

Since the state estimation module outputs the heading angle within the $[-180°, 180°]$ range, a sharp but physiological transition appears when the angle crosses the discontinuity. The Kalman filter effectively mitigates this effect, ensuring rapid convergence to the correct heading.

Additional performance indicators are summarised in Table 6.19. The experiment lasted approximately 15 minutes, during which the ASV completed more than two laps, covering over 650 m at an average speed of 0.76 m/s.

The ASV accurately tracked the prescribed polygonal path, reaching each waypoint within the acceptance radius with smooth transitions. These results confirm the effectiveness and robustness of the proposed guidance and control architecture, justifying its use in the subsequent, more demanding experimental scenario.



(a) Assigned waypoints test.

(b) Autonomous patrolling test.

Figure 6.52: Lake Lagaccio path results.

Table 6.20: Main features of the CAV experiment.

Distance	260.1	m
Total time	600.0	s
Avg. Speed	0.43	m/s
CPA	6.4	m

6.2.4.2 Collision avoidance experiment

The second experiment tested the ASV in a fully autonomous scenario, specifically designed to challenge its collision-avoidance module. The vessel was assigned to follow a polygonal route of approximately 206 m.

Along the path, buoys obstructed several legs of the polygon, preventing strict adherence to the planned trajectory. Upon detecting obstacles, the CAV module dynamically computed and updated collision-free trajectories, maintaining a 5 m safety distance as it progressed through the assigned waypoints.

It is also worth noting that additional uncontrolled activities occurred during the experiment, including other boats, ASVs, or swimmers entering the area, which introduced further uncertainty and complexity to obstacle detection and avoidance.

Figure 6.55 reports the ASV trajectory recorded during the collision avoidance experiment. Unlike the LOS test, the vehicle does not strictly adhere to the predefined polygonal segments shown by the dashed blue line, as multiple buoys obstruct the nominal route. Instead, the ASV follows the path modifications generated by the CAV module, which are updated every 3 s and omitted from the figure for clarity.

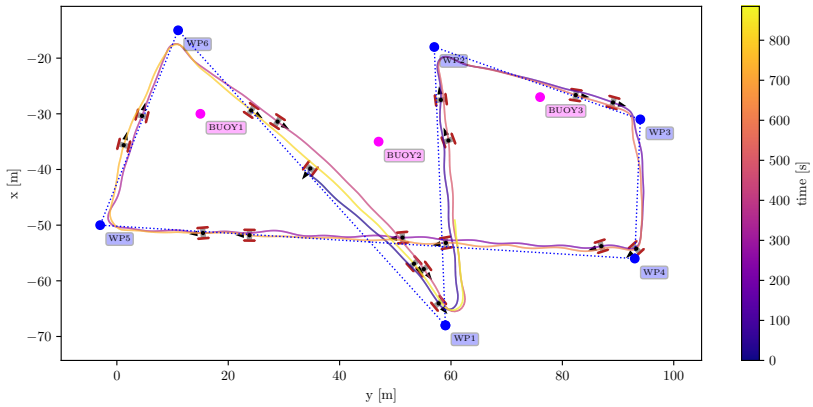
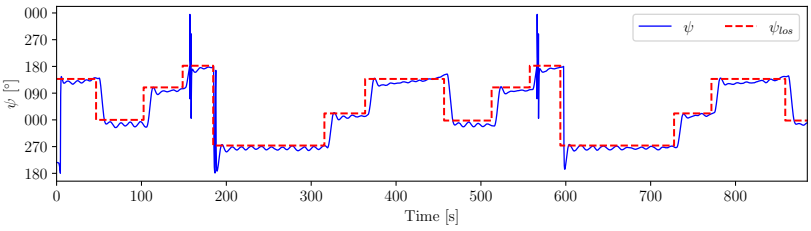
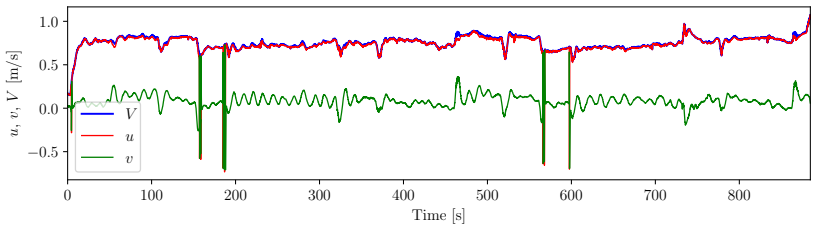


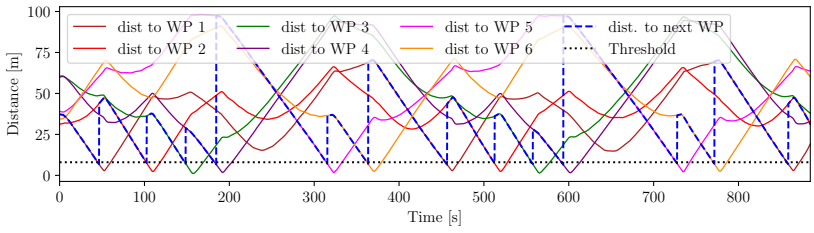
Figure 6.53: SWAMP's trajectory (solid line) and planned path (dashed blue line) during the LOS experiment.



(a)



(b)



(c)

Figure 6.54: Relevant data measured during the LOS experiment: SWAMP's heading (6.54a), speed and velocity components (6.54b), distances from the waypoints (6.54c).

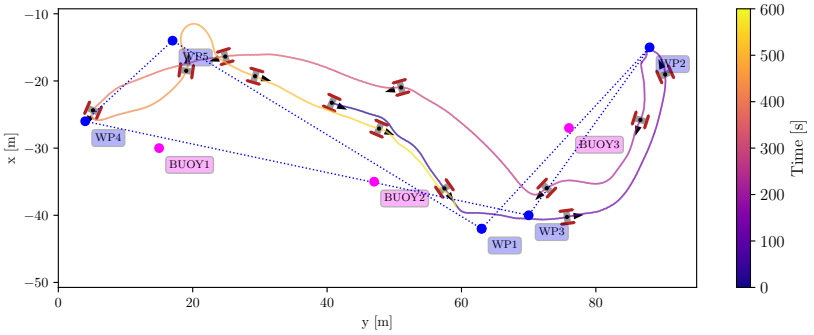


Figure 6.55: ASV's trajectory (solid line) and planned path (blue dashed line) during the collision avoidance experiment

The corresponding experimental measurements are shown in Figure 6.56. In this scenario, the ASV heading is commanded by the CAV module rather than by the LOS angle toward the target waypoint (Figure 6.56a), while still successfully reaching all assigned waypoints (Figure 6.56c) and preserving a minimum separation distance greater than 5 m from the obstacles (Figure 6.56d).

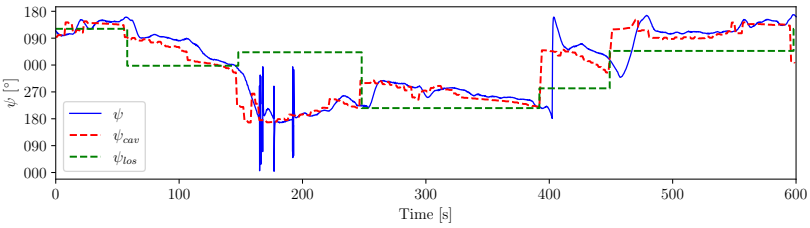
The trial lasted approximately 10 minutes; the average performance indicators, reported in Table 6.20, show that the ASV completed more than one loop of the polygon, travelling a total distance of about 260 m at a mean speed of 0.43 m/s.

Figure 6.57 presents a representative sequence demonstrating the collision avoidance system in operation. At 256 seconds into the experiment, the ASV has just reached waypoint 3 and is proceeding towards waypoint 4. The detection system identifies buoy 2 and computes a manoeuvre to avoid it (Figure 6.57a). Shortly afterwards, an unexpected obstacle—a swimmer—enters the LiDAR field of view (Figure 6.57b). The CAV module promptly replans the trajectory to maintain a safe distance from the swimmer (Figure 6.57c).

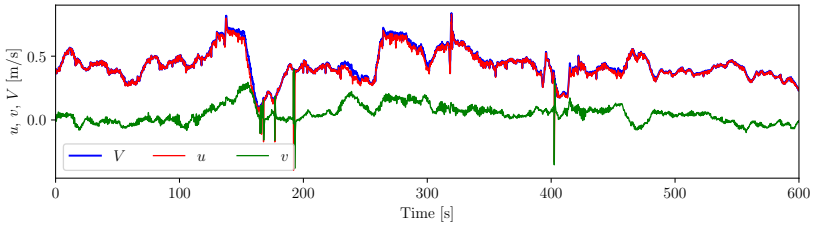
Once the swimmer moves out of the LiDAR's FoV (Figure 6.57d), the ASV resumes its path until buoy 1 comes into view (Figure 6.57e), triggering a further trajectory adjustment to ensure safe clearance (Figure 6.57f). This sequence demonstrates the system's ability to dynamically replan in real time, accommodating both static and unforeseen obstacles.

It is also worth noting that the obstacle detection system provides an accurate estimate of obstacle positions. However, the buoys' positions are approximate, as they drift continuously due to environmental factors such as wind. Consequently, in the current setup, it is not possible to distinguish between errors introduced by the perception and proprioception pipeline and those caused by the buoys' actual movement.

