FACULTY OF ENGINEERING OF UNIVERSITY OF PORTO

Multi-domain Contextual Awareness using Unmanned Surface Vehicles for Offshore Wind Farms Inspection

Daniel Filipe Barros Campos

Doctoral Program in Electrical and Computer Engineering

Supervisor: Professor Andry Maykol Gomes Pinto

Second Supervisor: Professor Aníbal Castilho Coimbra de Matos

October 7, 2024

© Daniel Campos, 2024

For my father.

Abstract

The rapid growth of maritime infrastructure, particularly offshore wind farms, underscores the need for efficient inspection and monitoring procedures to ensure safe and effective Operation and Maintenance (O&M) tasks. Current inspection practices for offshore wind farms are largely human-based, and requiring extensive resources (e.g., human assets, support vessels), leading to higher costs and increased safety risks. These O&M activities span large-scale environments with complex structures both above and below water, ranging from cable arrays, seabed and underwater structure surveys, for ensuring no structural damage and no sediment displacement exists. The introduction of autonomous vehicles, such as Unmanned Surface Vehicles (USVs) enable cost-effective O&M strategies that ensure repeatability and reduce risks to human life. Existing inspection solutions often rely on dedicated vehicles with specific sensor payloads, leading to inefficiencies in multi-domain inspections. Additionally, ensuring safe navigation near critical infrastructure is crucial.

This thesis introduces novel multi-domain USV systems designed to address the limitations of traditional single-domain USVs in offshore inspections. These vehicles are equipped with multi-domain sensor payloads capable of acquiring data from both surface and underwater environments, with a focus of enhancing scene understanding to increase inspection safety. The multi-domain mapping architecture (M2A) uses this data to create comprehensive 3D representations, while the FRONTIER approach enhances these representations by identifying missing data. The local knowledge representation generated from this data is then used for collision avoidance, ensuring safe navigation with minimal deviation from planned routes. This innovative system improves offshore inspections by reducing the need for multiple operations, thereby lowering costs, time, and mission-related risks.

Two multi-domain platforms SENSE and Nautilus USVs were developed, each providing modular and adaptable architectures tailored to specific operational requirements for data acquisition, navigation and perception. SENSE features a modular framework that allows easy transitions between configurations, making it versatile for both data acquisition and methodology validation under calm maritime conditions. Nautilus, with its compact monohull design, is optimized for offshore deployment, capable of beyond visual line-of-sight operations and resilient in moderate sea states. Both platforms share a standardized software architecture, MUSA, which defines a flexible four-layer structure (mission, task, skill, action) for mission description and USV control.

Extensive field tests at the Marina de Leça harbor and the ATLANTIS coastal test bed validated the navigation and data acquisition capabilities of both platforms. The ROAM@CRAS dataset, created from these tests, provides insights into challenges such as sensor fusion and obstacle detection. Nautilus demonstrated its resilience in harsh conditions during offshore tests, achieving a Technology Readiness Level (TRL) of 8. These tests confirmed Nautilus's reliability in varying environmental conditions and its suitability for autonomous operations on a floating offshore wind farm in the Atlantic.

M2A enables the simultaneous reconstruction of above-water and underwater environments,

facilitating comprehensive inspections of offshore wind farm infrastructure. FRONTIER addresses incomplete data by filling occluded regions and enriching local knowledge with semantic features. The M2A methodology integrates data from various sensors and filters outliers to achieve accurate 3D reconstructions. Tests demonstrated its efficiency, with positional errors of 0.049 meters in simulations and 0.07 meters in real-world data, and real-time reconstruction averaging 300 milliseconds. FRONTIER enhances semantic instance segmentation using a template-based framework, significantly improving processing time and classification accuracy.

A novel proactive obstacle avoidance architecture, AVOA, is being presented to leverage the USVs' multi-domain capabilities. This architecture uses a semantic-based topological representation for contextual awareness and a multicriteria heuristic function to ensure collision-free navigation with minimal route deviation. Comprehensive trials validated its effectiveness, showing substantial improvements in travel time and safety compared to state-of-the-art algorithms.

This thesis is addressing the limitations of autonomous multi-domain inspections using USVs in offshore wind farms. It presents a novel pipeline for safe and comprehensive inspections through modular hardware and software development for multi-domain USVs. This approach reduces human operator reliance, enables beyond visual line of sight (BVLOS) missions and improves cost-efficiency. The extensive experimental validation highlights the proposed technologies' accuracy and robustness, offering a significant advancement in autonomous maritime operations, particularly in demanding environments like offshore wind farms.

Resumo

O rápido crescimento das infraestruturas marítimas, particularmente dos parques eólicos offshore, destaca a necessidade de procedimentos eficientes de inspeção e monitorização para garantir a execução segura e eficaz das tarefas de Operação e Manutenção (O&M). As práticas atuais de inspeção para parques eólicos offshore são predominantemente baseadas no uso de operadores, exigindo vastos recursos (por exemplo, recursos humanos e embarcações de apoio), o que leva a custos mais elevados e ao aumento dos riscos de segurança. Estas atividades de O&M abrangem ambientes de grande escala com estruturas complexas tanto acima quanto abaixo da água, variando de inspeções de cabos, levantamentos de leito marinho e de estruturas subaquáticas para garantir a ausência de danos estruturais e deslocamento de sedimentos. A introdução de veículos autónomos, como Veículos de Superfície Não Tripulados (USVs), pode criar estratégias de O&M menos dispendiosas que garantem repetibilidade e reduzem os riscos para a vida humana. As soluções de inspeção existentes frequentemente dependem de veículos dedicados com cargas úteis de sensores específicas, resultando em ineficiências nas inspeções multi-domínio. Além disso, garantir a navegação segura próxima a infraestruturas críticas é fundamental.

Esta tese introduz novos sistemas de USV multi-domínio projetados para abordar as limitações dos USVs tradicionais de domínio único nas inspeções offshore. Estes veículos são equipados com cargas úteis de sensores multi-domínio capazes de adquirir dados tanto do ambiente de superfície como subaquático, com foco em aumentar a compreensão da cena para melhorar a segurança nas tarefas de inspeção. A arquitetura de mapeamento multi-domínio (M2A) utiliza esses dados para criar representações 3D abrangentes, enquanto que a abordagem FRONTIER aprimora essas representações identificando dados ocluídos. A representação de conhecimento local gerada a partir desses dados é então usada para evitar colisões, garantindo uma navegação segura com desvio mínimo das rotas planeadas. Este sistema inovador melhora as inspeções offshore ao reduzir a necessidade de múltiplas operações, diminuindo assim os custos, o tempo e os riscos relacionados às missões.

Duas plataformas multi-domínio, SENSE e Nautilus USVs, foram desenvolvidas, cada uma com arquiteturas modulares e adaptáveis, ajustadas aos requisitos operacionais específicos para aquisição de dados, navegação e percepção. O SENSE possui uma estrutura modular que permite transições fáceis entre configurações, tornando-o versátil tanto para aquisição de dados quanto para validação de metodologias em condições marítimas calmas. O Nautilus, com seu design compacto de casco único, é otimizado para utilização em offshore, sendo capaz de operar além da linha de vista e resiste a estados de mar moderados. Ambas as plataformas compartilham uma arquitetura de software padronizada, MUSA, que define uma estrutura flexível de quatro camadas (missão, tarefa, habilidade, ação) para a descrição de missões e controlo dos USVs.

Extensos testes de campo no porto da Marina de Leça e no centro de testes costeiro do ATLANTIS validaram as capacidades de navegação e aquisição de dados para ambas as plataformas. O conjunto de dados ROAM@CRAS, criado a partir desses testes, permite investigar desafios como fusão de sensores e detecção de obstáculos. O Nautilus demonstrou sua resiliência em

condições adversas durante os testes offshore, alcançando um Nível de Prontidão Tecnológica (TRL) de 8. Estes testes confirmaram a confiabilidade do Nautilus em várias condições ambientais e sua adequação para operações autónomas num parque eólico offshore flutuante no Atlântico.

O M2A permite a reconstrução simultânea de ambientes acima e abaixo da água, facilitando inspeções abrangentes da infraestrutura dos parques eólicos offshore. O FRONTIER aborda dados incompletos preenchendo regiões ocluídas e enriquecendo o conhecimento local com características semânticas. A metodologia M2A integra dados de vários sensores e filtra outliers para alcançar reconstruções 3D precisas. Os testes demonstraram a sua eficiência, com erros de posicionamento de 0,049 metros em simulação e 0,07 metros em testes reais. Além disso, permitiu uma reconstrução em tempo real com uma média de 300 milissegundos. FRONTIER aprimora a segmentação de instâncias semânticas usando uma estrutura baseada em templates, melhorando significativamente o tempo de processamento e a precisão da classificação.

Uma nova arquitetura proativa de evasão de obstáculos, o AVOA, é apresentada, aproveitando as capacidades multi-domínio dos USVs e sistemas de piloto automático existentes. Esta arquitetura usa uma representação topológica baseada em semântica para a criação do contexto e uma função heurística multicritério para garantir navegação sem colisões com desvio mínimo da rota. Testes abrangentes validaram a sua eficácia, demonstrando melhorias substanciais no tempo de viagem e segurança em comparação com algoritmos do estado da arte.

Assim, esta tese abordou as limitações das inspeções autónomas multi-domínio usando USVs em parques eólicos offshore, apresentando uma nova metodologia para inspeções seguras e completas através do desenvolvimento de hardware e software modular para USVs multi-domínio. Esta abordagem reduz a dependência de operadores humanos, permite missões além da linha de vista (BVLOS) e melhora a eficiência dos custos. A extensa validação experimental destaca a precisão e robustez das tecnologias propostas, oferecendo um avanço significativo nas operações marítimas autónomas, particularmente em ambientes desafiantes como os parques eólicos offshore.

Acknowledgments

First, I would like to start by thanking my supervisor and friend, Andry Pinto. Without him, this adventure would probably never have started. He is a great person who is not only there for the work but who was always available to scold, teach, guide, listen and encourage. So I thank him for always pushing me to do more and better and for sharing his experience gained through his hard work.

To my co-supervisor, Aníbal Matos, for welcoming me back to INESC TEC, namely at CRAS, who gave me the confidence to take the step towards this doctorate and ensuring the conditions to continue.

To my journey companions, some of whom I have seen since they were little Bambis until they became young adults, and others who came along the way, namely Renato Silva, Pedro Leite, Maria Inês, Rafael Claro, Pedro Almeida, João Marques, João Dionísio, Francisco Neves, Diogo Silva, Eduardo Gonçalves, Pedro Rocha and Lourenço Pinho. To all of you, thank you for the good times, camaraderie and fun. Not only for that, but also for the joint struggle in a doctorate as well as in all the moments from crazy missions that took us to the edge including in a boat (here I include our dreamer supervisor Andry Pinto), to the cleaning sessions and the convivialities. I have immense pleasure and pride in what you do and may come to do if you continue with a fighting spirit.

To my girlfriend and daughter's mother, Ana Menezes, thank you for fighting with me, putting up with the moments of despair, supporting me and for the unconditional love you gave me. These were difficult times, with a pandemic and a daughter in between, that were overcome by your side. To my daughter, Matilde Campos, who with her one year of life has already taught me so much and gives me the encouragement every day to grow bit by bit! Everything I do, I do it for you!

To my brother, Hugo Campos, for being there in the difficult moments we surpassed together on a personal level as well as in the good times! I am glad that beside brothers we could work together and for that thank you for helping me to fulfil a crazy mission of building a vehicle from scratch in just a few months, destroying it on a few occasions, until we managed to take it offshore and be the first to put an autonomous vessel on the offshore wind farm.

To my extremely loving mother, Júlia Campos, whom I admire and adore, thank you for everything! Even when all seemed difficult, even for her, she was always there and encouraged me to do more and be better! Thank you for instilling in me, together with my father, the values that make me the man I am today.

To my father, João Campos, my hero and example! Unfortunately, he was not able to physically accompany the last miles of this marathon, but I am sure he is following my path from where he is. He was my inspiration, a great man, with a patience and kindness that marked everything he touched! I hope you are as proud of me as I always were of you!

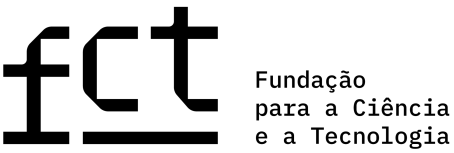
At last, to all who contributed directly or indirectly to this doctorate, a thank you.

Daniel Filipe Barros Campos

Official Acknowledgments

Daniel Filipe Campos acknowledges the Portuguese Government through the FCT (Foundation for Science and Technology) for the PhD grant SFRH/BD/144263/2019 (DOI: https://doi.org/10.54499/SFRH/BD/144263/2019), without which this research work would not have been possible.

This work was also financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme and by National Funds through the Portuguese funding agency, FCT (Foundation for Science and Technology), within project POCI-01-0145-FEDER-030010 and by the European Union's Horizon 2020 - The EU Framework Programme for Research and Innovation 2014-2020, under grant agreement No. 871571.



"Success is not final, failure is not fatal:
it is the courage to continue that counts."

Winston S. Churchill

Contents

1 Introduction **1**
- 1.1 Motivation . . . 4
- 1.2 Objectives . . . 7
- 1.3 Document structure . . . 7
- 1.4 Contributions . . . 9
  - 1.4.1 Achievements . . . 9
  - 1.4.2 Publications . . . 10
  - 1.4.3 Research outcomes . . . 11
  - 1.4.4 Videos . . . 12

2 Literature review **13**
- 2.1 General overview . . . 13
- 2.2 Unmanned Surface Vehicles . . . 18
- 2.3 Scene understanding . . . 23
  - 2.3.1 Multi-domain mapping . . . 24
  - 2.3.2 Semantic instance completion . . . 27
  - 2.3.3 Scene representation . . . 29
- 2.4 Safe navigation using collision avoidance for USVs . . . 30
- 2.5 Critical review . . . 33

3 Design of unmanned surface vehicles with multi-domain perception **37**
- 3.1 Introduction . . . 37
- 3.2 SENSE - Autonomous vessel for multi-domain inspection and maintenance . . . 39
- 3.3 Nautilus - An autonomous surface vehicle with a multi-layer software architecture for offshore inspection . . . 44
- 3.4 Software architecture . . . 50
  - 3.4.1 MUSA - Multi-layer universal software architecture . . . 51
  - 3.4.2 Software design . . . 54
- 3.5 Results . . . 60
  - 3.5.1 SENSE . . . 60
  - 3.5.2 Nautilus . . . 71
- 3.6 Conclusions . . . 81

4 Multi-domain 3D scene understanding **83**
- 4.1 Introduction . . . 83
- 4.2 Multi-domain mapping architecture . . . 86
  - 4.2.1 Emersed data preprocessing . . . 87
  - 4.2.2 Immersed data preprocessing . . . 88

4.2.3 Multi-domain map representation 91
4.3 FRONTIER - Offshore wind farm foundations data completion architecture . . . 92
4.3.1 Segmentation of potential candidates 93
4.3.2 Learning-based classification of offshore wind turbine foundations 95
4.3.3 Template-based instance registration 101
4.4 Results 102
4.4.1 M2A - Multi-domain mapping 103
4.4.2 FRONTIER - Offshore wind farm foundations data completion architecture 114
4.5 Conclusions 128

5 Safe navigation in offshore scenarios 131
5.1 Introduction 131
5.2 Proactive obstacle avoidance architecture 132
5.2.1 Local Knowledge Representation 133
5.2.2 AVOA - Adaptive velocity obstacle avoidance 138
5.3 Results 146
5.3.1 AVOA - Adaptive velocity obstacle avoidance 148
5.4 Conclusion 153

6 Conclusion 155
6.1 Future work 158

A Supplementary information 159
A.1 ROAM@CRAS dataset - Sensor placement and topics 159

List of Figures

1.1 Proposed concept of USV inspection of an offshore wind farm, including multi-domain inspection, data augmentation and collision-free navigation. 5

2.1 Offshore wind turbines foundation types and depth constraints. 14
2.2 Different domains of the marine environment. 15
2.3 Multi-domain operation pipeline schematic. 18
2.4 Available multi-domain USVs solutions. 20
2.5 Available single-domain USVs solutions. 21
2.6 Three layers of Pedersen *et al.* architecture: primitives, skills and tasks. 22
2.7 Albore *et al.* three layers architecture: decision, skills and functional. 23

3.1 SENSE USV operating in *Marina de Leça*. 39
3.2 Setup of sensors used on the SENSE. The navigation sensors coordinate systems (left), the SENSE (centre) and the perception sensors coordinate systems (right). The red, green and blue lines represent the X, Y and Z axis, respectively. 41
3.3 SENSE hardware architecture and functionalities layout 43
3.4 The 3D representation of a SENSE acquired frame, based on data from the LiDAR and Sonar sensors. Stereoscopic images are presented on top. 44
3.5 Nautilus USV operating in Viana do Castelo. 45
3.6 Nautilus CAD with the sensor payload attached. The blue, red, and green lines are distance, imaging, and navigation sensors, respectively. The pink lines are the actuators. 45
3.7 Navigation data from the Nautilus. The green line corresponds to the GPS positions and the attitude meter depicts the current roll, pitch and yaw of the vehicle. . 47
3.8 Nautilus perception data gathered in Viana do Castelo. 48
3.9 Nautilus hardware and functionalities layout. The blue and red dashed boxes are the navigation and communication, and perception submodules, respectively. The grey, green, blue, orange and yellow boxes are the processing units, communications, sensors interfaces, actuator drive units, and actuators, correspondingly. . . . 49
3.10 Nautilus internal and external communication links. The red, green, blue, orange and black lines represent the PoE, ethernet, serial, mavlink and power outputs. . . 50
3.11 MUSA: the four layers that make the proposed architecture. 52
3.12 The state machine diagram of the execution of each individual element of the proposed architecture. 53
3.13 Inspection mission using MUSA. Blue, green, yellow and red boxes represent mission, task, skill and action layers, respectively. 55
3.14 The state machine diagram of the continuous execution routine of the proposed autonomous operation skill, $\mathscr{S}_{11}$. 59

3.15 *Marina de Leça* harbor - 240 mooring places with entry and yacht basins depth of 4 m and 2-3.5 m, respectively. 60
3.16 Trajectory performed by the SENSE on *Marina de Leça*. The green line is the RTK observations and the red arrow points to the base station position. 62
3.17 Comparison between GPS modules and the RTK ground truth. The green, blue and purple lines are the RTK, SN-GPS and VK-GPS data, respectively. The error scale factor for the RTK and the SN-GPS is minimal, thus the lines are overlapped. 64
3.18 Data sample of the available IMUs. 65
3.19 Locations of the example snapshots. Each flag corresponds to data acquisition instants. 66
3.20 SENSE stereo camera sensor. 67
3.21 Camera data acquired in each location flagged in Figure 3.19. 67
3.22 LiDAR and MBES data acquired in each location flagged in Figure 3.19. 68
3.23 Odometry acquired through the GPS and IMU fusion. The green and red lines are the RTK observations and the GPS/IMU odometry, respectively. 69
3.24 SENSE bathymetry map. The rainbow presented goes from red (higher depths) to blue (lower depths). The green corresponds to intermediate values. 70
3.25 Multi-domain map on a partial sequence of the ROAM@CRAS. 70
3.26 ATLANTIS stream line for the validation and demonstration of robotic solutions. 71
3.27 Nautilus trial locations in Satellite view. 72
3.28 The ATLANTIS coastal test centre satellite view and shore view. 73
3.29 Nautilus on the transportation cart 74
3.30 Nautilus deployment on port and offshore. 74
3.31 Nautilus trials in the ATLANTIS coastal test centre. 75
3.32 Nautilus data sample acquired on different weather conditions. 76
3.33 Results of task level experiments performed in a port environment conducted in Viana do Castelo, Portugal. Light green circles represent the distance from the goal-points from which the continuous execution state of the skill enters its last stage, while darker green circles represent the maximum allowed margin of error for the skill to succeed. Each line represents the execution of a single skill. The green line represents the first skill to be executed, followed by a blue, a red and a purple line. Initial points are marked by a bigger red cross. 77
3.34 Multi-domain map created by Nautilus on the ATLANTIS coastal test centre with different tide conditions. The red to purple colour scale represents the Z-axis variation from minimum to maximum, respectively. 78
3.35 WindFloat Atlantic wind farm. 79
3.36 Nautilus deployment offshore. 80
3.37 Nautilus operations during the second and third offshore trials. 80
3.38 Nautilus data sample gathered during the third offshore trial on the Windfloat Atlantic wind farm. 81

4.1 Different assets for both domains of the marine environment from the Motion Robotics lab. 84
4.2 Incomplete multi-domain reconstruction of a jacket foundation. The emersed domain represents the dry structure in the splashing zone, the immersed domain depicts the submerged structure of the foundation and the seabed. The frontier region shows the information gap in the splashing zone. 85
4.3 Proposed multi-domain mapping architecture, M2A. 86

4.4 Reflected observations retrieved by the LiDAR on a docking structure. The blue rectangle highlights the outlier data and the scale depicts the height in meters. . . 87
4.5 Multibeam Echosounder scan. Rainbow-color line represents a scale variation, where the red to magenta color goes from higher to lower scales, respectively. . . 89
4.6 Sliding window iterations, where *S* is the MBES scan, and *SW* is the local map. Scans to forget are defined in red and to keep in green. 90
4.7 Sliding window management process. 90
4.8 Point cloud registration approach, where It=1 is the first iteration. 91
4.9 Flowchart depicting the FRONTIER pipeline for semantic instance completion of offshore turbine foundations based on multi-domain maps through detection (localization and classification) and alignment. 93
4.10 Offshore wind turbine foundations included in the classification model. 95
4.11 RI-PointTransf architecture design. The orange, red, green, purple and blue boxes are the patchifier, patch embedding, position embedding, transformer encoder, and task-specific head, respectively. 96
4.12 Oriented bounding box assessment using Durius. Initially computes the convex hull **(a)**, then uses its vertices as input for PCA **(b)**. In **(c)** is the resulting oriented bounding box. 97
4.13 Convolution stem architecture. $C1$, $C2$, $C3$, $C4$ and $C5$ are convolution blocks composed by a 1D convolutional layer, batch normalization and a Leaky-ReLU. $M1$, $M2$ and $M3$ are the max pool layers, and *SE* is a Squeeze-and-Excitation block. 99
4.14 Squeeze-and-Excitation architecture, where $\otimes$ represents matrix multiplication. . 99
4.15 Template-based instance registration. Red, green, black, blue and orange arrows represents the structure template point cloud, the instance observation point clouds, the detection output, the target bounding box centre and the output transformation, respectively. 101
4.16 Implementation and validation offshore wind farm scenario. 103
4.17 Inspection trajectory. The orange and green points depicts the path and the dock, respectively. 104
4.18 Odometry results during the sample trajectory execution. The orange, blue and green dots are the ground truth, GPS-IMU fusion and starting pose, respectively. . 105
4.19 LiDAR data acquired on the ATLANTIS coastal test centre. 107
4.20 Multi-domain map of the simulated scenario. 108
4.21 Scenario 1: Simulated scene and USV trajectory. 109
4.22 Multi-domain map of the *Marina de Leça* harbor yacht basin. 111
4.23 Aerial view of the multi-domain map of the *Marina de Leça* harbor yacht basin. . 112
4.24 Multi-domain mapping from Nautilus trials in the ATLANTIS coastal test centre and in the Windfloat Atlantic offshore wind farm. 113
4.25 Aerial view of the ATLANTIS coastal test bed. 114
4.26 Results of the clustering algorithms applied to LiDAR and MBES data acquired on the ATLANTIS coastal test center. 115
4.27 Comparison of the segmentation results. 116
4.28 ModelNet10 data samples and distribution. 118
4.29 Normalized confusion matrix for the ModelNet10 NR/NR test data. 120
4.30 Data acquisition procedures. 122
4.31 Foundation dataset performance evaluation: confusion matrix and most ambiguous class comparison. 124

4.32 Confusion matrix for Durius and Windfloat foundation, and background point clouds captured during the Nautilus trials. 125
4.33 Example of the detection output of the WindFloat data with 99.7% confidence. . 126
4.34 Registration of the CAD template (yellow points) to the detection output (blue points) for: fixed foundation JacketK on the left and two floating foundations Confloat in the middle and Durius on the right. 127
4.35 Result of the FRONTIER applied to a point cloud acquired during the Nautilus approval in principle trials. 127

5.1 Proactive obstacle avoidance architecture. 133
5.2 Example of a topological graph. Blue and green circles are the pose and landmark nodes, respectively. Dashed and solid arrows are prior and constraints edges, respectively. 134
5.3 Landmark node association rules. The yellow circle is the estimated landmark node pose, the light orange ellipse is the node covariance, the darker ellipse is the area for the covariance distance threshold, and the yellow star is the estimated feature. The dashed circle represents the distance threshold employed to determine if the estimated feature belongs to an existing landmark node. 135
5.4 Gazebo scenario for testing the topological graph. 136
5.5 Local knowledge representation of the scenario. Yellow and orange lines depict the pose-pose and pose-landmark edges. The blue and pink circles are the initial and end poses of the USV, respectively. 136
5.6 Approximate conditions of the scenario to be reconstructed, namely a port environment in the ATLANTIS coastal test centre with six docked vessels. 137
5.7 Topological representation of the scenario overlapped with the emersed domain 3D map. 138
5.8 Protective zone PZ_A^B of an obstacle B relative to agent A. 139
5.9 Protective zone representation, where the red, green, orange and yellow arrows denote the X-axis, Y-axis, obstacle velocity in its position and projected to the agent, respectively. The blue, green, and red circles represent the obstacle and agent shapes, and the respective Minkowski sums. The black region indicates the unfeasible areas and the HSV color space is used to denote velocity and heading for all non-colliding velocities. 141
5.10 Protective zone representation with kinematics, where the red, green, orange and yellow arrows denote the X-axis, Y-axis, obstacle velocity in its position and projected to the agent, respectively. The blue, green, and red circles represent the obstacle and agent shapes, and the respective Minkowski sums. The black region indicates the unfeasible areas and the HSV color space is used to denote velocity and heading for all non-colliding velocities. 142
5.11 Protective zones representation and local minima analysis. 144
5.12 Obstacle avoidance with static obstacles, where the green and red dots represents the desired goal location and the static obstacles, respectively. 147
5.13 Obstacle avoidance with a moving dynamic obstacle simulator, where the green and red dots represents the desired goal location and the static obstacles, respectively. The orange USV is the crossing vessel. 147
5.14 Obstacle avoidance with static obstacles. 148
5.15 Robot velocities for the cluttered environment. 149
5.16 Obstacle avoidance with a moving dynamic obstacle. The pink dot depicts the crossing vessel and the blue arrow its direction. 150

5.17 Robot velocities for the avoidance scenario B. 151
5.18 Robot velocities for the avoidance scenario C. 152

List of Tables

2.1 Performance comparison of USVs and other vehicles. 18
2.2 Comparison of available USVs, where: C - Catamaran; M - Monohull; E - Electric; and D - Diesel. 34

3.1 Technical specifications of the SENSE. 42
3.2 Technical specifications of the Nautilus. 47
3.3 Summary of the conditions during Nautilus trials. 72

4.1 Ripple configuration for simulation. 103
4.2 Odometry results with distinct ripple and wind configurations. 106
4.3 Multi-domain map RMSE score with distinct ripple configurations. 110
4.4 Comparison of segmentation results for the scenario depicted in figure 4.26a. . . 116
4.5 Experiment configurations. 117
4.6 Result comparison on ModelNet10 test dataset. NR/NR - No rotation; NR/AR - No rotation training, random rotation testing; Drop - Decay rate from NR/NR to NR/NA experiments. 119
4.7 Ablation study results using ModelNet10. NR/NR - No rotation, NR/AR - No rotation training, random rotation testing; Drop - Decay rate of rotation experiments; μ_{acc} - Class accuracy average for NR/NR; σ_{acc} - Class accuracy standard deviation for NR/NR; σ_{cls} - Standard deviation for each class in NR/NR. 121
4.8 Accuracy per class using foundation datasets. The global accuracy is 98.97% and the class accuracy standard deviation, σ_{acc}, of 2.02% 123

5.1 Wave configuration for simulation. 148

A.1 Sensors available in SENSE, respective frames and position in meters from figure 3.2. 159
A.2 List of the data available in the ROAM@CRAS and the respective ROS topics. . 160

Abbreviations

AHRS	Absolute Heading Reference System
ASV	Autonomous Surface Vehicle
AUV	Autonomous Underwater Vehicle
AVOA	Adaptive Velocity Obstacle Avoidance
BEVA-FPC	Bird's-Eye View Adaptive Farthest Point Clustering
BMS	Battery Management System
BVLOS	Beyond Visual Line-of-Sight
CAD	Computer Aided Design
CAGR	Compound Annual Growth Rate
CBAM	Convolutional Block Attention Module
CLS	Classification Token
CNN	Convolutional Neural Network
COLREG	Convention on the International Regulations for Preventing Collisions at Sea
CTV	Crew Transfer Vessel
DSG	Dynamic Scene Graph
DVL	Doppler Velocity Log
EKF	Extended Kalman Filter
FFN	Feed-Forward Network
FoV	Field of View
FPC	Farthest Point Clustering
FPS	Farthest Point Sampling
FRONTIER	oFfshore wind faRm fOundatioNs daTa completIon architEctuRe
GICP	Generalized Iterative Closest Point
GRP	Glass Reinforced Plastic
GRU	Gated Recurrent Unit
GVO	Generalized Velocity Obstacle
INS	Inertial Navigation System
ICP	Iterative Closest Point
IR	InfraRed
KNN	K-Nearest Neighbors
LBL	Long Baseline Acoustic Positioning Systems
LCOE	Levelized Cost Of Energy
LiDAR	Light Detection And Ranging
LM-ICP	Levenberg-Marquardt Iterative Closest Point
M2A	Multi-domain Mapping Architecture
MBES	MultiBeam EchoSounder
MDDPG	Modified Deep Deterministic Policy Gradient
MLP	MultiLayer Perceptron

MUSA	Multi-layer Universal Software Architecture
NavCom	Navigation and Communication
NDT	Non-Destructive Testing
NTP	Network Time Protocol
O&M	Operations and Maintenance
OHVS	Offshore High Voltage Station
OPEX	OPerating EXpense
OWF	Offshore Wind Farm
OWT	Offshore Wind Turbine
PCA	Principal Component Analysis
PNC	Perception, Navigation and Communication
POE	Power Over Ethernet
PPO	Pproximal Policy Optimization
PSO	Particle Swarm Optimization
PSU	Power Supply Unit
RC	Radio Controlled
RI-PointTransf	Rotation Invariant Point-cloud Transformer
RMSE	Root Mean Square Error
ROAM@CRAS	haRbor multidOmAin Mapping dataset
ROS	Robot Operating System
ROV	Remotely Operated Vehicle
RTK	Real-Time Kinematics
RVO	Reciprocal Velocity Obstacle
SCC	Shore Control Centre
SE	Squeeze-and-Excitation
SENSE	autonomouS vEssel for multi-domaiN inSpection and maintEnance
SOR	Statistical Outlier Removal
SSS	Side-Scan Sonar
TRL	Technology Readiness Level
UAV	Unmanned Aerial Vehicle
USV	Unmanned Surface Vehicles
UTM	Universal Transverse Mercator
UUV	Unmanned Underwater Vehicles
VGICP	Voxelized Generalized Iterative Closest Point
VO	Velocity Obstacle

Chapter 1

Introduction

The rising energy demands coupled with the urgent need to address climate change has motivated the global search for sustainable energy sources. International efforts such as the COP28 target to triple renewable energy by 2030 solidify this commitment[1]. Policymakers and international institutions acknowledge the urgency of expanding readily available renewable technologies like wind and solar photovoltaic to create a cleaner and more flexible energy system, aligning with the Paris Agreement trajectory. From both, wind energy stands out due to its minimal environmental footprint, high sustainability, and cost-effectiveness [1]. According to the 2024 Global Wind Report[2], a significant acceleration in annual wind energy installations is required, from 117 GW in 2023 to at least 320 GW annually by 2030. In this context, offshore wind energy emerges as a key solution. Compared to onshore wind turbines, offshore wind farms (OWFs) presents several advantages, namely abundant wind resources, lower turbulence, extensive space for deployment, lower transmission and distribution losses, and reduced visual and noise pollution [2]. OWFs are projected to experience a compound annual growth rate (CAGR) of 28% over the next five years, with an expected addition of 138 GW of capacity worldwide between 2024 and 2028[2].

Moreover, future offshore wind farms will be installed further from the coastal areas and at deeper locations [1]. While this exploration can incur in higher costs, floating platforms offer potential advantages such as enhanced wind conditions, reduced wave loading, lower installation costs, and minimal visual impact. Examples like the WindFloat Atlantic[3] highlights this approach viability. Commissioned in 2020, this OWF demonstrated successful operation with a total capacity of 25MW using three turbines in a 100 meters depth area at Viana do Castelo. It generated a total cumulative production of 259 GWh between July 2020 and the end of 2023.

Nevertheless, this emergent sector is driven by economic pressures from uncertainties associated with the inspection and maintenance of offshore assets which significantly contributes to an increase of the LCOE (levelized cost of energy). Operation and maintenance (O&M) costs

[1] COP28, 2024. *Global Renewables and Energy Efficiency Pledge.* URL: `https://www.cop28.com/en/global-renewables-and-energy-efficiency-pledge`

[2] Global Wind Energy Council, 2024. *Global Wind Report 2024*. URL: `https://gwec.net/wp-content/uploads/2024/04/GWR-2024_digital-version_final-1.pdf`

[3] Windplus. *World's first semi-submersible floating offshore wind farm.* URL: `https://www.windfloat-atlantic.com/`

constitute a large portion of an offshore wind farm's LCOE, accounting for 23% of the total investment compared to only 5% for onshore wind turbines [2]. Therefore, reducing O&M costs represents a critical strategy for controlling the LCOE of offshore wind farms. Maintenance costs, particularly the daily use of crew transfer vessels (CTVs) for round trips to and from the OWFs, and revenue losses associated with the duration of turbine downtime, are crucial areas for cost reduction [3]. Both the European Union and the wind power sector are actively pursuing ambitious cost-reduction goals. The development and implementation of novel technologies and improved solutions are key to achieving a more cost-efficient O&M process.

Current inspection practices for OWFs are mostly human-based and demands the use of a vast amount of resources (e.g. human assets, support vessels, etc) which leads to higher inspection costs and safety risks [4]. This O&M activities requires the inspection to be executed throughout a large scale environment with multiple complex structures, both bellow and above water. These operations range from cable arrays, seabed and underwater (immersed) structure surveys, for ensuring no structural damage and no sediment displacement exists, to inspecting above-water (emersed) structures like wind turbine blades, towers, and transition pieces to ensure the safety and longevity of the offshore wind farm. For such, Remotely Operated Vehicles (ROVs) and divers are used to clean and inspect the underwater components, while rope access and, more recently, tele-operated Unmanned Aerial Vehicles (UAVs) are used for the dry parts, namely the blades and tower [5]. Furthermore, inspections are heavily influenced by environmental conditions, with a limited operational window due to harsh weather with the access availability bellow 40% of the year using a small boat and an average operation window length of two days [6].

The application of autonomous systems, such as Unmanned Surface Vehicles (USVs), for OWF inspection offers significant potential benefits. Studies suggest that autonomous systems could reduce overall inspection costs by over 20% of a wind farm's total investment, and significantly reduce health and safety risks associated with offshore operations [3]. Moreover, their introduction can:

- Enhance safety: autonomous systems can mitigate these risks by performing inspections remotely or with reduced number of operators on-site. Human involvement in hazardous offshore environments inherently poses a risk of injury or loss of life. According to the incident report published by *G+ Global Offshore Wind* [4], in 2022 a total of 325 incidents occurred during turbine operations, where 44 required medical treatment and 19 demanded emergency response and medical evacuation. As well, collisions between vessels and offshore infrastructures are one of the most common accidents in water areas like OWFs, and 70% of cable faults are caused by unauthorized anchoring and fishing [7];

- Improve cost-effectiveness: autonomous systems can contribute to a more cost-effective O&M strategy by minimizing downtime and response times, both crucial factors for lowering operating expenses (OPEX). The *G+ Global Offshore Wind* [4] asserted that 24 work days were loss due to incidents in turbine operations, impacting the downtime. The unpredictable nature of weather forecasts also complicates long-term maintenance scheduling [8];

- Extend OWF lifespan: the autonomous systems capacity to conduct regular structural inspections can contribute to extending the operational life of an OWF. This translates to a direct reduction in the LCOE by minimizing maintenance costs over the lifetime of the wind farm.