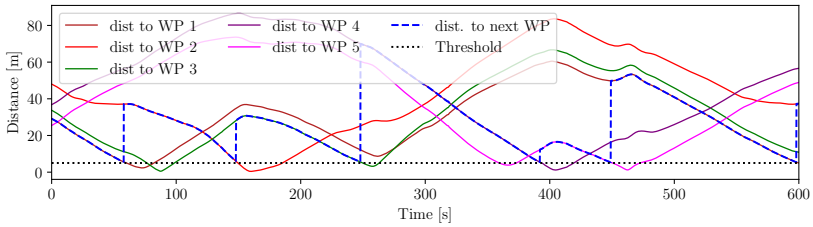
Finally, Figure 6.58 shows examples of obstacles captured by the onboard sensors. Specifically, Figures 6.58a, 6.58c, and 6.58e display RGB images of two buoys and the swimmer mentioned above, acquired from the ASV-mounted camera. The corresponding LiDAR point clouds are shown in Figures 6.58b, 6.58d, and 6.58f. Although the swimmer is barely visible in the RGB images, the LiDAR successfully detects it, even when it is mostly submerged, albeit with a lower point-cloud density than the buoys.



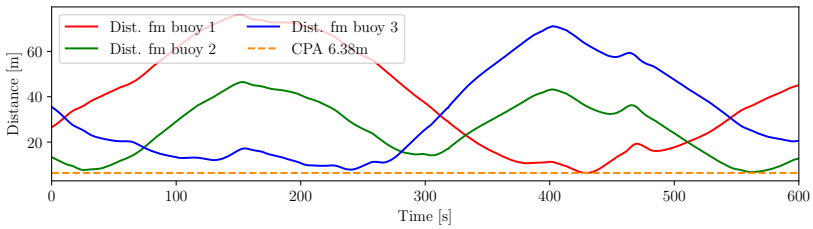
(a) ht



(b) ht

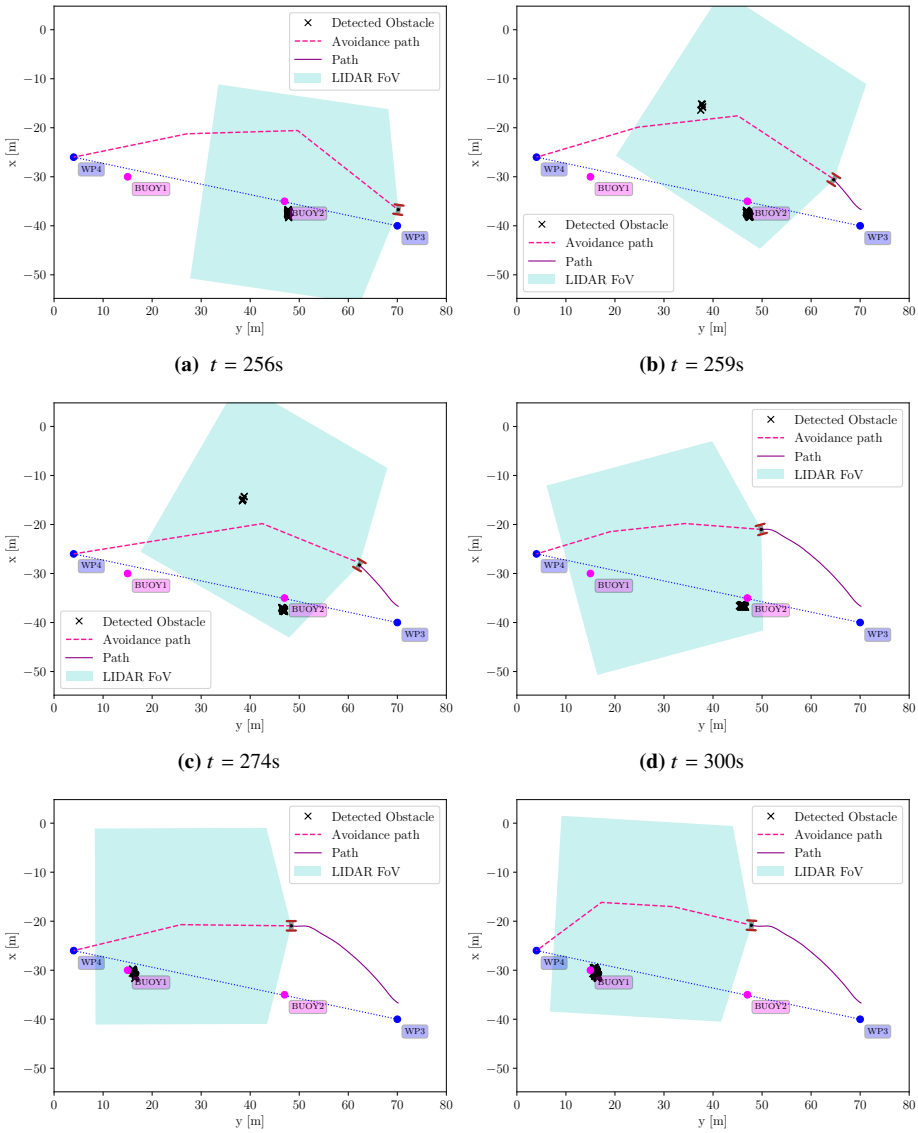


(c) ht



(d) ht

Figure 6.56: Relevant data measured during the collision avoidance experiment: ASV's heading (6.56a), speed and velocity components (6.56b), distances from the waypoints (6.56c), distances from the obstacles (6.56d).



(a) t = 256s **(b)** t = 259s

(c) t = 274s **(d)** t = 300s

(e) t = 303s **(f)** t = 304s

Figure 6.57: An example of the collision avoidance system in action: (6.57a) shows the planned manoeuvre after detecting the first buoy; (6.57b) An unexpected swimmer enters the path. (6.57c) The trajectory is modified to avoid the swimmer safely. (6.57d) The second buoy has not yet been detected by the LiDAR. (6.57e) The LiDAR detects the second buoy. (6.57f) The trajectory is updated to navigate safely around the second buoy.

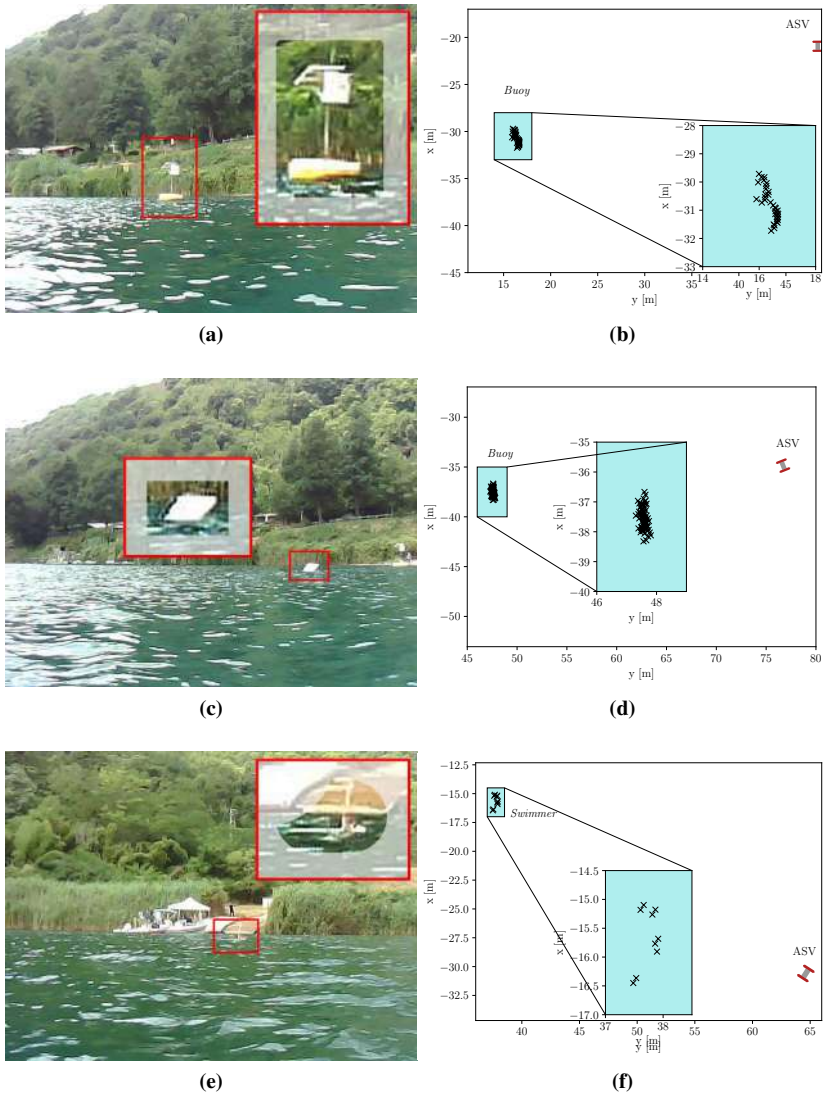


Figure 6.58: Examples of detected obstacles: RGB images captured by the ASV's onboard camera, alongside the corresponding subsampled LiDAR point clouds. A large buoy is shown in (6.58a, 6.58b), a smaller buoy in (6.58c, 6.58d), and a swimmer crossing the ASV's path in (6.58e, 6.58f). Note the variation in point cloud density depending on whether the obstacle is mostly above or partially submerged below the water surface.