Therefore, the introduction of autonomous vehicles can create cost-effective O&M strategies ensuring repeatability and less risk for human life. In the context of offshore wind farms, research has highlighted the utility of USVs for inspecting foundations and monitoring mooring lines from the water surface, thereby assisting in maintenance scheduling decisions [9]. These systems offer a versatile solution for assessing infrastructure conditions across both above-water and underwater domains. However, these distinct domains of the marine environment present unique challenges for sensor availability and performance on autonomous vehicles, namely for localization technologies (e.g. GPS is only available on the surface) and perception sensors (e.g. 3D LiDAR is used for surface exploration, while sonar is for underwater). Consequently, tasks demanding different sensor capabilities often require the use of separate vehicles. Typically, USVs are tailor-made for specific tasks within a single domain, such as using a multibeam echosounder (MBES) for bathymetric surveys [10] or performing 3D reconstructions of emersed components in marine structures [11]. Additionally, for operating in close proximity to critical infrastructures requires safe navigation, being crucial to detect possible obstacles and hazardous trajectories to avoid collisions in harsh offshore environments. Nevertheless, current technology still didn't reached the TRL (Technology readiness level) required by this sector to reduce the cost of deployment. USVs represent a promising research area within maritime robotics with the potential to revolutionize various maritime applications, including inspection, transportation, and patrolling [12, 13, 14, 15, 7]. Moreover, their ability to perform inspections across diverse sectors has already been demonstrated, where the deployment of USVs has significantly enhanced marine monitoring capabilities, offering a cost-effective and safe method [16, 17].

The increased autonomy and flexibility provided by USVs can significantly reduce the need for personnel to transition to the foundation, dive, or work in hazardous offshore conditions, thereby enhancing operational safety. Specifically, USVs can inspect scour protection conditions by performing bathymetric surveys of the seabed. Currently, this task requires a working crew of ten operators and a dedicated service vessel [18]. Furthermore, USVs can address the challenging task of inspecting the dry areas of the splash zone in the transition piece by conducting close-range visual and 3D inspections using multiple sensors, such as visual cameras and LiDAR. This task is presently carried out by a working crew of two qualified personnel requiring a crew transfer vessel (CTV) to transition to the foundation [18].

Therefore, endowing USVs with the capability to navigate safely in close proximity to offshore wind turbines (OWTs) and collect multi-domain data simultaneously will enhance the overall reliability and sustainability of offshore wind energy O&M. By integrating multi-domain sensors and advanced safety features, USVs can potentially reduce the number of required assets on-site by at least half and the number of operators by at least three times, thereby improving both safety and operational efficiency. Introducing this solution into offshore scenarios will reduce O&M

risks and enable the creation of a comprehensive representation of the scenario using a single system. Furthermore, ensuring fully electric operation offers the added benefit of reducing the CO_2 footprint compared to traditional inspection methods that rely on fossil fuel-powered assets. For example, the use of CTVs for OWF O&M contributes approximately 3.6 tonnes of CO_2e[4] per GWh of energy generated by the wind farm per year [19].

1.1 Motivation

In recent years, marine robotic perception techniques have advanced significantly, providing solutions for monitoring biodiversity in oceans, reef corals, harbors, and assisting in the inspection of moored vessels, offshore sites, and submarines [20, 21, 22, 23]. These solutions are typically tailored to address specific technological challenges and are constrained by the operational environment. However, environmental perception in marine contexts can be affected by field-of-view limitations and partial or noisy information, which leads to incomplete scenario characterization. Moreover, the dual-domain nature of marine environments (above-water and underwater) presents additional challenges, as different sensor requirements and limitations exist in each domain.

As a result, marine robotic perception remains a challenging task, with recent research focusing on data acquisition, object detection, and 3D reconstructions, primarily for single-domain applications. USVs are no exception; they are predominantly designed for single-domain tasks, such as bathymetric surveys. Consequently, using this technology often requires multiple specialized solutions, increasing the risk of missing operational windows and escalating associated costs. As such, addressing this challenge is pivotal to maximize operational effectiveness and information completeness while minimizing asset requirements for O&M operations. Moreover, autonomy is key in enabling beyond visual line-of-sight (BVLOS) missions, allowing USVs to operate autonomously at considerable distances from operators.

Therefore, enhancing scenario representation to capture detailed information from both the emersed and immersed domains simultaneously is crucial for ensuring mission success and safety in complex and demanding environments. Creating 3D multi-domain maps facilitates the structural assessment and inspection of both the splash zone and the integrity of scour protection. Due to sensor payload limitations, such as field of view constraints and challenges associated with domain transitions, elements of the structure can be occluded, increasing the risk of collision. Detecting and completing the 3D data of surrounding structures can mitigate this risk by providing a more comprehensive depiction of the scenario. These multi-domain representations improve USVs' BVLOS capabilities near infrastructures. Moreover, USVs must be able to anticipate and respond to unforeseen events and effectively mitigate potentially hazardous situations, extending beyond the capabilities of conventional autopilots that follow predefined routes. Integrating preventive collision avoidance algorithms will enhance USVs' autonomy by leveraging the increased multi-domain contextual awareness. These methodologies enable USVs to dynamically adjust

[4] CO_2 equivalent

their motion in response to both static and dynamic obstacles, thereby significantly improving overall operational safety.

This thesis proposes USV technologies for extending multi-domain perception capabilities and safe autonomous navigation to revolutionize the inspection processes of offshore wind farms. The focus lies on creating a multi-domain contextual awareness system for both inspection and safe navigation of USVs within OWFs. This framework aims to improve safety, expand mission capabilities, facilitate autonomous operations and ultimately reduce the dependency on on-site human intervention. Figure 1.1 illustrates the proposed concept for offshore USV operation, showcasing multi-domain structural inspection, data augmentation, and predictive collision avoidance on a floating platform within an OWF.

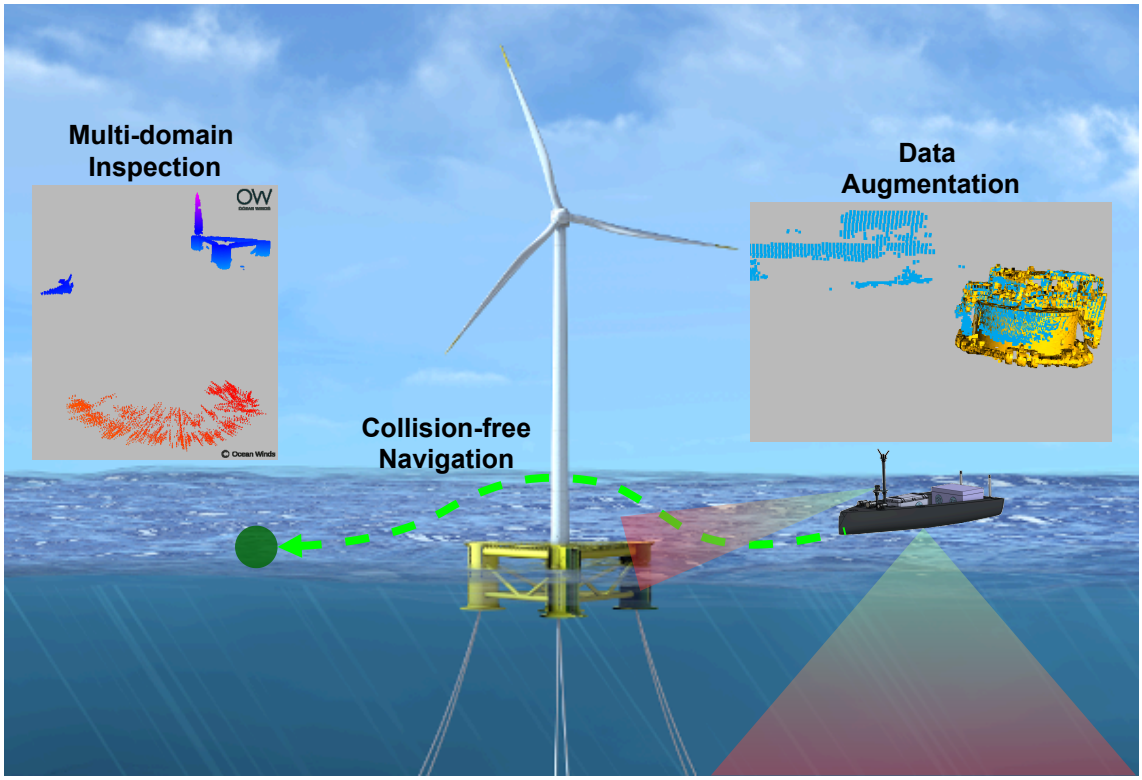


Figure 1.1: Proposed concept of USV inspection of an offshore wind farm, including multi-domain inspection, data augmentation and collision-free navigation.

Routine inspections are critical for maintaining the health and productivity of OWFs. By enabling early detection of defects or irregularities in wind turbines, a USV can play a vital role in optimizing O&M practices, presenting benefits, such as proactive maintenance, minimizing downtime and associated revenue losses, and informed decision-making, where the detailed inspection data allows for more precise scheduling of maintenance strategies, optimizing resource allocation and cost management. A key focus of this thesis is the development of USVs equipped with multi-domain sensor capabilities to comprehensively address the unique inspection requirements of OWFs. These capabilities will enable USVs to:

- Create multi-domain 3D representations by simultaneously collecting data from both above-water and underwater domains USVs will generate comprehensive 3D representations of the OWF infrastructure. This complete observation reduces the logistical complexity and associated costs of traditional inspection methods;
- Enhance inspection efficiency since multi-domain USVs will capture a wider range of critical information during inspections, including details of the transition piece and seabed bathymetry. This level of detail is essential for thorough and effective monitoring and for enabling BVLOS;
- Increase safety as USVs with robust collision avoidance methodologies contribute to safer operations in close proximity to critical infrastructures, such as the OWTs. By using multi-domain information to create contextual awareness of the scenario, USVs can proactively prevent collisions, enhancing the overall safety of the operations even BVLOS.

Therefore, the proposed approach leverages the multi-domain capabilities of the USV to acquire 3D information and to create a comprehensive map of the environment, capturing the splash zone components, the seabed and immersed parts in a unified representation. This comprehensive 3D mapping is crucial for the structural assessment of OWTs. However, the USV's field-of-view limitations can result in information gaps in the generated 3D representations, especially in the transitional areas between domains. To address these challenges, the proposed solution incorporates semantic instance completion principles to augment the multi-domain representation, thereby depicting otherwise occluded components. Given that the structures installed offshore are known, the captured information can be supplemented by detecting the structure and using its CAD (Computer-Aided Design) models to fill in the missing details. This augmented multi-domain information further allows for the definition of semantic features from the collected sensor data, contributing to the creation of local knowledge about the OWF, enabling safe and efficient autonomous operations. This local knowledge is enhanced by integrating multi-modal information provided by object detection algorithms, such as LiDAR-based dock detectors [24] and camera-based vessel detectors [25]. Each detector offers unique insights, requiring the creation of a standardized representation to unify the data from various detectors into a cohesive format.

Consequently, a semantic-based representation is proposed, using high-level features with a uniform structure to correlate the USV's pose with the surrounding objects. This standardized representation supports collision avoidance methodologies, enabling proactive measures to prevent hazardous situations and perform evasive actions upon detecting potential imminent collision threats [26]. To this end, this thesis introduces a proactive obstacle avoidance algorithm that not only assesses the current state of the surroundings but also anticipates future risks. The collision avoidance approach uses the standardized semantic-based representation to define a configuration space, which facilitates the selection of a safe trajectory for the USV. By integrating information such as object pose and velocity, the predictive obstacle avoidance architecture can make informed decisions, anticipate potential collisions in advance and initiate appropriate maneuvers proactively. This ensures that the USV navigates safely, even in dynamic and uncertain maritime environments.

1.2 Objectives

The main goal for this work is to contribute for the development of a next-generation USV. It will be endowed with a multi-domain sensor payload to enable the creation of a comprehensive and coherent reconstruction of marine assets, namely offshore wind farms. This research work is focused in five objectives:

- Design and development of modular USVs for autonomous multi-domain data acquisition in marine inspections. The USVs will prioritize modular hardware and software with multi-domain sensor payload configuration, operation under diverse conditions, enhanced weather resistance, advanced BVLOS capabilities, and real-time data processing;
- Propose a versatile multi-domain mapping architecture for USVs that integrates heterogeneous sensors for emersed and immersed domains. This architecture will focus on comprehensive 3D representations, enhanced robustness to noise and inconsistencies in sensor data, and reliable operation across diverse scenarios;
- Enhance BVLOS perception through multi-sensor algorithms that extends the contextual awareness using semantic instance completion principles and creates a local knowledge representation with semantic-based features for collision avoidance architectures;
- Expand the safety of USV autonomous navigation through a real-time predictive obstacle avoidance architecture that performs early threat detection and enables evasive maneuvers in dynamic and uncertain environments;
- Perform extensive experiments in near-real and offshore scenarios to demonstrate the USVs' ability to execute multiple inspection tasks simultaneously, such as transition piece and seabed mapping or scour protection inspection, and the deployability from the port or a vessel in offshore.

The proposed framework seeks to facilitate autonomous inspections by USVs, thereby minimizing the demand for on-site human intervention. By eliminating technological barriers, it opens the way for USVs to conduct beyond visual line-of-sight missions. This innovative solution tackles key challenges in multi-domain inspections, representing a pioneering advancement that enhances operational efficiency and reliability, particularly in challenging environments like in offshore wind farms.

1.3 Document structure

This document is organized as follows:

Chapter 2 provides a comprehensive literature review focusing on challenges and implications of employing USVs for inspection tasks in offshore wind farms. This chapter offers an extensive

overview of existing inspection methodologies used within OWFs. Additionally, it explores the available solutions involving USVs, examines methodologies for scene understanding, particularly by emphasizing the development of multi-domain 3D representations and semantic instance completion techniques. Furthermore, this chapter surveys collision avoidance architectures, with a particular emphasis on reactive solutions endowed with predictive capabilities.

Chapter 3 outlines the development of the designed USVs, that represents the next generation of modular multi-domain USVs. This research focus on the design of two distinct USVs, namely SENSE [27] and Nautilus [28], engineered to operate under varied environmental conditions. The chapter describes the modular multi-domain hardware design of each vehicle, along with the shared multi-layer software architecture. Moreover, it encompasses a comprehensive presentation of experiments and validations conducted to assess the overall design and performance of the systems in both near-real and real environments, including offshore trials.

Chapter 4 focuses on multi-domain architectures designed to increase the inspection capabilities and the scenario representation completeness. Initially, the chapter outlines the multi-domain mapping architecture (M2A) [29, 30], which focuses on establishing a generic pipeline for generating 3D multi-domain representations. This involves using USV localization to facilitate the preprocessing and fusion of heterogeneous sensors, specifically multibeam echosounder (MBES) and light detection and ranging (LiDAR). Subsequently, the chapter introduces the FRONTIER (oFfshore wind faRm fOundatioNs daTa completIon architEctuRe), a semantic instance completion architecture designed to fill in raw data within the frontier region. This aims to reduce information gaps and reveal hidden obstacles. The FRONTIER uses customized clustering algorithms for LiDAR and MBES to segment objects, followed by a novel deep learning architecture for classifying foundations. Furthermore, the chapter presents the results for both architectures, validated through simulation, near-real, and real environments. This provides a comprehensive multi-domain inspection tool for the mission and lays the groundwork for creating high-level semantic features to be used in generating contextual awareness for the safe navigation architecture.

Chapter 5 introduces the proactive obstacle avoidance architecture, designed to operate effectively in dynamic and uncertain scenarios. The chapter outlines the safe navigation pipeline, beginning with the creation of a local knowledge representation. It defines a feature representation and consolidates custom detections from various sources into a unified representation. A semantic topological representation using the USV's odometry and semantic features. Subsequently, the chapter presents the AVOA (adaptive velocity obstacle avoidance) algorithm [31], which uses the local knowledge representation as input. Furthermore, the chapter provides an extensive set of experimental validations demonstrating the performance of AVOA in terms of smoothness, safety, and effectiveness of the avoidance maneuver. This ensures safe navigation with preventive capabilities, enabling the anticipation of potential collisions and the execution of appropriate maneuvers proactively.

Chapter 6 provides the major conclusions derived from this thesis, including the novelty of the work and its achievements. Moreover, a set of guidelines for future works is addressed.

Appendix A presents a supplementary information about the ROAM@CRAS dataset.

1.4 Contributions

The contributions resulting from this thesis, namely the achievements, related publications, research outcomes, and the most relevant videos for dissemination, are listed in the following subsections.

1.4.1 Achievements

- Development of a multi-domain mapping architecture (M2A) for USVs, to reconstruct the scenario using heterogeneous sensors, operating in both the emersed and immersed regions. It generates 3D maps using a MBES for the immersed domain and sparse point clouds from a LiDAR for the emersed region. It tackles challenges associated with data fusion related to low update rates, and noisy and non-correlated measurements due to the limited field-of-view;

- The FRONTIER algorithm leverages CAD templates to approximate real-world structures for 3D semantic instance completion. It employs multi-domain data to segment objects, classify them by foundation type or background, and uses registration techniques to align templates with the object of interest detections;

- Definition of a 3D deep learning classification method, the RI-PointTransf (Rotation Invariant Point-cloud Transformer), featuring a transformer architecture with five key components: a patchifier, patch embedding, position embedding, a transformer encoder consisting of multiple transformer layers, and a task-specific head. This framework achieves nearly complete rotational invariance and surpasses the performance of state-of-the-art 3D classification methods;

- A local knowledge representation that standardizes multi-modal detectors information into an unified feature representation used to create a semantic topological graph. This provides the input for proactive obstacle avoidance algorithms;

- A proactive obstacle avoidance methodology for USVs, AVOA, designed to operate in real-time. The method enables a safe interaction between the USV and the scenario. It introduces a protective zones concept to describe a position and velocity configuration space model of the detected obstacles, and uses an optimization procedure to estimate an effective velocity through a minimization of a multicriteria condition that characterize both the collision risk and the course deviation;

- The ROAM@CRAS (haRbor multidOmAin Mapping dataset) [32] is the first dataset to offer simultaneous data collection from both surface and underwater domains in the maritime environment, using a single vehicle. Designed to be ROS-Compliant, this dataset serves as a resource for evaluating and addressing various data fusion challenges, particularly in odometry estimation and multi-domain mapping techniques. The dataset encompasses real-world application scenarios, including navigating through waves generated by larger vessels, maneuvering in close proximity to structures, dynamic environments, and instances of missing data. This dataset can be accessed through a public repository at `https://rdm.inesctec.pt/dataset/nis-2020-002`;

- Design of a compact USV, the Nautilus, with a multi-domain sensor payload and capable of operating in offshore conditions and beyond visual line-of-sight. It has a fully electric design and contributes to a environmentally sustainable technology aiming to reduce the need for traditional vessels, thereby minimizing CO_2 emissions. It uses an integrated and adaptable multi-layer software architecture designed for O&M operations, MUSA (Multi-layer Universal Software Architecture). The architecture comprises four distinct levels - mission, task, skill, and action - designed to ease the programming task for developers and engineers, making it faster and simpler to program different autonomous agents, while avoiding code redundancy;

- Nautilus was the first USV to operate autonomously at a floating offshore wind farm located in the Atlantic, namely the WindFloat Atlantic wind farm owned by Ocean Winds;

- Design of a modular USV, the SENSE (autonomouS vEssel for multi-domaiN inSpection and maintEnance), fitted to navigate on harbors and offshore infrastructures under calm sea states to acquire multi-domain data for inspection, surveying and mapping tasks.

1.4.2 Publications

The main publications made related to this research are presented below.

- [31] D. F. Campos, A. Matos, and A. M. Pinto, "An Adaptive Velocity Obstacle Avoidance Algorithm for Autonomous Surface Vehicles," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8089–8096, IEEE, nov 2019

- [29] D. F. Campos, A. Matos, and A. M. Pinto, "Multi-domain Mapping for Offshore Asset Inspection using an Autonomous Surface Vehicle," in *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 221–226, IEEE, apr 2020

- [30] D. F. Campos, A. Matos, and A. M. Pinto, "Multi-domain inspection of offshore wind farms using an autonomous surface vehicle," *SN Applied Sciences*, vol. 3, p. 455, 4 2021

- [27] D. F. Campos, A. Matos, and A. M. Pinto, "Modular multi-domain aware autonomous surface vehicle for inspection," *IEEE Access*, vol. 10, pp. 113355–113375, 2022

- [28] D. F. Campos, E. P. Gonçalves, H. J. Campos, M. I. Pereira, and A. M. Pinto, "Nautilus: An autonomous surface vehicle with a multilayer software architecture for offshore inspection," *Journal of Field Robotics*, vol. 41, no. 4, pp. 966–990, 2024

Moreover, supplementary publications highlight relevant contributions that support the research, namely:

- [33] R. Silva, P. Leite, D. Campos, and A. M. Pinto, "Hybrid approach to estimate a collision-free velocity for autonomous surface vehicles," in *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 1–6, 2019
- [34] P. N. Leite, R. J. Silva, D. F. Campos, and A. M. Pinto, "Dense disparity maps from rgb and sparse depth information using deep regression models," in *Image Analysis and Recognition* (A. Campilho, F. Karray, and Z. Wang, eds.), (Cham), pp. 379–392, Springer International Publishing, 2020
- [16] D. F. Campos, M. Pereira, A. Matos, and A. M. Pinto, "Diius - distributed perception for inspection of aquatic structures," in *OCEANS 2021: San Diego – Porto*, pp. 1–5, 2021
- [17] A. M. Pinto, J. V. A. Marques, D. F. Campos, N. Abreu, A. Matos, M. Jussi, R. Berglund, J. Halme, P. Tikka, J. Formiga, C. Verrecchia, S. Langiano, C. Santos, N. Sá, J.-J. Stoker, F. Calderoni, S. Govindaraj, A. But, L. Gale, D. Ribas, N. Hurtós, E. Vidal, P. Ridao, P. Chieslak, N. Palomeras, S. Barberis, and L. Aceto, "Atlantis - the atlantic testing platform for maritime robotics," in *OCEANS 2021: San Diego – Porto*, pp. 1–5, 2021
- [35] F. S. Neves, H. J. Campos, D. F. Campos, R. M. Claro, P. N. Almeida, J. V. Marques, and A. M. Pinto, "Shore control centre for multi-domain heterogeneous robotic vehicles," in *OCEANS 2023 - Limerick*, pp. 1–4, 2023
- [36] A. M. Pinto, J. V. A. Marques, N. Abreu, D. F. Campos, M. I. Pereira, E. Gonçalves, H. J. Campos, P. Pereira, F. Neves, A. Matos, S. Govindaraj, and L. Durand, "Atlantis coastal testbed: A near-real playground for the testing and validation of robotics for o&m," in *OCEANS 2023 - Limerick*, pp. 1–5, 2023
- [37] A. M. Pinto, A. Matos, J. V. A. Marques, D. F. Campos, M. I. Pereira, R. Claro, E. Mikola, J. Formiga, M. E. Mobachi, J.-J. Stoker, J. Prevosto, S. Govindaraj, D. Ribas, P. Ridao, and L. Aceto, "Promoting the use of robotics in the inspection and maintenance of offshore wind," in *Robotics and Automation Solutions for Inspection and Maintenance in Critical Infrastructures* (K. Loupos, ed.), pp. 202–240, Now Publishers, 2024

1.4.3 Research outcomes

This thesis has contributed several key elements to the field, including:

a) Robotic platforms (2):

(i) SENSE USV

(ii) Nautilus USV

b) Open-source datasets (1):

(i) ROAM@CRAS

c) Software applications (6):

(i) MUSA (Multi-layer Universal Software Architecture)

(ii) M2A (Multi-domain Mapping Architecture)

(iii) BEVA-FPC (Bird's-Eye View Adaptive Farthest Point Clustering)

(iv) RI-PointTransf (Rotation Invariant Point-cloud Transformer)

(v) FRONTIER (oFfshore wind faRm fOundatioNs daTa completIon architEctuRe)

(vi) AVOA (Adaptative Velocity Obstacle Avoidance)

1.4.4 Videos

Relevant videos related to this research are presented below:

- *AVOA - Adaptive Velocity Obstacle Avoidance* - URL: `https://youtu.be/jOTsiBlszuA`
- *SENSE - The Autonomous Vessel for Multi-domain Inspection and Maintenance* - URL: `https://youtu.be/vkJ-MVCSpdA`
- *Multi-domain mapping of Leça harbor* - URL: `https://youtu.be/EfqGu5nDuXs`
- *Nautilus USV* - URL: `https://youtu.be/12niamOnmL8`
- *Nautilus USV - Campaigns* - URL: `https://youtu.be/c-DaojSg9SI`

Chapter 2

Literature review

Offshore wind energy is a rapidly growing renewable energy source. However, ensuring the safe and efficient operation of offshore wind farms presents significant challenges. Traditional inspection methods rely heavily on human operators, exposing them to inherent risks associated with the harsh maritime environment. This literature review investigates the potential of unmanned surface vehicles (USVs) to address these challenges by enabling multi-domain inspections of offshore wind farms, where multi-domain refers to the ability to collect data from both above-water (emersed) and underwater (immersed) environments, encompassing critical components like wind turbine towers, transition pieces, foundations, mooring lines, and scour protection.

Throughout this chapter, the literature review is structured as follows: Section 2.1 provides an overview of current practices for inspecting offshore wind farms, highlighting their limitations and defining the solutions to enable autonomous inspection to be explored; Section 2.2 explores the current state of the art in USV technology and its application in O&M tasks; Section 2.3 surveys 3D scene understanding solutions, targeting multi-domain solutions for maritime inspection, completing missing information with data enhancement procedures, and towards comprehensive 3D scene representation; Section 2.4 explores methods for safe USV navigation in complex maritime environments, focusing on collision avoidance techniques. Finally, section 2.5 provides a critical review of the available solutions for USV-based multi-domain inspections of offshore wind farms, identifying limitations and opportunities for future research.

2.1 General overview

Operation and maintenance activities are considered among the most critical tasks for offshore wind farms, and they pose several challenges for various reasons. Firstly, the distance from an OWF to a port or shore reduces accessibility and increases downtime. Owning or hiring a maintenance fleet and employing a larger number of technicians can be costly. Weather conditions, particularly significant wave heights (maximum of 1.5 m) and wind speeds (maximum of $10\,\mathrm{m\,s^{-1}}$), further limit accessibility to OWFs for service vessels and personnel transfer from the

crew transfer vessel (CTV) to the offshore wind turbine, and results in higher failure rates of turbine components [38]. Moreover, if a maintenance task must be postponed due to weather issues, a longer waiting period and greater loss of power generation during downtime are likely. Even without considering the effects of weather, offshore wind turbine maintenance costs are higher than equivalent tasks onshore due to the specialized equipment required. Furthermore, the increasing size of OWTs in recent decades, aimed at improving power generation efficiency, requires larger and more specific devices for offshore maintenance and repairs [2]. Therefore, efforts to develop and implement new inspection technologies and methodologies in the offshore industry should be made to provide more efficient and cost-effective solutions while minimising human risk [39].

Currently, offshore wind turbines are supported by a variety of foundation structures, categorized into bottom-fixed and floating foundations. Bottom-fixed structures are designed for installation at depths less than 60 m, while floating structures are used for greater depths [40], as illustrated in figure 2.1 from the DHI Group[1]. These include:

- Monopile: a large-diameter steel tube driven into the seabed using a hydraulic hammer;
- Gravity based: a heavy displacement structure typically made of concrete, applying vertical pressure to the seabed;
- Jacket: a lattice-type steel structure, square in footprint, constructed of thin tubes, primarily used for large water depths;

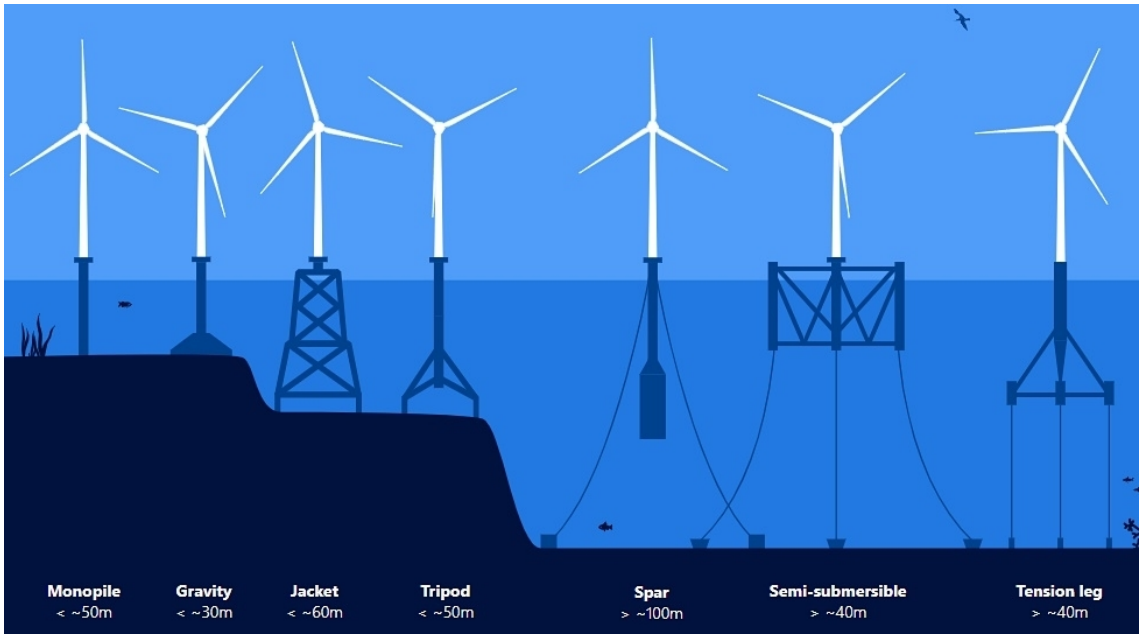


Figure 2.1: Offshore wind turbines foundation types and depth constraints[1].

[1]DHI, 2024. *Decode offshore wind engineering challenges at every stage*. URL: `https://blog.dhigroup.com/decode-offshore-wind-engineering-challenges-at-every-stage/`

- Tripod: a steel tube protruding from the ocean surface, with an anchor pile driven into the seabed to support the foundation on three legs;
- Spar-buoy: composed of a ballasted cylinder to maintain balance, with mooring lines and anchors to secure the foundation in position;
- Semi-submersible platforms: consisting of large columns, attached via beams to submerged pontoons. Columns provide hydrostatic stability through ballast tanks, while pontoons offer additional buoyancy. Mooring lines and anchors maintain the foundation's position;
- Tension leg platforms: comprises a slim cylindrical buoy with columns and arms connected to anchors via mooring lines.

These structures can be divided into two main domains (figure 2.2): the emersed (above-water) domain, consisting of the blades and tower, and the immersed (underwater) domain, comprising submerged components such as mooring lines, array cables, or scour protection. The transition piece and the floating structure are situated between both domains, being partially submerged, particularly in the splashing zone.

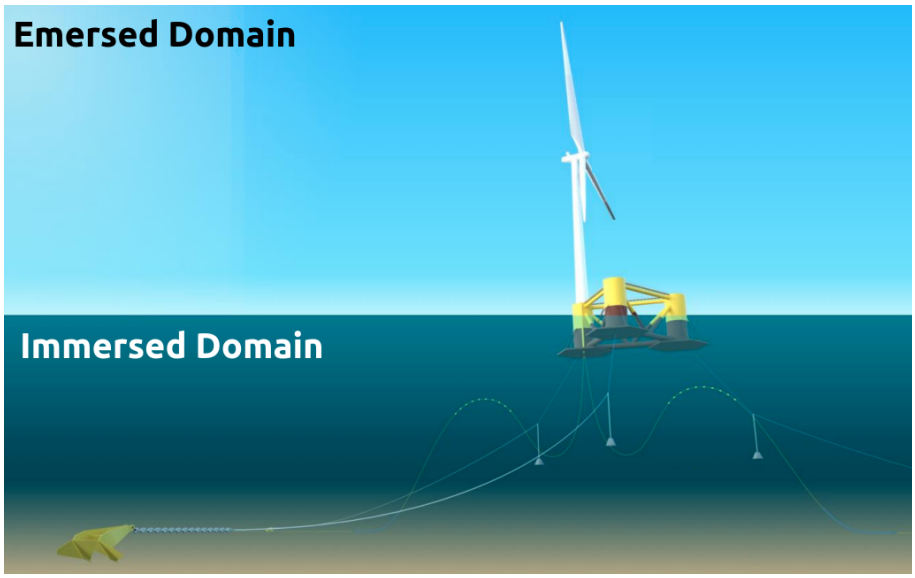


Figure 2.2: Different domains of the marine environment.

In the emersed domain, traditional inspection techniques rely on CTVs to transport workers. For the tower and blades inspections, operators equipped with rope and harness climb down the wind turbine to assess the infrastructure health, having also limited access to the blades [41]. This requires non-destructive testing (NDT) techniques, such as visual inspection, ultrasound, thermal imaging, LiDAR mapping, and electromagnetic testing to verify their integrity [42]. For the transition piece or floating structure, visual inspection in the dry area involves climbing workers, while divers and ROVs are used in the wet area for cleaning biofouling and conducting NDT. Immersed structures are inspected using divers or ROVs for structural components and bathymetric readings using service vessels or ROVs for scour monitoring [39].

Among all components, the transition piece (or floating structure) and scour protection are seen as critical elements defining the structural integrity of the entire system. The transition piece secures the turbine tower, while the scour protection ensures the stability of bottom-fixed foundations [43]. The splashing zone is particularly susceptible to environmental deterioration, including atmospheric and sea-related factors leading to issues, such as cracks, plastic deformation, galvanic corrosion and excessive marine growth. Inspections in this region involve three levels of detail [18]:

- General visual inspection: provides an overview of the integrity of the part of the support structure above water. Conducted from a CTV, it aims to identify obvious mechanical, fatigue, or corrosion damage within the main access platform and the transition piece;
- Close visual inspection: detects corrosion or fatigue damage in specific areas of the transition piece, main access platform, and access systems above water. This inspection is conducted closer to the structure (at a meter distance) to detect smaller defects and determine if NDT techniques are required. It can use specialized rope climbers towards the inspection;
- Detailed visual inspection: determines the extent of fatigue damage when cracks are detected at pre-selected welds. NDT is employed at this stage. This is carried out as a reactive measure when there is strong suspicion or evidence of fatigue damage at welds.

Regarding scour, the presence of fixed foundations alters water current flow patterns, leading to local sediment transport and the formation of scour holes. Scouring compromises foundation stability and may expose seabed cables due to erosion. Two strategies can address this issue: predicting scour hole size and dimensioning the foundation to reach equilibrium, or implementing scour protection systems around the foundation. Periodical monitoring campaigns assess the functionality of scour protection and ensure scour holes are within estimated parameters. Protection systems include riprap, geocontainers, or concrete mattresses [44]. Scour inspections monitor changes in seabed topology around bottom-fixed foundations, identifying and removing seafloor objects and debris close to the structure. Two methods are commonly used: Multibeam echosounder (MBES) bathymetry survey and side scan sonar survey using service vessels or ROVs [45].

Therefore, offshore wind energy offers a promising path towards renewable energy goals, but presents challenges in operation and maintenance due to harsh environments and human dependence. Recent advancements in offshore robotics offer promising alternatives. The growing interest on research on autonomous inspection systems for wind turbines reflects this potential [16, 17]. In this context USVs present a compelling solution for wind farm O&M, offering potential benefits, such as:

- Reduced operational costs: the ability to automate inspections with USVs can significantly reduce dependence on human personnel and enable more frequent and consistent inspections, leading to better preventive maintenance and improved uptime for OWTs. Currently, close visual inspection of the transition piece requires a working crew of two qualified personnel, a CTV, and daily rental expenses with equipment leading up to a total cost of 4870

€ per day. As for scour protection monitoring, a working crew of ten personnel and a service vessel is required leading up to a total cost of 8100 € per day, with an additional mobilization cost of 70000 € [18];

- Enhanced safety: by removing technicians from hazardous environments, USVs can significantly reduce the risks associated with offshore wind farm inspections. Currently, performing both the transition piece inspection and scour protection monitoring requires a CTV and a service vessel, along with a combined working personnel of twelve people, excluding the vessels' crew [18].

Furthermore, USVs possess distinct advantages for marine data acquisition tasks due to their communication, observation, perception, and onboard computing capabilities [46]. Their versatility and cost-effectiveness compared to crewed vessels have attracted significant research and development efforts across various sectors [47, 48, 49]. Moreover, recent studies started exploring the utility of USVs for inspecting foundations and monitoring mooring lines, aiding in critical maintenance decisions [9]. However, safe and efficient USV navigation in offshore wind farms presents unique challenges. The complex marine environment characterized by unpredictable weather conditions such as wind, waves, and currents, coupled with the substantial response times attributed to USV inertia can significantly hinder navigation [50]. Additionally, these environments span through different domains, impacting sensor performance and availability during inspections, adding to the complexity of the task.