6.2.4.3 Limitations and Sources of Uncertainty

A key limitation of the present study is the lack of ground-truth data for the USV trajectory. In addition, no ground truth is available for the positions of obstacles in the environment, which prevents an exact quantitative comparison of the estimated trajectories and detected obstacles with the real scenario.

As a result, the evaluation presented in this section primarily demonstrates the functionality and qualitative behaviour of the estimation and perception modules, rather than providing a fully validated numerical assessment.

Furthermore, the perception system was tested for integration using only LiDAR, treating obstacles as instantly stationary. This aspect still leaves some pipeline modules in a dynamic environment untested.

Although the obtained results are encouraging and indicate that the system behaves consistently, the lack of repeated trials and real-world ground-truth data limits the ability to generalise the findings and assess the approach's reproducibility. Future work will include systematic field campaigns with precise measurements of both the USV trajectory and environmental features to enable comprehensive validation of the methods and quantify their accuracy across diverse operating conditions.

6.2.5 Gulf of Naples results

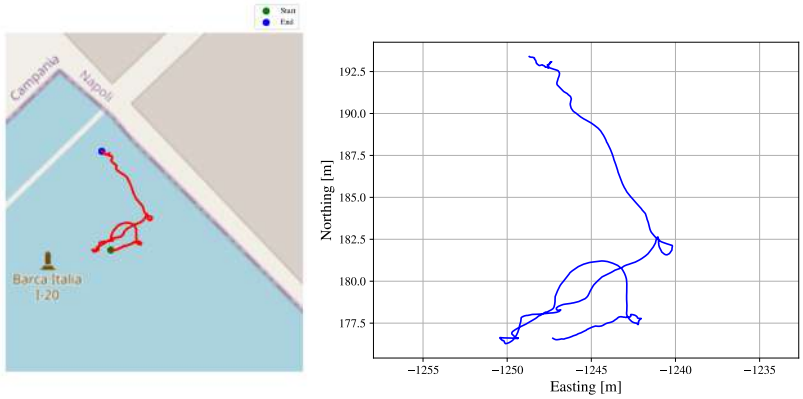
The first result presented is the path of the SWAMP vehicle during remotely controlled navigation operations. In particular, Figure 6.59a shows an example of one test carried out, where the vehicle's path expressed in latitude and longitude obtained from the on-board INS/GNSS sensors is superimposed on a map obtained using Open Street Map. Next to Figure 6.59b shows the same path in local coordinates with respect to an arbitrary reference point chosen for convenience, according to Section 5.2.4. The distinctive feature of these trajectories is that the first is recorded directly by the onboard computer, whereas the second is received by the ground nodes through the communication system after filtering and transformation into local coordinates. The continuity and close overlap of the two trajectories empirically demonstrate the system's transmission capability.

In addition, Figure 6.60a shows the heading and yaw rate estimated by the on-board node, transmitted and recorded by the on-shore nodes along the path shown above, as provided by the feedback system.

Finally, to demonstrate the feedback system's ability to reliably estimate even unmeasured variables, Figure 6.60b shows the velocity components (Northing and Easting) and the vehicle's speed modulus. .

Although empirical observation of the measured, estimated, and transmitted quantities demonstrates the system's effectiveness, a more detailed analysis was conducted. In particular, the three main nodes of the state estimation network were identified and distributed throughout the system, so that each processing step required data transmission and reception over the communication network.

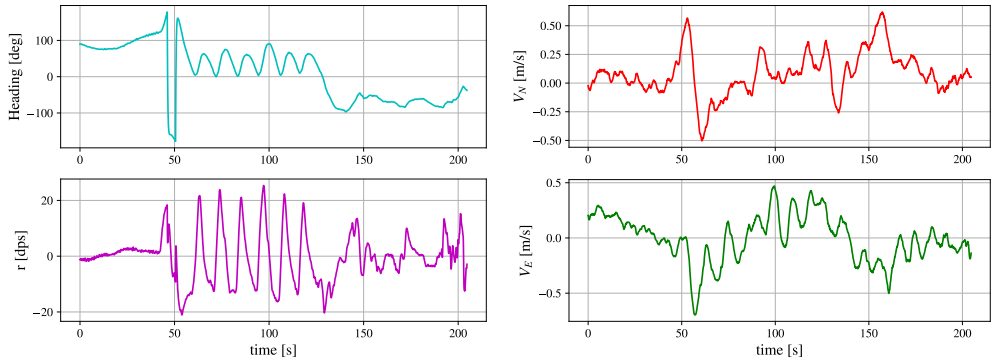
Specifically, the nodes were:



(a) Path superimposed to open street map.

(b) Path related to the local reference system.

Figure 6.59: Path of the open sea trials in the Gulf of Naples



(a) Heading angle and yaw rate.

(b) Velocity components.

Figure 6.60: State estimation results.

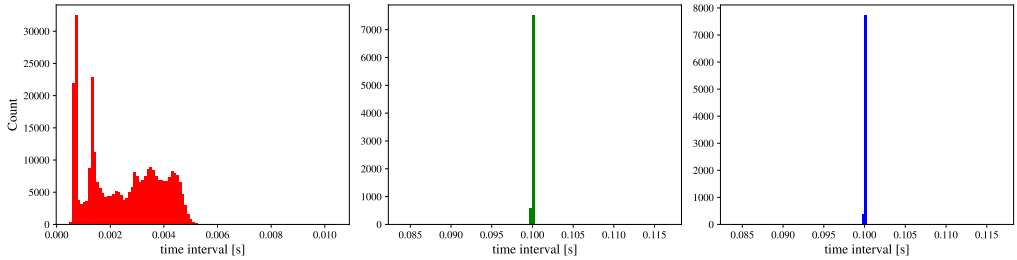


Figure 6.61: Time interval between messages in the feedback system.

- IMU/GNSS data acquisition and transmission at a reference frequency of 100 Hz;
- Subsampling of sensor data with buffering at 10 Hz;
- Filtering of the subsampled data at 10 Hz.

The time interval between consecutive messages at each of the three nodes was monitored. The corresponding histogram is shown in Figure 6.61.

As shown, the signal parsing node recorded a timestamp higher than expected. This phenomenon is mainly due to the high network traffic during testing, caused by the simultaneous use of several computers and sensors with large data streams. However, the increase in the time interval did not affect the operating frequency set for the downstream nodes, ensuring feedback at 10 Hz under all conditions. This phenomenon is further evident from the data reported in the Table 6.21.

Node	Reference	Recorded Mean	Recorded std
Raw measure	0.01	0.01248	0.00135
Subsampled measure	0.1	0.10007	0.00029
Filtered feedback	0.1	0.10007	0.00028

Table 6.21: Time interval between signals for each node.

The final test focused on the interaction among the nodes distributed in Genoa and La Spezia and the operational system running in Naples.

The following tests were conducted:

- Ping between all nodes: the test was considered successful if the measured round-trip time was below 100 ms.
- Remote start and stop of sensor acquisition processes on the onboard computer via SSH: the test was considered successful if the operator-issued remote command resulted in confirmed data reception at the local node.
- Remote start and stop of logging and post-processing processes on the shore station via SSH: the test was considered successful if the operator-issued remote command resulted in confirmed data reception at the local node.

All interaction tests yielded positive results.

6.2.5.1 Limitations and Sources of Uncertainty

The experimental evaluation presented in this section focused exclusively on testing the communication framework and the propagation of sensor data across distributed nodes. Sea trials involving autonomous operations of the SWAMP vehicle were not conducted during this phase and are planned for future work.

A significant limitation of the current assessment is the lack of ground-truth data for both the USV trajectory and the positions of environmental obstacles. This prevents a precise quantitative evaluation of the state estimation performance. Consequently, the results reported herein primarily illustrate the functional behaviour and reliability of the communication and data processing framework under controlled experimental conditions. While the observed performance is encouraging, further extensive tests are necessary to validate the system's accuracy, reproducibility, and robustness in realistic operational scenarios.

7. Conclusions and Future Directions

7.1 Synthesis of Results

This thesis focused on the development and testing of perception and communication systems for autonomous surface vessels. The goal was to support higher levels of autonomy in real-world scenarios. The methods were evaluated through a step-by-step experimental approach. This allowed a gradual transition from algorithm development to operational validation.

A key outcome of this research is the demonstrated effectiveness and robustness of the developed perception pipelines. The majority of the proposed algorithms were tested using real-world sensor data acquired during dedicated experimental campaigns, rather than relying exclusively on simulated environments.

Despite the limited availability of open-source datasets in the maritime domain, sufficiently large and diverse datasets were collected and employed to validate the pipelines under realistic operating conditions.

In particular, LiDAR data were collected at 8 distinct locations, yielding a total of 11,680 scans. From these, 920 samples were manually classified, and the dataset has been made publicly available to support open research initiatives. Furthermore, a dedicated experimental campaign was conducted to capture human-in-water scenarios using a multimodal acquisition setup. This campaign included both physical and virtual data augmentation to simulate environmental disturbances, yielding a training dataset of 6,000 samples. To the best of the author's knowledge, this represents the first dataset structured in this manner.
Extensive testing of the LiDAR detection and classification pipeline confirmed the robustness of the proposed methods against sensor noise, measurement uncertainties, and environmental variability, demonstrating their suitability for deployment in real marine environments.