Addressing these challenges is crucial for unlocking the full potential of USVs in offshore wind farm O&M. For such, it is essential to focus on advancements in three key areas (figure 2.3), including: enhance the situational awareness by equipping USVs with multi-domain sensor payloads to create a comprehensive picture of their surroundings, providing a multi-domain scene understanding, enabling robust decision-making during inspections; and increase the USV autonomy by developing robust collision avoidance capabilities, minimizing reliance on human intervention. Therefore, to explore the available solutions this review will highlight the current state of the art in USV technology for offshore wind farm inspections, focusing on:

- Available USV solutions: examines the capabilities of USVs to overcome limitations of traditional methods and explore different USV platforms suitable to navigate in offshore environments and with the sensor payload needed for the splashing zone inspection and scour protection monitoring;

- Scene understanding: study techniques for processing multi-domain sensor data from USVs, focusing on methods to address the creation of multi-domain 3D representations, semantic instance completion methodologies to augment the information, and scene representation techniques to be used as input for collision avoidance methodologies;

- Safe navigation: analyse algorithms for collision avoidance in complex maritime environments using USVs, particularly with proactive behaviors to avoid both static and dynamic obstacles.

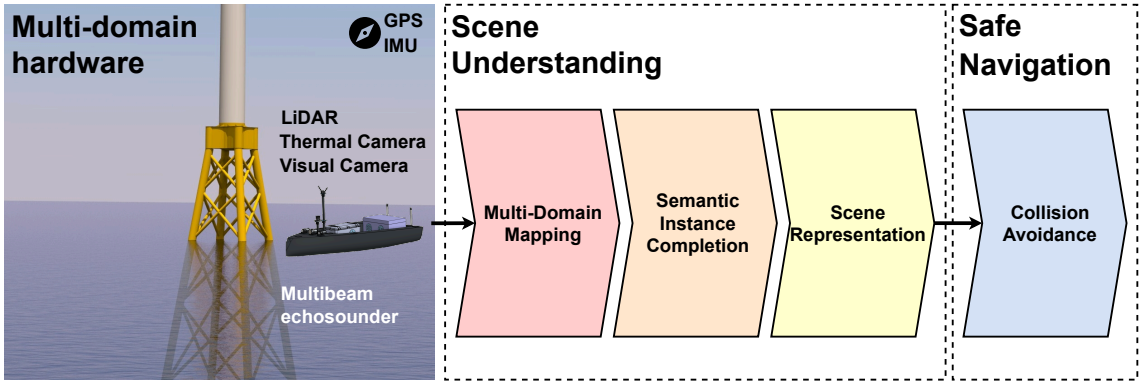


Figure 2.3: Multi-domain operation pipeline schematic.

2.2 Unmanned Surface Vehicles

USVs play a crucial role across various domains due to their cost-effectiveness, compact size, agility, and intelligence, comprising a platform and mission payload system oriented towards a specific task. This vehicles find applications in diverse civil and military fields, including environmental monitoring, search and rescue operations, hydrological exploration, regional inspection, mine clearance, or anti-submarine warfare [51]. As highlighted by Liu *et al.* (2016) [52], USVs offer several advantages over other manned or unmanned systems, such as unmanned underwater vehicles (UUVs), UAVs, float platforms, or manned ships, including: (i) ability to perform longer and more hazardous missions than manned vehicles; (ii) lower maintenance costs and greater personnel safety due to the absence of onboard crew; (iii) enhanced maneuverability and deployability in shallow waters (e.g., coastal areas) due to their low weight and compact dimensions; (iv) greater potential payload capacity enabling deeper water depth monitoring and sampling compared to other aircraft/UAVs and satellites. These advantages can be observed on the comparison provided in table 2.1 [52].

Table 2.1: Performance comparison of USVs and other vehicles.

Attributes	UUVs	Float Platforms	Satellites	Manned Ships	UAVs	Manned Aircrafts
Endurance	●	○	○	◑	●	●
Payload capacity	●	◑	●	◑	●	◑
Maneuverability	●	●	●	●	○	●
Deployability	●	●	●	●	○	●
Water depth measurement	○	◑	●	◑	●	◑
Autonomy Requirement	◑	●	●	●	◑	●

● Clear advantage of USVs ◑ Near parity ○ Clear disadvantage of USVs

Moreover, USVs leverage a combination of exteroceptive and proprioceptive sensors to precisely determine their position and attitude above water. This information is crucial for autonomous navigation and mapping. Additionally, they can be equipped with various payload sensors depending on their mission. The most common sensor for localization and mapping in

unmanned vehicles is the LiDAR. It creates point clouds that describe the shape and position of surrounding objects. Several studies have explored LiDAR integration with USVs for structural monitoring and topographic surveying in inland and coastal environments [53, 54, 55]. For underwater environments, sonar is the standard sensor payload. Similar to a 2D LiDAR, the MBES offers a series of sonar ranges and bearing information about the seafloor, which can be transformed into a point cloud. Extensive research has been conducted on using MBES with underwater vehicles and ships for ocean mapping [56, 57, 58, 59].

Therefore, the offshore wind energy industry would benefit from the employment of USVs equipped with multi-modal sensors for a cost-effective O&M of offshore wind farms, minimizing risks to human life [60, 61]. To fulfil this needs, USVs designed to operate under moderate sea states (around Douglas sea scale 4 [62]) and equipped with multi-domain sensors (LiDAR, sonar, cameras) are crucial for the inspection data acquisition. Research and commercial multi-domain solutions are currently emerging to fulfil these requirements, namely:

- *SCOUT* (Figure 2.4a): Papadopoulos *et al.*(2014) [22] developed SCOUT, a kayak with stabilizing pontoons, 3 m length, 0.5 m width and weights 90 kg. It is fully electric with a single servo-steered thruster and is equipped with GPS, IMU and DVL as navigation sensors, and five 2D laser scanners, two of them mounted on a rotating shaft, and an imaging sonar, used for perceiving the surface and underwater domains and reconstructing partially submerged structures. SCOUT is designed for calm waters with an estimated Douglas sea state of 2;

- *ASAROME sailboat* (Figure 2.4b): Plumet *et al.*(2015) [63] presented a 3.6 m sailing boat equipped with a visual camera, sonar, hydrophones, anemometer, GPS and IMU for perceiving the scenario within the ASAROME framework. It can detect obstacles and perform path planning, and was tested on a lake with calm waters (Douglas sea state 2);

- *Jung et al. USV* (Figure 2.4c): Jung *et al.* (2023) [15] used a fully electric catamaran USV measuring 4.1 m x 2.5 m, and weighing 400 kg. It is equipped with GPS, IMU, stereo cameras, LiDAR, and MBES, capable of reconstructing multi-domain scenarios. This USV was tested in rough sea conditions (Douglas sea scale 5) on the offshore wave power plant testbed owned by the Korea Research Institute of Ships and Ocean Engineering (KRISO);

- *Teledyne Z-Boat 1800RP* (Figure 2.4d): The Teledyne Z-Boat 1800RP[2] is a small USV, 1.8 m x 1.0 m, weighting 38 kg produced by Teledyne Marine. It is used for high-resolution shallow-water hydrographic surveying and incorporates MBES, LiDAR, visual camera, GPS, and IMU. The Z-Boat 1800R is designed to operate in calm waters up to a Douglas sea state of 2;

[2]Teledyne, 2023. *Z-Boat 1800RP*. URL: `https://tinyurl.com/z-boat1800rp`

- *HydroCat-180* (Figure 2.4e): Seafloor Systems' HydroCat-180[3] is a fully electric USV with a catamaran design, differential drive, measuring 5.5 m x 2.5 m, and weighing 362.87 kg. It gathers data from a sonar, and LiDAR or thermal camera, suitable for nearshore applications and navigating up to Douglas sea state 4;
- *C-Worker 7* (Figure 2.4f): L3Harris' C-Worker 7[4] is a robust offshore workclass USV designed for inspection applications. It weighs 4900 kg, has a footprint of 7.5 m x 2.3 m, and can navigate up to sea state 5. It acquires data from sonar, visual, and thermal cameras, allowing autonomous or remote navigation with collision avoidance procedures.

(a) SCOUT USV [22]. (b) ASAROME sailboat [63]. (c) Jung *et al.* USV [15].

(d) Z-Boat 1800RP[2]. (e) Hydrocat 180[3]. (f) C-Worker 7[4].

Figure 2.4: Available multi-domain USVs solutions.

Despite the potential of USVs for multi-domain applications, single-domain use cases remain widespread in both academic and commercial sectors including:

- *Kongsberg Geoswath Plus* (Figure 2.5a): Kongsberg Geoswath Plus USV[5] is a fully electric monohull USV with 1.7 m x 0.8 m and 55 kg equipped with GPS, IMU and sonar to perform swath bathymetry under calm conditions (Douglas sea state 2);

[3]Seafloor Systems, 2023. *HydroCat-180*. URL: `https://tinyurl.com/seafloorsystems-hydrocat180`

[4]L3Harris, 2023. *C-Worker 7*. URL: `https://tinyurl.com/c-worker7-datasheet`

[5]Kongsberg, 2023. *Geoswath Plus USV*. URL: `https://tinyurl.com/57yws7zx`

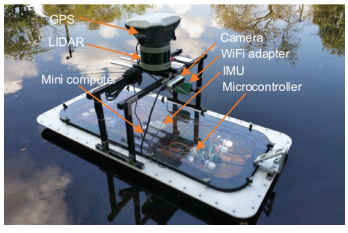
- *DriX* (Figure 2.5b): Eudaline (2019) [64] presented DriX a commercial USV from Exail for bathymetric surveys under harsh conditions (Douglas sea state 5), being equipped with GPS, IMU and MBES. It is a diesel vehicle with 7.7 m x 0.8 m and 1380 kg;
- *ROAZ* (Figure 2.5c): Ferreira *et al.* (2009) [65] uses ROAZ to perform underwater surveys. This fully electric catamaran has a footprint of 4.5 m x 2.2 m and weights 400 kg, being able to operate under a Douglas sea scale of 3;
- *Aragon USV* (Figure 2.5d): Aragon USV introduced by Han *et al* (2020) [66] for collision avoidance is a diesel-driven monohull with a footprint of 8.0 m x 2.4 m and 3000 kg. It is equipped with GPS, IMU, LiDAR and a visual camera and is able to operate under harsh conditions, Douglas sea state 5;
- *Roboat* (Figure 2.5e): Wang *et al.* (2019) [67] Roboat is designed for operating in dynamic GPS-attenuated urban waterways, and uses LiDAR, camera, and IMU for localization. It is a small vehicle designed to operate at sea state 2 with a 0.9 m x 0.5 m footprint and 15 kg;
- *Roboat III* (Figure 2.5f): Wang *et al.* (2023) [68] Roboat III is similar to Roboat regarding the sensor payload and application, however uses a larger hull. This vehicle is still designed to operate under a sea state of 2 with a 4.0 m x 2.0 m footprint and 1050 kg.

(a) Kongsberg Geoswath Plus[5].

(b) DriX [64].

(c) ROAZ [65].

(d) Aragon [66].



(e) Roboat [67].

(f) Roboat III [68].

Figure 2.5: Available single-domain USVs solutions.

Furthermore, layered robot control architectures are crucial for achieving adaptable systems capable of handling diverse operations and scenarios. These architectures offer several advantages including: (i) researchers can progressively add complexity to existing systems by introducing new functionalities or enhancing flexibility; (ii) layered architectures simplify robot programming, making them more accessible to scale-up by reducing the programming burden.

Pioneering work by Brooks (1986) [69] established the foundation for layered control systems in mobile robots. His architecture allowed higher levels to override lower levels by injecting inputs signals and inhibiting their outputs, enabling greater control. Hasegawa *et al.* (1992) [70] further emphasized the concept of skills as building blocks for robot tasks, separating low-level control from high-level functionalities.

Pedersen *et al.* (2016) [71] made a significant contribution by proposing a hierarchical skill-based architecture specifically for industrial robots. Their primary goal was to simplify robot programming for non-experts. The architecture divided robot operations into three distinct layers: tasks (complex actions), skills (building blocks of tasks), and device primitives (low-level controls), figure 2.6 [71]. Each layer of the architecture communicated with its lower layer, in a hierarchical fashion. A task was, therefore, composed of a sequence of pre-programmed skills, and defined the complex operations of the autonomous agents. While effective, this approach had limitations in execution speed compared to specialized robots. However, it has become a reference point for researchers, particularly in human-robot collaboration [72, 73].

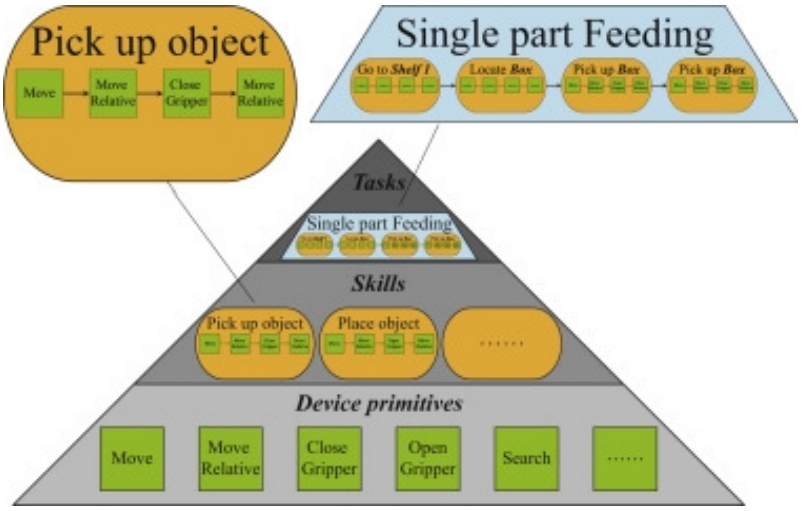


Figure 2.6: Three layers of Pedersen *et al.* [71] architecture: primitives, skills and tasks.

Building upon this concept, Huang *et al.* (2018) [74] proposed a skill-based programming system for robotic furniture assembly. They categorized skills into primitive and compound skills (sequences of primitive skills). This approach aimed to improve software reusability by allowing developers to combine pre-programmed skills for more complex assembly tasks.

Research on skill-based architectures for mobile robots remains limited. Leite *et al.* (2018) [75] implemented a fault-detecting skill within the existing skill model for industrial robots, incorporating a safety monitoring module. This skill was later tested with an USV for a fault-detecting GoTo function. Albore *et al.* [76, 77] developed a skill-based software architecture with resilient behaviors for UAVs, capable of handling sensor failures and communication losses. The architecture comprised three layers: decision, skill, and functional, figure 2.7 [76]. The decision layer, implemented as a Behavior Tree, determined which skills to execute. Skills were chosen from motion and perception skill sets and interacted with the functional layer for data exchange.

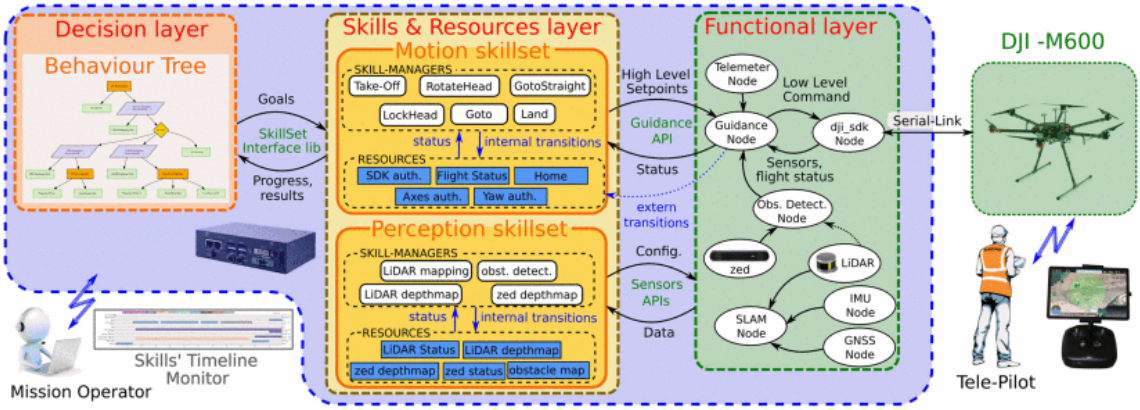


Figure 2.7: Albore *et al.* [76] three layers architecture: decision, skills and functional.

2.3 Scene understanding

Humans possess a natural ability to understand 3D environments, using prior knowledge to estimate geometry and semantics, even in large occluded areas. This task is more challenging for automated solutions using computers, which has drawn significant interest from researchers in recent years [78]. Scene understanding focus on analysing objects in context with the 3D structure of the scene, including its layout, and the spatial, functional, and semantic relationships between objects. This is a fundamental feature for various applications, from robotic navigation to augmented reality, where geometrical and semantic representations are key to interact with the real world [79].

Therefore, scene understanding is crucial for creating the USV's contextual awareness, providing the capabilities required for close-range inspection of the splashing zone and scour monitoring. Specifically, it enables the reconstruction of the environment [80], the completion of occluded information [81] and the detection of objects within the scenario [82]. Consequently, scene understanding enhances contextual awareness for:

- Visual inspection: 3D representations enable detailed visual inspections, such as examining offshore wind turbine transition pieces or scour protection integrity [83, 84];
- Improved safety: scenario reconstruction can provide data for obstacle avoidance algorithms, increasing operational safety [85, 86];
- Enhanced autonomy: a well-reconstructed environment facilitates autonomous deployment and recovery procedures, such as USV docking [87, 88].

For the inspection of the splash zone and scour protection monitoring, creating 3D reconstructions can significantly enhance the NDT capabilities for O&M experts. Benefiting from multi-domain data acquisition, mapping methodologies can depict both maritime domains simultaneously within a single representation. To achieve this, the inspection would benefit from using metric structures, such as point clouds, voxel grids, or meshes, as they capture the exact spatial layout of the environment, including the shapes and sizes of objects and their relative positions [89].

This detailed spatial information is particularly useful for inspecting complex structures or environments, which include infrastructure (e.g., foundations and docks) and surrounding geophysical elements (e.g., seabed).

However, multi-domain mapping often captures incomplete representations of the scene, providing only partial observations of the world due to factors such as limited field of view (FoV), sparse sensing, measurement noise, and the physical transition between maritime domains [90]. Therefore, the use of semantic instance completion can promote the rendering of a more detailed scenario representation. This approach can improve multi-domain mapping, creating a more comprehensive depiction for safe navigation by including otherwise occluded structures.

Furthermore, by using this complete information, the scene representation can be built to create local knowledge with details about static and dynamic objects in the scenario, which can then be used as input for collision avoidance algorithms. Leveraging information about the elements' location, shape, and velocity provides a more comprehensive understanding, thereby improving the overall observability of the USV and enhancing decision-making. This access to the semantic object information can significantly boost the capability to operate in large-scale dynamic scenarios, such as OWFs, through the use of semantic topological maps rather then the traditional metric representations [91].

These three scene understanding stages can then be used to: render a 3D environment model encompassing both the emersed (above water) and immersed (underwater) domains; enable safer navigation through obstacle detection; and enhance the overall observability of the USV for improved decision-making.

2.3.1 Multi-domain mapping

Unlike aerial or ground robots, which typically operate within a single domain, marine robots must navigate and interact with both the water and air environments, where each domain has its own set of sensor requirements and constraints for mapping technologies. Traditionally, marine robotics solutions have focused on addressing challenges within a single domain, tailor-made for specific needs [17]. Within USV systems single domain solutions are predominant, with the main focus being on the immersed domain, particularly for bathymetric surveys. Currently, the mapping of the emersed domain primarily focuses on land or aerial applications, with limited exploration of these algorithms on USVs. The maritime environment presents additional challenges compared to land-based contexts, such as wind, waves, currents, and sea fog, making the evaluation at sea more complicated [92].

Han *et al.* (2019) [93] proposes a SLAM approach for the navigation and mapping of USVs in coastal waters under GPS-restricted situations. In this method, coastline features from noise-corrupted radar images are extracted and parametrized using B-splines for memory-efficient map representation. The control points of each spline curve are incorporated into an EKF-based SLAM filter to estimate both the map parameters and the vehicle's motion. The proposed navigation

system demonstrated satisfactory performance, closely aligning with ground-truth GPS measurements. Additionally, the reconstructed coastlines accurately corresponded with the actual coastlines of the surveyed area.

Wang *et al.* (2022) [94] proposes a SLAM system for the Roboat USV that integrates data from a 3D LiDAR, an IMU, and a GPS, utilizing a factor graph approach to address the multi-sensor fusion challenge. This algorithm is designed for urban waterways, specifically the Amsterdam canals. The factor graph employs three types of factors - lidar odometry, GPS, and loop closure factors - to optimize localization and construct an accurate map of the environment. The proposed SLAM system achieves a relative translation error of 0.17 m when the three factors are enabled. However, with only the LiDAR odometry, the system fails to produce relevant results and is affected by a significant drift in a featureless environments, and when considering GPS factor the altitude suffers vertical drift up to 100 m.

Wang *et al.* (2023) [95] propose a framework for estimating the state of a ship using a scanning registration algorithm. Their work involves comparing the performance of ten commonly used registration methods within the context of maritime environments. The study concludes that for scan-to-scan registration, the ICP algorithm and its variant, the Levenberg-Marquardt ICP (LM-ICP) [96], demonstrated the best performance. However, for scan-to-map registration, the best results were achieved using the Voxelized Generalized ICP (VGICP) [97] algorithm.

Furthermore, the immersed domain inspection solutions using USVs are more focused in bathymetric surveys, emerging as a promising alternative for conventional methods [98]. Traditionally, the seabed bathymetry relies heavily on manned survey vessels, which implies significant costs. These expenses primarily stem from the high investment required for equipment, personnel, mobilization and demobilization costs [18].

Jung *et al.* (2022) [99] present a method for creating bathymetry maps of marine environments using a USV equipped with MBES, GPS, IMU, and DVL. The system estimates the dead reckoning using the IMU and DVL to obtain the USV's trajectory relative to an initial reference point given by the GPS. To create the seabed reconstruction, the authors generate submaps by assembling point clouds obtained from the MBES data according to the dead reckoning trajectory during a fixed time window. Subsequently, a regularized Iterative Closest Point (ICP) algorithm is employed to find the best match between these submaps. This matching process serves as a loop closure constraint, allowing the system to correct the USV's trajectory errors. Finally, a pose-graph optimization approach is used to refine the USV's trajectory. This optimization considers both loop closure edges, obtained from the regularized ICP matching, and dead reckoning edges, resulting in an improved and corrected trajectory estimate for the USV. This bathymetric SLAM obtained linearly growing errors instead the exponential growth from the dead reckoning.

Jung *et al.* (2022) [100] propose a method that uses multi-modal underwater sensors, specifically a MBES and a side-scan sonar (SSS), to determine the position and shape of power lines lying on the seafloor. The USV's pose is obtained from GPS data, and its orientation is derived from a survey-grade AHRS (Absolute Heading Reference System). The acquired pose information is applied to generate the submaps by aggregating the MBES scans over time and filtering

outlier data using vertical statistical outlier removal. A similar process is applied to the data obtained from the SSS. The submaps from each sensor are independent with the reference frame aligned along a common baseline. The SSS maps depicts detailed information (e.g. anchor bags) through the acoustic shadows created, which cannot be seen in the MBES generated maps. The field tests were conducted in a ocean environment and validated the performance of the developed USV inspection system, being able to observe entire segment of the power cable lines.

Merging both emersed and immersed maps to recreate the full scenario is still a research line with small dissemination. Leedekerken *et al.* (2014) [23] used a USV equipped with a set of 2D laser scanners and an imaging sonar, combined with a high-accuracy localization unit that fuses GPS, IMU, and DVL data. An ICP algorithm was subsequently employed to find the best match between the emersed map and the current data. The 3D representation of the data from both domains was then combined, considering this alignment, to create a multi-domain map. The proposed method show promising 3D reconstructions from both parts of the marine structure, despite suffering from symmetry in the longitudinal direction that increase the reconstruction error.

Papadopoulos *et al.* (2014) [22] validated the use of LiDAR and MBES for multi-domain mapping of civil structures in river and coastal environments. They used LiDAR to estimate the USV's odometry through ICP and merged the LiDAR and MBES data to a single 3D mesh representation. For this integration, they simplified the emersed data by considering only points below a certain distance threshold and reducing them to voxels. Instead of providing bathymetric data, the MBES was oriented forward to scan the below-water part of the marine structure of interest. The proposed method show consistent 3D reconstructions from both parts of the marine structure.

Han *et al.* (2019) [101] introduce a feature extraction scheme for the 3D reconstruction of a marine floating platform. The method employs an Extended Kalman Filter (EKF) for localization, integrating data from GPS, Inertial Navigation System (INS), and planar features extracted from LiDAR scans. These planar features are derived from the column structures of the platform through segmentation and classification methodologies. Motion sensor measurements, such as those from a Doppler Velocity Log (DVL), are then used to refine the trajectory estimation. In the reconstruction process, the emersed and immersed structures of the platform are reconstructed. This is achieved by combining measurements from LiDAR and MBES data, with the optimized trajectory of the vehicle. The reconstruction has maximum error of 1.3% for the column structures above the waterline and 2.3% for the pontoon structures below the waterline. Two major error source were identified, namely error associated with misalignment and calibration factors, and the error from the motion of the mapping vehicle.

Jung *et al.* (2023) [15] propose a method that employs multi-domain sensors, specifically LiDAR and MBES, to create a 3D multi-domain map of offshore structures. The method estimates the USV's pose using GPS data for vehicle position and a survey-grade AHRS for heave and orientation. The pose information is used to generate submaps by aggregating the MBES point cloud in the world frame over time. Outlier data is filtered using vertical statistical outlier removal. These submaps are then appended to the final bathymetric map. A similar process is applied to the

data obtained from the LiDAR. Each sensor's maps are created independently, but the reference frames are aligned along a common baseline. The LiDAR maps provide information about the emersed domain, while the MBES maps depict information about the immersed domain. This approach allows for detailed reconstruction and monitoring of both above-water and underwater structures. The proposed filtering method effectively removed outliers in real-time, significantly reducing the need for manual data handling during post-processing. Field tests conducted in a real ocean environment validated the feasibility of the inspection system, demonstrating its capabilities in waypoint tracking and multi-modal mapping.

2.3.2 Semantic instance completion

Perception sensors often provide partial and incomplete observations due to their limited field of view, sparse sensing capabilities, and measurement noise. This results in an incomplete representation of the scene. To mitigate this issue, scene completion techniques are employed to infer the complete geometry of a scene from one or more 2D/3D observations. Specifically, semantic instance completion focuses on detecting individual instances within a scene and inferring their complete object geometry. However, it is limited to object instances and does not extend to completing the scene background, such as walls or floors [90]. Most existing methods rely on instance detection followed by completion using CAD models [102, 103] or a completion head [104, 105].

Gupta *et al.* (2015) [106] represent objects in an RGB-D scene using corresponding 3D models from a library. They achieve this by initially detecting and segmenting object instances in the scene and subsequently using a convolutional neural network (CNN) to predict the pose of the object. The top k inferred pose hypotheses are then employed to initialize a search over a small set of 3D models, including their scales and exact placements. For this task, a modified iterative closest point (ICP) algorithm is used, demonstrating reasonable results even when operating at the level of object categories rather than exact instances. This approach solely relies on 2D annotations on the image for training all models, and during testing, it generates a comprehensive 3D representation of the scene. The final output comprises a set of 3D models aligned with the objects present in the image.

Li *et al.* (2017) [107] adopt a distinctive approach to non-rigid alignment. Their methodology begins by aligning both the template model and the incomplete model in an upright position, aiding the initial alignment process. Subsequently, they perform rigid alignment by translating and scaling the template to fit the bounding box of the scan. It is followed by an exhaustive search for the orientation that minimizes the distance between corresponding points. To refine the match further, they employ embedded deformation on the template model and define a deformation energy function comprising matching error, shape smoothness, and rigidity cost. This process is iterated for each selected model, and the one with the best score is retained. However, the outcome of this procedure may be coarse due to inherent disparities between the input and the deformed model. Therefore, they introduce an optimization step where they patch the model to produce a smoother final model.

Liu *et al.* (2019) [108] employ key point extraction from 3D models for identification and pose estimation purposes. Their approach uses an unsupervised learning algorithm wherein they propagate the 3D model through a 1D Encoder CNN network to obtain a feature vector, followed by a decoder with reversed dimensions to reconstruct the model. By reconstructing the model from the feature vector, they compare it with the original to compute a loss function for backpropagation. Leveraging the feature information, they search a database of encoded models to obtain accurate pose estimation in six Degrees of Freedom, subsequently patching the incomplete model.

Hou *et al.* (2020) [105] propose RevealNet, an end-to-end, fully differentiable network designed for semantic instance completion from RGB-D scans. RevealNet integrates instance detection and object completion into an unified framework. The network detects and regresses object bounding boxes, classifies objects, and predicts complete object geometry using a shared backbone. This approach allows information to flow from detection to completion, using both color image data and 3D geometry. RevealNet employs an encoder-decoder backbone to extract bounding boxes, which are then fed into an instance completion head to achieve object reconstruction. The results showed that RevealNet outperformed state-of-the-art approaches for semantic instance completion on both real and synthetic scan data.

Rosinol *et al.* (2021) [109] introduce Kimera, a method for generating a 3D dynamic scene graph (DSG) from visual-inertial data, integrating both metric and semantic elements of a dynamic environment. To enhance the depiction of static objects, Kimera uses semantic instance completion. Objects are initially identified through 2D semantic segmentation of camera images, and this information is applied to label the 3D mesh using Bayesian updates. These labelled meshes are then compared against a CAD database. If the segmented object matches one in the database, the CAD model is aligned using TEASER++[110] to find the best fit between the point clouds, even with significant outliers present. The result of these steps is an accurate 3D pose of the object.

Nie *et al.* (2021) [111] introduce RfD-Net, a reconstruction-from-detection framework designed for end-to-end semantic instance completion directly from raw point clouds. This method integrates a shape reconstruction head with a 3D detector backbone, leveraging the sparsity of point clouds to focus on predicting shapes detected with high objectness while ignoring free scene space. The RfD-Net first learns a denoiser using PointNet [112] layers to remove background points. The cleaned points are then extended by box features through a skip connection, before being encoded into new proposal features to decode spatial occupancy values. Object meshes are obtained by applying the Marching Cubes [113] algorithm to the predicted occupancy grids. This design enables implicit function learning to reconstruct surfaces at a much higher resolution.

Wan *et al.* (2023) [114] introduced PMNet, a completion-then-reconstruction network designed for semantic instance completion from real-world point clouds. PMNet comprises a 3D detection backbone, a point cloud completion module, and a mesh generation module. The 3D detection backbone learns features with rich semantic information and confidently predicts instances. These instances' local point clouds are encoded to capture local geometric features. A feature transformation module aligns the proposal features with the local geometric features. Once aligned, these features are fused and sent to the completion module. The final completed

point cloud is processed through a mesh generator. PMNet achieves the best reconstruction performance on the real-world indoor point cloud dataset ScanNetv2.

Instance-level completion can preserve the structures and details of objects. However, detecting objects from partially observed scenes is inherently challenging due to complex perception scenarios, small or occluded objects, and negative-positive imbalance sampling [115]. Currently, multiple end-to-end methodologies have been proposed for 3D detection, building on the success of deep learning techniques applied to point clouds [112, 116]. These methodologies propose diverse architectures to detect 3D objects directly from different 3D representations, such as point clouds [117, 118, 119], grid-based formats (e.g., voxels and pillars) [120, 121, 122], and range images [123, 124, 125]. For point clouds, the typical approach involves first passing the point cloud data through a point-based backbone network, where points are gradually sampled and features are learned using point cloud operators. Subsequently, 3D bounding boxes are predicted based on the downsampled points and their features. In grid-based 3D object detection, conventional 2D convolutional neural networks or 3D sparse neural networks are employed to extract features from the grids, culminating in a detection head. Range images, on the other hand, provide a dense and compact 2D representation where each pixel encodes 3D distance information instead of RGB values. The detection problem in range images can be tackled from two perspectives: designing new models and operators specifically tailored for range images, or adapting existing models from 2D object detection to handle range images effectively [126]. Nevertheless, the lack of annotated data for detection in real scenarios, the negative-positive imbalance inherent to the environment, and the limited computational resources present significant challenges to end-to-end 3D object detectors. To address these challenges and create lighter detection methods capable of performing under limited resources, a two-part approach can be used, where first segments the scenario using clustering algorithms to find proposals for the available objects, and then use 3D object classification to identify the objects within the cluster [127, 128].

2.3.3 Scene representation

Scene representation outlines how an environment's map is organized, facilitating spatial understanding for USV operations. It can be classified into three primary types: 3D metric maps, topological maps, and hybrid maps that combine semantics with topological or metric information [129].

Metric representations provide a detailed, quantitative depiction of the environment, capturing its exact geometric properties and making them ideal for precise operations. However, they suffer from computational overhead and lack semantic information, limiting their application in high-level task planning. These representations can be categorized based on input sensor data and mapping dimensions, including grid-based or point-based methods [130]. Grid-based representations use dense grids [131, 132, 133], hierarchical octrees [134, 135, 136], and voxel hashing [137, 138, 139] to create memory-efficient solutions. These grids are constructed using neighborhood lookups and context aggregations, offering fast and straightforward methods. However,

their main limitation is the need to specify grid resolution beforehand, which cannot be easily adjusted during reconstruction, even with octrees. This can lead to suboptimal resolutions, wasting memory in low-complexity areas while failing to resolve fine details beyond the chosen resolution. Point-based representations, on the other hand, provide high-resolution models of 3D geometry, making them suitable for visualization and rendering. They have been successfully applied to 3D reconstruction [140, 141, 142], allowing point density to vary across the scene without prior specification. Although visually appealing, these models are computationally expensive due to the large amount of data they require and typically offer low-level geometric descriptions, neglecting aspects such as obstacle topology.

Topological maps represent the environment as a graph, achieved by discretizing the continuous world into a finite set of places (nodes) connected based on their relative positions (edges). These maps prioritize the connectivity and relationships between different places over their precise geometric layout. They offer a compact representation as only distinctive locations within the environment are encoded, resulting in lower computational demands. Navigation commands naturally follow from the graph structure. However, a key challenge of this method is perceptual aliasing, where distinct locations may appear identical to the robot's sensors. Various techniques, including SLAM algorithms, are commonly used to build topological maps, allowing the robot to construct the map while exploring and interacting with the environment [143, 144, 145].

Hybrid maps are becoming increasingly popular, since it incorporate semantic information into metric representations. These maps leverage the ability to simultaneously estimate the 3D geometry of a scene and assign semantic labels to objects and structures. This approach enhances robot autonomy and robustness, facilitates more complex tasks (e.g., avoid getting close to foundations while navigating), shifts the focus from path-planning to task-planning, and enables advanced human-robot interaction [146]. However, these maps remain computationally intensive. To address these challenges, 3D Dynamic Scene Graphs (DSGs) are being explored as a unified representation for actionable spatial perception [147, 109, 148, 149]. DSGs are layered directed graphs where nodes represent spatial concepts (such as objects, rooms, and agents) and edges represent pairwise spatio-temporal relations. The graph is layered, meaning that nodes are grouped into layers corresponding to different levels of scene abstraction, creating a hierarchical representations [150].

2.4 Safe navigation using collision avoidance for USVs

As noted by Khalid *et al.* [61], the inspection of the splashing zone and the monitoring of scour protection using multi-domain USVs is a promising concept. However, it remains challenging in terms of large-scale command and navigation. Consequently, there is an increased need to enhance the USVs autonomy to operate under these harsh scenarios. Among these improvements, safe navigation technology, particularly collision avoidance, is important for achieving true unmanned operation, enabling USVs to navigate independently, efficiently, and safely in various scenarios.

Enhanced autonomy not only expands the utility and safety of USVs across different applications but also underscores their role in diverse operational settings [151].

Numerous classical algorithms exist for collision avoidance in USVs, broadly categorized into static path planning and dynamic collision avoidance strategies [152]. Static path planning involves generating a guidance path for the USV based on a map of the known environment and target information. This method considers the overall route and long-term navigation to ensure that the USV avoids collisions with obstacles or other vessels while reaching its destination. Classical algorithms such as Dijkstra [153] and A* [154] have been extensively explored for this purpose. However, given the dynamic nature of offshore wind farm inspections, dynamic collision avoidance offers more effective approaches to safely operating near other vehicles and infrastructures by providing a more reactive behavior. This group focuses on determining short-term avoidance maneuvers in response to immediate obstacles or dynamic changes in the environment.

Fox *et al.* (1997) [155] presents a collision avoidance method for mobile robots with synchro-drives. It simplifies robot trajectories into sequences of circular arcs and uses a dynamic window approach to limit the search space to safe, admissible velocities. By considering the robot's ability to stop and its acceleration constraints, the method ensures safety while optimizing the robot's speed and trajectory to avoid obstacles and stay on course toward its goal.