In particular, the DBSCAN algorithm proved to be the most suitable clustering methodology among those analysed. Specifically, with a computational cost significantly lower than that of the other methods, it achieved the best performance in relevance to the ground truth, yielding an ARI of 0.93 with a noise ratio of 0.0006.

The Random Forest-based classification module achieved a 5-fold cross-validation accuracy of 93%, outperforming standard classifiers such as SVMs and MLPs. Evaluation of the complete pipeline reported overall accuracies of 93% and 98%, depending on the scenario. Additionally, progressive truncation tests revealed a clear relationship between classification accuracy and target truncation: the highest robustness was observed for the *ship* class, with misclassifications occurring at 55% truncation, whereas the *sailing boat* class was the most sensitive, with misclassifications appearing at 40% truncation.

The multimodal decision-level fusion approaches yielded promising results for target detection and tracking applications, with no track switching or fragmentation. The multimodal fusion strategy

effectively leveraged the complementary characteristics of the two sensing modalities, resulting in a substantial reduction of false positives compared to single-sensor vision-based processing. Furthermore, the results highlight that the inclusion of a tracking mechanism within the pipeline is essential to stabilise system outputs during temporary detection losses.

Due to the cascade detection, confirmation, and tracking mechanism, delays in target tracking were observed up to 15 frames (3s) in the worst case, exceeding the physiological delay required for the tracker to function. Despite these drawbacks, the system remained stable due to trace propagation, which effectively absorbed the disturbance. Considering the more challenging scenario, the detection, fusion, and tracking chain produced the expected number of active tracks in 84% of the frames with respect to the ground truth. This percentage increases to 92% when compared with the expected system behaviour, excluding the four frames required for track initialisation.

Small boats operating in semi-planing mode have been shown to generate significant noise in LiDAR measurements. Their small size further increases their detectability, making them among the most challenging to identify. Although a detailed analysis of the effect of distance is not yet available, it can be expected that detection performance is strongly influenced by range, given the divergent nature of the LiDAR sensor channels.

The LiDAR–Radar fusion strategy successfully maintained continuous tracking of all targets without track switching, track loss, or fragmentation, even under deliberately injected missed detections, demonstrating the tracker's robustness. The worst RMSE value was recorded at 4.28 m for the position and 1.83 m/s for the velocity. The variation in tracking performance across targets highlighted the dependence of these metrics on speed and distance, with the worst results observed for small targets moving away at high speed. In this case as well, the multimodal approach ensured continuous tracker updates even in the absence of instantaneous detections, further demonstrating the importance of avoiding reliance on a single sensing modality for robust target tracking.

Extensive testing of the raw data fusion framework for human-in-water detection demonstrated precise and weather-robust performance in detection, classification, and localisation. Five-fold cross-validation yielded a precision of 93% and a recall of 91%. Augmentation experiments further showed that the proposed method maintains superior performance under challenging environmental conditions (via harsh augmentation), outperforming a thermal-imaging-only approach by 6–15% across all metrics. Moreover, the fusion framework substantially reduced the number of false positives per image (from 3.3 to 1.1) compared to the thermal-only method under adverse conditions, while the number of false negatives remained unchanged.

All detection pipelines evaluated for real-time compliance exhibited end-to-end computational times, from sensor acquisition to result output, below 200 ms, which is considered the acceptable reference threshold.

The state estimation framework based on the Kalman filter was tested on real data, with residuals checked for normality, thereby enabling reliable state estimation at a frequency that consistently meets the 10 Hz requirement. Its low computational cost ensures rapid, efficient operation, making it suitable even for embedded onboard PCs with limited processing power. Given its greater complexity, a framework based on the Unscented Kalman Filter (UKF) was also evaluated on the same dataset. The UKF demonstrated the ability to overcome discontinuities in

angle representation and to avoid the spurious innovation peaks observed with the standard Kalman filter. However, these results should be taken as a general indication, as there is no ground truth against which to check them.

The validation activities were conducted at multiple levels, ranging from standalone performance assessment to cross-validation across heterogeneous data sources and experimental setups. This multi-level validation strategy provided strong evidence of the reliability and consistency of the estimation and perception processes, and allowed critical aspects of the pipelines to be assessed under realistic constraints.

Beyond validating individual pipelines, the research emphasised their integration into an increasingly complex system-level testing framework. The developed technologies were progressively deployed in experimental setups characterised by growing communication distances, distributed processing nodes, and more demanding operational tasks. This approach enabled the assessment of pipelines not only as isolated components but also as integral parts of a distributed autonomous system operating under realistic communication constraints.

The integration of the proposed technologies into the SWAMP autonomous platform further highlighted their role in enabling system-level autonomy. The developed perception and communication pipelines helped close critical loops within the Guidance, Navigation, and Control (GNC) architecture, enabling other researchers in the group to experimentally validate their control and decision-making algorithms in real-world conditions.

In particular, the indoor and outdoor state estimation systems enabled a series of experimental tests. The Indoor Positioning System provided data at a high frequency (over 40 Hz); the consistency of these measurements allowed trajectories to be reliably superimposed across repeat tests.

Initial outdoor experiments at Lake Lagaccio demonstrated the effectiveness of the INS/GNSS-based state estimation system, enabling uninterrupted patrolling for over 30 minutes.

The experimental campaign at Lake Nemi further validated the platform by integrating perception and state-estimation systems for the first time, thereby confirming the system's full functionality and enabling the execution of collision-avoidance logic. During the tests, the vessel travelled approximately 650 m for the LOS experiment and 260 m for the collision-avoidance experiment, with average speeds of 0.76 m/s and 0.43 m/s, respectively. In the collision-avoidance experiment, the system successfully detected both fixed and moving obstacles, including a swimmer, and replanned the route without collisions.

Finally, the conclusive test conducted at Lake Naples demonstrated the system's capability to estimate vessel state in a challenging environment with high external disturbances, while distributing network nodes and maintaining feedback at a consistent 10 Hz, in full compliance with real-time requirements.

Overall, this research demonstrates that the proposed methodologies constitute effective, flexible, and experimentally validated solutions for perception and communication in marine autonomous systems. By bridging the gap between algorithmic development and operational deployment and enabling realistic system-level testing, this work advances the maturity of autonomous surface ship technologies and supports their transition to real-world applications.

7.2 Limitations of the Study

The experimental results presented in Chapter 6 demonstrate strong performance under the tested conditions; however, they also reveal specific limitations that constrain the generalisation and operational robustness of the proposed architecture.

A first limitation concerns LiDAR-based perception under sparse-return conditions. While clustering achieved an ARI of 0.93 with a very low noise ratio (0.0006) in controlled scenarios, performance degraded in open-water environments and for small semi-planing boats at longer ranges. In these cases, the divergence of LiDAR beams and the limited vertical resolution led to sparse, noisy point clouds, thereby reducing cluster separability and increasing sensitivity to background reflections. The degradation was particularly evident for small, fast-moving targets receding from the sensor, where the tracking RMSE reached 4.28 m in position and 1.83 m/s in velocity. These findings indicate that detection and tracking accuracy are strongly range-dependent and influenced by target size and motion profile, limiting direct extrapolation to long-range or high-speed operational scenarios.

A second limitation emerges from the classification robustness analysis. Although cross-validation yielded overall accuracies of 93%–98% across scenarios, truncation experiments showed that performance degrades significantly when targets are partially observed. Misclassification occurred at approximately 40% truncation for sailing boats and 55% for ships, highlighting sensitivity to partial occlusions and incomplete point clouds. This suggests that the classifier assumes sufficiently complete geometric information, an assumption that may not hold in crowded or occluded maritime environments.

The tracking framework, while robust to missed detections and free from track switching or fragmentation, exhibited delays of up to 15 frames (approximately 3 s) due to the cascade nomination and confirmation mechanism. Although trace propagation stabilised the system, this latency may reduce responsiveness in high-risk collision scenarios requiring rapid reaction. Moreover, the expected number of active tracks was achieved in 84% of frames (92% excluding initialisation), indicating residual sensitivity to detection intermittency.

Another structural limitation arises from the adopted planar state-estimation model. The Kalman-based estimator met the 10 Hz requirement and yielded statistically consistent residuals. However, the absence of full 6-DoF modelling limits robustness under rough sea states where attitude dynamics are non-negligible. Furthermore, the lack of ground truth prevents a rigorous analysis of the metrics.

System-level validation also remains partial. Although the perception, fusion, and communication modules were experimentally verified, their integration with collision-avoidance logic was not quantitatively assessed in a closed-loop framework.

Finally, scalability and regulatory compliance were not fully addressed. Experimental validation was conducted in controlled or moderately complex environments, without systematic evaluation in dense multi-vessel traffic or under sustained long-duration missions. While communication latency and packet loss were monitored, stress testing under high network load was not performed. Furthermore, compliance with the IMO MASS framework was not formally assessed, and

redundancy or fail-operational analyses were beyond the scope of this work.