Huang *et al.* (2019) [156] describes a modified generalized velocity obstacle (GVO) algorithm for ship collision prevention that takes into account the dynamics of ships. Building on Berg *et al.* (2011) [157], which uses the velocity obstacles representation to derive sufficient conditions for collision-free motion by reducing the problem to solving a low-dimensional linear program, this modified algorithm incorporates the specific dynamics of ships to avoid holonomic constraints. Additionally, it integrates a collision avoidance system to ensure compliance with COLREGs (Convention on the International Regulations for Preventing Collisions at Sea) regulations. Simulation experiments have demonstrated its effectiveness in various maritime environments.

Shaobo *et al.* (2020) [158] propose a collision avoidance decision-making system for Maritime Autonomous Surface Ships (MASS) using a modified velocity obstacle (VO) algorithm. This system combines the VO with a Finite State Machine to respond to the dynamic behavior of other ships during the avoidance process. The system architecture is divided into front-end and back-end components, defining the decision process to consider various limiting factors in the real marine environment, including ship maneuverability, multi-ship encounter situations, COLREGS, course maintenance, and good seamanship, with a particular focus on the actions of other ships. The system's effectiveness is verified through simulation experiments under 10 different scenarios.

Zhuang *et al.* (2021) [159] present a collision avoidance approach for USVs in scenarios with multiple moving obstacles based on the velocity obstacle method. This approach considers the relationship between the speed of the USV and the collision cone of each obstacle to determine a safe heading that maintains the vehicle's current velocity. Simulation experiments demonstrate that the algorithm can avoid hazardous situations quickly and accurately even in scenarios with multiple ships present.

Cho *et al.* (2022) [160] introduce a collision avoidance system that defines and classifies

encounter types to ensure pairwise-role-symmetry: give-way and stand-on. This system assumes that all encountered ships use the proposed algorithm, which adheres to COLREGs. The procedure comprises two layers: the first layer provides waypoint tracking behavior and evasive maneuver behavior to define the ship's responsibility and uses a probabilistic velocity obstacle algorithm to generate a set of collision-free trajectory options, taking into account the velocity uncertainty for both the USV and obstacles. The second layer employs a decision-making method, designed using a cost function and the results of the role classification, to resolve role conflicts in accordance with existing maritime traffic rules. The results showed better performance compared to methods that do not consider uncertainty.

Zhou *et al.* (2022) [161] defines a modified deep deterministic policy gradient (MDDPG) strategy based on a reinforcement learning method. This algorithm assumes that USVs operate in an environment with incomplete perception, where they are only aware of obstacles within a certain detection distance and may have limited visibility due to occlusions. Additionally, the USVs can select actions arbitrarily within an allowable range, and the obstacle avoidance algorithm should be universal. The MDDPG method is developed as the USV's decision-making controller under these incomplete perception conditions, enabling continuous control output for the USV even in the presence of dynamic obstacles.

Han *et al.* (2022) [162] describes a distributed multi-robot navigation approach that combines velocity obstacles with a deep reinforcement learning framework to handle reciprocal collision avoidance tasks within a shared workspace with limited surrounding information. This method uses VO vectors to represent both dynamic agents and static obstacles and employs bidirectional gated recurrent units (GRUs) based neural networks for feature extraction and action computation within continuous space. Additionally, the reward function is designed based on the reciprocal velocity obstacle (RVO) area [163]. The proposed approach is evaluated through trials conducted in both simulation and real-world scenarios involving ground robots, demonstrating a superior collision avoidance capability and time efficiency compared to other approaches in crowded environments.

Xue *et al.* (2023) [164] implement a proximal policy optimization (PPO) with RVO for distributed control of multi-USVs in compliance with COLREGs. They use RVO vectors to represent the information of surrounding USVs and obstacles, enabling global environmental awareness. Each USV in the system can register its information (position, velocity, etc.) and autonomously avoid collisions without requiring communication with other USVs. A GRU-based neural network is employed to extract changes in the number of USVs at risk of collision, serving as input for the PPO algorithm. The reward function of the PPO is enhanced based on the RVO area and expected collision time, enabling collision avoidance behavior and balancing collision risk against navigation time. Experimental simulation results demonstrate the efficiency of the proposed algorithm in navigating multi-USVs to their target points while avoiding collisions with other USVs within the system and with external static and dynamic obstacles.

2.5 Critical review

The ability of USVs to perform safe autonomous inspections is significantly influenced by advancements in multi-domain data acquisition, scene understanding, and collision avoidance capabilities. The growing trend of offshore wind farms being placed farther from shore presents logistical challenges, requiring the development of compact USVs capable of operating in up to Douglas Sea State 4 conditions. Specifically, these USVs should handle significant wave heights of up to 1.5 m and wind speeds up to $10\,\mathrm{m\,s^{-1}}$. Equipping them with a multi-domain sensor payload - including GPS, IMU, LiDAR, sonar, visual, and thermal cameras - can reduce operational costs and minimize the required working crew. Enhancing scene understanding capabilities will further boost the vehicle's autonomy, enabling it to perform multi-domain inspection tasks and create comprehensive representations of the scenario for safe navigation. Additionally, the integration of collision avoidance technologies will allow USVs to operate BVLOS in close proximity to critical infrastructures, such as offshore wind turbines. While this technology offers significant advantages for inspecting the splashing zone and monitoring scour protection, several challenges remain to be addressed to fully realize its potential.

Ideally, the USV should be compact for efficient transportation and capable of seamless integration with existing CTVs to eliminate energy consumption during transit. It should also be equipped with a multi-domain sensor payload. However, as shown in Table 2.2, only a few of the reviewed USVs meet the sea state requirement, and most of these are large and do not include the full sensor payload for NDT. Therefore, there is the need for a compact, robust USV with comprehensive sensor capabilities to enhance the efficiency and safety of offshore inspections. Additionally, the software architecture of USVs should be modular and flexible, allowing for easy adaptation to multiple missions with minimal programming effort. Establishing a development standard for inspection missions, particularly with maritime robots, is essential. While skill-based architectures show promise for this purpose, they can have limited execution speed compared to specialized robots. Thus, it is crucial to standardize the design while maintaining the performance of tailor-made robotic tasks.

To enhance USVs scene understanding capabilities, three key aspects need to be addressed: multi-domain mapping, semantic instance completion, and scene representation. Regarding the multi-domain mapping, most existing USV solutions focus on single-domain applications, such as bathymetric surveys or surface reconstruction. While recent advancements have seen the growth of multi-domain mapping algorithms, these typically rely on expensive sensors like DVL and survey-grade AHRS to estimate USV location and transform LiDAR and MBES data over time. This reliance poses challenges in GPS-denied scenarios, where any outlier measurements can introduce significant drift in the representation. Additionally, the limited field of view (FoV) of sensors often results in incomplete representations, particularly at the waterline boundary separating different domains. The use of semantic instance completion can fill in occluded or partially observed objects, contributing to a more complete scene representation and enhance safety. Deep learning approaches offer versatility, capable of inferring high-level structural details without di-

Table 2.2: Comparison of available USVs, where: C - Catamaran; M - Monohull; E - Electric; and D - Diesel.

USV	Hull Type	Power Source	Dimensions (m)	Douglas Sea State	GPS	IMU	LiDAR	Sonar	Visual Camera	Thermal Camera
ROAZ [65]	C	E	4.5 x 2.2	3	✓	✓	✗	✓	✗	✓
HydroCat-180[3]	C	E	5.5 x 2.5	4	✓	✗	✓	✓	✓	✓
Jung *et al.* [15]	C	E	4.1 x 2.5	5	✓	✓	✓	✓	✓	✗
SCOUT [22]	M	E	3.0 x 0.5	2	✓	✓	✓	✓	✗	✗
ASAROME [63]	M	E	3.6 (length)	2	✓	✓	✗	✓	✓	✗
Roboat [67]	M	E	0.9 x 0.5	2	✓	✓	✓	✗	✓	✗
Roboat III [68]	M	E	4.0 x 2.0	2	✓	✓	✓	✗	✓	✗
DriX [64]	M	D	7.7 x 0.8	5	✓	✓	✗	✓	✗	✗
Aragon [66]	M	D	8.0 x 2.4	5	✓	✓	✓	✗	✓	✗
Kongsberg Geoswath Plus[5]	M	E	1.7 x 0.8	2	✓	✓	✗	✓	✗	✗
Teledyne Z-Boat 1800RP[2]	M	E	1.8 x 1.0	2	✓	✓	✓	✓	✓	✗
C-Worker 7[4]	M	D	7.5 x 2.3	5	✓	✓	✗	✓	✓	✓
SENSE (ours) [27]	C	E	1.5 x 1.1	2	✓	✓	✓	✓	✓	✗
Nautilus (ours) [28]	M	E	3.0 x 1.4	4	✓	✓	✓	✓	✓	✓

rectly copying from prebuilt CAD models, making them suitable for objects with many variants or highly deformable features [165]. These methods avoid the need for a priori model selection, bypassing the classification step on incomplete models and costly model alignment procedures. However, training neural networks for generative tasks remains challenging, and greater generalization may not be necessarily useful for completing models of known objects with limited variations. In such cases, template matching approaches can provide more robustness. Furthermore, existing methods often struggle with partial point clouds that exhibit morphological diversity and discontinuities [166]. Given the well-defined nature of OWT foundations, using computationally less complex methodologies can yield fast, accurate, and deployable results. A framework that clusters the scene, classifies proposals, and matches them with respective CAD templates can be beneficial. Moreover, towards scene representation, most current methods utilize metric representations to depict scenarios, offering detailed portrayals particularly useful for inspection tasks. However, for collision avoidance in large-scale dynamic environments, these representations may lack relevant information. Hybrid semantic-topological representations can address this gap, depicting not only the position and shape of objects (e.g., bounding boxes) but also their velocity and instance classification. This approach can enable for a more comprehensive representation of large-scale scenarios, facilitating safer navigation and enhanced decision-making.

To increase safety for close-range inspection using USVs, collision avoidance capabilities need to be explored. One of the primary challenges in USV operation within dynamic and variably structured environments is the ability to address the unmodeled changes in the scenario efficiently while integrating mission-specific information into the local decision-making process [167]. To address this, obstacle avoidance methodologies should incorporate proactive responses to dynamic obstacles. By using scenario representation information, such as the pose, shape, and velocity of objects, USVs can model dynamic changes in the environment. This can enable the selection of optimal motions to avoid obstacles early, preventing overly conservative maneuvers or potential collisions. Implementing these strategies effectively will enhance the USV's ability to navigate

complex and changing environments autonomously, thereby improving operational safety and efficiency.

Therefore, this thesis explores the design of multi-domain USVs to operate under distinct environmental conditions, specifically tailored for inspecting offshore wind farms. The proposed USV will enhance multi-domain scene understanding and data acquisition through a robust multi-domain mapping architecture that provides metric representations independent of the localization source, ensuring reliability even in GPS-denied environments. To address occluded data challenges, a semantic instance completion procedure is suggested, clustering the scenario to detect regions of interest, classifying proposed objects, and aligning them with CAD templates from a database. For safe navigation, a semantic topological map representation is created to inform the collision avoidance algorithm, incorporating detailed information about object positions, shapes, and velocities. Additionally, a collision avoidance method is presented based on defining a protective zone that represents the configuration space using object pose and velocity, allowing proactive selection of optimal motions to avoid obstacles. These integrated methodologies aim to develop a comprehensive framework for a multi-domain USV capable of detailed OWF inspections, enhancing operational safety and efficiency.

Chapter 3

Design of unmanned surface vehicles with multi-domain perception

The offshore wind energy industry would benefit from the employment of autonomous vehicles for monitoring and inspection endowed with multi-modal sensors to create cost-effective O&M strategies and ensure repeatability for data collection and reduce the risk of human losses [60]. Unmanned vehicles, namely Unmanned Aerial Vehicles (UAV), Unmanned Surface Vehicles (USV), Remotely Operated Vehicles (ROV), and Autonomous Underwater Vehicles (AUV), are being developed and deployed to perform O&M activities, such as mapping [168], underwater interventions [169] or visual inspection [170]. Current procedures uses Remotely Operated Vehicles (ROVs) and divers for observing the subsea structures, and rope access and tele-operated Unmanned Aerial Vehicles (UAVs) for the aerial parts, where the infrastructures state are usually assessed through Non-Destructive Testing (NDT), such as visual inspection, ultrasonic testing, infrared thermography, LiDAR mapping or electromagnetic testing [42].

This chapter is organized as: section 3.1 gives a brief introduction to the problem and proposed solutions; section 3.2 presents the SENSE USV and its hardware architecture; section 3.3 describes the Nautilus hardware architecture; section 3.4 defines the software architecture, namely a modular software architecture (MUSA) designed to be transversal to different intelligent systems applied to the USVs; section 3.5 illustrates the results for SENSE, Nautilus, and MUSA; section 3.6 highlights some brief conclusions.

3.1 Introduction

The inspection of the splash zone and the scour protection is crucial to ensure the safety and efficiency of the foundations. These are harsh operations that are normally human-dependent and present multiple constraints, such as a maximum wave height of 1.5 m for using a small boat on site or wind of at most $5\,\mathrm{m\,s^{-1}}$ to climb the tower [6]. The splash zone is one of the most challenging inspection area since it is the interface between the surface and underwater domains being subjected to both sea and atmosphere corrosion, abrasion and can suffer impacts above the

waterline. This inspection typically uses rope access and divers for cleaning and NDT inspection, starting to emerge solutions such as ROVs and crawlers for this task [171]. As for the scour protection affects fixed foundations and it is used to avoid scour holes caused by changes within the water current flow pattern which results in local sediment transport that compromises the stability of the foundations and may expose the cables on the seabed due to the erosion. A typical inspection uses manned vessels equipped with sensors capable of mapping the seabed (e.g. multi-beams, single-beams, sub-bottom profilers) to perform a bathymetry on the region of interest for evaluating the performance of the scour protection and to quantify the edge scour development [172].

Hence, the utilization of Unmanned Surface Vehicles can greatly facilitate the inspection of these crucial sections in offshore wind farms. These vehicles outfitted with a multi-domain sensor payload are capable of acquiring data from both surface and underwater domains which eliminates the necessity for multiple operations and distinct assets, consequently leading to a reduction in costs, time, and mission-related risks.

This chapter introduces two USV that were developed, namely the SENSE and the Nautilus. The vehicles were designed and developed to acquire multi-domain data which is crucial for the inspection of the splash zone and the scour protection. Both have a modular design and can operate autonomously under different conditions, where:

- SENSE is an USV with catamaran design capable of operating under calm and favourable maritime conditions. It features a modular architecture with an easily adaptable sensor payload, capable of acquiring multi-domain data. This provides a platform for testing and data acquisition in near-shore conditions with an easy deployment due to its small dimensions.

- Nautilus is a compact monohull USV capable of operating in moderate sea states and easily deployable, designed to operate in offshore and perform multi-domain inspection. It is capable of operating beyond visual line-of-sight (BVLOS) and in scenarios with reduced luminosity. Additionally, it allows communication with a Shore Control Centre (SCC) and broadcasts the vehicle state and data via wi-fi.

Aiming to promote resource efficiency and eliminate redundant development efforts, a skill-based software architecture is proposed, the Multi-layer Universal Software Architecture (MUSA), which provides a standard for distinct vehicles namely SENSE and Nautilus. This approach divides the architecture into several layers. The bottom layer, composed of actions or primitives, interconnects the hardware to the decision making layer, where skills sit. This way, skills output data that is independent to the hardware, creating a level of abstraction that is much desired when software re-usability is wanted. On top of the skill level, a task level usually concatenates skills in a sequentially arranged fashion, allowing for distinct complex operations to be run one after the other.

3.2 SENSE - Autonomous vessel for multi-domain inspection and maintenance

The SENSE (autonomouS vEssel for multi-domaiN inSpection and maintEnance) USV, figure 3.1, provides the versatility to adapt to the most suitable payload according to the task necessities and aims to improve the easiness of construction and replication. The USV is endowed with a sensor payload that allows the acquisition of multi-domain data, from the surface and underwater domains simultaneously. It uses a 3D LiDAR (Light Detection And Ranging), a stereoscopic camera and a multibeam echo sounder (MBES), as well as navigation sensors, namely GPS and IMU.

Figure 3.1: SENSE USV operating in *Marina de Leça*.

The SENSE design provides a stable platform with a catamaran configuration with a small footprint to be easy to transport and deploy. The hull design as well as the navigation payload allows it to operate under favourable conditions within the range of sea state 3, up to 0.6 m maximum wave height, in the Douglas sea scale [62].

Therefore, the flexibility provided by SENSE offers the capability to adapt to different applications allowing to exchange the sensor payload, the available energy or the drive system by switching independent modules rather than redesigning the all system. This enhances the potential to be used also as research platform that can be easily replicated by any other institution. Moreover, it can boost the development of cooperative USVs with heterogeneous configurations using the same base design. The ROS-based software architecture allows a simple integration of new packages for researching new methodologies, such as global path planning [173], collision avoidance [31] and multi-domain mapping [29]. In parallel with the hardware customization, the SENSE provides a powerful development and testing platform for boosting the technology readiness level for the autonomy of USVs, as well as a downscaled use case to validate situational

awareness methods for autonomous ships.

The SENSE is a small sized USV with 1.5 m length, approximately 50 kg, 2 h operation time and differential drive. The structural construction is based on the INESC TECs' Zarco USV [53] and is composed by anodized aluminium profiles linked to each other to form the rigid frame which can be attached to two polyethylene pontoons with snap buttons, creating the flotation system for the catamaran configuration. Each pontoon provides a net buoyancy of 50 kg. Thus, with the minimal configuration, energy and navigation modules, the SENSE weights approximately 50 kg providing an extra 50 kg for any additional payload. The flotation system will affect the maximum velocities and the operational limits, where bulkier pontoons with higher stability, as used currently in SENSE, will reach lower velocities while thinner pontoons will provide higher speed at cost of lower stability. The operation requirements for SENSE to operate corresponds a sea state 3 (small wavelets) in the Douglas sea state, requiring favourable conditions to operate (e.g. in the Mediterranean Sea during Summer time).

The drive system is composed by two thrusters configured for differential drive at the stern of the vessel. It uses standard trolling motors with a maximum thrust around 250 N with a maximum power consumption of 360 W each. The motors are connected to a commercial brushed dc motor drive with dual-channel to receive commands from an on-board PC, while sending power inputs, using PWM control, to the motors. This allows the SENSE to reach speeds up to 1.5 $\mathrm{m\,s^{-1}}$ using the pontoons depicted in figure 3.1.

Regarding the sensor payload, SENSE is composed by multi-domain as well as navigational sensors[1], namely:

- Sparkfun Razor IMU 9DOF, 15 Hz, 9 axis, 1.0°/2.0° angular accuracy. This is available in the navigation module and is a low-cost IMU solution presenting a lower accuracy;
- Xsens MTi-30 IMU, 200 Hz, 9 axis, 0.2°/0.5° angular accuracy. This is available in the localization module and is a high-end IMU solution with high accuracy;
- Vk-162 Glonass GPS, up to 10 Hz, GNSS, 2.0m horizontal accuracy. This is available in the navigation module and is a low-cost GPS solution with lower accuracy, more susceptible to signal loss and noisy measurements. No external antenna is available;
- Swift Navigation Piksi Multi GPS, up to 10 Hz, L1/L2 RTK, 0.01m horizontal accuracy and 0.015m vertical accuracy. This is available in the localization module and is a high-end GPS solution with higher accuracy and with a more stable signal. This sensor when paired with a base station also provides a L1/L2 RTK technology with an Euclidean error lower than 0.05 m that provides the ground-truth for localization based methods.
- Imagenex "Delta T" 837B, Multibeam Echosounder with a rate up to 20 Hz, 120 beams, able to acquire data from a minimum range of 0.5 m to as far as 100 m, maximum distance. Resolution of 0.2% of the range;

[1]The sensors specifications are available on the vendors' datasheets.

- Mynt Eye D, stereo camera with a rate up to 60 fps, and a maximum resolution of 1280x720;
- Velodyne VLP-16, LiDAR with a rate of 10 Hz, capable of measuring distance up to 100 m and field of view of 360° horizontal and 30° vertical. Accuracy of 0.03 m.

The sensor payload and setup installed in SENSE to acquire the dataset is depicted in figure 3.2, as well as the coordinate frames for their physical placement, where the red, green and blue lines represent the X, Y and Z axis, respectively. Table 3.1 presents a summary of the SENSE characteristics, where the total weight, additional payload and operation time corresponds to the minimal configuration (energy and navigation modules only).

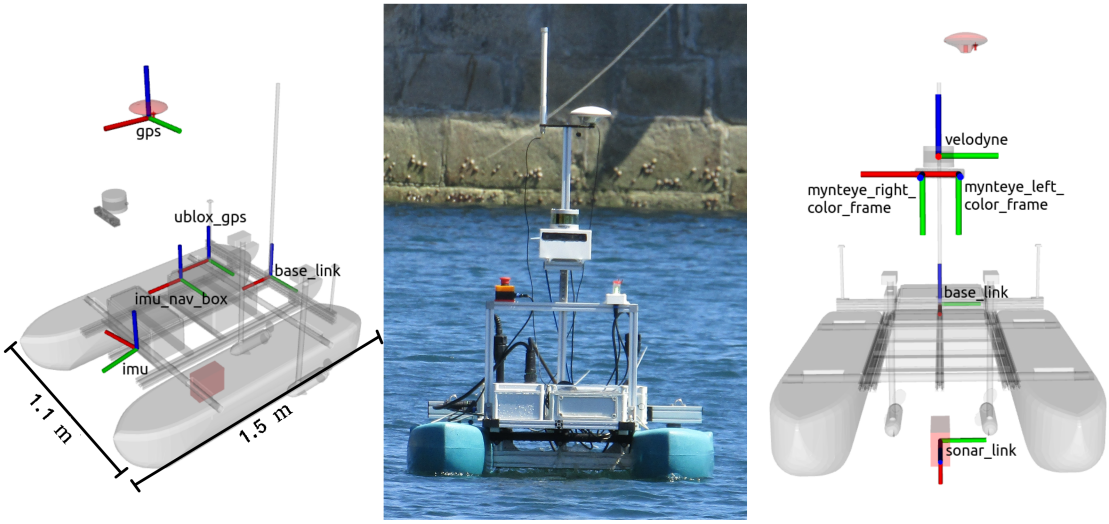


Figure 3.2: Setup of sensors used on the SENSE. The navigation sensors coordinate systems (left), the SENSE (centre) and the perception sensors coordinate systems (right). The red, green and blue lines represent the X, Y and Z axis, respectively.

The SENSE hardware architecture is subdivided in four modules, namely energy, navigation, perception and localization, as depicted in figure 3.3. Each one of them is built in a single watertight enclosure that is attached to the aluminium frame. The minimal configuration required for the USV to be fully functional is constituted by the energy and navigation modules, which provides the required power and control systems for the vehicle to operate, as well as a remote wi-fi access point. The addition of the other modules, perception and localization, enhances the vessel potential beyond the navigation ability by introducing the capacity to acquire and process perception data and improving the localization and navigation sensors accuracy. This modularity yields the advantage of allowing a straightforward modification of single modules independently, without requiring a major re-engineering effort to the vehicle itself. Moreover, its important that the weight of the modules is distributed along the rigid frame to ensure that the centre of mass is near the middle of the vehicle, thus avoiding any unevenness that can compromise the vehicle safety and stability.

The energy module is responsible for providing power to all available modules and is composed by the battery and the power distribution systems. As illustrated in figure 3.3a, it is linked

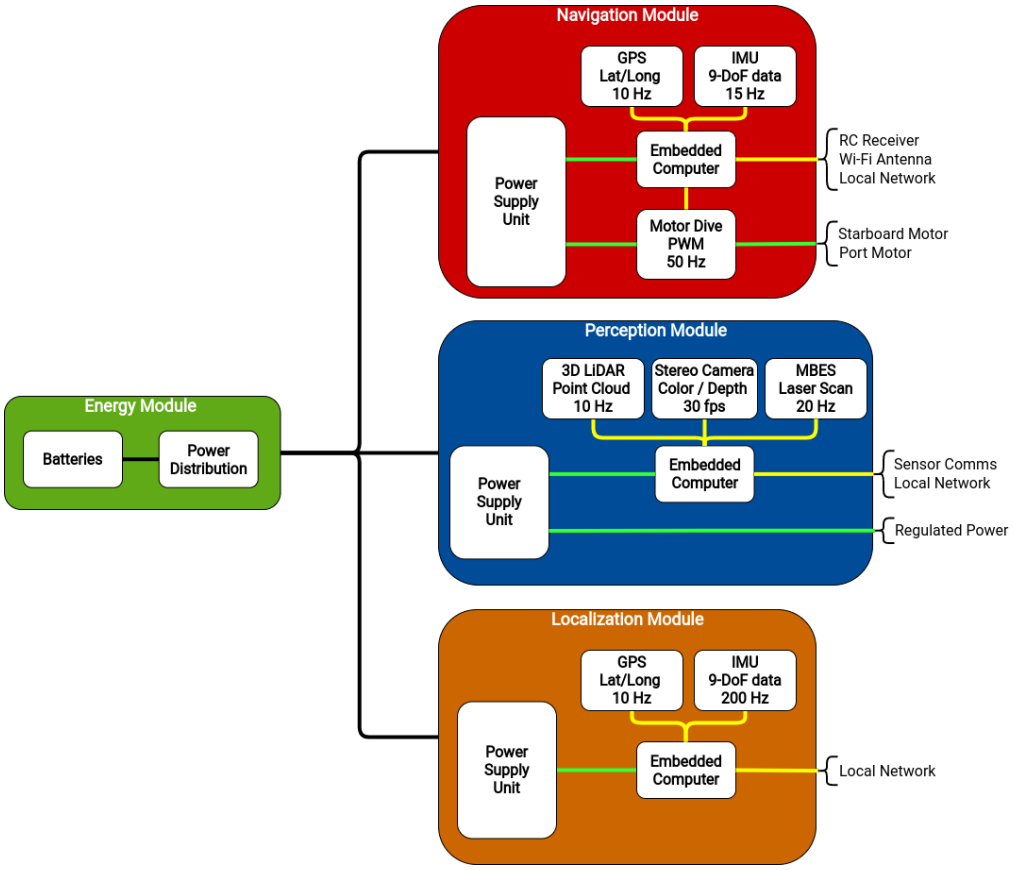
Table 3.1: Technical specifications of the SENSE.

Characteristics	**Specifications**
Dimensions (L x W x H)	1.5 m x 1.1 m x 1.9 m
Total Weight*	∼50 kg
Available Payload*	∼50 kg
Drive Mode	Differential drive Trolling motors
Max Speed	1.5 $m\,s^{-1}$
On-board sensors	Navigation: GPS, IMU Perception: 3D LiDAR, MBES, Stereo camera
Power Supply	4S rechargeable LiPo batteries
Operation Time*	∼3.5 h
Operation Time (Full config)**	∼2.5 h
Douglas Sea Scale***	Sea State: 3 (up to 0.6 m)

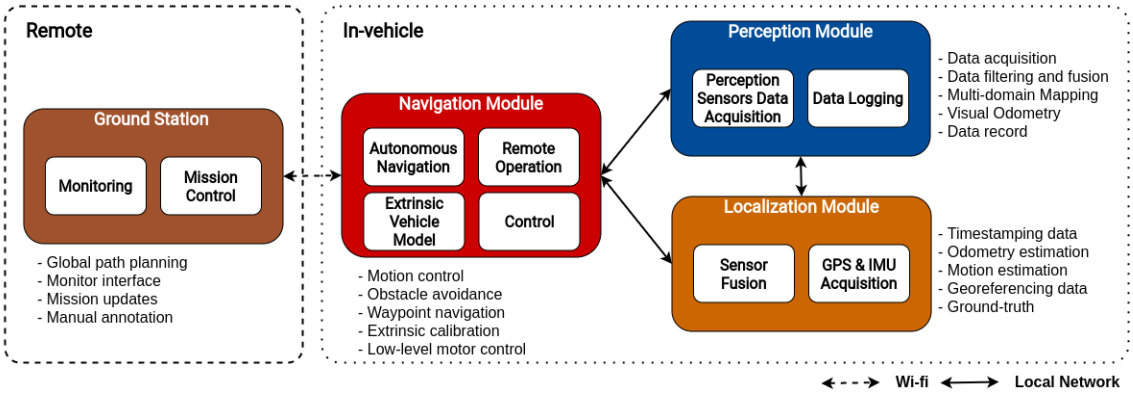
* Values acquired using the minimal configuration, energy and navigation modules only.
** Values acquired using the full configuration, energy, navigation, perception, and localization modules.
*** Dependant on the type of pontoon.

to all the modules available in the system and is composed by several lithium batteries in parallel with a total capacity of 2220 Wh. Currently, it uses 4S LiPo batteries with 10 000 mAh each. This payload offers an autonomy around 3.5 h for the minimal configuration. This operation time is dependent on the modules in use, their composition and the installed pontoons. Moreover, the energy module can easily be replaced by a similar assembly during the mission.

The navigation module provides the motor drive controller and allows the remote or autonomous operation of the SENSE, while serving as the communication relay with a remote ground station. This enclosure is composed by a power supply unit (PSU), an embedded computer, the navigation sensors (GPS and IMU) and the motor drive. The PSU regulates the power for all the components and protects the system from undervoltage and overvoltage states. It also provides an emergency switch option that cuts-off the motor power while maintaining all processing capabilities. An embedded computer (Odroid XU4) is used to send commands to the motor drive, to interact with the radio controlled (RC) transmitter for remote operation and to create a local network to interface with the other modules. Additionally, the Odroid creates a wi-fi network that broadcasts the vehicle information to a remote location, namely a ground station. The on-board GPS and IMU sensors are connected to the embedded computer for their acquisition and processing, and can be used for the autonomous navigation if no other solution is available. Moreover, the computer can be used to execute more advanced features, such as waypoint navigation, collision avoidance and geopositioning.



(a) Hardware architecture. Yellow, green and black lines are the communications, regulated power and energy supply connections, respectively.



(b) Functionalities layout.

Figure 3.3: SENSE hardware architecture and functionalities layout

As for the perception module, the PSU outputs the power for the computer and regulates the voltage to 5 V, 12 V and 24 V, which then can be used to power the available sensors on the selected payload. The embedded computer employs a mini-itx motherboard solution equipped with an Intel i5-8400@2.80 GHz CPU, a 16 Gb RAM and 1 Tb SSD storage, as well as Ethernet and USB 3.0 communications. This provides a platform with higher computational capacities to acquire, process and store sensor data to be used during real-time operation or to be analysed

offline. An example of the data gathered by the perception module with the corresponding sensor alignment in the SENSE is depicted in figure 3.4. This provides one example of the data obtained by the available perception sensor payload, namely a pair of images from the stereo camera, the LiDAR point cloud and the MBES scan (line under SENSE) retrieved on cloudy day in *Marina de Leça*. The localization module provides extra navigation sensors and an embedded computer to read the data and estimate the localization of the vehicle. This adds redundancy to the USV positioning system with high-end GPS and IMU sensors, that allows a more accurate localization estimation with access to new technologies, for instance real-time kinematics (RTK), which can be used as ground-truth due to its centimetrical precision.

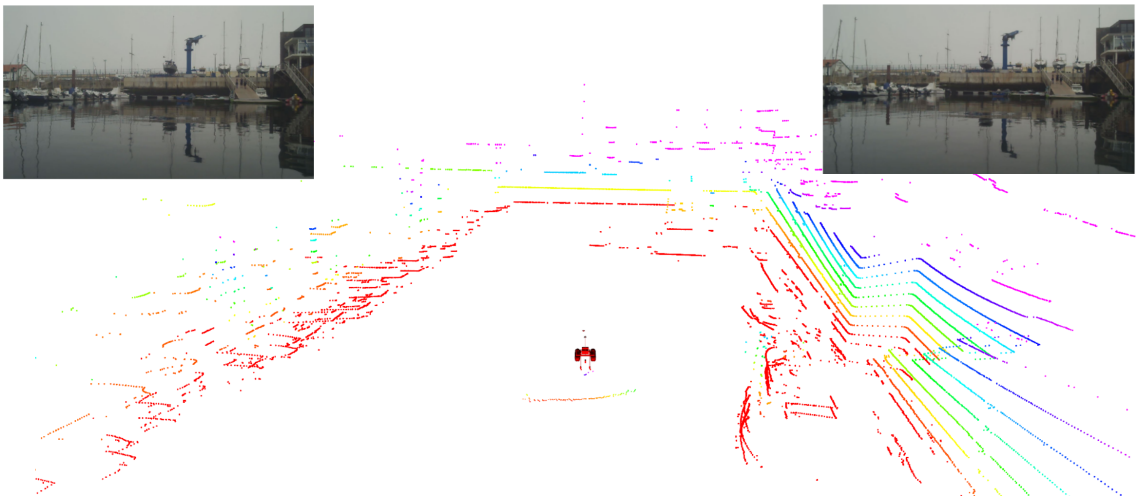


Figure 3.4: The 3D representation of a SENSE acquired frame, based on data from the LiDAR and Sonar sensors. Stereoscopic images are presented on top.

The addition of new modules besides the ones presented here can be achieved by connecting the new box to the local network and to the energy module. For instance, it can be added a communication relay module for the interaction with underwater vehicles. A remote ground station, with Wi-fi connectivity, can be used to transmit missions for the vehicle, to monitor the current state and to read the sensor data, such as images and point clouds.

3.3 Nautilus - An autonomous surface vehicle with a multi-layer software architecture for offshore inspection

The Nautilus, depicted in figure 3.5, is an USV designed for patrolling and the inspection of maritime scenarios, such as harbors and offshore wind farms. It uses multi-domain perception sensors, namely MBES, LiDAR (Light Detection And Ranging), visual and thermal cameras. This payload allows to reconstruct the environment, inspect faulty structures, and extend the vehicle situational awareness capabilities. Navigation sensors, in particular GPS and IMU, are also available for localising the vehicle and performing autonomous operations.

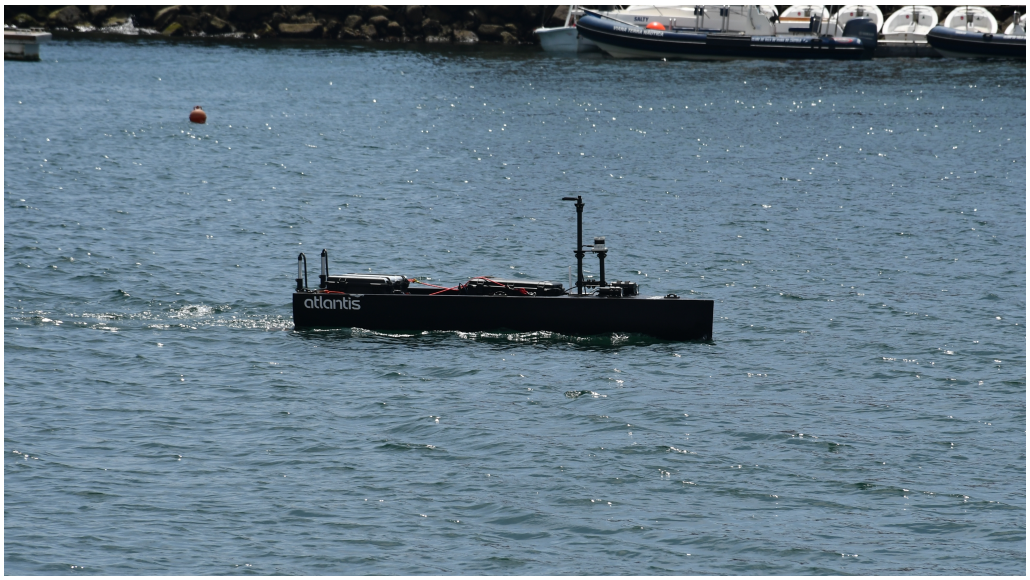


Figure 3.5: Nautilus USV operating in Viana do Castelo.

The USV design provides a stable and robust platform with a monohull configuration, while maintaining a small footprint to be easy to transport and deploy. The hull design as well as the navigation payload allows it to operate within the range of sea state 4, up to 2.5 m maximum wave height, in the Douglas sea scale [62]. Thus, Nautilus works within the expected operation window on offshore wind farms for boats with small hulls, i.e. a maximum wave height of 1.5 m. Nautilus' modular hardware architecture allows to easily include, remove or replace sensors according to the operation requirements. The current design provides heterogeneous data from the surface and the underwater domains that can be visualised on a Shore Control Centre (SCC) through wi-fi communication [35]. This SCC also enables the definition, supervision and control of the vehicles' missions. Moreover, the USV can be also tele-operated with Beyond Visual Line-Of-Sight (BVLOS) capabilities through a 2.4 GHz radio controller with a 20 km range.

The Nautilus is a compact vessel with 3.0 m length and approximately 130 kg, figure 3.6. It has a monohull configuration, designed by Composite Solutions, built using GRP (glass reinforced

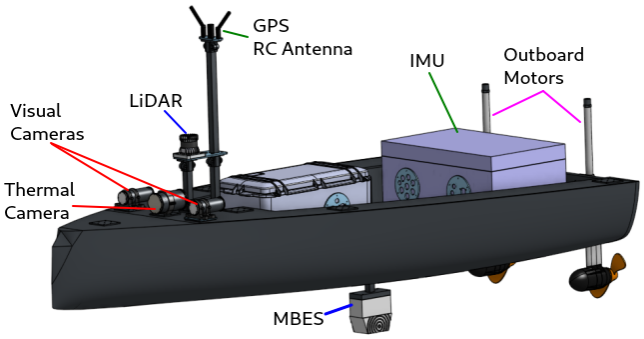


Figure 3.6: Nautilus CAD with the sensor payload attached. The blue, red, and green lines are distance, imaging, and navigation sensors, respectively. The pink lines are the actuators.

plastic) composite sandwich making up a weight of approximately 45 kg, and a net buoyancy of 250 kg. It is equipped with several linking unions, where the modules boxes can be attached, and maritime-grade stainless steel fixation plates to support the surface sensors. The underwater sensors can be placed through a moon pool hatch placed on the centre of the hull with a size of 373 x 373 mm. Its modules are powered through a 24 V and 150 Ah LiFePO4 battery, stored in a *Peli* case. This provides an autonomy of approximately 3 h with the current payload, considering the motors are working around half-throttle. The energy module can be easily replaced or charged externally without requiring to open the case.

Nautilus has a differential drive configuration via two brushless outboard motors placed at the stern of the hull, equivalent to two 3 hp petrol outboards, controlled by a commercial VESC solution. Each motor produces a propulsive power of 520 W with a power consumption of at most 1100 W. This configuration allows the USV to reach velocities up to $3\,\mathrm{m\,s^{-1}}$. As for the current sensor payload, it provides multi-domain and navigational data, being composed by:

- A GPS (Holybro F9P), with RTK technology at a rate of 3 Hz. The accuracy with GNSS is 1.5 m and with RTK 0.01 m. It provides the localisation of the vehicle and allows to geo-tag inspected flaws.

- A 9 degrees-of-freedom IMU (Xsens MTi 630) that provides the attitude of the vehicle, the angular velocity and linear acceleration with a rate of 20 Hz and a ENU (East-North-Up) referential. The sensor fusion provides an error of 0.2 deg RMS in roll and pitch, and 1.0 deg for yaw. It can be employed for the vehicle localisation and the perception data alignment.

- A 3D LiDAR (Ouster OS1-128 Rev7), with 128-channel LiDAR, a range up to 200 m, a 360° x 45° Field of View (FoV) and a rate of 10 Hz. The sensor has mean range error of 2.5 cm. It is applied to reconstruct the surface environment, detect nearby obstacles, and navigate near structures.

- Two visual cameras (Hikrobot MV-CA016-10GC), with a 12.5 mm lens, a resolution up to 1440 x 1080, a rate of 30 Hz and external triggering. These sensors are synchronised via the external trigger allowing to be used for stereo perception with a baseline of 50 cm, capable of tracking surrounding elements, detect structural flaws, provide stereoscopic information, and supply visual data for BVLOS tele-operation.

- A thermal camera (Flir Vue Pro), with a 13 mm lens, a resolution of 336 x 256 and a rate of 30 Hz. It provides thermographic data required for inspection and detection of flaws in offshore structures and, along side the LiDAR, facilitates night operations.

- A multibeam echosounder (Imagenex "Delta T" 837B), with a frequency 260 kHz, range up to 100 m, an angular FoV of 120° and a 20 Hz rate. The sensor has a range resolution of 0.2% of the range. It allows to inspect underwater structures, create bathymetry information, and geo-localise submerged structures.

Table 3.2 depicts a summary of the Nautilus characteristics. A sample of the navigation data (GPS and IMU) taken in one of the trials is depicted in figure 3.7, where it can be observed the vehicle trajectory, represented by the green line, a attitude meter, placed on the bottom that represents the vehicle roll, pitch and yaw, and an image, retrieved by the visual camera.

Table 3.2: Technical specifications of the Nautilus.

Characteristics	**Specifications**
Dimensions (L x W x H)	3.0 m x 1.4 m x 1.5 m
Total weight	∼130 kg
Available payload	∼120 kg
Moon pool size	373 mm x 373 mm
Drive mode	Differential drive Outboard motors
Max speed	3 $m\,s^{-1}$
On-board sensors	Navigation: GPS, IMU Perception: 3D LiDAR, MBES, Visual Cameras (x2), Thermal Camera
Power supply	LiFePO4 24 V@150 Ah
Power consumption (max.)*	∼2700 W
Operation time	∼3 h
Douglas sea scale	4 (up to 2.50 m)

* Maximum power consumption values for each individual component were considered to estimate the overall energy consumption.

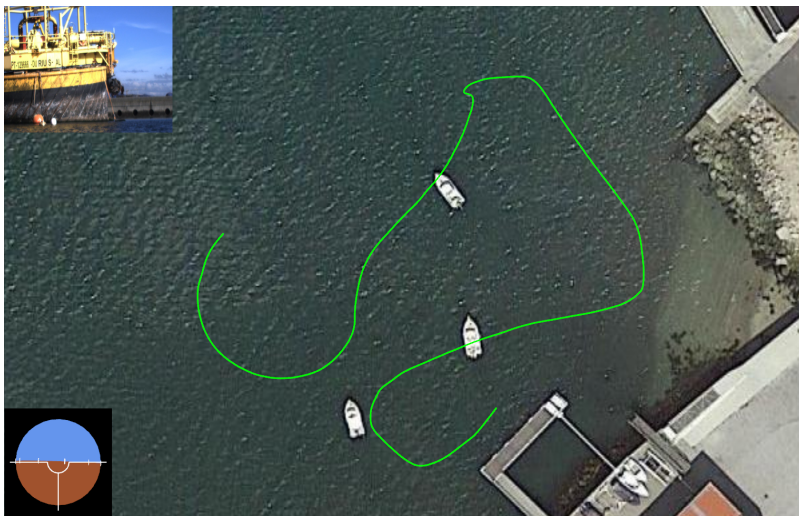


Figure 3.7: Navigation data from the Nautilus. The green line corresponds to the GPS positions and the attitude meter depicts the current roll, pitch and yaw of the vehicle.

Figure 3.8 depicts a sample of the perception data. It provides visual information through the cameras (right and left) with different perspectives to be able to perform stereo matching. Nevertheless, this information is highly constrained by the lighting conditions and fog. The LiDAR and the MBES provide distance information using infrared (IR) and sonar technology, respectively, being less susceptible to environmental changes. The thermal camera provides an IR image which allows to retrieve the structures' temperature and operate in low-lighting conditions.

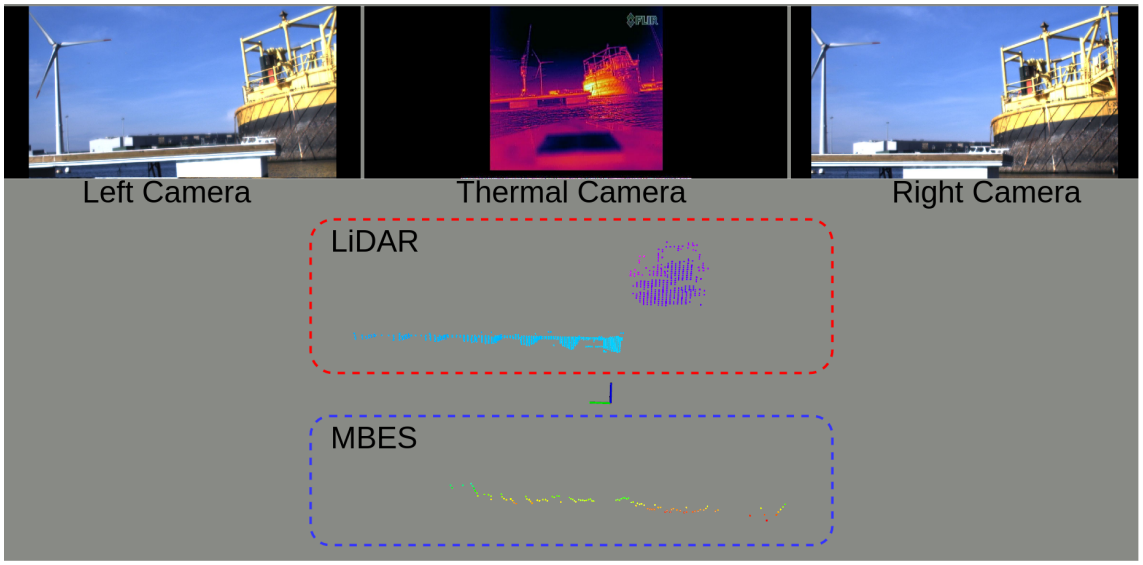
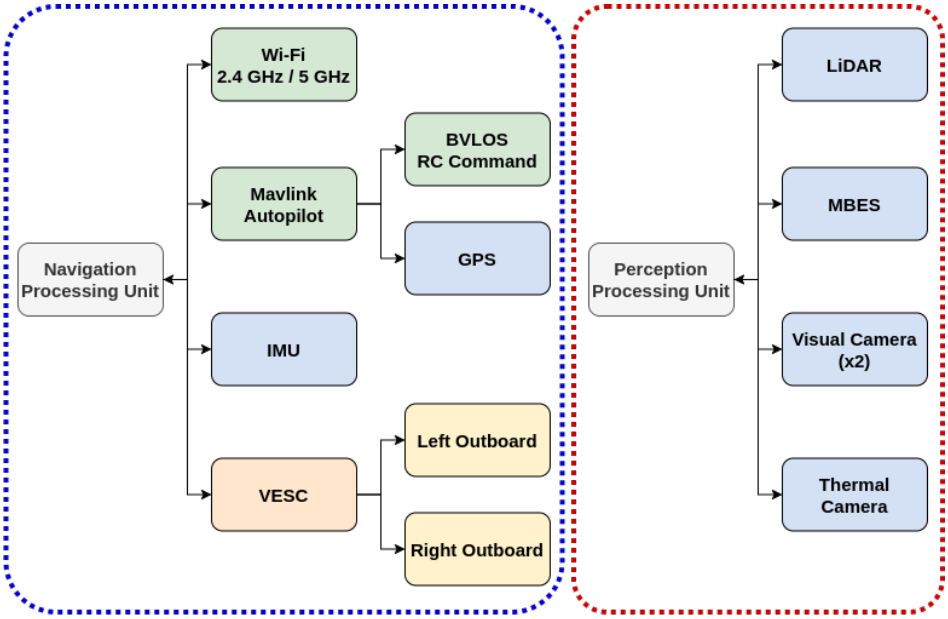


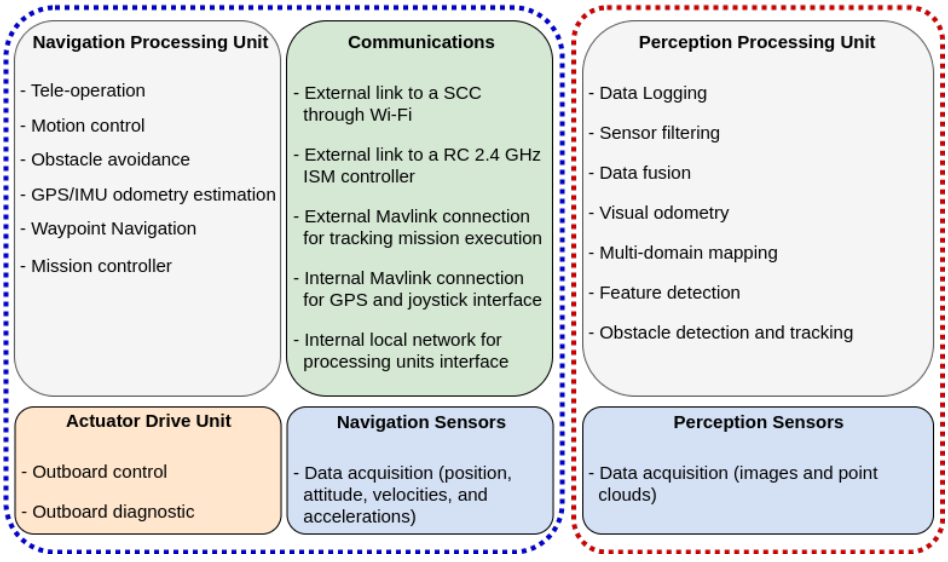
Figure 3.8: Nautilus perception data gathered in Viana do Castelo.

Nautilus hardware is distributed between two watertight cases containing the: energy module, and perception, navigation and communication (PNC) module. The first is solely composed by the battery with the integrated Battery Management System (BMS) and the system power cut-off system. The second is arranged into distinct submodules that creates the interfaces with the sensors and actuators, defines the internal and external communications, and executes the defined mission, figure 3.9. All physical interfaces between modules and external connections are made through side panels adaptable to suit the required connectors.

The PNC module, as depicted in figure 3.9a, is divided in two submodules, navigation and communication, and perception. Each one is endowed with a processing unit, linked through a gigabit local network. The navigation and communication (NavCom) submodule is designed as the master processing unit to ensure the safety by design, where a critical fault from the navigation components halts the vehicle, whilst a defect from the perception parts allows the vehicle to return to base. As such, the NavCom creates the internal and external communications, controls the outboards, acquires navigational data and executes the tele-operation and autonomous navigation procedures, figure 3.9b. This submodule uses an embedded computer as the processing unit that acquires the IMU data and communicates through Mavlink to acquire the RC and GPS inputs, Gigabit Ethernet to create the local network and wi-fi to communicate with the SCC. The perception submodule is designed for higher computational capacity to withstand the larger data



(a) Nautilus PNC module structure.



(b) Nautilus functionalities layout.

Figure 3.9: Nautilus hardware and functionalities layout. The blue and red dashed boxes are the navigation and communication, and perception submodules, respectively. The grey, green, blue, orange and yellow boxes are the processing units, communications, sensors interfaces, actuator drive units, and actuators, correspondingly.

acquisition and processing requirements. This is connected to the NavCom through the local network and uses a laptop as a processing unit. As such, this unit is responsible for acquiring data from the perception sensors, filtering the data, perform skills to increase the situational awareness and inspection methodologies. Moreover, this submodule logs all the data retrieved by Nautilus, that can be accessed and retrieved via wi-fi.

All communications, internal and external, use distinct protocols being connected as depicted in figure 3.10, where all timestamps between assets are synchronised using Network Time Protocol. Both processing units are connected through Ethernet and use the ROS framework to create the data bridges between them, while defining the NavCom as the master unit and establishing an external bridge connection through a wi-fi extender. The sensors are linked through their predefined interface, either Ethernet, power over Ethernet (PoE) or serial. The Mavlink autopilot is equipped with a redundant IMU and allows to retrieve the telemetry from the vehicle, acquiring the GPS data and synchronising through a wi-fi link with the RTK base station. Moreover, the autopilot also provides the communication between the BVLOS RC command and the navigation processing unit. This BVLOS system communicates through a 2.4 GHz link and allows not only to control the vehicle, but also to monitor the telemetry and stream a visual image using the Mavlink protocol.

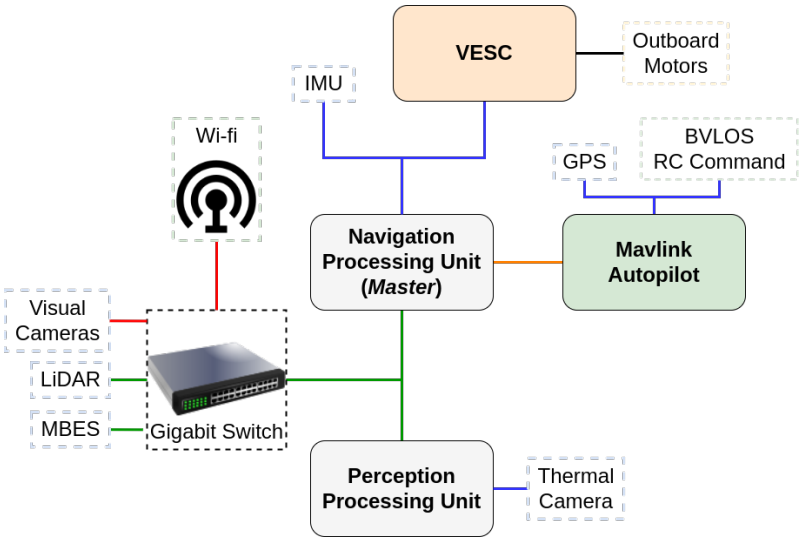


Figure 3.10: Nautilus internal and external communication links. The red, green, blue, orange and black lines represent the PoE, ethernet, serial, mavlink and power outputs.

3.4 Software architecture

The hardware architecture endows both SENSE and Nautilus with a modular framework capable of integrating distinct sensors and execute different tasks adapted to the designed mission. With the current design, the USVs are capable of performing skills like mapping and localisation near structures [174], autonomous docking [24] or object tracking [175]. Thus, it demands a substantial flexibility from the software architecture to adapt to the different hardware and skills that the USVs are able to fulfil. Moreover, offshore O&M presents challenging conditions to the autonomous agents that perform them. The different domains of operation - air, surface and underwater - expose autonomous vehicles to environments with the most distinct dynamics and constrains. Most of the times, new robots are re-programmed from scratch, discarding work that

could potentially serve the same purpose. Aiming at reducing waste of resources and the need to redo redundant work, skill-based approaches come into hand. With focus on industrial robotics, such architectures usually divide the architecture into several layers. The bottom layer, composed of actions or primitives, interconnects the hardware to the decision making layer, where skills sit. This way, skills output data that is independent to the hardware, creating a level of abstraction that is much desired when software re-usability is wanted. On top of the skill level, a task level usually concatenates skills in a sequentially arranged fashion, allowing for distinct complex operations to be run one after the other.

Mobile robot applications of such architectures are, however, still very scarce. The Multi-layer Universal Software Architecture (MUSA) aims at filling that void. Its main goal is to ease the programming task of mobile robots, and it tackles challenges such as software portability and re-usability, and data source agnosticism. In other words, it aspires at making the code interchangeable and independent from the protocol that drives data from sensors and to actuators. Thus, the underlying principle of MUSA is to foster software portability and establish an abstraction from the hardware in the control loop, where it can work stand-alone or be integrated in existing operating systems for robots, such as ROS and ROS2. For such, MUSA has to fulfil the following requirements:

- Each level of the hierarchy has to share information with the level atop or beneath it;
- A templated backbone for each level of the hierarchy must provide a mechanism that encompasses the basic operation of that each element of that level;
- Every level has to be agnostic to the levels within reach, meaning each component of the hierarchy might be swapped for a more (or less) complex component, without loss of the system's functionality.

3.4.1 MUSA - Multi-layer universal software architecture

MUSA provides an architecture that can cope with complex O&M operations. It is composed of four levels: a Mission ($\mathscr{M}$), a Task ($\mathscr{T}$), a Skill ($\mathscr{S}$) and an Action ($\mathscr{A}$) level, figure 3.11. This creates a hardware abstraction layer which separates from conventional architectures that are usually dependent of the autopilot in use. MUSA is designed to be able to cope with any hardware structure treating each dependency at the lower level of the architecture, the action layer.

The actions are closely related to hardware, they are responsible for turning information into commands that the hardware level, through its drivers, can interpret. A skill, $\mathscr{S}$, used in certain domains and environments can only depend on specific actions that are physically viable for the robot hardware and with the operations to be performed. The skills add functionalities to a robot and are defined as a set of individual and self contained actions. Having some sort of control process, they interpret information from the environment using the data collected by the sensors or control the vehicle sending commands to the actuators. Thus, $\mathscr{S}$ can be seen as a set of n

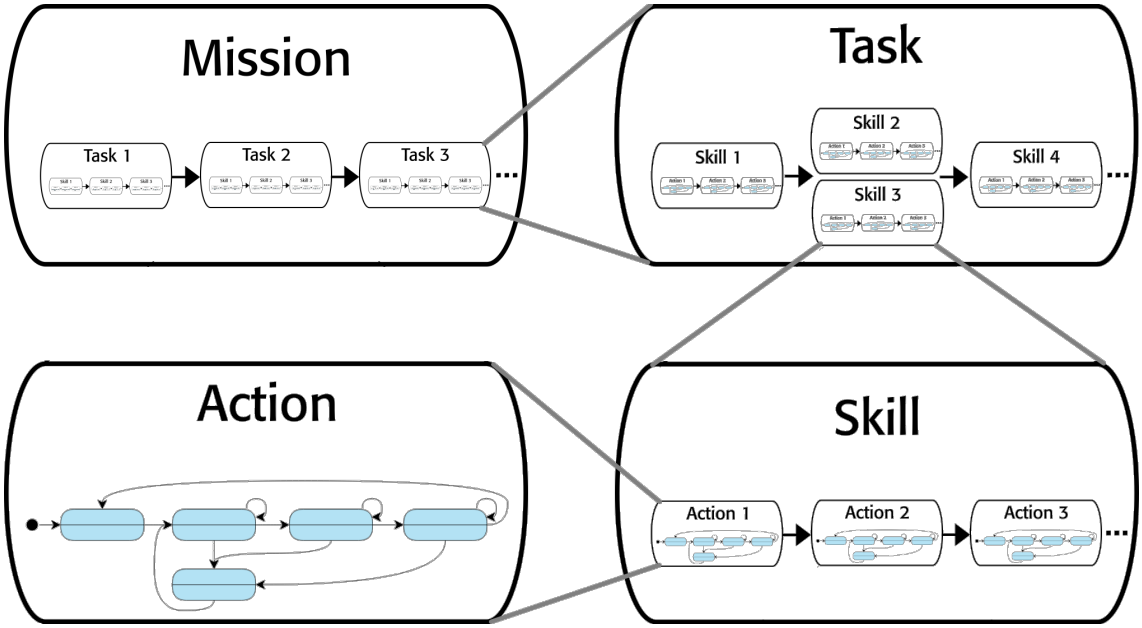


Figure 3.11: MUSA: the four layers that make the proposed architecture.

actions:

$$\mathscr{S} = \{\mathscr{A}_1, ..., \mathscr{A}_n\}. \tag{3.1}$$

$\mathscr{S}$ can be either persistent, i.e. runs continuously in background parallel to other skills until the task is stopped (e.g. odometry), or volatile, i.e. runs once until the goal condition is reached (e.g. positioning system). To avoid execution repetitions, persistent skills can be shared between tasks. Programming at this level requires a comprehensive knowledge of the operation to be conducted. It is also required that the dynamics and limitations of the domain to which the skill is to be applied is taken into account. It is at this level that information is turned into commands to be fed into the action layer. Therefore, routines such as motion planning, motion execution and obstacle avoidance should be implemented at skill level.

MUSA starts to differ from the already existing skill-based approaches at task level. Unlike previous works, tasks are defined, not as a sequence of skills, but as a set of skills, which might be organised either sequentially, concurrently, or as combination of both. Hence, tasks are a set of m skills planned to overcome a specific goal, i.e.:

$$\mathscr{T} = \{\mathscr{S}_1, ..., \mathscr{S}_m\}. \tag{3.2}$$

They also define input parameters for skills. Unlike statically placed industrial robots, mobile robots might have to perform several skills concomitantly (e.g. hoovering while a manipulator performs an operation). With this approach, MUSA takes advantage from concurrency, meaning mobile robots can benefit from skills that execute at the same time.

Above task level, it is proposed a Mission layer, defined as a concatenation of tasks that are, again, run sequentially. Missions further extend the applicability of the proposed architecture,

allowing for high-level programming of tasks, by simple definition of parameters. Thus, it can be defined as the set of k tasks:

$$\mathscr{M} = \{\mathscr{T}_1, ..., \mathscr{T}_m\}. \tag{3.3}$$

Upon the successful completion of all k tasks $\mathscr{T}$, it can be guaranteed that $\mathscr{M}$ was also successfully concluded. Nonetheless, these tasks are not exclusive to only mission, i.e., two different missions, $\mathscr{M}_1$ and $\mathscr{M}_2$, can share the same task $\mathscr{T}$ or, even, being defined by the same set of tasks.

To better understand the division, one can picture the scenario in which an USV has to inspect an offshore wind farm autonomously. The whole inspection operation composes the entire mission, which can be broken into smaller operations - tasks - such as the inspection of individual wind turbines. The inspection of each wind turbine requires the USV to be able to perform navigation skills, such as reaching a location and hovering around a fixed point. Skills send velocity commands to actions, which convert that information into commands for the USV motors. Zooming in on each individual element of every layer (an individual mission, task, skill or action), one can find that the main workflow follows a similar sequence, which goes as follows:

1. Upon start, an initialisation routine is called. It is responsible for checking if the pre-conditions for the element's normal execution are met, and, if so, if it can proceed. Otherwise, the execution stops, returning an error;

2. The continuous execution routine is executed after the initialisation procedures are completed. This routine is non-blocking, meaning its execution is triggered by an object of a layer on top of it or, if it is the case of the highest layer, at a certain rate. Once again, in case of failure, the execution is halted, requiring a new start command to be sent;

3. Before concluding the execution, a post-condition checking routine is called. It is responsible for checking the overall execution status of the process and, eventually, for sending the last commands to the action layer.

In case of failure of any of the aforementioned routines, the execution is enters a failure routine, which halts the operations. A visual representation of the described workflow is illustrated in Figure 3.12, in the form of a state machine diagram.

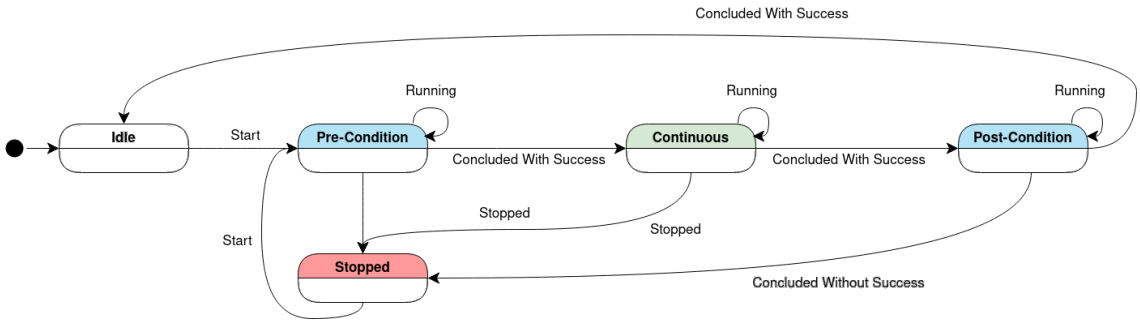


Figure 3.12: The state machine diagram of the execution of each individual element of the proposed architecture.

3.4.2 Software design

The on-board software architecture describes the autonomous navigation and remote operation methodologies, as well as the middleware used to communicate with the sensors, defined with the MUSA architecture. It uses an event based communication, namely Qt's signals and slots[2], to interchange information between layers and ROS [176] messaging system to interface with hardware drivers with a hybrid communication network configuration, where the local network is distributed through gigabit Ethernet switches that links all nodes between modules, and the remote access uses a centralized configuration, available through a wi-fi connection. To ensure robustness to updates, the ROS tools used are the most widely integrated and stable, to minimize the impact on the software, since they are the less prone to suffer changes and the adaptations are easily applied. Moreover, the data buffering and timestamping provided by ROS allows to minimize the impact of packet loss, where all timestamps are synchronized with the master module using a Network Time Protocol (NTP) designed to perform well with intermittent network connections, and heavily congested networks, having a sub-microsecond accuracy. To ensure the mission safety, the navigation processing unit of each vehicle are used to create the ROS master, that tracks all publishers, subscribers and parameters. Thus, in case of a system failure on any other module it is always possible to control the USVs. In the event of a navigation failure, the mission is aborted and the thrusters are halted.

The software functionalities for both in SENSE and Nautilus are divided in three main tasks, namely navigation, localization and perception, distributed through each in-vehicle processing units, figure 3.13. As such, the architecture was fit into the USVs in the form of an inspection mission for maritime environments, that can be divided as:

- $\mathscr{M}$ - **Inspection**, navigates, gathers data and reconstructs the scenario in maritime environments. The mission defines the type of navigation, either remotely operated or autonomous, which can be commuted online. This requires the tasks:
  - $\mathscr{T}_0$ - **Localization**, gathers the vehicle localization information. This requires the skill:
    - $\mathscr{S}_{00}$ - **GPS/IMU Odometry**, localization method based on GPS/IMU fusion. Its a persistent skill that is executed continuously. This requires the actions:
      - $\mathscr{A}_{000}$ - Get GPS data.
      - $\mathscr{A}_{001}$ - Get IMU data.
  - $\mathscr{T}_1$ - **Navigation**, tele-operated or drive through a list of goal points task (autonomous). This requires the skills:
    - $\mathscr{S}_{10}$ - **Remote Operation**, uses the RC command data and converts it into velocities, safety stop command and navigation mode switching. Its a persistent skill that is executed continuously. This requires the actions:
      - $\mathscr{A}_{100}$ - Calculate kinematics.

[2]Qt, 2023. *Signals & Slots*. URL:`http://tinyurl.com/qtsigslot`

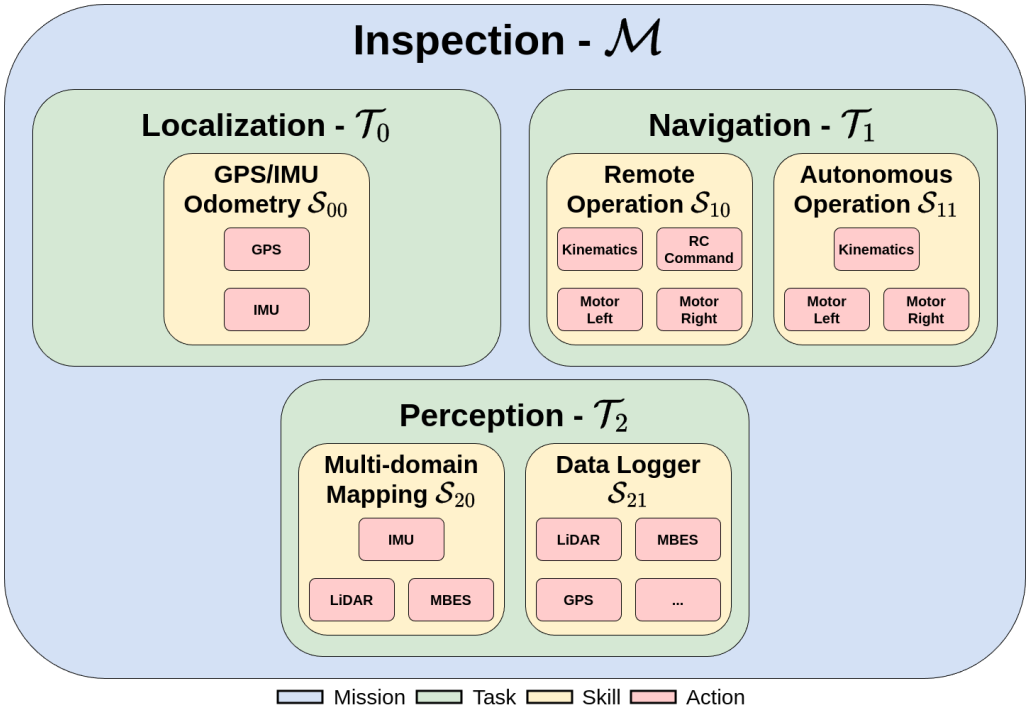


Figure 3.13: Inspection mission using MUSA. Blue, green, yellow and red boxes represent mission, task, skill and action layers, respectively.

- $\mathscr{A}_{101}$ - Get RC command data.
- $\mathscr{A}_{102}$ - Control left motor.
- $\mathscr{A}_{103}$ - Control right motor.

* $\mathscr{S}_{11}$ - **Autonomous Operation**, algorithm based on the error between the current position and the desired waypoint. Its a volatile skill that is executed once until the waypoint position is reached. This requires the actions:
  - $\mathscr{A}_{110}$ - Calculate kinematics.
  - $\mathscr{A}_{111}$ - Control left motor.
  - $\mathscr{A}_{112}$ - Control right motor.

– $\mathscr{T}_2$ - **Perception**, creates the scenario reconstruction and logs the required data. This requires the skills:

* $\mathscr{S}_{20}$ - **Multi-domain Mapping**, uses the data from the LiDAR, MBES and IMU as well as the vehicle pose to create a multi-domain representation with the method proposed by Campos *et al.* (2021) [30]. Its a persistent skill that is executed continuously. This requires the actions:
  - $\mathscr{A}_{200}$ - Get IMU data.
  - $\mathscr{A}_{201}$ - Get LiDAR data.
  - $\mathscr{A}_{202}$ - Get MBES data.

* $\mathscr{S}_{21}$ - **Data Logger**, stores relevant data from all modules into a solid-state drive (SSD). Its a persistent skill that is executed continuously. This requires the actions:
  - $\mathscr{A}_{210}$ - Get LiDAR data.
  - $\mathscr{A}_{211}$ - Get MBES data.
  - $\mathscr{A}_{212}$ - Get GPS data.
  - $\mathscr{A}_{21n}$ - Get n required topics.

The mission $\mathscr{M}$ coordinates the navigation mode, defines the trajectories, monitors task execution, and provides real-time feedback on the vehicle's status. In autonomous mode, the trajectory can either be loaded from a file or defined dynamically through the SCC. To switch between navigation modes, the mission either utilizes the SCC interface with $\mathscr{M}$ or actively monitors the navigation task $\mathscr{T}_1$ mode flag, which can be modified via the RC command.

The localization task $\mathscr{T}_0$, monitor and share the vehicle pose obtained from the odometry skill $\mathscr{S}_{00}$. In this skill, the absolute position is estimated with the GPS, action $\mathscr{A}_{000}$, and the linear acceleration, angular velocity and orientation of the USV is obtained from the IMU sensor, action $\mathscr{A}_{001}$. The data acquired from these sensors is fused to estimate a 6-DOF pose using an Extended Kalman Filter (EKF) [29]. This filter employs a 15-dimensional state vector (X_k) given by equation 3.4, comprising the vehicle's 3D pose $P = (x, y, z)$, orientation $\omega = (\phi, \psi, \theta)$, where ϕ is roll, ψ is pitch and θ is yaw, along with their respective velocities $\dot{P} = (\dot{x}, \dot{y}, \dot{z})$ and $\dot{\omega} = \left(\dot{\phi}, \dot{\psi}, \dot{\theta}\right)$, and linear accelerations $\ddot{P} = (\ddot{x}, \ddot{y}, \ddot{z})$. The EKF's prediction model, used to project the system state forward in time, is based on a 3D kinematic model derived from Newtonian mechanics, given by equation 3.5, where R_{ZYX} represents the ZYX Tait-Bryan angles rotation matrix and Ω is defined in equation 3.6.

$$X_k = \begin{bmatrix} x & y & z & \phi & \psi & \theta & \dot{x} & \dot{y} & \dot{z} & \dot{\phi} & \dot{\psi} & \dot{\theta} & \ddot{x} & \ddot{y} & \ddot{z} \end{bmatrix}^T, \tag{3.4}$$

$$\begin{cases} P_k = P_{k-1} + R_{ZYX} \cdot \left(\dot{P}_{k-1}\Delta t + 0.5\ddot{P}_{k-1}\Delta t^2\right) \\ \omega_k = \omega_{k-1} + \Omega \cdot \dot{\omega}_{k-1}\Delta t \\ \dot{P}_k = \dot{P}_{k-1} + \ddot{P}_{k-1}\Delta t \\ \dot{\omega}_k = \dot{\omega}_{k-1} \\ \ddot{P}_k = \ddot{P}_{k-1} \end{cases}, \tag{3.5}$$

$$\Omega = \begin{bmatrix} 1 & sin(\phi)tan(\psi) & cos(\phi)tan(\psi) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi)cos(\psi)^{-1} & sin(\phi)cos(\psi)^{-1} \end{bmatrix}. \tag{3.6}$$

The EKF predicts the USV poses throughout time when no sensor measure exists by using this constant velocity model. When a new sensor reading is perceived the estimate is corrected. To support distinct sensors, it is assumed that each sensor produces measures of the estimated state variables, thus the H observation matrix is an identity matrix. Nevertheless, the sensors are

asynchronous and do not measure all state vector variables independently, partial updates of the state vector are performed. This is achieved by reducing the observation matrix H to a m by 15 matrix with the only non-zero values (in this case set to 1) being the m measured variables [177]. The EKF filter inputs are the IMU attitude (roll, pitch and yaw) and angular velocity data, and the positional information of an odometry inferred from the GPS and IMU initial state. The latter is acquired by projecting the GPS data to the Universal Transverse Mercator (UTM) using the transformation defined in the USGS Bulletin 1532 [178]. This is given by the equations 3.7 and 3.8.

$$x = k_0 \cdot N \cdot \left(A + \frac{(1 - T + C) \cdot A^3}{6} + \frac{5 - 18T + T^2 + 72C - 58e'^2}{120} \right), \tag{3.7}$$

$$y = k_0 \cdot \left[M - M_0 + N \cdot tan\phi \cdot \left(\frac{A^2}{2} + \frac{(5 - T + 9C + 4C^2) \cdot A^4}{24} + \frac{(61 - 58T + T^2 + 600C - 330e'^2) \cdot A^6}{720} \right) \right], \tag{3.8}$$

where k_0 is the scale on central meridian for UTM, e the first eccentricity and ϕ the latitude in radians. e'^2, N, T, C, and A are defined on equations 3.9 to 3.13, where a is the semi-major axis for WGS84, λ is the longitude in radians and λ_0 the origin longitude for the current UTM zone. M is the true distance along the central meridian from the Equator to ϕ obtained from equation 3.14 and $M_0 = M$ calculated for ϕ_0, the latitude crossing the central meridian λ_0, at the origin of the x, y coordinates.

$$e'^2 = \frac{e^2}{1 - e^2}, \tag{3.9}$$

$$N = \frac{a}{\sqrt{(1 - e^2 \cdot sin^2\phi)}}, \tag{3.10}$$

$$T = tan^2\phi, \tag{3.11}$$

$$C = e'^2 \cdot cos^2\phi, \tag{3.12}$$

$$A = cos\phi \cdot (\lambda - \lambda_0), \tag{3.13}$$

$$\begin{aligned} M = a \cdot [&(1 - e'^2/4 - 3e'^4/64 - 5e'^6/256) * \phi \\ &- (3e'^2/8 + 3e'^4/32 + 45e'^6/1024) \cdot sin(2\phi) \\ &+ (15e'^4/256 + 45e'^6/1024) \cdot sin(4\phi) \\ &- (35e'^6/3072) \cdot sin(6\phi)] \end{aligned}. \tag{3.14}$$

With the UTM projection the odometry is estimated from the position evolution throughout time in relation to the first GPS coordinate acquired and the correspondent initial rotation from the IMU.