Overall, the results confirm the technical feasibility and robustness of the proposed approach within the tested conditions. However, performance remains dependent on dataset coverage, completeness of target geometry, planar-motion assumptions, and moderate traffic density. These constraints define the operational envelope within which the reported metrics are reliable.

7.3 Assessment of the Research Questions

This section systematically revisits the research questions introduced in Chapter 1 and evaluates the extent to which they have been addressed, based on the experimental evidence presented in Chapter 6.

RQ1: Integration of Advanced Sensing Technologies

How can advanced sensing technologies, including LiDAR, optical, and infrared sensors, be effectively integrated and adapted to provide robust perception capabilities in complex and dynamic maritime environments?

This research addressed RQ1 by developing and validating multiple perception pipelines using LiDAR, thermal imaging, RGB cameras, and radar. Both decision-level and raw-data fusion strategies were implemented and experimentally evaluated. The results demonstrated the superiority of the multimodal approach over the single-sensor approach, as well as greater resilience to false positives and false negatives. In general, the use of LiDAR and optical sensors enables the acquisition of rich image-plane features, which can be combined with precise LiDAR-derived distance measurements. Furthermore, the use of two 3D sensors, such as LiDAR and Radar, provides complete redundancy in position estimation, increasing confidence in the presence of double detections and mitigating the risk of sensor failure.

Reduced performance under sparse LiDAR returns, long-range detection, and highly dynamic small targets is only partially evaluated. Full robustness under extreme sea states remains unvalidated.

RQ2: Marine Dataset Limitations and Data Collection Strategies

What strategies can be developed to overcome the current limitations in marine data availability, and how can new datasets be collected and annotated to support the development of reliable perception and decision-support systems onboard ships and yachts?

To address this question, dedicated experimental campaigns were conducted to collect maritime LiDAR and multimodal datasets. A LiDAR dataset comprising 11,680 scans acquired at eight locations was created, including 920 manually annotated samples. Additionally, a structured multimodal dataset for human-in-water detection was generated, including 6,000 augmented samples designed to simulate environmental disturbances.

The datasets enabled cross-validation procedures, robustness testing under artificial perturbations, and quantitative evaluation of the pipelines and classification accuracy.

Environmental diversity remains limited (moderate sea states, moderate traffic density, lim-

ited long-duration missions). Extreme-weather, high-traffic, and offshore scenarios were not systematically covered.

RQ2 is therefore partially addressed: new datasets were acquired and successfully used, but their representativeness for full-scale deployment remains incomplete.

RQ3: Sensor Fusion and AI for Maritime Detection and Tracking

Algorithmic analysis of the LiDAR detection pipeline highlighted the approaches most suitable for maritime applications, balancing computational cost and detection accuracy. Supervised non-deep classifiers provided robust performance even on sparsely populated datasets, avoiding the high training costs of deep learning models and leveraging a modular architecture for system integration.

Decision-level fusion proved optimal for multi-object detection, enabling asynchronous operation of multiple sensor pipelines and improving overall system robustness. Conversely, early fusion significantly expanded the available feature space, thereby enhancing detection precision even under severe environmental disturbances.

Key limitations include incomplete validation under real-world disturbed conditions and the absence of full integration testing, which could further refine algorithm selection and verify end-to-end system performance. Due to limited data availability, deep learning approaches were not evaluated, although they could potentially achieve high performance with careful deployment and training strategies.

In summary, RQ3 has been addressed at the component and subsystem level, but full closed-loop system validation, including complete integration testing under high-risk maritime scenarios, remains necessary.

RQ4: Transferability of Cross-Domain Perception Algorithms

Which perception algorithms and methodologies, originally developed for other domains such as automotive or aerospace, can be effectively transferred to the maritime domain? What modifications, adaptations, or additional measures are necessary to ensure their robustness and reliability in marine environments, and which approaches are unsuitable without substantial re-engineering?

Several algorithms originally developed in other domains, such as automotive, were successfully adapted to maritime LiDAR and vision-based sensing, albeit with careful domain-specific adjustments.

At the LiDAR level, clustering algorithms commonly used in automotive applications are transferable, but require particular attention. Density-based methods, such as DBSCAN, are preferred due to their robustness to noise and irregular cluster shapes. In the maritime domain, noise characteristics are strongly influenced by wakes and foam, which differ substantially from urban or road environments, making density-aware clustering more effective.

For classification, widely adopted deep learning models from the automotive field are currently impractical in maritime scenarios without extremely large annotated datasets. Among non-deep supervised classifiers, Random Forests emerged as the most suitable choice. The maritime context enables the extraction of predictive statistical features arising from the high variance in target

shapes and geometry, which can be leveraged effectively for class discrimination.

In vision-based approaches, directly transferring YOLO-type networks trained on conventional datasets is challenging because of the wide variety of maritime targets that lie outside the original training distribution. Maritime-oriented networks are thus preferable; however, in the absence of adequate datasets, decision-level sensor fusion mitigates the tendency toward false positives and enables effective deployment of multi-purpose networks in marine settings. For thermal imaging associated with SAR operations, constructing a dedicated dataset is essential, as the appearance of humans at sea differs significantly from that in other domains, rendering transfer learning largely ineffective.

RQ4 is therefore substantially addressed, with clearly identified adaptation requirements and operational boundaries.

7.4 Direction for Future Research

Future research should directly target the specific shortcomings identified in this work. First, the observed dataset limitations, particularly reduced LiDAR performance in sparse-return and open-water scenarios, call for extended data acquisition campaigns that include higher sea states (e.g., sea state 4–5), longer operational ranges, and high-traffic environments with multiple dynamic targets. Such datasets would enable systematic cross-scenario robustness analysis and allow quantitative evaluation of detection degradation as a function of range, wave-induced spray, and vessel dynamics.

Second, the planar assumption adopted in the state estimation framework should be extended to a full six-degree-of-freedom (6-DoF) model that explicitly incorporates roll, pitch, and heave dynamics. This extension would allow improved motion compensation of LiDAR and optical measurements, particularly under rough sea conditions where attitude variations directly affect perception accuracy.

Third, the partial integration between perception outputs and collision-avoidance logic should be completed to enable end-to-end quantitative validation of autonomous navigation performance. This includes measuring closed-loop metrics and reaction latency under realistic operational loads.

Furthermore, scalability assessments must be conducted in dense multi-vessel scenarios to evaluate communication latency, packet-loss resilience, and the stability of distributed state estimation.

Finally, future validation should explicitly consider regulatory and safety constraints associated with the IMO MASS framework, including redundancy analysis, fail-safe versus fail-operational behaviour, and systematic testing under sensor degradation or failure modes.

Addressing these targeted research directions will transform the proposed architecture from a validated experimental framework into a deployment-ready autonomous maritime perception and communication system.

List of Publications

1. Faggioni, N., Leonardi, N., Ponzini, F., Sebastiani, L., & Martelli, M. (2021). Obstacle detection in real and synthetic harbour scenarios. In International Conference on Modelling and Simulation for Autonomous Systems (pp. 26-38). Cham: Springer International Publishing. DOI: 10.1007/978-3-030-98260-7_2.

2. Martelli, M., Faggioni, N., & Ponzini, F. (2022). Detecting and Tracking Multi-Object in Real Marine Environment. In Conference Proceedings of iSCSS (Vol. 2022). DOI: 10.24868/10707.

3. Faggioni, N., Ponzini, F., & Martelli, M. (2022). Multi-obstacle detection and tracking algorithms for the marine environment based on unsupervised learning. Ocean Engineering, 266, 113034. DOI: 10.1016/j.oceaneng.2022.113034.

4. Ponzini, F., Zaccone, R., & Martelli, M. (2023). A multi-sensor indoor tracking system for autonomous marine model-scale vehicles. In Journal of Physics: Conference Series (Vol. 2618, No. 1, p. 012008). IOP Publishing. DOI 10.1088/1742-6596/2618/1/012008.

5. Ponzini, F., Fruzzetti, C., & Sabatino, N. (2024). Real-time critical marine infrastructure multi-sensor surveillance via a constrained stochastic coverage algorithm. In Conference Proceedings of iSCSS (Vol. 2024). DOI: 10.24868/11141.

6. Ponzini, F., Zaccone, R., & Martelli, M. (2025). LiDAR target detection and classification for ship situational awareness: A hybrid learning approach. Applied Ocean Research, 158, 104552. DOI: 10.1016/j.apor.2025.104552.

7. Ponzini, F., Van Hamme, D., & Martelli, M. (2025). Human detection in marine disaster search and rescue scenario: a multi-modal early fusion approach. Ocean Engineering, 340, 122341. DOI: 10.1016/j.oceaneng.2025.122341.

8. Martelli, M., Faggioni, N., & Ponzini, F. (2025). ARNOLD Annotated Repository of Navigational Obstacles from LiDAR Data. Autonomous Transportation Research. DOI: 10.1016/j.atres.2025.08.001.

9. Ponzini, F., Martelli, M. (2025). Marine Obstacles Multi-Modal Detection, Classification and Tracking via Camera-LiDAR Late Fusion. IEEE MetroSea 2025. DOI: 10.1109/MetroSea66681.2025.11245686.