The navigation task $\mathscr{T}_1$ controls the vehicle motion either on remote or autonomous operation. Each mode provides linear and angular velocities - v and ω, respectively - for the USV and at

the task level a prioritization is made, where the remote commands have higher priority than the autonomous mode to control the vehicle. This allows an on-mission mode switching, increasing the flexibility and safety of the vehicle by allowing to halt the mission and pass the control for a trained pilot, for instance in case of a system failure or if a new goal outside the mission scope is required. In autonomous mode, $\mathscr{T}_1$ sends one goal-point at a time to the skill $\mathscr{S}_{11}$, and waits for its execution to be completed to send the next command.

For the remote operation skill $\mathscr{S}_{10}$, the RC transmitter is used to alternate to tele-operated mode and to send the velocity commands. The controller joystick states, acquired from $\mathscr{A}_{101}$, are mapped to the linear and angular velocities by using the linear regression presented in equation 3.15, where y is the velocity (linear, y_v, or angular, y_w), s is the stick scale and the subscripts *min* and *max* defines the minimum and maximum values for the velocity and the scale. This transmitter also allows to override all autonomous navigation commands or to activate a safety procedure that halts the USV motion during emergency situations.

$$y = \frac{(y_{max} - y_{min}) \cdot (s - s_{max})}{s_{max} - s_{min}} + y_{max}. \tag{3.15}$$

As for the autonomous navigation skill $\mathscr{S}_{11}$, it uses the localization estimation from $\mathscr{T}_0$ and receives goal-points as parameters, in the form of 2D coordinates, and outputs linear and angular velocity commands to the action layer. Hence, position is represented by its x, y coordinates, and orientation is expressed by θ. Its continuous execution state consists of a finite state machine model with five states, namely an initial state, a rotation phase, a state where the angular velocity is reduced, a cruising state, and a final state, where the USV's velocity is decreased. The skill's continuous execution behaviour is illustrated in Figure 3.14. At the initial state, the USV checks if its position relative to the goal-point is within the limits set by a user defined parameter, which bounds the maximum Euclidean distance, $\|\Delta(x,y)\|_2$, from which the system advances directly to the final state. If $\|\Delta(x,y)\|_2$ is bigger than the defined parameter, the execution state changes to a rotation stage. At this stage, the USV rotates until it is oriented to the direction given by a line that crosses both the vehicle's base line and the goal-point. Given the dynamics of the environment where the USV operates, and the lack of a finer velocity controller, the USV stops the rotation when the module of its deviation in orientation , $|\Delta\theta|$, is smaller than a user defined parameter. Then, in the third state, the USV waits for a time interval to pass, or until the error in orientation reduces in half, to start moving with linear velocity. The time interval duration is, once again, a parameter of the skill that a user can easily set or modify. In the following state, the USV cruses at the maximum defined linear velocity. The angular velocity is calculated by a proportional controller, with a gain that was tuned during the conducted tests. Finally, as the USV reaches the goal-point coordinates, and the skill's continuous execution enters its last stage, linear velocity is reduced to a specified value, and angular velocity is adjusted proportionally to $|\Delta\theta|$.

The final linear and angular velocities are then converted to the velocity per thruster required for the USV to reach that velocities, to be sent by actions $\mathscr{A}_{102}$, $\mathscr{A}_{103}$, $\mathscr{A}_{111}$ and $\mathscr{A}_{112}$. As such, a differential drive kinematic model is applied, actions $\mathscr{A}_{100}$ and $\mathscr{A}_{110}$, where the speeds v_r^t and

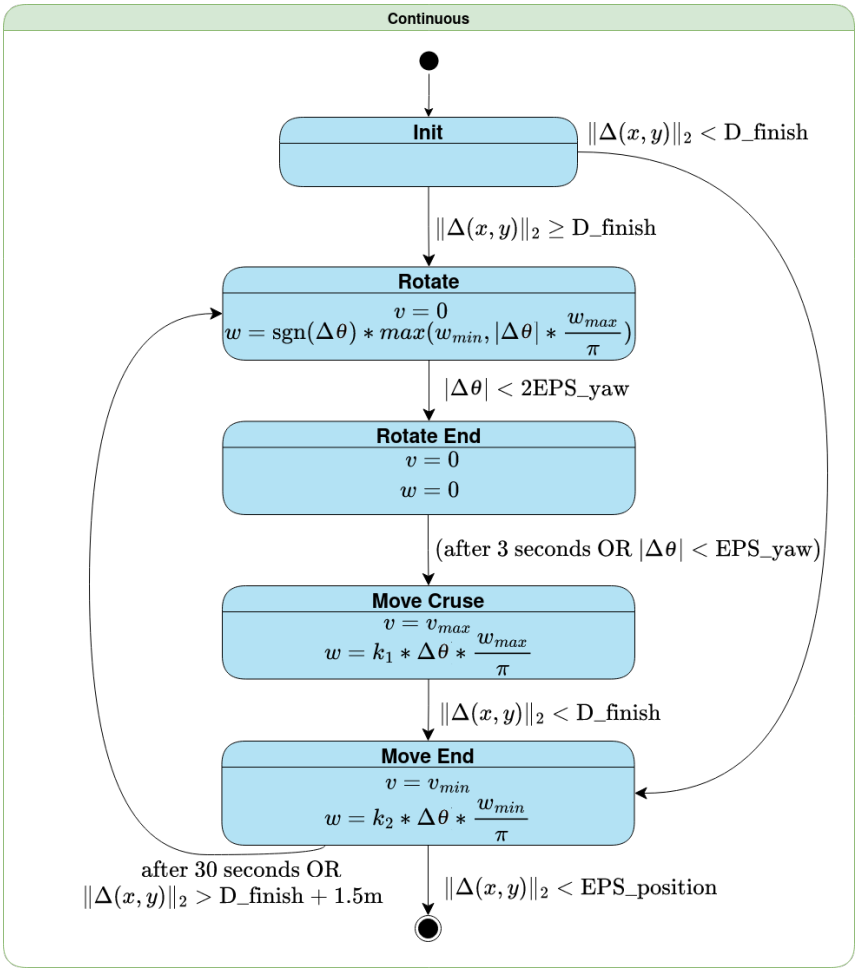


Figure 3.14: The state machine diagram of the continuous execution routine of the proposed autonomous operation skill, $\mathscr{S}_{11}$.

v_l^t affected to each thruster can be determined from equations 3.16 and 3.17, where α and β are constants for thrusters placement and configuration within the USV [31].

$$v_r^t = \alpha \cdot v + \beta \cdot \omega, \tag{3.16}$$

$$v_l^t = \alpha \cdot v - \beta \cdot \omega. \tag{3.17}$$

The perception task $\mathscr{T}_2$, performs the multi-domain mapping skill, $\mathscr{S}_{20}$, proposed by Campos *et al.* (2021) [29] and interface with all relevant information, either low-level (e.g. raw sensor data) or high-level (e.g. mission states) through the data logger skill, $\mathscr{S}_{21}$. It runs in the perception processing units due to its higher computational power available. This task can also execute the skills to increase the situational awareness of the vehicle, such as multi-domain mapping [29], detection of docks [179], vessel detection and tracking [180], and obstacle detection [181].

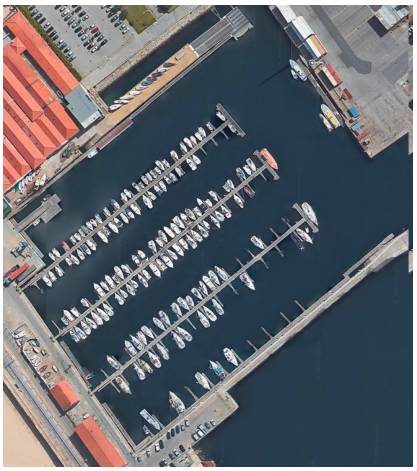
3.5 Results

To demonstrate the data acquisition capabilities and manoeuvring proficiency, a series of comprehensive tests were conducted on both USVs, spanning various conditions and locations. This section illustrates trials performed with the SENSE and Nautilus USVs, alongside a sample application of the MUSA software architecture for a positioning system, including corresponding tests employing the USVs.

Initially, the SENSE USV outcomes is showcased, with emphasis on the data obtained from the first publicly available multi-domain dataset, the ROAM@CRAS dataset[3] [32]. This demonstrates the USV's data acquisition and navigation capabilities. Subsequently, an evaluation of the Nautilus USV's deployability under distinct conditions, encompassing onshore and offshore scenarios, is presented, as well as its ability to operate effectively, even with low illumination conditions. Lastly, the MUSA framework is showcased through a positioning task conducted at the ATLANTIS test center, featuring both the SENSE and Nautilus USVs.

3.5.1 SENSE

The heterogeneous payload available on the SENSE provides information of the vehicle state and of both domains in the maritime environment, namely surface and underwater. As such, distinct modalities, information, rates and noise affect the available data. Two field tests were performed with SENSE on the biggest port for yachting on the North of Portugal, namely the *Marina de Leça* harbor depicted in figure 3.15. It has 240 mooring places available on site and has a depth of 4 m



(a) Satellite view (41°11'13.8"N 8°42'18.3"W)

(b) Dataset acquisition conditions.

Figure 3.15: *Marina de Leça* harbor - 240 mooring places with entry and yacht basins depth of 4 m and 2-3.5 m, respectively.

[3]INESC TEC, 2023. *ROAM@CRAS - A haRbor multidOmAin Mapping dataset*. URL:https://rdm.inesctec.pt/dataset/nis-2020-002

at the entry basin and 2-3.5 m at the yacht basin[4]. On the first test the minimal configuration was validated and on the second the full architecture was used. With the minimal configuration no data was collected only the navigation was tested, while part of the second test resulted on the data collection displayed on the ROAM@CRAS dataset.

Images of the two field tests performed with SENSE and a preview of the dataset can be found in the video available on footnote[5]. All the presented data on this section were retrieved from the ROAM@CRAS dataset.

Currently, the availability of datasets with multi-domain data in the literature is very limited and confidential. This lack of datasets along with the complexity of developing and deploying USVs with multi-domain perception makes it more difficult to explore several open challenges for new situation awareness methodologies. Thus, the SENSE was used to gather the first publicly available dataset at the biggest yachting harbor in the north of Portugal, the ROAM@CRAS. It not only provides visual and 3D range data from the surface, 2D range underwater data, and navigation data from GPS and IMU, as also supplies an accurate positional ground-truth with centimetric precision. The available data illustrates the environmental challenges on the harbor use case scenario, from the weather and tidal effects on the vessel to the impact of crossings with larger vehicles and approaching big constructions. The ROAM@CRAS enables the exploration of new state-of-the-art methods in the research areas of sensor fusion, multi-domain mapping and perception-based collision avoidance techniques, while in the presence of intermittent data, interactions with other vessels, static and dynamic features, and distinct navigational challenges from the harbor operations.

The ROAM@CRAS presents the result of a mission performed during the field tests of SENSE at the harbor with the full modules and sensor payload presented. A tele-operated route is performed on this harbor and provides data obtained by several sensors allowing the acquisition of navigation information through GPS and IMU from the localization module, plus perception data of the two domains with a 3D LiDAR and a stereo camera (surface) and a MBES (underwater).

This dataset was acquired on a partially cloudy day, figure 3.15b, with slow wind speeds (4km/h North) and good illumination circumstances. As depicted in figure 3.15a from Google Maps[6], the test site presents heterogeneous scenarios with several dock types (e.g. U-Shaped and I-Shaped), static features (e.g. bridge columns), dynamic features (e.g. docks with distinct vessels and states), open areas with few features and various depth conditions. The dataset provides a comprehensive ensemble of conditions that illustrates a realistic set of typical events on a fully functional harbor namely:

- The SENSE docking and undocking;
- Crossings with other vessels;

[4]Marina Porto Atlântico, 2023. *Marina Porto Atlântico*. URL: `https://www.marinaportoatlantico.net/en`.

[5]Daniel Campos, 2023. *SENSE - The Autonomous Vessel for Multi-domain Inspection and Maintenance*. URL:`https://youtu.be/vkJ-MVCSpdA`.

[6]Google Maps, 2023. *Marina de Leça*. URL: `https://goo.gl/maps/CAXmn8Qg921EUJKs6`.

- Observation of distinct types of docks and vessels;
- Perception of the two dock states, occupied and free.

The ROAM@CRAS is ROS-compliant and all sensor data collected from scenarios is stored in bag files. ROS is a well adopted technology, mainly in the fields of mobile and industrial robotics, and preserves the output rate of sensors, saves the timestamp of the data and describes the extrinsic relations between sensors, being ready to use online and offline. A public repository from INESC TECs' CRAS with the dataset is available at `https://rdm.inesctec.pt/dataset/nis-2020-002` [32].

The data collection was acquired through the remote operation by a trained driver on board of a crewed support vessel. It reached velocities up to $1.2\,\mathrm{m\,s^{-1}}$ and stored the data from all on-board sensors and some internal states of the vehicle. The operation time span was roughly 20 min ($\sim$1196 s) stored in multiple 14 s patches. The mission starts with the USV moored for a few seconds then it launches through the canal. The SENSE observes distinct harbor features, namely multiple types of docks, mainly U-shaped and I-shaped, with occupation states that changes throughout time, several vessels either berthed or moving, and fixed structures, such as the bridge with column foundations at the southeast side and the breakwater surrounding the harbor. At the end, the SENSE returned to the initial docking area. The trajectory performed by the SENSE during the acquisition of the ROAM@CRAS dataset is depicted in figure 3.16, where

Figure 3.16: Trajectory performed by the SENSE on *Marina de Leça*. The green line is the RTK observations and the red arrow points to the base station position.

the green line is the RTK observations and the red arrow points to the base station position on the harbor[7].

Along the route distinct situations are covered trying to maximize the generalization of common challenges on harbor scenarios and to test the observability limitations, namely:

- The SENSE presents variable velocities from halt periods, to ensure the safety in intersection situations, to faster velocities;
- Crossings with other vessels operating in the harbor, that results on halt situations;
- Steeper waves caused by larger vehicles in transit;
- Temporary data occlusion by dynamic elements;
- Navigation in close range to metallic structures with the possibility of noisier sensor estimations by the navigation sensors;
- Acquisition of close-range data with more detailed measurements from the perception sensors;
- Limited the MBES field-of-view range to 5 m to create data incompleteness situations. For instance, at the entry basin few or even no observations are made in multiple scans.

Thus, a comprehensive set of occurrences are illustrated throughout the full operation, that provides several challenges for applications, such as mapping, feature extraction, structure classification and obstacle detection. The data gathered from all the sensors on SENSE are available in the ROAM@CRAS dataset and can be subscribed through the ROS topics listed on table A.2, where *VK-GPS* is the Vk-162 GPS, *SN-GPS* is the Swift Navigation GPS, *R9-IMU* is the Razor 9DOF IMU and *XS-IMU* is the Xsens IMU.

The sensor payload extrinsic calibration and synchronization was represented using ROS available tools for transform and timestamp the data. The localization module sensors, the Velodyne and the Mynt Eye were calibrate and synchronized following the same procedures as in a previous work by Gaspar et al. [182] for acquiring a dataset on an urban scenario. The extrinsic calibrations for all sensors are available on the dataset repository[8]. It represents the physical displacements and orientations of each sensor in relation to the coordinate system of the base frame, previously presented in figure 3.2. The stereoscopic system, Mynt Eye D, was calibrated according to Wang et al. [183], using a chessboard to calculate the extrinsic parameters from corner extraction. The intrinsic parameters of the LiDAR are provided by the vendor[9]. To acquire the relation between the cameras and the LiDAR coordinate frames it was used the single value decomposition of a set of 3D correspondences from the work of Dhall et al. [184]. Finally, the IMUs and the GPSs frames were manually measured in relation to a reference coordinate system.

[7] Base station positioned on the coordinates 41°11'13.7"N 8°42'18.4"W.

[8] Daniel Campos, 2023. *ROAM@CRAS*. URL:`https://github.com/danielfbcampos/ROAM-CRAS`

[9] VeloView-VLP-16-HiRes.xml on December 29, 2016

The synchronization of sensors can be verified through the timestamp information provided in the ROS messages. This timestamp returns the moment that data is acquired by the middleware for each of the installed sensors which presents asynchronous output rates.

Navigation sensors: The navigation sensors available in the SENSE are composed by two GPSs (SN-GPS and VK-GPS) and two IMUs (XS-IMU and R9-IMU) that provide redundant data with distinct accuracies and rates.

The SN-GPS is used to acquire both RTK and GPS data. This sensor publishes the RTK data at a rate of 10 Hz with a centimetrical precision that can be used as ground-truth. As for the GPS data, the rate is approximately 8 Hz. Since both positional data are given by the same sensor, almost no difference is observed between the GPS and RTK measures, as depicted in figure 3.17, where the GPS readings, blue line, overlaps with the ground-truth, green line (covered by the GPS trajectory). This overlap is also given by the scale factor since the error is within centimetrical precision, while the travel distance is approximately 620 m. As for the VK-162 GPS, the data is acquired at a rate of 4 Hz presenting a lower accuracy in comparison to the SN-GPS. As illustrated in figure 3.17, where the purple is the VK-GPS, the data presents major drifts in relation to the RTK. Since no external antenna is available for this sensor higher signal attenuation affects the data while being more susceptible to noise from the surrounding structures.

The XS-IMU produces high frequency data at approximately 200 Hz with smooth orientation evolution through time and responsive angular velocities and linear accelerations readings. On the

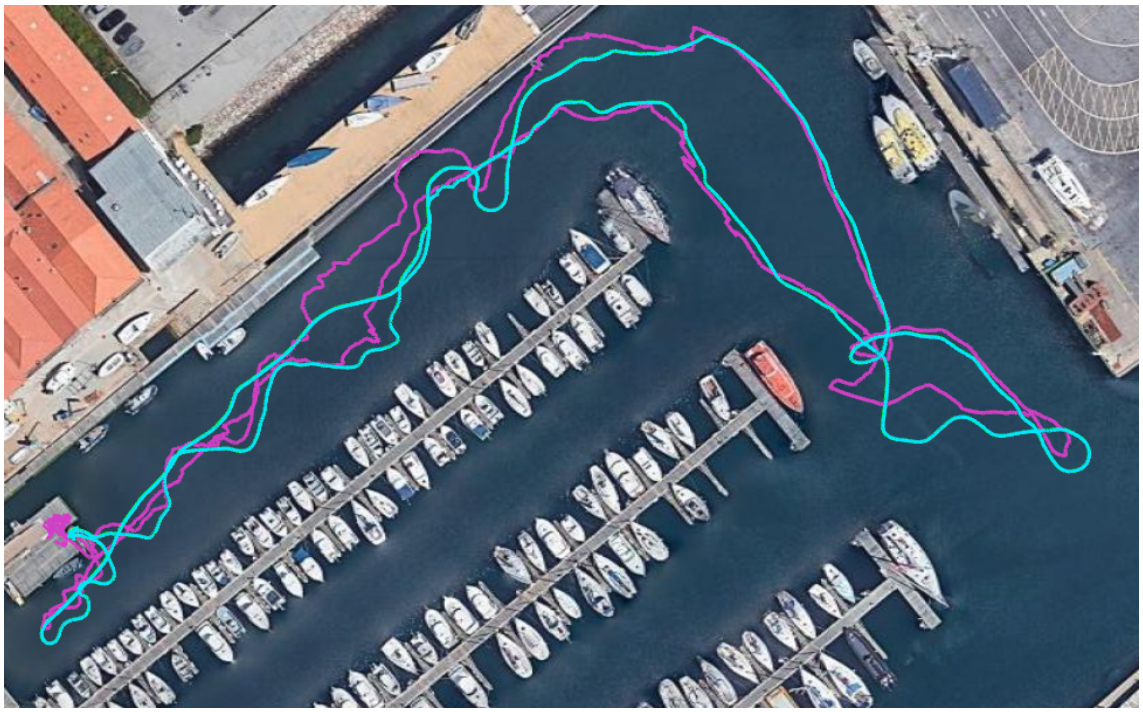
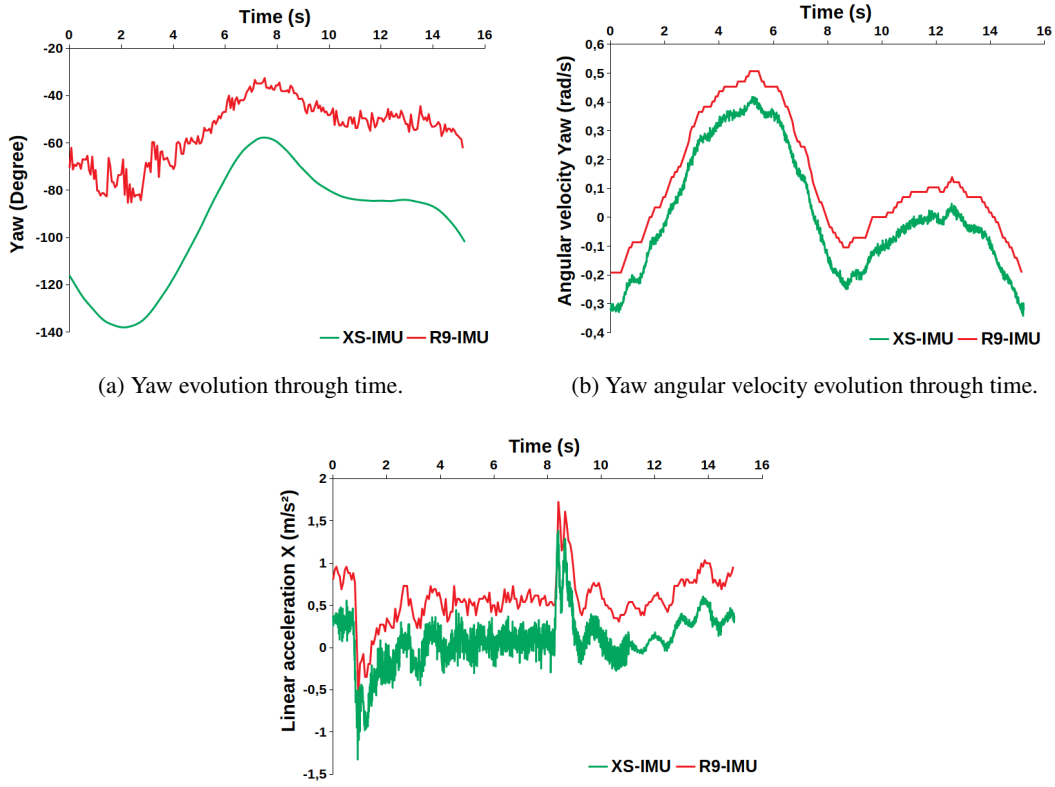


Figure 3.17: Comparison between GPS modules and the RTK ground truth. The green, blue and purple lines are the RTK, SN-GPS and VK-GPS data, respectively. The error scale factor for the RTK and the SN-GPS is minimal, thus the lines are overlapped.

other end, the R9-IMU provides a lower rate, around 15 Hz with noisier orientation data and less sensitive angular velocities and linear accelerations.

Two distinct situations where selected to be illustrative of the qualitative difference of performance for both IMUs. In the first, the SENSE rotates in yaw with almost no linear velocity. In figure 3.18a and figure 3.18b are depicted the yaw, in degrees, and the corresponding angular velocity, in rad s^{-1}, acquired by both IMUs. The orientation given by the XS-IMU, green line, provides a smoother yaw evolution than the R9-IMU, red line. Thus, the Xsens IMU's performance aligns more closely with the vehicle's kinematic limitations, preventing the occurrence of implausible large angular variations within short periods of time, whereas the Razor IMU exhibits such discrepancies. Examining the angular velocities provided by the gyroscopes of each IMU reveals a similar overall trend, albeit with minor deviations in magnitude. The Xsens IMU's readings exhibit superior responsiveness, capable of detecting finer angular variations due to its acquisition rate, which is over ten times faster than that of the Razor IMU.

Finally, when the SENSE halts its movement and then resumes the mission, a distinct variation in the vehicle's linear acceleration along the X-axis is observed. This variation is consistent with the change in linear velocity vector imposed by the differential drive. Both IMUs effectively capture these variations, as shown in figure 3.18c. The SENSE stops around the first second and



(a) Yaw evolution through time.

(b) Yaw angular velocity evolution through time.

(c) Linear acceleration in the X-axis evolution through time.

Figure 3.18: Data sample of the available IMUs.

resumes motion shortly after the eighth second. The data provided by both sensors accurately represent these two distinct moments. However, similar to the angular velocity measurements, the Xsens IMU demonstrates superior responsiveness, enabling it to detect finer variations in the acceleration signal with less discretization. Additionally, a small discrepancy in magnitude is observed between the readings of the two IMUs.

Perception sensors: The ROAM@CRAS dataset offers an insight towards the data perceived by the SENSE on the harbor. To illustrate the data gathered by the perception sensors, namely the stereo camera, LiDAR and MBES, three snapshots were selected to highlight some scenario features. Figure 3.19 depicts the position for each one of them, where the flags with distinct color placed on the ground-truth trajectory represents the locations.

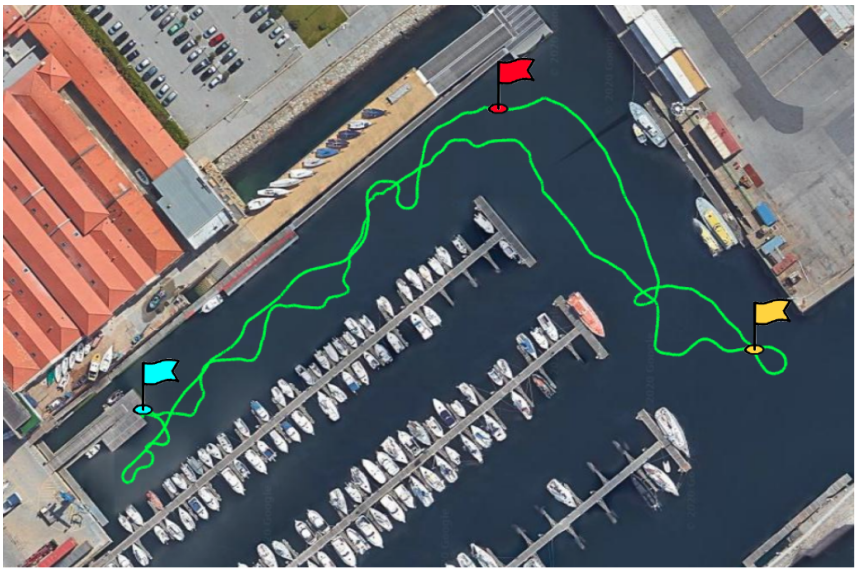
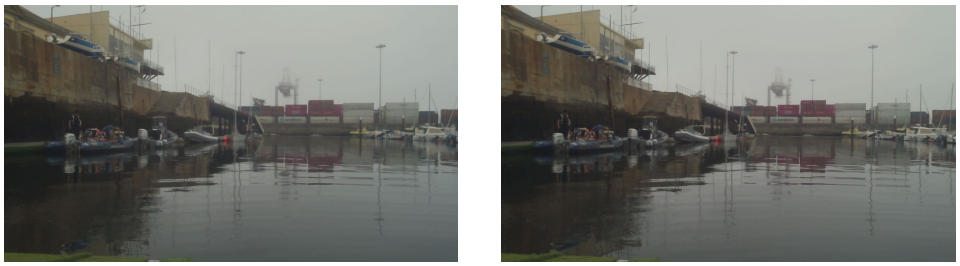


Figure 3.19: Locations of the example snapshots. Each flag corresponds to data acquisition instants.

The stereo camera system, Mynt Eye D, provides visual information from the scenario. The images retrieved by the Mynt Eye D have a resolution of $1280x720$ published with a rate of approximately 20 fps. Since the sensor has a short baseline, it generates a high overlap between the left and right cameras, as shown in figure 3.20. As for the data collected throughout the mission, it portrays different challenges with both dynamic and static observations, during which the camera is affected by noise, varying lighting conditions and by the vehicle motion, causing motion blur.

In figure 3.21 are depicted the images from the right camera retrieved on the locations flagged on figure 3.19. At the first image, blue flag, the SENSE is docked. As it can be seen, there are several moored vessels and the yacht basin is clear. The second one, yellow flag, was gathered near the entry basin, where larger vehicles are moored and shipping containers from the Port of Leixões are on the line-of-sight. At last, the third frame, red flag, exhibits one static infrastructure available in the scenario, which can provide features that can be used as feedback to test visual inspection techniques or other perception based methodologies, such as visual odometry.



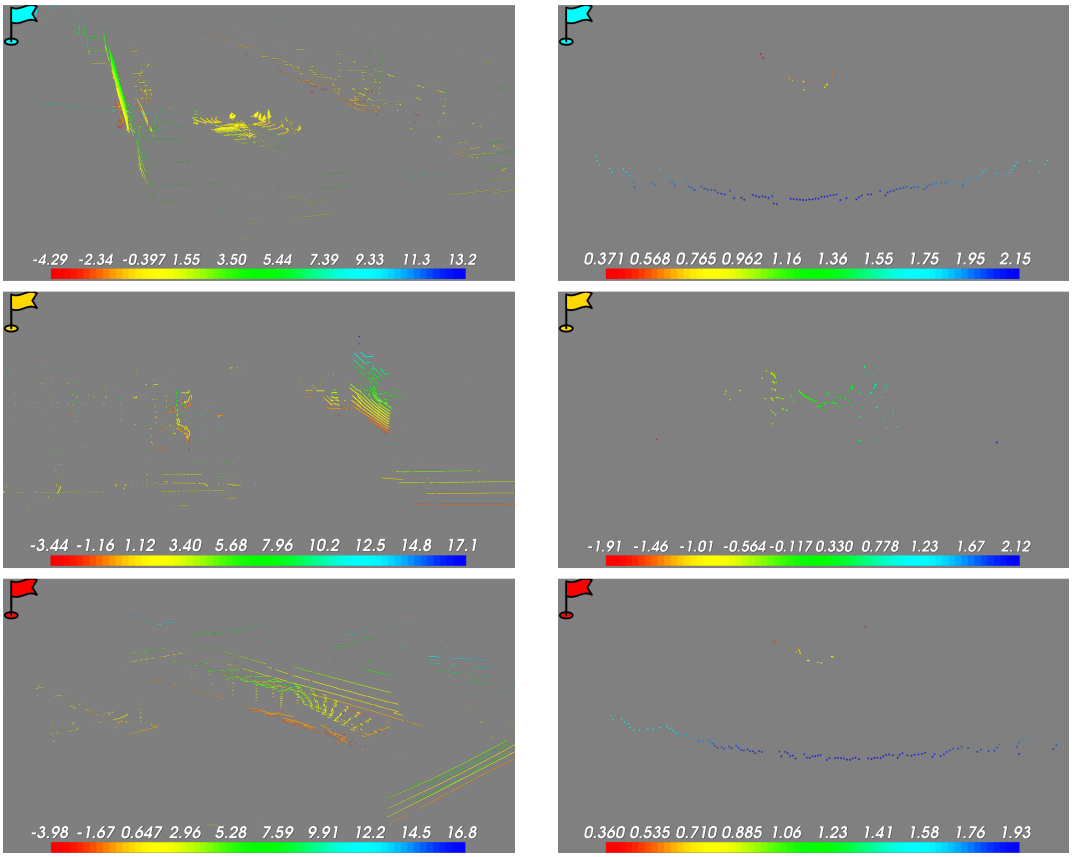
(a) Left camera image. (b) Right camera image.

Figure 3.20: SENSE stereo camera sensor.

Figure 3.21: Camera data acquired in each location flagged in Figure 3.19.

Another available sensor is the LiDAR, Velodyne VLP-16, that provides 3D distance information of the surroundings with a frequency of 10 Hz and a range up to 120 m depending on the material reflectivity. It has a FoV of 330° horizontal, due to an occlusion resulting from the antennas mounting system, and 30° vertical. This sensor publishes high accuracy point clouds, where the higher the distances the sparser data becomes. Moreover, the sensor wave length is 905 nm, as such it can only penetrate 1 cm of the water surface providing no reflections, thus the wave impact will translate into missing data instead of spurious data [30]. In figure 3.22a are depicted some examples of the surface point clouds from the SENSE. In the first snapshot it is possible to visualize the harbor facilities, such as piers, access ramps and stairs, several moored vessels, as well as people on the piers and on the support vessel. As seen, the data is denser near to the sensor itself, becoming sparser at further distances. This offers a challenge, since in some cases, when the SENSE navigates in wider areas, such as in the second snapshot, it will have less visible features to be used or even incomplete data. This is verified specially during operations in offshore locations where the scenarios are mostly open sea. For this, the use of a RADAR would be better to increase the range, mainly for bigger ships to detect objects earlier and increase the safety of the navigation. Nevertheless, during missions with several man-made structures that provide an elevated number of features, as well as constructions to inspect, the LiDAR provide accurate metric information for inspection purposes. The third snapshot is one example, where the bridge elements are visible, namely the pillars, the beams, the platform and the lower level piers. Also, throughout the dataset several docking spaces are depicted.

At last, the available multibeam echosounder produces a 2D profile of the sea-bottom. The sensor is configured to publish data from 120 beams with a maximum range of 5 m at a rate of approximately 12 Hz. The selected beamforming method used on the Imagenex sensor detects the



(a) LiDAR data acquired in each location flagged. The scale illustrates the Z-axis (height) range in relation to the sensor.

(b) MBES data acquired in each location flagged. The scale illustrates the X-axis (depth) range in relation to the sensor.

Figure 3.22: LiDAR and MBES data acquired in each location flagged in Figure 3.19.

first echoes returned by each beam. Since the MBES is pointing downwards and the tidal waves were almost flat, the beams in the centre will correspond roughly to the depth of the seabed in relation to the sensor, which is placed 0.4 m bellow the water level. The beamforming results at each snapshot location is illustrated in figure 3.22b. The first and last images captures the seabed near the yacht basin. The data acquired provides readings of 2.15 m and 1.93 m in each location to add up to the 0.4 m of the sensor placement below the water level, which is consistent with the information detailed by the harbor, where the yacht basin has a depth of about 2-3.5 m. As for the MBES scan in the yellow flag instant, the seabed depth is above the maximum range, where the available measurements are spurious measures given by erroneous reflections during beamforming. This will produce missing data during the SENSE passage through the entry basin, where the depth is higher than the expected 4 m.

The ROAM@CRAS provides multi-modal data that enables the exploration of new state-of-the-art methods in the research areas of sensor fusion, multi-domain mapping and perception-based collision avoidance techniques, while in the presence of intermittent data, interactions with other vessels, static and dynamic features, and distinct navigational challenges from the har-

bor operations. Some potential applications are presented bellow, namely odometry estimation, bathymetry, and multi-domain mapping.

Odometry estimation: Odometry provides useful navigational information that can be used to track the vehicle position evolution throughout time. It can be obtained from the navigation sensors, for instance using the method presented in the navigation module section, or from the perception sensors, namely the cameras and the LiDAR, e.g. the work proposed by Forster et al. [185], Teixeira et al. [186], Skjellaug et al. [187]. Since the SENSE is operating in areas with GPS availability the application of the GPS and IMU fusion method presented in the navigation module section provides an accurate odometry. As seen in figure 3.23, it produces a similar trajectory to the available RTK ground-truth. Using the metric proposed by Zhang and Scaramuzza [188], this odometry estimation trajectory produces an average RMSE of 0.099 m and a standard deviation of 0.069 m for a travelled distance around 620 m.

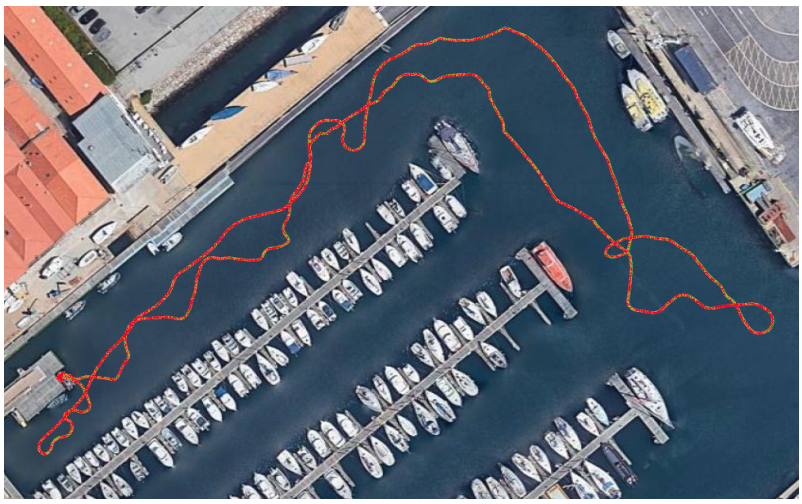
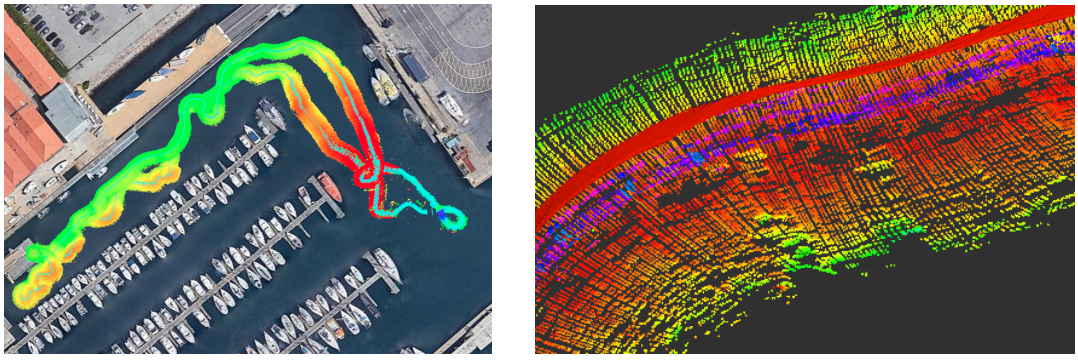


Figure 3.23: Odometry acquired through the GPS and IMU fusion. The green and red lines are the RTK observations and the GPS/IMU odometry, respectively.

Bathymetry: The data provided in the ROAM@CRAS dataset can also be used to develop and validate bathymetry methodologies. This can be applied to evaluate and explore the underwater domain. Moreover, it provides crucial information for infrastructure inspections, namely for representing the scour condition on offshore wind turbines or bridges foundations, and for depth estimation. As an example, a simple bathymetry given by the association between the GPS and IMU odometry, acquired from the localization module, and the MBES scans without any filter is shown in figure 3.24a. This georeferenced MBES profiles were concatenated through time. In the performed trajectory it is possible to observe the depth increase as SENSE approaches the entry basin, where the data becomes unavailable, only noise readings were concatenated. In figure 3.24b a close-up of the bathymetry is depicted, where the seabed is concave with the deepest part in the centre of the observed route. Moreover, some artefacts are visible, for instance the isolated green measures on the red plane, which can be stones placed on the ground or the accumulation of sediments due to the currents.



(a) SENSE bathymetry acquisition.

(b) Close-up of the SENSE bathymetry.

Figure 3.24: SENSE bathymetry map. The rainbow presented goes from red (higher depths) to blue (lower depths). The green corresponds to intermediate values.

Multi-domain mapping: The creation of multi-domain representations of the maritime environment can increase the situational awareness for USVs by depicting both the surface and underwater regions on a single map. As such, by applying the method proposed by Campos et al. [30] to a partial sequence of the ROAM@CRAS it was possible to create a multi-domain map of the harbor. This approach uses the SENSE localization estimation to provide the initial seed of the system state. Then the data acquired through the LiDAR passes through an outlier filter, while the MBES data is georeferenced and concatenated through a predetermined time providing a sliding window that is composed partially by old data, as well as new scans to create an overlap region. The filtered data from both sensors are registered using an ICP-based approach and concatenated to a single representation. In figure 3.25 is depicted the resultant 3D map. From the representation it is possible to observe several moored vessels, as well as the pillars, ramps and walls from the harbor infrastructures. Moreover, the underwater domain map shows the seabed with depths from 1.9-3.0 m, which are in accordance with the official harbor depth rates for the yacht basin.

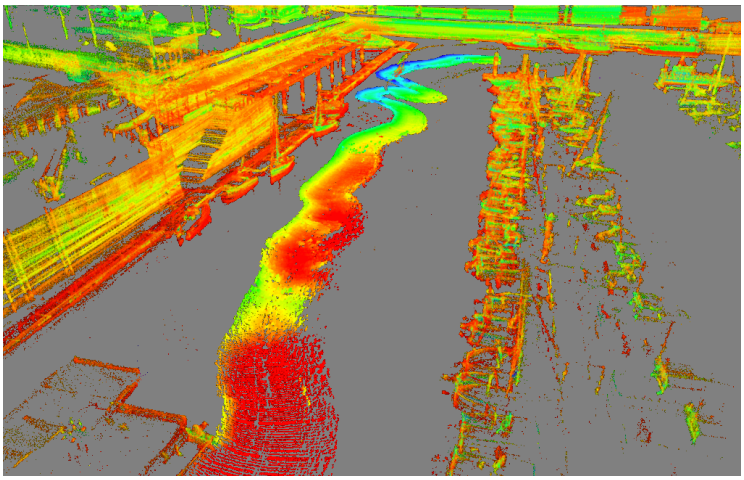


Figure 3.25: Multi-domain map on a partial sequence of the ROAM@CRAS.

3.5.2 Nautilus

The validation of Nautilus was conducted in accordance with the ATLANTIS technology certification procedures, figure 3.26. This rigorous testing framework provides a systematic approach that considers not only the robotic system's characteristics and testing objectives (defined by the Test Centre users or end-users), but also the specific task and associated performance indicators. This comprehensive approach ensures that testing results meet the highest standards of reliability, efficiency, and safety [17]. The ATLANTIS testing process is specifically designed for entities developing technologies or services related to offshore wind. It aims to validate the demonstrated technology or methodology, ensuring its compliance with existing safety regulations and its viability as a performance-enhancing alternative to traditional methods. The ATLANTIS testing methodology comprises two distinct stages closely linked to the TRL (Technology Readiness Level) of the robotic solution being tested. The first stage, Approval in Principle, involves validations conducted exclusively on the Coastal Testbed. The second stage, Technology Qualification, requires validations on the Offshore Testbed. Obtaining Technology Qualification necessitates the endorsement of external entities (e.g., asset owners). Prior to proceeding to the Offshore Testbed, a critical step involves validating the technology's deployment and recovery methodologies in open sea.

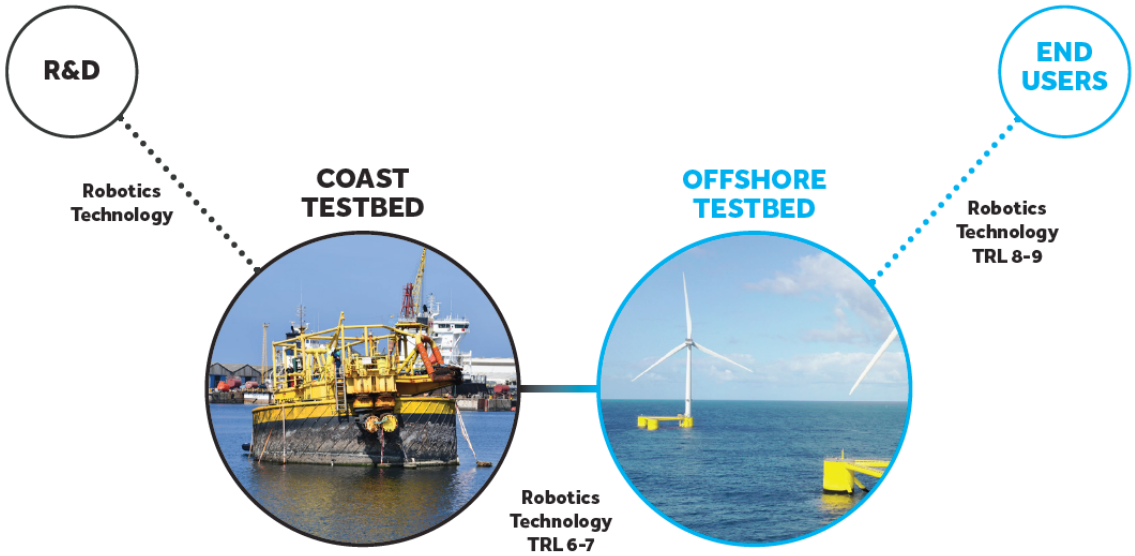


Figure 3.26: ATLANTIS stream line for the validation and demonstration of robotic solutions.

The Nautilus successfully completed all testing stages, engaging in four trials at the sites depicted in figure 3.27, specifically, the ATLANTIS Test Centre and Offshore Trials 1, 2, and 3, each with varying distances to the shoreline. While TRLs were lower (up to 5), the USV's performance underwent repeated testing at the ATLANTIS coastal test centre [36] situated on the port of Viana do Castelo, Portugal. During the Approval in Principle stage, first trial at the ATLANTIS Test Centre, Nautilus reached TRL 6 showcasing its ability to operate in a relevant scenario. During the first offshore trial the deployment methodology was validated in open waters under moderate

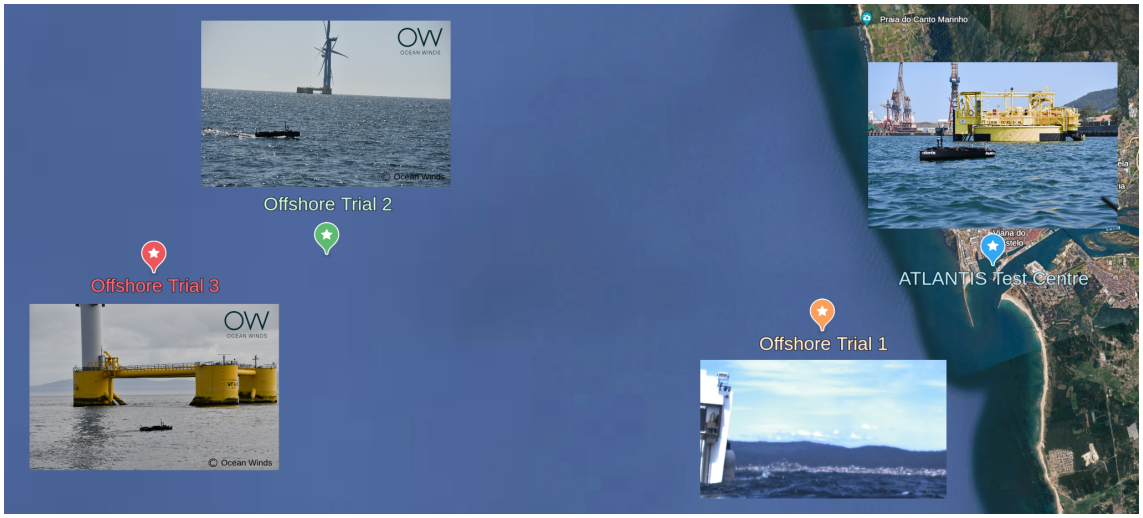


Figure 3.27: Nautilus trial locations in Satellite view.

conditions. For the Technology Qualification stage, two offshore trials were conducted in collaboration with the WindFloat Atlantic (WFA) owners, where Nautilus reached TRL 8. Offshore trial 2 demonstrated deployment near WFA farm, while offshore trial 3 performed an inspection mission on a WindFloat foundation. According to available information, Nautilus became the world's first USV to perform autonomous tasks on a floating foundation offshore wind farm in the Atlantic. This final trial was conducted at a distance of 18 km from the shoreline, on the Windfloat Atlantic wind farm, encountered an estimated H_s during the operation of 1.1 m, and wind velocities averaging $5\,\mathrm{m\,s^{-1}}$.

The offshore operations happened according to strict safety protocols, specially, related to weather and sea-state conditions: up to 1.5 m of H_s and $10\,\mathrm{m\,s^{-1}}$ of wind speed [38]. All experiments of Nautilus followed a state-of-the-art testing procedure at ATLANTIS Test Centre that considers all these parameters and conditions imposed by the industry.

The four trials, namely ATLANTIS Test Centre, Offshore Trial 1, Offshore Trial 2 and Offshore Trial 3, presented varying wind and sea states, and distance to the shoreline, as summarized in table 3.3. As can be seen, the validation of Nautilus occurred in real world conditions close to the maximum level of sea-state and wind speed allowed by the safety standards followed by the industry.

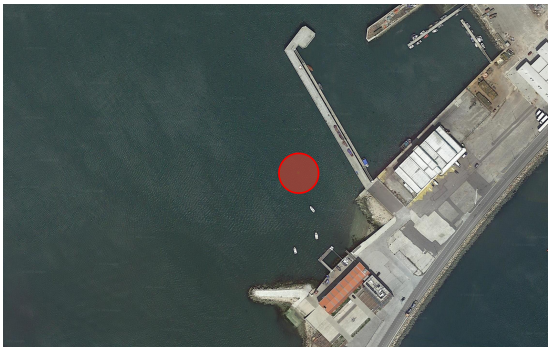
Table 3.3: Summary of the conditions during Nautilus trials.

Trial Locations (Figure 3.27)	**Distance from Shore (km)**	**Douglas Sea State**	**Significant Wave Height (m)**	**Wind Speed (ms^{-1})**
ATLANTIS Test Centre	-	2	0.2	2 - 10
Offshore Trial 1	3	4	1.3	10
Offshore Trial 2	14	3	1	7
Offshore Trial 3	18	3	1.1	5

Nautilus design doesn't include a keel in order to facilitate the deployment of this USV at the sea, and although, it is not expected to operate Nautilus under more challenging weather and sea-state conditions than the industry safety standard, it is believed that it would be possible to operate Nautilus up to 1.6 m Hs and wind speed up to 15 $m\,s^{-1}$.

3.5.2.1 Approval in Principle: Near-real Environment Validation

The ATLANTIS coastal test bed [36] situated on the port of *Viana do Castelo*, Portugal, depicted in figure 3.28, served as the venue for all onshore trials throughout the development process and for the Approval in Principle stage. This site comprises the port's infrastructure and the Durius, an offshore floating structure repurposed from the Oil & Gas industry to emulate the floating foundation of the offshore wind energy industry. A video with snippets from some Nautilus trials in this scenario can be found on the link in footnote[10].



(a) ATLANTIS coastal test bed satellite view. The red circle indicates the floating structure, Durius, approximate location.

(b) ATLANTIS coastal test bed shore view.

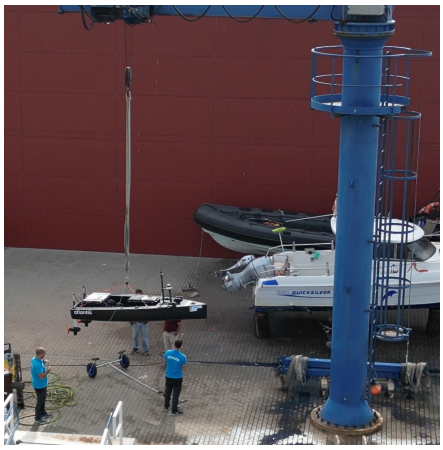
Figure 3.28: The ATLANTIS coastal test centre satellite view and shore view.

Nautilus is a portable and deployable system, owing to its compact design. It can be transported on a standard cargo van with a storage space length of at least 3.1 m, a dimension commonly found in vehicles with storage volumes of 10 m^3 or more. During transit, a transportation cart is employed, as depicted in figure 3.29, with the outboard motors disassembled for ease of handling. The hardware modules can also be detached from their fixation frames, further reducing the overall weight and simplifying transportation tasks for the operator.

At the designated deployment site, the various components and connectors are meticulously assembled and securely fastened, and the power switch is activated to initiate the preliminary safety assessments. The deployment procedure involves the employment of either a crane or an A-frame mechanism, both equipped with a rapid-release snap hook. This snap hook is attached to a 4-point sling that is connected to Nautilus' ring bolts. This arrangement facilitates the lifting and subsequent controlled descent of Nautilus onto the water's surface, as depicted in figure 3.30. The inclusion of the quick-release system eliminates the necessity for a support vessel during the

[10]MOTION Robotics, 2022. *Nautilus USV*. URL:`https://youtu.be/12niamOnmL8`

Figure 3.29: Nautilus on the transportation cart



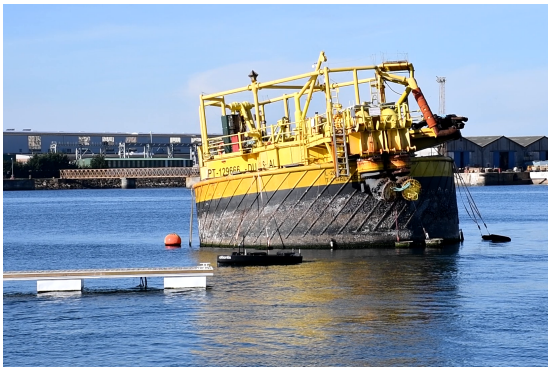
(a) Nautilus deploy through the ports' crane.

(b) Nautilus deploy system close-up.

Figure 3.30: Nautilus deployment on port and offshore.

deployment of the USV. For the recovery phase, a telescopic boat hook complete with a supplementary carabiner support is employed. This apparatus serves to securely attach the winch cable to the previously mentioned sling.

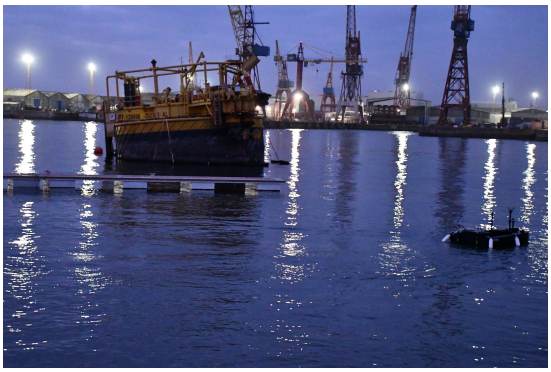
During the trials on the ATLANTIS test centre, Nautilus was subjected to distinct weather and light conditions to validate the navigational capabilities and the data integrity. Figure 3.31 depicts three trials under different operational scenarios: clear sky, clouded and night. Nautilus during these trials presented a swift manoeuvrability allowing to perform quick heading and speed variations due to the hull shape and the differential drive. Despite of the absence of environmental waves in *Viana do Castelo* port, it is a high traffic area for vessels, as such the wake caused by their displacement can cause several ripples with high frequency and heights dependant on the size and velocity of the vehicle, usually with a H_s between 0.1 and 0.2 m. During these trials, Nautilus



(a) Nautilus trial with clear weather.

(b) Nautilus trial with clouded weather.

(c) Nautilus trial at night.



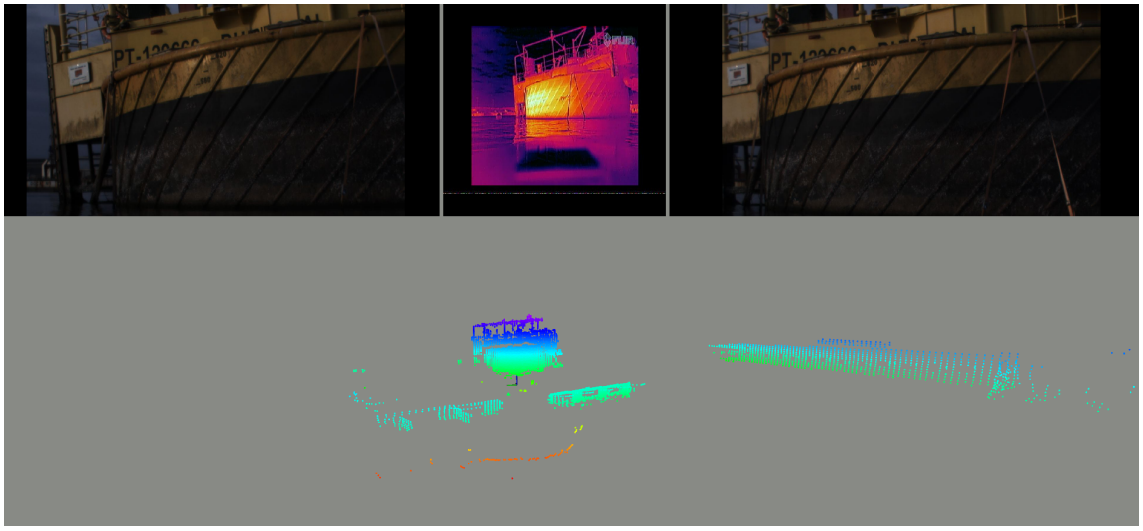
(d) Nautilus trial at night environment conditions.

Figure 3.31: Nautilus trials in the ATLANTIS coastal test centre.

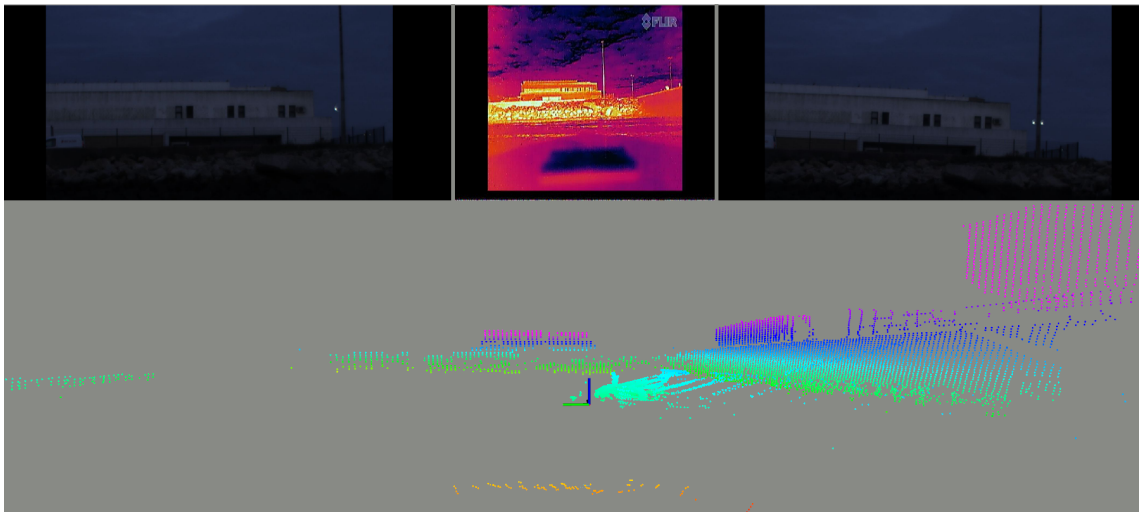
was affected by this phenomena showing a high capacity to endure the wake swells, maintaining its structural integrity and safety.

The major effects of the different operational scenarios are noticed on the data acquisition and processing capabilities. Nautilus has a heterogeneous payload capable of acquiring information about the vehicle state and both domains in the maritime environment, namely surface and underwater, which are affected differently by the conditions. As seen in figure 3.8, Nautilus was able to acquire high quality data under clear weather. When under low lighting conditions, the visual cameras images are the most affected requiring higher exposure times, as it can be seen in figure 3.32. The use of the LiDAR and thermal camera are complementary to the visual information allowing to operate even at night, since the performance is not affected by the illumination conditions.

The results from the proposed MUSA architecture are presented and discussed. To provide a metric evaluation of the navigation task ($\mathscr{T}_1$) in autonomous mode ($\mathscr{S}_{11}$) it was used two different sequences as examples. First, the discussion focuses on a smaller and easier succession of goal-points, and then, a larger sequence, constrained to harsher environment conditions, such as stronger undulation and wind, is also analysed. All data was acquired under port conditions. For these experiments, we defined the following parameters:



(a) Nautilus data sample acquired on a cloudy day.



(b) Nautilus data sample acquired at night.

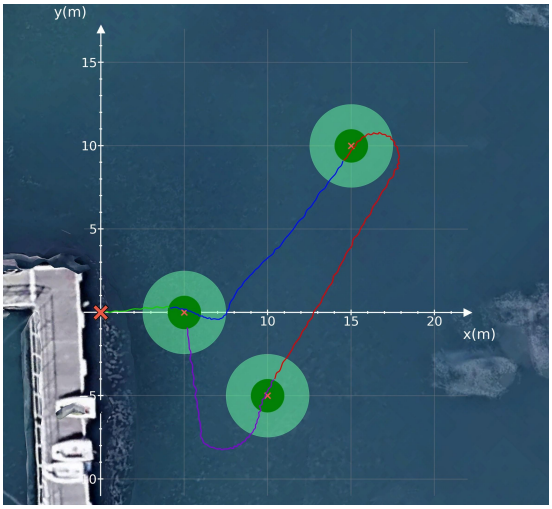
Figure 3.32: Nautilus data sample acquired on different weather conditions.

- A parameter which bounds the maximum $\|\Delta(x,y)\|_2$ from which the system advances directly to the final state. It was set to 2.5 m;
- The maximum allowed deviation in orientation was set to 0.1 rad;
- The maximum allowed deviation in position was set 1 m.

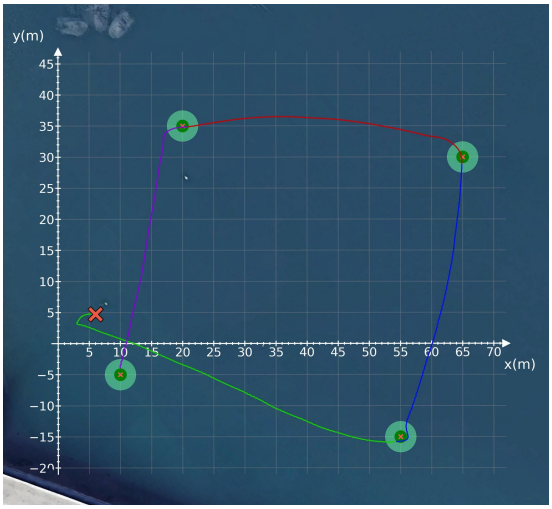
Pose information was acquired with the localization task $\mathscr{T}_0$. Initially, a local odometry frame was defined at the start of the operation. This frame was fixed to the geographic coordinates where the USV was initialised, hence, the goal-points were expressed in relation to the vehicle's starting pose, forming a set $P^O = \{p_1^O, ..., p_N^O\}$, with $p_n^O = (x_n^O, y_n^O), \forall n \in \{1, ..., N\}$, where '$O$' superscript represents the fixed odometry coordinate frame. The first experiment consisted of a sequence of the goal-points composed by the set $P^O = \{(5,0), (15,10), (10,-5), (5,0)\}$, and the path followed by

the USV is depicted in figure 3.33a. The movement of the vehicle was constrained to a small area, forcing it to rapidly adjust to any deviations in orientation when heading to any of the goal-points. The USV was initially placed at coordinates $(0,0)$ on the odometry fixed frame, and started its trajectory oriented to the first goal-point $(5,0)$, meaning the execution started directly on the cruising state, at maximum linear velocity. Every point, except for the first one, forced the USV to change its orientation, making sure that every stage of the continuous execution was performed. The USV was able to successfully execute the task, reaching the 1 m tolerance defined at the beginning of the experiment. A second experiment was conducted, simulating an inspection operation around the structure visible in figure 3.28b. As it can be seen in figure 3.33b, the scale of the movement was increased, easing constraints on the movement of the vehicle, but exposing it to harsher maritime conditions. The set of goal-points was defined as $P^O = \{(55,-15),(65,30),(20,35),(10,-5)\}$, chosen so that the USV trajectory could circumscribe the Durius floating structure. At the beginning, and contrarily to the first experiment, the vehicle was placed facing the opposite direction to the first goal-point, with coordinates $(6,5)$. As it rotated, the vehicle was being dragged by sea currents, as it can be observed in the first segment of the traced trajectory, represented by the green line in figure 3.33b. Despite the harsher conditions, the maximum acceptable margins of error were not changed. The task succeeded, meaning the USV reached every goal-point of the sequence with a maximum error lower than 1 m.

The use of Nautilus available sensor payload allows for multiple applications from a simple



(a) Task defined by the sequence of goal-points $P^O = \{(5,0),(15,10),(10,-5),(5,0)\}$



(b) Task defined by the sequence of goal-points $P^O = \{(55,-15),(65,30),(20,35),(10,-5)\}$

Figure 3.33: Results of task level experiments performed in a port environment conducted in Viana do Castelo, Portugal. Light green circles represent the distance from the goal-points from which the continuous execution state of the skill enters its last stage, while darker green circles represent the maximum allowed margin of error for the skill to succeed. Each line represents the execution of a single skill. The green line represents the first skill to be executed, followed by a blue, a red and a purple line. Initial points are marked by a bigger red cross.

visualisation tool to autonomous navigation and mapping. Figure 3.34 illustrates the multi-domain map of Durius' neighbourhood obtained with the skill $\mathscr{S}_{20}$, where the red to purple colour scale represents the Z-axis variation from minimum to maximum, respectively. For such, a 3D LiDAR odometry, the CT-ICP [189], was used to estimate the USV position given that it was operating in proximity to structures, which allows for a more accurate localisation and, then, using the method proposed by Campos *et al.* [30] to reconstruct the multi-domain map. As it can be seen, the seafloor representation is sparse (red and orange area). This is due to the water salinity variation inside the port caused by the mixture between fresh and salt waters that affects the MBES sensor. It causes the sensor to acquire multi-echoes and, consequently, spurious data which needs to be filtered. Additionally, the soft nature of the seabed further hinders the data acquisition process. To address the challenges of the water salinity and seafloor material, employing a MBES system with a lower operation frequency would provide better swath coverage and enhance the overall data quality[11]. The blue trace around Durius in the emersed domain is due to a support vessel moving near Nautilus during the mission execution, which was not filtered from the data.

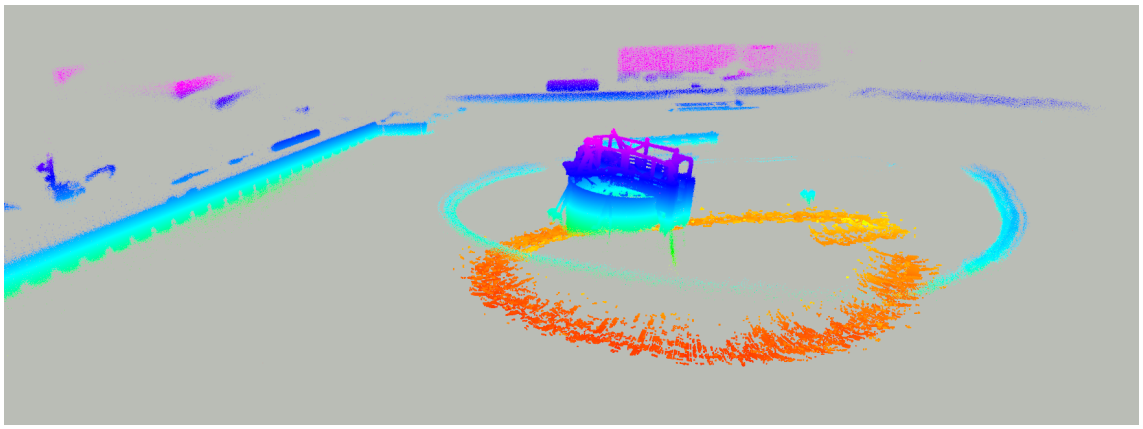


Figure 3.34: Multi-domain map created by Nautilus on the ATLANTIS coastal test centre with different tide conditions. The red to purple colour scale represents the Z-axis variation from minimum to maximum, respectively.

Another use case is the positioning system designed according to MUSA, which estimates Nautilus' position through the GPS/IMU fusion and performs a navigation skill to autonomously move from the current position to the mission waypoint. Thus, the vehicle trials allowed to validate the USV deployability and demonstrated its capacity for executing a vast range of applications due to the data acquisition capabilities and the structural integrity of the hull. Moreover, the design allows for a smooth manoeuvring capable of withstanding multiple environmental conditions. To further enhance bathymetric quality, employing a sonar with a lower or adjustable frequency would be beneficial, as it could diminish the influences of water salinity variations and soft seabed conditions, while also expanding the detection range.

[11] R2Sonic, 2023. *How to Best Use the Multibeam Echosounder Manual*. URL:`http://tinyurl.com/mbes-best-practice`

3.5.2.2 Technology Qualification: Real Environment Validation

Nautilus successfully completed three offshore trials, spanning varying distances from the shoreline and sea states, as depicted in figure 3.27 and summarized in table 3.3. The first trial was conducted at 3 km, followed by a second trial at 14 km, and the third and most relevant trial occurred at 18 km within the WindFloat Atlantic offshore wind farm. These trials provided a rigorous testing environment to validate and assess the USV's deployment capabilities and maneuverability under diverse conditions, fulfilling the requirements of the Technology Qualification stage of the ATLANTIS testing process.

The final trial was conducted directly on the WindFloat Atlantic wind farm, which comprises three fully operational floating foundations, with the total capacity of 25 MW, figure 3.35. The mission was carried out on the farthest turbine from shore, situated approximately 18 km from the coastline, in water depths reaching 100 m [12].

Figure 3.35: WindFloat Atlantic wind farm.

The deployment in offshore condition follows similar steps as the near-real environment. For safe operations in offshore conditions, stabilizing ropes are used being strategically positioned to mitigate the effects of undulation-induced motion and maintain the stability of Nautilus during its elevation. figure 3.36 depicts the deployment procedure of Nautilus adjacent an offshore wind farm located in *Viana do Castelo*, with operational conditions within the limits for deployment, namely 1.5 m or less significant wave height (H_s) for small boats [6].

[12] WindFloat Atlantic, 2023. *WindFloat Atlantic | Offshore wind energy*. URL: `https://www.windfloat-atlantic.com/~`

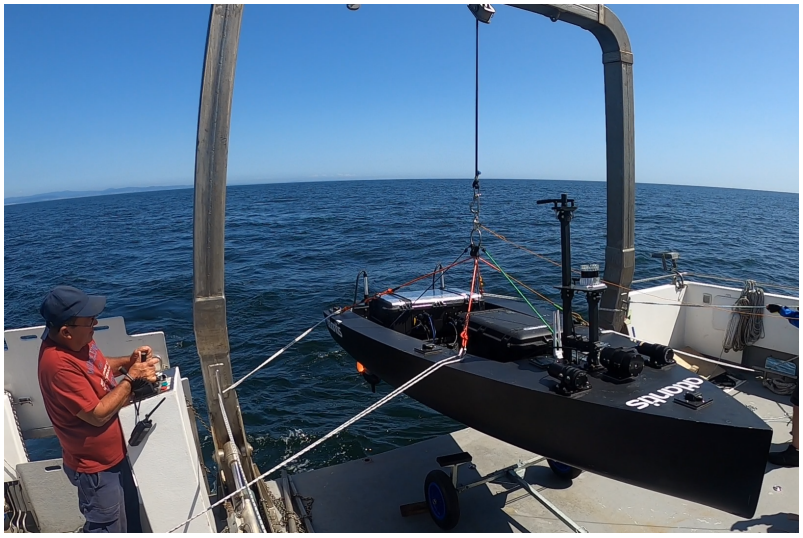


Figure 3.36: Nautilus deployment offshore.

The mission focused on evaluating the comprehensive procedure encompassing the USV's loading from port onto the transport vessel Mar Profundo[13], its on-site preparation, and the subsequent deployment and recovery protocol in open waters. The Nautilus deployment and recovery operation was effectively accomplished in three trials offshore. Figure 3.37 illustrates the USV operation during the second and third offshore trials. Moreover, a video of Nautilus operations portraying several applications of Nautilus demonstrated during the Approval in Principle stage, the deployment procedure and segments of the offshore trials during the Technology Qualification stage can be seen in the link on footnote[14].

During the offshore trials, the USV demonstrated structural integrity and safety, operating

(a) Nautilus operating in offshore during the second trial.