10. Zaccone, R., Ponzini, F., & Martelli, M. (2026). Experimental validation of a modular navigation architecture for marine autonomous surface vehicles with reactive collision avoidance. Applied Ocean Research, 166, 104903. DOI: 10.1016/j.apor.2025.104903.

Nomenclature

List of Symbols

A Classification accuracy.

A_{ij} Accuracy for grid combination (i, j).

a_x, a_y Planar acceleration components.

B_i Magnetic intensity along body axes b_i.

B_x, B_y Planar magnetic field components.

$\mathbf{B}$ Magnetic field vector.

b_1, b_2, b_3 Body-fixed reference frame axes.

C Geary's spatial autocorrelation coefficient.

$C_{\mathbf{svm}}$ SVM regularisation parameter.

c $\cos(\psi)$.

d Distance.

$e_{i,j}$ Ellipticity ratios between PCA eigenvalues.

f_s Sampling frequency.

$\mathbf{f}(\cdot)$ Process model.

$\mathbf{h}(\cdot)$ Measurement model.

H Kruskal–Wallis test statistic.

I Moran's spatial autocorrelation index.

I_i Reflectivity (intensity) of LiDAR point i.

$I_{\max}$ Maximum LiDAR intensity.

K Kalman gain.

k Discrete-time index.

k_i Point cloud kurtosis on principal direction i.

k_I Intensity kurtosis.

l_{pp} Length between perpendiculars.

n State dimension.

n_p Number of LiDAR points in a cluster.

n_1, n_2, n_3 North-East-Down axes.

O_b Origin of body frame.

O_n Origin of Earth frame.

p_i i-th LiDAR point.

$\mathbf{P}_{k|k}$ Updated covariance matrix.

$\mathbf{P}_{k|k-1}$ Predicted covariance matrix.

$\mathbf{Q}$ Process noise covariance matrix.

R_ψ Rotation matrix (body to Earth).

$\mathbf{R}$ Measurement noise covariance matrix.

r Yaw rate.

$r_{\mathbf{acc}}$ Waypoint acceptance radius.

S_0 Sum of spatial weights.

s $\sin(\psi)$.

T Total computational time.

T_1 Feature extraction time.

T_2 Classification time.

$\mathbf{T}$ Computational cost matrix.

u Control input.

v_x, v_y Planar velocity components.

V_N, V_E North and East velocity components.

$\mathbf{W}$ Spatial weights matrix.

w_{ij} Element of spatial weights matrix.

$\mathbf{x}$ State vector.

x, y, z Cartesian coordinates.

$\hat{\mathbf{x}}_{k|k}$ Updated state estimate.

$\hat{\mathbf{x}}_{k|k-1}$ Predicted state estimate.

$\chi^{(i)}$ i-th sigma point.

$z_{\mathbf{sea}}$ Estimated sea level.

$\alpha_{\mathbf{mlp}}$ MLP regularisation parameter.

γ_I Intensity skewness.

γ_i Point cloud skewness on principal direction i.

δ Generic deviation.

δ_z Vertical tolerance.

Δt Sampling time.

ε DBSCAN neighbourhood radius.

λ UKF scaling parameter.

$\lambda_1, \lambda_2, \lambda_3$ PCA eigenvalues.

μ Mean value.

ι_i Reflectivity value of point i.

ι Mean reflectivity.

ρ Density.

σ Standard deviation.

τ_I Relative intensity threshold.

θ Generic angle.

ω Angular velocity.

$\dot{r}$ Yaw acceleration.

List of Acronyms

AIS Automatic Identification System.

ANOVA Analysis of Variance.

ARI Adjusted Rand Index.

ASV Autonomous Surface Vehicle.

BEV Bird's-Eye View.

CAV Collision Avoidance module.

CPA Closest Point of Approach.

CNR-INM National Research Council – Institute of Marine Engineering.

DBSCAN Density-Based Spatial Clustering of Applications with Noise.

EDA Exploratory Data Analysis.

EMSA European Maritime Safety Agency.

ENC Electronic Navigational Chart.

FOV Field of View.

GNSS Global Navigation Satellite System.

HDBSCAN Hierarchical Density-Based Spatial Clustering of Applications with Noise.

Hz Hertz.

IMU Inertial Measurement Unit.

KDE Kernel Density Estimation.

KPI Key Performance Indicator.

LiDAR Light Detection and Ranging.

LOS Line Of Sight.

MASS Maritime Autonomous Surface Ships.

MDI Mean Decrease Impurity.

MLP Multi-Layer Perceptron.

NED North-East-Down.

OPTICS Ordering Points To Identify the Clustering Structure.

PCA Principal Component Analysis.

RADAR Radio Detection and Ranging.

RGB Red-Green-Blue.

RMSE Root Mean Square Error.

SD Standard Deviation.

SI International System of Units.

SVM Support Vector Machine.

UKF Unscented Kalman Filter.

USCG United States Coast Guard.

USV Unmanned Surface Vehicle.

WMO World Meteorological Organisation.

YOLO You Only Look Once.

Declaration on generative AI and AI-assisted technologies in the writing

While writing this paper, the author used generative AI and AI-assisted technologies, specifically ChatGPT-5, DeepL, and Grammarly Pro, to improve language and readability. The authors reviewed and edited the generated text as needed and take full responsibility for its content.

Data Protection and Privacy Statement

This research was conducted in compliance with the European Union General Data Protection Regulation (EU Regulation 2016/679 – GDPR). The study did not involve the processing of personal data beyond anonymised technical sensor data. All datasets used were either publicly available or collected in controlled experimental environments without identifying individuals.
During field data acquisition, no biometric or personally identifiable information was intentionally collected. Any incidental data were anonymised prior to storage and analysis.
Data were stored on secure institutional servers and used exclusively for research purposes within the scope of this doctoral project.
No data were transferred outside the European Economic Area.

Bibliography

[1] United Nations Conference on Trade and Development, "Review of Maritime Transport 2017," Geneva, Switzerland, 2017.

[2] ——, "Review of Maritime Transport 2022," Geneva, Switzerland, 2022.

[3] European Maritime Safety Agency (EMSA), "Annual overview of marine casualties and incidents 2025," European Maritime Safety Agency, Tech. Rep., Nov 2025, accessed Feb 20, 2026. [Online]. Available: https://www.emsa.europa.eu/publications/item/5352-annual-overview-of-marine-casualties-and-incidents-2024.html

[4] BIMCO and International Chamber of Shipping, "Seafarer workforce report : The global supply and demand for seafarers in 2021," BIMCO and International Chamber of Shipping, Copenhagen, Denmark, Industry Report, 2021, accessed: 2026-01-07. [Online]. Available: https://www.bimco.org/products/publications/

[5] DNV, "Maritime safety trends 2014–2024," DNV, Høvik, Norway, Industry Report, 2024, accessed: 2026-01-07. [Online]. Available: https://www.dnv.com/maritime/publications/maritime-safety-report-2014-2024-download/

[6] J. A. García Maza and R. P. Argüelles, "Colregs and their application in collision avoidance algorithms: A critical analysis," *Ocean Engineering*, vol. 261, p. 112029, 2022.

[7] U. C. Guard, "Recreational boating statistics," Online, 2023, accessed April 9, 2025. [Online]. Available: https://www.uscgboating.org/library/accident-statistics/Recreational-Boating-Statistics-2023-Ch2.pdf

[8] K. Liu, Q. Yu, Z. Yuan, Z. Yang, and Y. Shu, "A systematic analysis for maritime accidents causation in chinese coastal waters using machine learning approaches," *Ocean & Coastal Management*, vol. 213, p. 105859, 2021.

[9] P. Brandt, Z. H. Munim, M. Chaal, and H.-S. Kang, "Maritime accident risk prediction integrating weather data using machine learning," *Transportation Research Part D: Transport and Environment*, vol. 136, p. 104388, 2024.

[10] P. Panagiotidis, K. Giannakis, N. Angelopoulos, and A. Liapis, "Shipping accidents dataset: Data-driven directions for assessing accident's impact and improving safety onboard," *Data*, vol. 6, no. 12, 2021.

[11] L. Zhang, H. Wang, Q. Meng, and H. Xie, "Ship accident consequences and contributing factors analyses using ship accident investigation reports," *Proceedings of the Institution of Mechanical Engineers, Part O*, vol. 233, no. 1, pp. 35–47, 2019.

[12] Criminal Court of Livorno, Section I, "Judgment no. 179," 1998, issued on October 31, 1998, Case No. 66/95 RG, No. 542/91 NR.

[13] B. D. Bryony Gooch, "North sea: Everything we know about oil tanker and cargo vessel collision off yorkshire coast," 2025, accessed: 2025-04-10. [Online]. Available: https://www.independent.co.uk/news/uk/home-news/north-sea-oil-tanker-cargo-collision-

[14] E. Commission, "Eu Wind Power Action Plan," 2023. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52023DC0669

[15] R. Chataut, "Undersea cables are the unseen backbone of the global internet," April 2024. [Online]. Available: https://theconversation.com/undersea-cables-are-the-unseen-backbone-of-the-global-internet-226300

[16] J.-B. Jouffray, R. Blasiak, A. V. Norström, H. Österblom, and M. Nyström, "The Blue Acceleration: The Trajectory of Human Expansion into the Ocean," *One Earth*, vol. 2, no. 1, pp. 43–54, 2020.

[17] Å. Fritzon, K. Ljungkvist, A. Boin, and M. Rhinard, "Protecting europe's critical infrastructures: problems and prospects," *Journal of Contingencies and Crisis Management*, vol. 15, no. 1, pp. 30–41, 2007.

[18] M. Clare, "Submarine Clable Protection and the Environment. An Update from the ICPC," 2021. [Online]. Available: https://www.iscpc.org/publications/submarine-cable-protection-and-the-environmentICPC_Public_EU_March%202021.pdf

[19] C. Bueger, "Russian Spy Ship in North Sea raises concerns about the vulnerability of key maritime infrastructures," 2023. [Online]. Available: https://theconversation.com/russian-spy-ship-in-north-sea-raises-concerns-about-the-vulnerability-of-key-maritime-infrastructure-204205

[20] M. Knights, "Assessing the Houthi War Effort Since October 2023," April 2024. [Online]. Available: https://www.washingtoninstitute.org/policy-analysis/assessing-houthi-war-effort-october-2023

[21] G. Kantchev, "Sweden Says Second Undersea Cable Damaged in Baltic Sea," October 2023. [Online]. Available: https://www.wsj.com/world/europe/sweden-says-second-undersea-cable-damaged-in-baltic-sea-d9f21fea

[22] T. Nguyen, "The challenges of dark ships to the safety and security of commercial shipping and the way forward," *Asia-Pacific Journal of Ocean Law and Policy*, vol. 8, no. 2, pp. 310 – 328, 2023.

[23] C. Bueger and T. Liebetrau, "Critical maritime infrastructure protection: What's the trouble?" *Marine Policy*, vol. 155, p. 105772, 2023.

[24] F. Ponzini, C. Fruzzetti, and N. Sabatino, "Real-time critical marine infrastructure multi-sensor surveillance via a constrained stochastic coverage algorithm," in *Conference Proceedings of iSCSS*, vol. 2024, 2024, pp. 62–74.

[25] A. Odetti, G. Bruzzone, R. Ferretti, S. Aracri, F. Carotenuto, C. Vagnoli, A. Zaldei, and I. Scagnetto, "Lake environmental data harvester (led) for alpine lake monitoring with autonomous surface vehicles (asvs)," *Remote Sensing*, vol. 16, no. 11, 2024.

[26] S. Luo, Y. Singh, H. Yang, J. H. Bae, J. E. Dietz, X. Diao, and B.-C. Min, "Image processing and model-based spill coverage path planning for unmanned surface vehicles," in *OCEANS 2019 MTS/IEEE SEATTLE*, 2019, pp. 1–9.

[27] A. Z. Akbar, C. Fatichah, and R. Dikairono, "Autonomous surface vehicle in search and rescue process of marine casualty using computer vision based victims detection," in *2022 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, 2022, pp. 1–6.

[28] H. Mansor, M. H. Norhisam, Z. Z. Abidin, and T. S. Gunawan, "Autonomous surface vessel for search and rescue operation," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 3, pp. 1701–1708, 2021.

[29] Y. Singh, M. Bibuli, E. Zereik, S. Sharma, A. Khan, and R. Sutton, "A novel double layered hybrid multi-robot framework for guidance and navigation of unmanned surface vehicles in a practical maritime environment," *Journal of Marine Science and Engineering*, vol. 8, no. 9, 2020.

[30] A. Boretti, "Unmanned surface vehicles for naval warfare and maritime security," *The Journal of Defense Modeling and Simulation*, vol. 0, no. 0, p. 15485129241283056, 0.

[31] N. Faggioni, F. Ponzini, and M. Martelli, "Multi-obstacle detection and tracking algorithms for the marine environment based on unsupervised learning," *Ocean Engineering*, vol. 266, p. 113034, 2022.

[32] N. Faggioni, N. Leonardi, F. Ponzini, L. Sebastiani, and M. Martelli, "Obstacle Detection in Real and Synthetic Harbour Scenarios," in *Modelling and Simulation for Autonomous Systems*. Cham: Springer International Publishing, 2022, pp. 26–38.

[33] F. Ponzini, R. Zaccone, and M. Martelli, "Lidar target detection and classification for ship situational awareness: A hybrid learning approach," *Applied Ocean Research*, vol. 158, p. 104552, 2025.

[34] F. Ponzini, D. Van Hamme, and M. Martelli, "Human detection in marine disaster search and rescue scenario: a multi-modal early fusion approach," *Ocean Engineering*, vol. 340, p. 122341, 2025.

[35] M. Martelli, N. Faggioni, and F. Ponzini, "Detecting and Tracking Multi-Object in Real Marine Environment," in *Proceedings of the International Ship Control Systems Symposium*, 2022.

[36] F. Ponzini and M. Martelli, "Marine obstacles multi-modal detection, classification and tracking via camera-lidar late fusion," in *2025 IEEE International Workshop on Metrology for the Sea; Learning to Measure Sea Health Parameters (MetroSea)*, 2025, pp. 235–240.

[37] R. Zaccone, F. Ponzini, and M. Martelli, "Experimental validation of a modular navigation architecture for marine autonomous surface vehicles with reactive collision avoidance," *Applied Ocean Research*, vol. 166, p. 104903, 2026.

[38] M. Martelli, N. Faggioni, and F. Ponzini, "Arnold annotated repository of navigational obstacles from lidar data," *Autonomous Transportation Research*, 2025.

[39] F. Ponzini, R. Zaccone, and M. Martelli, "A multi-sensor indoor tracking system for autonomous marine model-scale vehicles," *Journal of Physics: Conference Series*, vol. 2618, no. 1, p. 012008, oct 2023.

[40] G. Gennarelli, C. Noviello, G. Ludeno, G. Esposito, F. Soldovieri, and I. Catapano, "24 ghz fmcw mimo radar for marine target localization: A feasibility study," *IEEE Access*, vol. 10, pp. 68 240–68 256, 2022.

[41] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: Why and how you should (still) use dbscan," *ACM Trans. Database Syst.*, vol. 42, no. 3, jul 2017.

[42] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining*, J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 160–172.

[43] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," *SIGMOD Rec.*, vol. 28, no. 2, p. 49–60, 1999.

[44] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[45] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural Networks*, vol. 1, no. 2, pp. 119–130, 1988.

[46] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[47] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into deep learning*. Cambridge University Press, 2023.

[48] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[49] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015.

[50] I. Sobel, G. Feldman *et al.*, "A 3x3 isotropic gradient operator for image processing," *a talk at the Stanford Artificial Project in*, vol. 1968, pp. 271–272, 1968.

[51] H. Scharr, "Optimal operators in digital image processing," Ph.D. dissertation, University of Heidelberg, Germany, 2000. [Online]. Available: http://archiv.ub.uni-heidelberg.de/volltextserver/volltexte/2000/962/pdf/Diss.pdf

[52] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[53] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[54] V. A. Feraru, R. E. Andersen, and E. Boukas, "Towards an autonomous uav-based system to assist search and rescue operations in man overboard incidents," in *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2020, pp. 57–64.

[55] S. Cheong, W. Jung, Y. S. Lim, and Y.-H. Park, "Thermal-infrared remote-target detection system for maritime rescue using 3-d game-based data augmentation with gan," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–13, 2024.

[56] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 4015–4026.

[57] L. Wanninger, "Introduction to network rtk," *IAG Working Group*, vol. 4, no. 1, pp. 2003–2007, 2004.

[58] G. Boquet, X. Vilajosana, and B. Martinez, "Feasibility of providing high-precision gnss correction data through non-terrestrial networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–15, 2024.

[59] J. F. Zumberge, M. B. Heflin, D. C. Jefferson, M. M. Watkins, and F. H. Webb, "Precise point positioning for the efficient and robust analysis of gps data from large networks," *Journal of geophysical research: solid earth*, vol. 102, no. B3, pp. 5005–5017, 1997.

[60] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 116–121.

[61] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 4372–4378.

[62] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 1–4.

[63] X. Zhang, H. Zhai, J. Liu, Z. Wang, and H. Sun, "Real-time infrared and visible image fusion network using adaptive pixel weighting strategy," *Information Fusion*, vol. 99, p. 101863, 2023.

[64] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures*, P. S. Schenker, Ed., vol. 1611, International Society for Optics and Photonics. SPIE, 1992, pp. 586 – 606.

[65] M. Magnusson, "The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection," Ph.D. dissertation, Örebro universitet, 2009.

[66] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[67] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[68] L. I. Kuncheva, *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2014.

[69] A. P. Dempster, *Upper and Lower Probabilities Induced by a Multivalued Mapping*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 57–72.

[70] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[71] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.

[72] R. Luo and M. Kay, "Multisensor integration and fusion in intelligent systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 901–931, 1989.

[73] S. S. Blackman and R. Popoli, "Design and analysis of modern tracking systems," *(No Title)*, 1999.

[74] Y. Bar-Shalom, T. E. Fortmann, and P. G. Cable, "Tracking and data association," 1990.

[75] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2001.

[76] D. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[77] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[78] E. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 2000, pp. 153–158.

[79] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *IEE Proceedings F (Radar and Signal Processing)*, vol. 140, pp. 107–113, 1993.

[80] J.-S. Ha, S.-R. Im, W.-K. Lee, D.-H. Kim, and J.-K. Ryu, "Radar based obstacle detection system for autonomous unmanned surface vehicles," in *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, 2021, pp. 863–867.

[81] G. Gennarelli, C. Noviello, G. Ludeno, G. Esposito, F. Soldovieri, and I. Catapano, "24 ghz fmcw mimo radar for marine target localization: A feasibility study," *IEEE Access*, vol. 10, pp. 68 240–68 256, 2022.

[82] G. Ludeno, G. Esposito, I. Catapano, F. Soldovieri, and G. Gennarelli, "Short range k-band radar for maritime security," in *2023 IEEE Conference on Antenna Measurements and Applications (CAMA)*, 2023, pp. 30–33.

[83] J.-S. Ha, S.-R. Im, W.-K. Lee, D.-H. Kim, and J.-K. Ryu, "Radar based obstacle detection system for autonomous unmanned surface vehicles," in *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, 2021, pp. 863–867.

[84] L. Deng, T. Guo, H. Wang, Z. Chi, Z. Wu, and R. Yuan, "Obstacle detection of unmanned surface vehicle based on lidar point cloud data," in *OCEANS 2022, Hampton Roads*, 2022, pp. 1–8.

[85] Y. Xie, C. Nanlal, and Y. Liu, "Reliable lidar-based ship detection and tracking for autonomous surface vehicles in busy maritime environments," *Ocean Engineering*, vol. 312, p. 119288, 2024.

[86] Y. Eustache, C. Seguin, A. Pecout, A. Foucher, J. Laurent, and D. Heller, "Marine object detection using lidar on an unmanned surface vehicle," *IEEE Access*, vol. 13, pp. 121 658–121 669, 2025.

[87] X. Chen, Y. Liu, and K. Achuthan, "Wodis: Water obstacle detection network based on image segmentation for autonomous surface vehicles in maritime environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.

[88] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[89] D. Nunes, J. Fortuna, B. Damas, and R. Ventura, "Real-time vision based obstacle detection in maritime environments," in *2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2022, pp. 243–248.

[90] B. Bovcon and M. Kristan, "Wasr—a water segmentation and refinement maritime obstacle detection network," *IEEE Transactions on Cybernetics*, vol. 52, no. 12, pp. 12 661–12 674, 2022.

[91] W. Liu, Y. Liu, B. A. Gunawan, and R. Bucknall, "Practical moving target detection in maritime environments using fuzzy multi-sensor data fusion," *International Journal of Fuzzy Systems*, vol. 23, no. 6, pp. 1860–1878, 2021.

[92] T. Clunie, M. DeFilippo, M. Sacarny, and P. Robinette, "Development of a perception system for an autonomous surface vehicle using monocular camera, lidar, and marine radar," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 14 112–14 119.

[93] Øystein Kaarstad Helgesen, K. Vasstein, E. F. Brekke, and A. Stahl, "Heterogeneous multi-sensor tracking for an autonomous surface vehicle in a littoral environment," *Ocean Engineering*, vol. 252, p. 111168, 2022.

[94] L. Stanislas and M. Dunbabin, "Multimodal sensor fusion for robust obstacle detection and classification in the maritime robotx challenge," *IEEE Journal of Oceanic Engineering*, vol. 44, no. 2, pp. 343–351, 2019.

[95] Z. Jian, Z. Liang, Z. Li-nan, and L. Nan, "The design and research of intelligent search and rescue device based on sonar detection and marine battery," in *2017 International Conference on Computer Network, Electronic and Automation (ICCNEA)*, 2017, pp. 383–387.

[96] C. M. Kang, L. C. Yeh, S. Y. R. Jie, T. J. Pei, and H. Nugroho, "Design of usv for search and rescue in shallow water," in *Intelligent Robotics and Applications*. Cham: Springer International Publishing, 2020, pp. 351–363.

[97] J. Taipalmaa, J. Raitoharju, J. P. Queralta, T. Westerlund, and M. Gabbouj, "On automatic person-in-water detection for marine search and rescue operations," *IEEE Access*, vol. 12, pp. 52 428–52 438, 2024.

[98] Y. Wang, W. Liu, J. Liu, and C. Sun, "Cooperative usv–uav marine search and rescue with visual navigation and reinforcement learning-based control," *ISA Transactions*, vol. 137, pp. 222–235, 2023.

[99] A. Ancy Micheal and S. Sivaramakrishnan, "Human detection and tracking for drone based marine surveillance," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2024, pp. 1–7.

[100] V. Tsekenis, C. K. Armeniakos, V. Nikolaidis, P. S. Bithas, and A. G. Kanatas, "Machine learning-assisted man overboard detection using radars," *Electronics*, vol. 10, no. 11, 2021.

[101] E. Lygouras, N. Santavas, A. Taitzoglou, K. Tarchanidis, A. Mitropoulos, and A. Gasteratos, "Unsupervised human detection with an embedded vision system on a fully autonomous uav for search and rescue operations," *Sensors*, vol. 19, no. 16, 2019.

[102] D. Li, L. Yu, W. Jin, R. Zhang, J. Feng, and N. Fu, "An improved detection method of human target at sea based on yolov3," in *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, 2021, pp. 100–103.

[103] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[104] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The apolloscape dataset for autonomous driving," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 1067–10 676.

[105] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, "Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 1993–2016, 2017.

[106] M. Kristan, V. Sulić Kenk, S. Kovačič, and J. Perš, "Fast image-based obstacle detection from unmanned surface vehicles," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 641–654, 2016.

[107] B. Bovcon, R. Mandeljc, J. Perš, and M. Kristan, "Stereo obstacle detection for unmanned surface vehicles by imu-assisted semantic segmentation," *Robotics and Autonomous Systems*, vol. 104, pp. 1–13, 2018.

[108] Autonomous Underwater Vehicles Laboratory, "MIT Sea Grant Marine Perception Dataset," public dataset. [Online]. Available: https://seagrant.mit.edu/auvlab-datasetsmarine-perception-1/

[109] Y. Xie, "Maritime LiDAR-based Ship Detection Datasets," *Figshare*, 7 2024.

[110] D. Chung, J. Kim, C. Lee, and J. Kim, "Pohang canal dataset: A multimodal maritime dataset for autonomous navigation in restricted waters," *The International Journal of Robotics Research*, vol. 42, no. 12, pp. 1104–1114, 2023.

[111] A. Odetti, G. Bruzzone, M. Altosole, M. Viviani, and M. Caccia, "Swamp, an autonomous surface vehicle expressly designed for extremely shallow waters," *Ocean Engineering*, vol. 216, p. 108205, 2020.

[112] A. Odetti, M. Altosole, G. Bruzzone, M. Caccia, and M. Viviani, "Design and construction of a modular pump-jet thruster for autonomous surface vehicle operations in extremely shallow water," *Journal of Marine Science and Engineering*, vol. 7, no. 7, 2019.

[113] R. A. Light, "Mosquitto: server and client implementation of the mqtt protocol," *Journal of Open Source Software*, vol. 2, no. 13, p. 265, 2017.

[114] P. A. P. Moran, "Notes on continuous stochastic phenomena," *Biometrika*, vol. 37, no. 1/2, pp. 17–23, 1950.

[115] R. C. Geary, "The contiguity ratio and statistical mapping," *The Incorporated Statistician*, vol. 5, no. 3, pp. 115–146, 1954.

[116] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[117] R. A. Fisher, *Statistical Methods for Research Workers*. New York, NY: Springer New York, 1992, pp. 66–70.

[118] W. H. Kruskal and W. A. Wallis, "Use of Ranks in One-Criterion Variance Analysis," *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.

[119] T. K. Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, 1995, pp. 278–282 vol.1.

[120] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 1st ed. USA: Prentice Hall PTR, 1994.

[121] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273—-297, 1995.

[122] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[123] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[124] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[125] R. Zaccone, "A dynamic programming approach to the collision avoidance of autonomous ships," *Mathematics*, vol. 12, no. 10, 2024.

[126] P. A. Thomas, J. Barr, B. Balaji, and K. White, "An open source framework for tracking and state estimation ('stone soup')," in *Signal Processing, Sensor/Information Fusion, and Target Recognition XXVI*, vol. 10200. SPIE, 2017, pp. 62–71.

[127] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[128] J. P. Snyder, "Map projections: A working manual," in *Professional Paper*. U.S. Government Printing Office, 1987, vol. 1395.

[129] P. S. Maybeck, *The Kalman Filter: An Introduction to Concepts*. New York, NY: Springer New York, 1990, pp. 194–204.

[130] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[131] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: A hands-on survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 12, pp. 2633–2651, 2016.

[132] World Meteorological Organization, "World meteorological organization," https://wmo.int, 2025, accessed on July 24, 2025.

[133] J. M. Rivera Velázquez, L. Khoudour, G. Saint Pierre, P. Duthon, S. Liandrat, F. Bernardin, S. Fiss, I. Ivanov, and R. Peleg, "Analysis of thermal imaging performance under extreme foggy conditions: Applications to autonomous driving," *Journal of Imaging*, vol. 8, no. 11, 2022.

[134] S. Nirgudkar, M. DeFilippo, M. Sacarny, M. Benjamin, and P. Robinette, "Massmind: Massachusetts marine infrared dataset."