(b) Nautilus operating in offshore during the third trial.

Figure 3.37: Nautilus operations during the second and third offshore trials.

[13]INESC TEC, 2022. *Launching of a research ship to test marine technologies*. URL: `http://tinyurl.com/mar-profundo`.

[14]ATLANTIS Testing Platform for Maritime Robotics, 2023. *Nautilus USV*. URL:`http://tinyurl.com/nautilus-off`.

under significant wave height (H_s) conditions of up to 1.3 meters. Figure 3.38 presents a sample of the data gathered during the third trial, which is highlighted in figure 3.37b. As shown, the LiDAR, visual, and thermal cameras provided distinctive data from the floating foundation. However, the multibeam echosounder (MBES) returned highly sparse data with a low quantity of valid beams. This is due to the sensor's range limitation of 100 meters, which is insufficient to capture the seafloor, which is estimated to be deeper than 96 meters in the trial zone, according to the Portuguese Navy Hydrographic Institute charts[15].

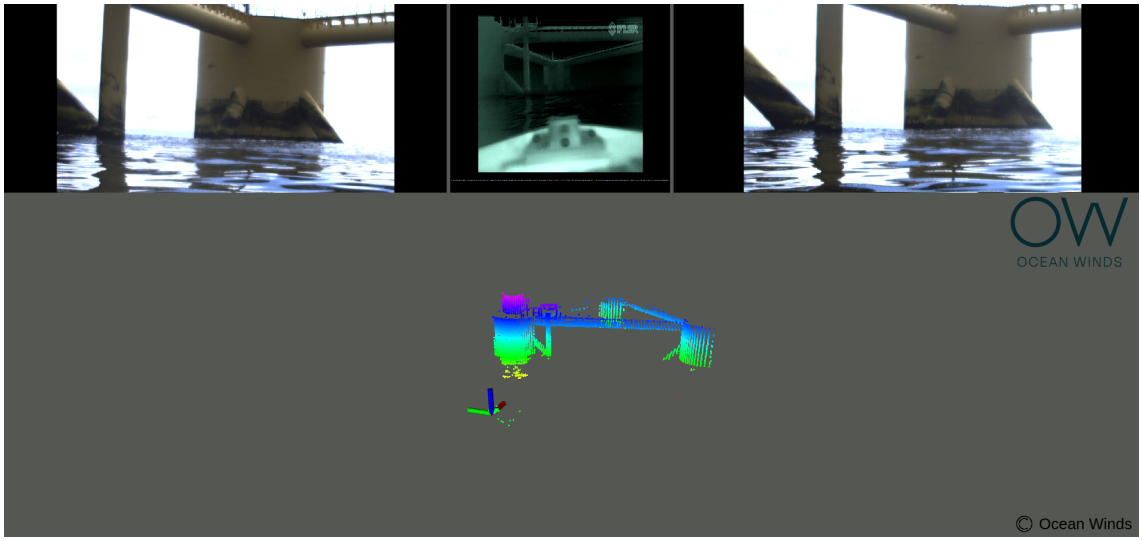


Figure 3.38: Nautilus data sample gathered during the third offshore trial on the Windfloat Atlantic wind farm.

3.6 Conclusions

This chapter introduces two innovative USVs designed to address multiple challenges in offshore inspections for wind farms. The SENSE and Nautilus platforms offer modular and flexible architectures, each with unique capabilities, contributing to advancements in multi-domain data acquisition, navigation, and perception.

The SENSE platform's modular design enables seamless transitions between configurations, where changes to the sensor payload, navigation systems, energy availability and localization systems can be made easily. This presents the advantage of providing a testing platform for data acquisition, validating different methodologies and explore new technologies that may emerge throughout time. The catamaran design allows to operate under calm and favourable maritime conditions and is easy to deploy. Thus, the SENSE can be used as a data acquisition vehicle as well as a platform for test and validation of low TRL methodologies.

On the other hand, the Nautilus USV maintains the modular architecture while demonstrating a higher operational capacity for deployment on offshore locations. The compact monohull ensures

[15]Hydrographic Institute, 2023. *Hidrográfico Mais*. URL:`http://tinyurl.com/geomarhidro`

an easy transportation and deployment, and withstands moderate sea states (State 4 in Douglas Sea State). Moreover, it is capable of performing beyond visual line-of-sight operations, both remote and autonomous, while broadcasting data to a Shore Control Centre. Nautilus was designed to operate within the safety standards of the O&M industry for ocean winds, namely the maximum significant weight of 1.5 m (Douglas sea state 4) and wind speeds of $10\,\mathrm{m\,s^{-1}}$.

Despite the distinct hardware and functionalities on both USVs, most of the software is interchangeable between them. Thus, MUSA, a multi-layer software architecture, ensures a standardised development structure for different types of vehicles based on four distinct layers: mission, task, skill and action. MUSA provides a flexible structure to describe a mission as a set of tasks with different skills performing several actions, as well as an unified software architecture for O&M operations. Its design fully describes the complexity levels for controlling the vehicle, going from the low-level system actuation to complex high-level configurations, namely from actions to mission.

Extensive field tests in the *Marina de Leça* harbor and the ATLANTIS coastal test bed demonstrated the potentialities of both USVs, and validated the navigation and data acquisition capabilities. This resulted in the creation of the first multi-domain public dataset acquired at biggest port for yachting on the north of Portugal, the ROAM@CRAS. This ROS-compliant dataset provides one complete mission in a fully functional harbor demonstrating several challenges for sensor fusion, tracking, mapping and obstacle detection. Moreover, Nautilus withstand three offshore tests that demonstrated its operational capacity to endure in harsh scenarios. As well, MUSA provided an easy architecture to adapt to different vehicles requiring only to modify the actions layer to the available hardware, where the positioning task developed according to the architecture allowed the USVs to autonomously navigate around Durius and stop with an error below 1 m of the final waypoint given by the task.

The ATLANTIS testing procedure was used in all demonstrations and validations with Nautilus in near-real and real environments. The USV deployment procedures and navigability of Nautilus were validated under distinct environmental conditions, namely a Douglas sea state up to 4 and wind speeds up to $10\,\mathrm{m\,s^{-1}}$, within the O&M industry safety standards. Through this procedure the vehicle reached a TRL 8 being demonstrated on the WindFloat Atlantic offshore wind farm to the end-users. Nautilus became the first USV to perform autonomously on a floating offshore wind farm in the Atlantic.

Chapter 4

Multi-domain 3D scene understanding

Traditional O&M practices for offshore wind farms rely on specific technologies for inspecting different wind farm components. The emergence of vehicles capable of acquiring data from the surface and underwater domains presents a significant opportunity to enhance O&M practices. These vehicles provide a more comprehensive assessment of the entire offshore wind farm, including the foundations and the seabed. Thus, the integration of multi-domain inspection capabilities for USVs can significantly enhance the safety of navigation and reduce the number of vehicles required for inspecting the splashing zone and monitoring scour protection systems, leading to increased safety and reduced operational costs.

This chapter is organized as: section 4.1 introduces the motivation of this chapter; section 4.2 presents the multi-domain mapping architecture, namely the M2A (Multi-domain Mapping Architecture); section 4.3 defines a semantic instance completion architecture, namely the FRONTIER (oFfshore wind faRm fOundatioNs daTa completIon architEctuRe); section 4.4 illustrates the results for M2A and FRONTIER; and, section 4.5 highlights the main conclusions.

4.1 Introduction

Inspecting large-scale infrastructures installed offshore presents significant challenges due to various factors including the distance from shore, challenging weather conditions (e.g., waves and winds), the necessity for multiple equipment and operators, and the need for different payloads for each inspection domain. Furthermore, many components of offshore infrastructures require inspection span multiple domains, involving both emersed and immersed elements. This demands different vehicle types for each task, as illustrated in figure 4.1, where:

- Inspecting the emersed domain requires the use of aerial and surface robots equipped with a range of navigation and perception sensors, including GPS, IMU, visual and thermal cameras, and 3D LiDAR. However, the operation of these vehicles in this environment is often disrupted by waves and prevailing winds. In the context of offshore wind farms, the emersed domain includes the structural and visual inspection of several components through NDT,

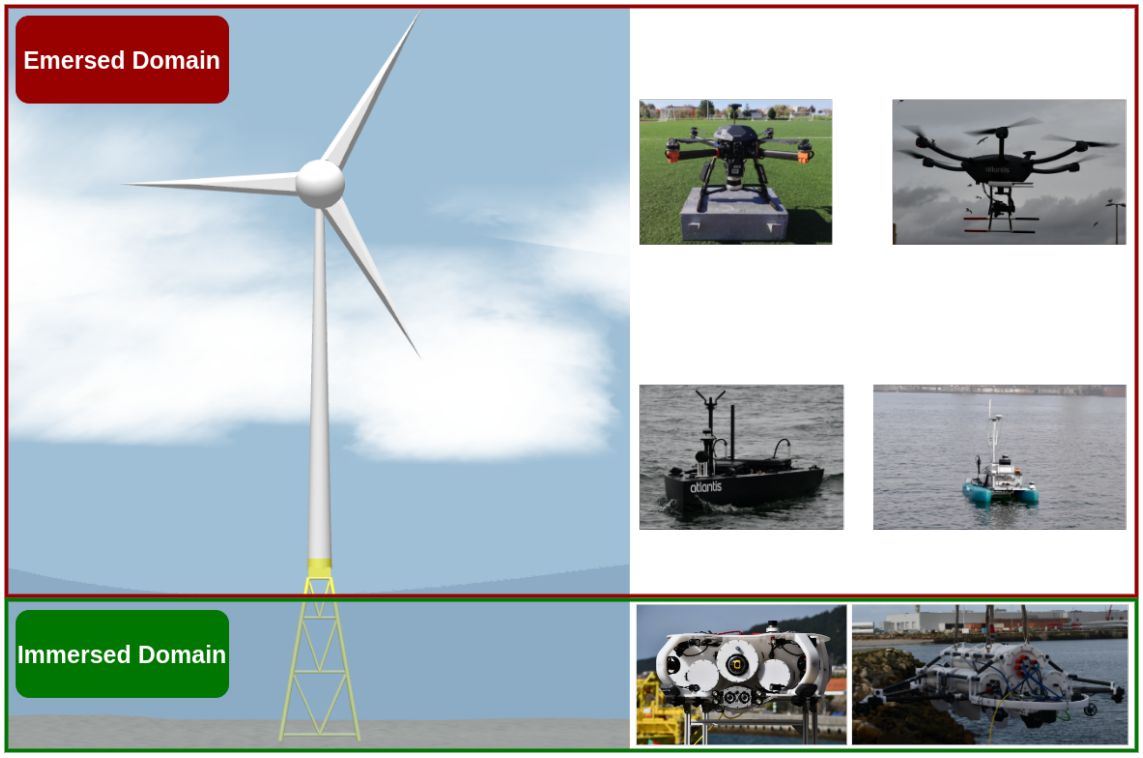


Figure 4.1: Different assets for both domains of the marine environment from the Motion Robotics lab.

namely the turbine towers and blades with UAVs and the dry area of the splash zone in transition piece or the floating structure using USVs.

- Inspecting the immersed domain requires the use of underwater robots, which face constraints imposed by available sensors, increased levels of noise and a restricted field-of-view [190]. These vehicles can be equipped with MBES, imaging sonars and cameras with external lighting systems. Besides the IMU, the navigation sensors requires specialized equipment such as DVL and Long Baseline Acoustic Positioning Systems (LBL) due to the absence of GPS signals. In the context of offshore wind farms, the immersed domain includes the inspection of various components through NDT using ROVs or AUVs, such as the submerged structure of offshore wind turbine foundations, array cables, mooring lines, anchors, and scour protection mechanisms.

Therefore, the use of vehicles capable of acquiring multi-domain information, such as SENSE and Nautilus, enables a more comprehensive representation of the environment, including elements like the splashing zone and scour protection. Nevertheless, automatically acquiring data from multiple domains simultaneously demands the use of sensors with heterogeneous capabilities (e.g., signal-to-noise ratio, range, rate) that operate below and above the waterline to accurately reconstruct the scene surrounding the USV. Although this multi-modal approach significantly enhances the vehicle's situational awareness, several issues still need to be formally addressed. These

include limitations imposed by the sensor payload's FoV and the disruptive effects of wave action, which introduce noise, occlusions, and perspective shifts. These factors lead to information gaps across the frontier region in the splashing area, which is the physical transition between the emersed and immersed domains, as illustrated in figure 4.2.

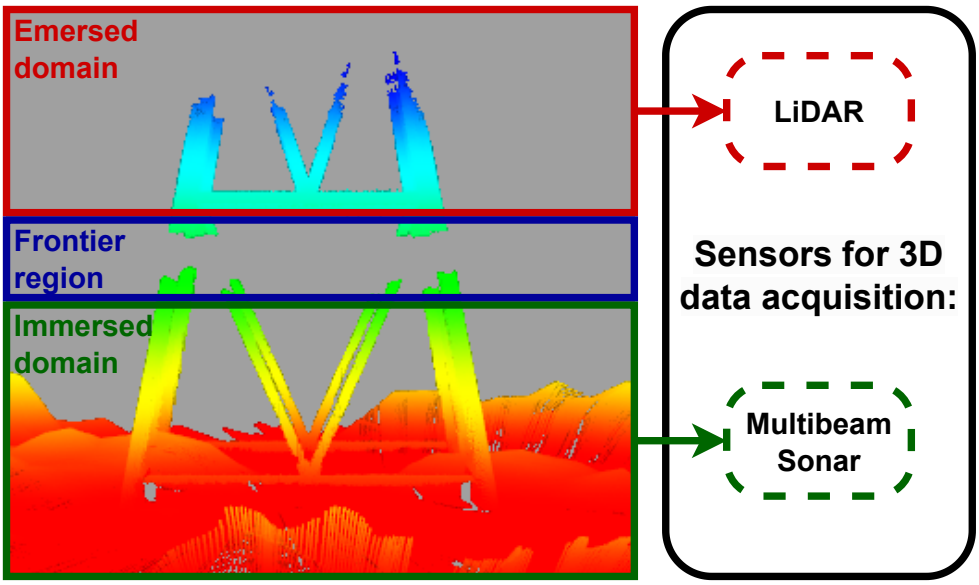


Figure 4.2: Incomplete multi-domain reconstruction of a jacket foundation. The emersed domain represents the dry structure in the splashing zone, the immersed domain depicts the submerged structure of the foundation and the seabed. The frontier region shows the information gap in the splashing zone.

This chapter improves the 3D situation awareness of a surface vehicle by introducing a dual-stage multi-domain architecture formed by:

- The Multi-domain Mapping Architecture (M2A) which leverages the USV's GPS and IMU data for vehicle localization within the environment. Subsequently, 3D data from LiDAR is used to reconstruct the emersed domain, while 3D data from the MBES is used for the immersed domain. The localization data is employed for point cloud filtering and provides an initial estimate for the scenario alignment to improve the quality of the 3D registration process.

- Semantic instance completion that leverages the prior knowledge of the structural configuration of offshore wind farm foundations to fill in occluded information. The offshore wind farm foundations data completion architecture, FRONTIER, is designed to complete missing data, identifying previously the type of foundation even from partial observations of both the emersed and immersed domains, such as single LiDAR and MBES readings near the structure. Once the foundation type is identified, a corresponding 3D model is selected from a pre-existing database and projected onto the multi-domain map. This process augments situational awareness by completing the missing data given by limited observation

capabilities. This enhanced representation not only improves safety for USVs by providing a more comprehensive view of the environment but also offers high-level semantic features for depicting the scenario.

4.2 Multi-domain mapping architecture

The GPS, RTK and IMU measurements are used to localize the USV with modest accuracy. These sensors can produce noisy, imprecise, spurious measurements and deprivation situations, particularly nearby large metallic infrastructures. Moreover, the dynamic nature of the offshore environment related to the waves, currents and wind can produce motion-induced distortion on the sensor measures and outlier data. Moreover, the sensors operating in distinct domains present disparate information which requires different preprocessing approaches.

Therefore, the Multi-domain Mapping Architecture (M2A), shown in figure 4.3, performs a robust data association for data collected by the sensors from each domain. Overall, the odometry estimation of the USV is calculated using the available sensors. Then, the point clouds from the LiDAR and MBES submaps are registered to create a multi-domain map taking the odometry as the initial seed. Thus, an estimation of the error relative to the current pose of the USV for the emersed and immersed domains (e_l and e_m, respectively) is acquired and the RMSE (Root mean square error) fitness of the alignment for each domain (c_l and c_m) provides a measure of the confidence. This architecture provides a guideline to improve scene understanding for O&M.

The initial step of the proposed architecture is the estimation of the vehicle odometry. Since the maritime environment usually allows for GPS and IMU readings when operating offshore, the localization method presented in section 3.4 can be used to estimate the pose of the vehicle. However, there are scenarios where the accuracy of the GPS can be reduced, such as harbors.

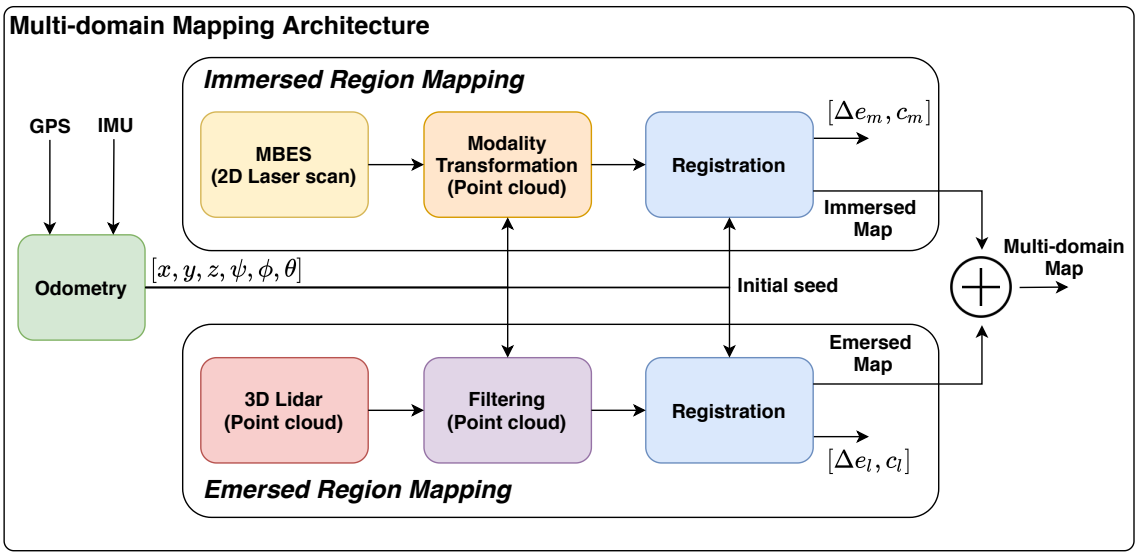


Figure 4.3: Proposed multi-domain mapping architecture, M2A.

In these scenarios the LiDAR or camera based odometry algorithms can be applied (e.g. CT-ICP [189] or ORB-SLAM3 [191]). The vehicle pose is used for preprocessing point clouds as well as to increase the convergence of the registration.

The framework employs a decoupled configuration, wherein distinct domains are endowed with individual pipelines to create the single domain maps and then converge into a unified 3D scene representation. The following subsections detail each procedural step.

4.2.1 Emersed data preprocessing

The LiDAR sensor provides 3D information near the infrastructures. These sensors wave length is 905 nm which can only penetrate 1 cm of the water surface providing no measurement of the sea surface and the point clouds produced are susceptible to outlier data due to reflections detected that project phantom objects within the water [192], as illustrated in figure 4.4. Consequently, this can yield less precise portrayals of the perceived surroundings, a discrepancy that becomes more noticeable in sparsely populated environments like offshore wind farms, where structural elements are distanced thousands of meters to each other. These outliers introduce unreliable features to be registered.

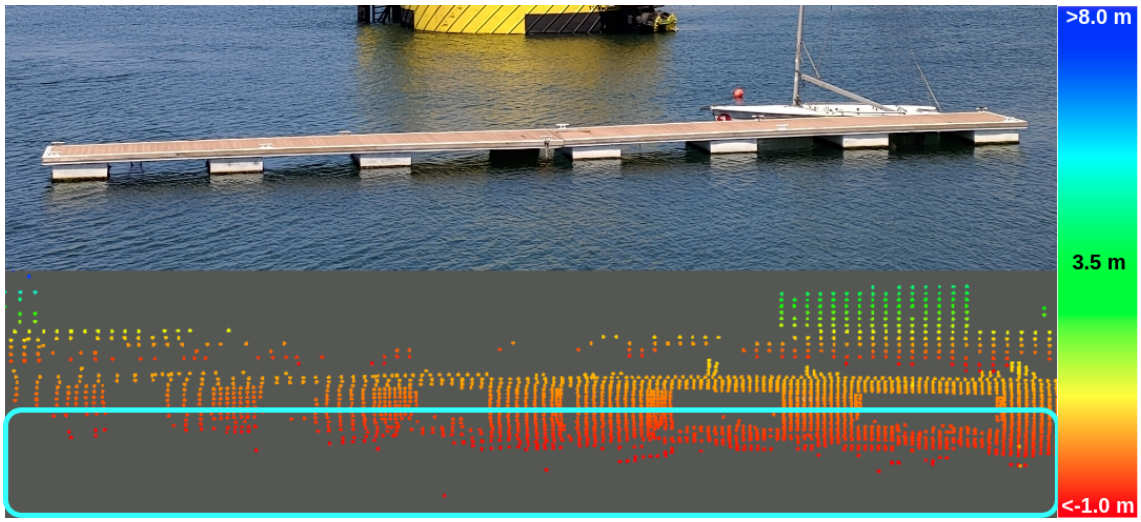


Figure 4.4: Reflected observations retrieved by the LiDAR on a docking structure. The blue rectangle highlights the outlier data and the scale depicts the height in meters.

Since the vehicle experiences undulation, outlier data must be removed by projecting the point clouds onto the absolute frame representing the world frame, based on localization data. This separation of data according to the hydrographic zero is essential. Therefore, spurious measurements are removed using a passthrough filter that eliminates all measurements with an altitude below z_{min} or above z_{max}. This process assumes that the Z-axis in both the sensor frame and the world frame are parallel. This is ensured by first estimating the static homogeneous transformation matrix between the sensor frame and the robot base frame (T_v^r) and then evaluating the homogeneous

transformation matrix describing the USV position in the world (T_r^w), considering only the rotational component. The transformation matrix is defined as shown in equation 4.1, where R_r^w is a 3x3 rotation matrix of the robot relative to the world.

$$T_r^w = \begin{bmatrix} & & & 0 \\ & R_r^w & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.1}$$

Both homogeneous transformations will then be applied to each point in the point cloud to ensure the alignment of the data to filter, equation 4.2, where p_i' is the projection of the point p_i in the world. By projecting the point cloud data into the absolute frame, errors induced by undulation can be mitigated.

$$p_i' = T_r^w T_v^r p_i. \tag{4.2}$$

At last, the passtrough filter is applied to each p_i' point creating a filtered point cloud to be projected back to the sensor frame using the inverse homogeneous transformations, equation 4.3.

$$p_i = (T_v^r)^{-1}(T_r^w)^{-1} p_i'. \tag{4.3}$$

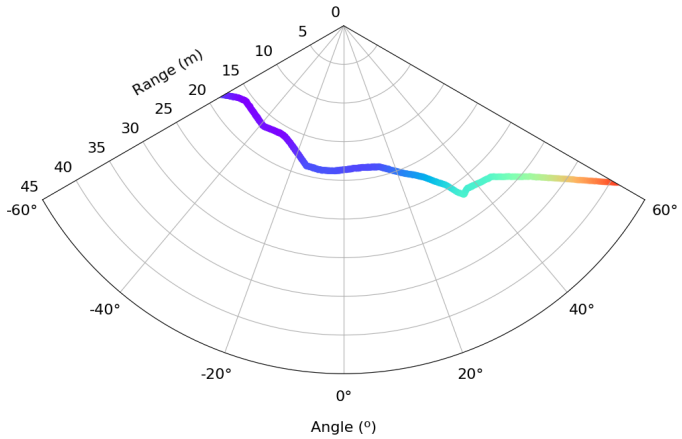
In addition to removing outlier data at the water level, the passthrough filter should also be employed to eliminate points observed within the USV footprint. To achieve this, during the passthrough filter stage, the X and Y axes can be constrained as well. This involves removing all points p_i' that fall below a minimum threshold for each axis. Furthermore, long-range measurements can also be filtered out due to their sparseness, which can introduce low-feature data that is less accurate and more prone to motion estimation errors.

4.2.2 Immersed data preprocessing

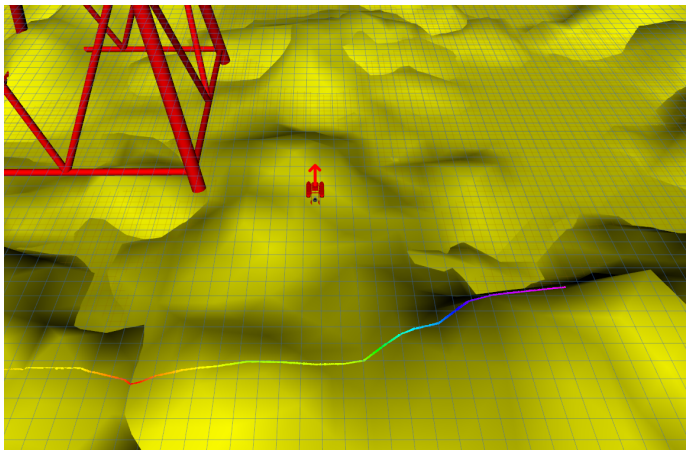
Mapping the underwater scenario is obtained with the MBES. This sensor provides data in polar coordinates (figure 4.5a), with a range measure for each beam placed at different angles. This polar coordinate system of the MBES data is converted to a Cartesian coordinate system using the equation 4.4, where $r_{n,t}$ represents the sonar range from nth beam measured by the scan at time t, b represents the beam spacing angle and s the sonar scan width.

$$p_{n,t} = \begin{bmatrix} x_{n,t} \\ y_{n,t} \\ z_{n,t} \end{bmatrix} = \begin{bmatrix} r_{n,t} \cos\left(n\,b - \frac{s}{2}\right) \\ r_{n,t} \sin\left(n\,b - \frac{s}{2}\right) \\ 0 \end{bmatrix}. \tag{4.4}$$

A two-dimensional scan is created with the depth information along a line perpendicular to the heading direction of the vessel, as depicted in figure 4.5b. The sensor modality and position on the USV will generate multiple situations where the observations are not related between consecutive frames.



(a) Scan in polar coordinates. Rainbow-color line representing range.



(b) Scan in world. Rainbow-color line representing depth.

Figure 4.5: Multibeam Echosounder scan. Rainbow-color line represents a scale variation, where the red to magenta color goes from higher to lower scales, respectively.

Assuming that the odometry error, $E(\Delta p)$, is small when the displacement, Δp, approaches zero, equation 4.5, a 3D representation with several scans can provide overlap areas between different samples.

$$\lim_{\Delta p \to 0} E(\Delta p) = 0. \tag{4.5}$$

Thus, a 3D sliding window provide a neighborhood that can be used to estimate alignment errors between an existing map and the current observation. To build the sliding window each MBES scan (S_i) is associated to the corresponding odometry pose. Once the number of scans (N) required for the window are available, a local map can be established (SW_i). Moreover, to ensure an overlapping regions between consecutive local maps, a number of the latest scans will be kept from one SW to the next, as depicted in figure 4.6, where the red area is the scans to be forgotten and the green area the one to hold in memory.

Before inserting a scan in SW_i a pre-defined threshold for the relative displacement of the robot must be secured between consecutive scans, namely if the displacement (Δp) and the vari-

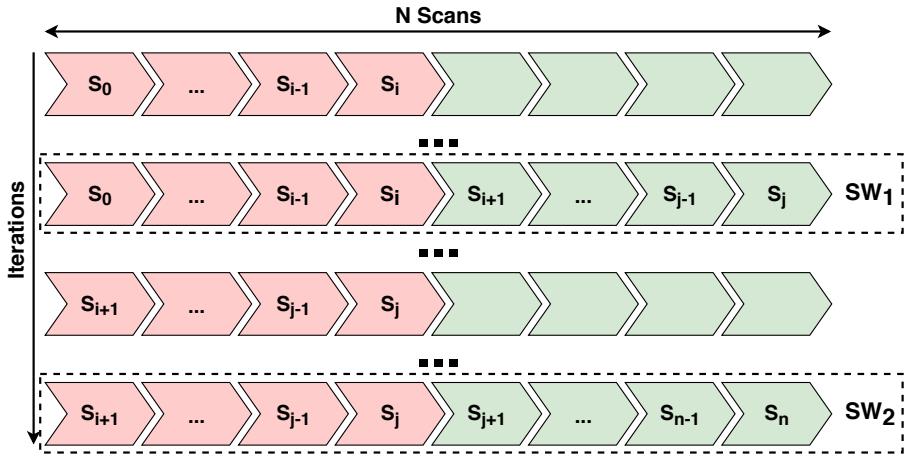


Figure 4.6: Sliding window iterations, where *S* is the MBES scan, and *SW* is the local map. Scans to forget are defined in red and to keep in green.

ation in heading ($\Delta\theta$) are superior to some predetermined thresholds, d_m and θ_m respectively. If the minimum required motion is detected, the measure (S) is accepted. Further a homogeneous transformation matrix from the MBES sensor to the world frame (T_w^m) is used to correct the sliding window measurements. Figure 4.7 illustrates the step-by-step method for the sliding window management.

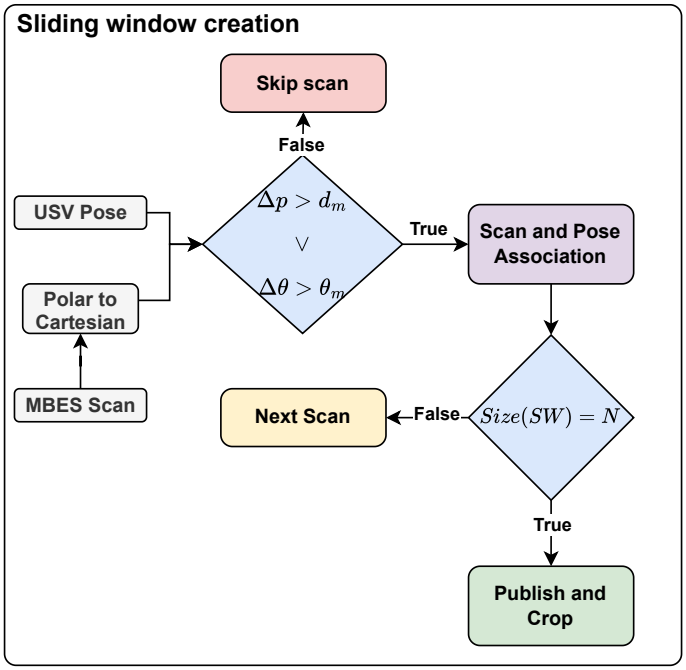


Figure 4.7: Sliding window management process.

4.2.3 Multi-domain map representation

A registration approach that removes outliers, registers and organizes an input point cloud was proposed. As depicted in figure 4.8, the input point cloud for both domains is preprocessed to remove spurious and noisy observations, reducing the propagation of errors in the reconstructed map through the registration process. Thus, a Statistical Outlier Removal (SOR) filter performs the statistical analysis of the neighborhood of each point in the dataset, computing the average distance to the k nearest neighbors of each point [193]. Posteriorly, the mean ($\overline{d}$) and standard deviation (σ_d) of all these distances are computed to determine a distance threshold t_d equal to equation 4.6, where m_σ is a standard deviation multiplier.

$$t_d = \overline{d} + \sigma_d \cdot m_\sigma. \tag{4.6}$$

The points will then be classified as inlier or outlier according to their average neighbor distance. When a distance is above the threshold t_d the point is considered an outlier and as such it will be trimmed.

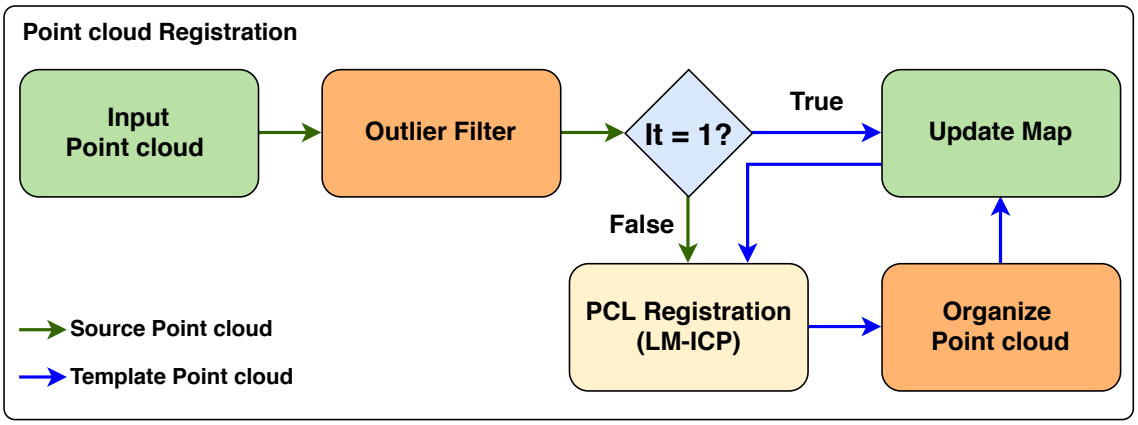


Figure 4.8: Point cloud registration approach, where It=1 is the first iteration.

Following the outlier removal process of the input point cloud, the registration method is employed to refine its alignment with the global 3D map (used as reference) that is being built. The initial point cloud (first iteration, It=1) is directly integrated into the map, while subsequent point clouds undergo alignment and concatenation. To accomplish this task, a non-linear variant of the Iterative Closest Point (ICP) algorithm is employed, specifically the Levenberg-Marquardt optimization-based approach (LM-ICP). This methodology, as proposed by Fitzgibbon [96], aims to minimize the registration error by iteratively matching points in the input cloud to their closest counterparts in a reference point cloud. The LM-ICP algorithm does the registration by establishing correspondences between points in the input cloud and the reference template. These correspondences serve as the basis for estimating the optimal rotation and translation parameters, which are then applied to align the input point cloud with the global 3D map. The process is repeated

iteratively until a predefined fitness threshold is reached or a maximum number of iterations is exceeded.

The global map quickly grows in the number of points and corresponding memory consumption, leading to increased computational time for the algorithm. To address this issue, after the alignment of the filtered input point cloud, an organization method is applied to speed up the concatenation process. This is achieved through octree division which performs spatial partitioning [194]. This spatial organization significantly reduces the complexity of the registration process by enabling more efficient neighborhood searches. By clustering points into small neighborhoods based on their spatial proximity, the precess avoids unnecessary interactions between distant points, improving the computational efficiency. All voxels are converted into centroid representations. This simplification involves approximating the entire voxel's point cloud with its centroid, effectively reducing the data volume and processing requirements.

This registration method ensures a visually accurate representation of the scenario and enables the fusion of both the immersed and emersed representations, using the shared information from the GPS-IMU odometry while mitigating the inherent noise associated with sensor data.

4.3 FRONTIER - Offshore wind farm foundations data completion architecture

The application of semantic instance completion techniques provides valuable insights for both navigation and inspection tasks, such as collision avoidance and structural inspections. By incorporating prior knowledge about the structures, these techniques can effectively fill information gaps, enhancing the digital representation of the environment. Identifying the type of foundations for offshore wind turbines is a crucial task, as it determines the 3D model to be found within the prior knowledge database. This enables the alignment of the model with the observed foundation, effectively filling in any missing or occluded information.

This work employs a semantic instance completion approach by exploiting the premise that all infrastructures installed at sea are built in shipyards or at factories, meaning that their structural dimension and shape are well established. The FRONTIER (oFfshore wind faRm fOundatioNs daTa completIon architEctuRe) requires a vehicle equipped with multi-domain sensing capabilities. Initially, a preprocessing stage converts the input data into the same modality, point cloud, and eliminates spurious information from the sensors. Subsequently, a clustering algorithm is applied to segment the elements present in the scene. The segmented objects are further filtered based on the height and number of points to eliminate superfluous objects, such as birds, small vessels or docks. The LiDAR points of the remaining clusters are fed into a multi-class classifier. This classifier utilizes the LiDAR data to identify the foundation type, as LiDAR provides a more comprehensive representation of the surrounding environment. It returns the detection bounding box, cluster points, class and confidence. This information serves to filter the clusters in the immersed data to a region of interest. The final clusters points for both the emersed and submerged domains are then concatenated to augment the data. For registration, the detected foundation type

serves as the basis for selecting the corresponding CAD template from the database and aligning it with the LiDAR and MBES foundation detections. A flowchart depicting the pipeline is illustrated in figure 4.9.

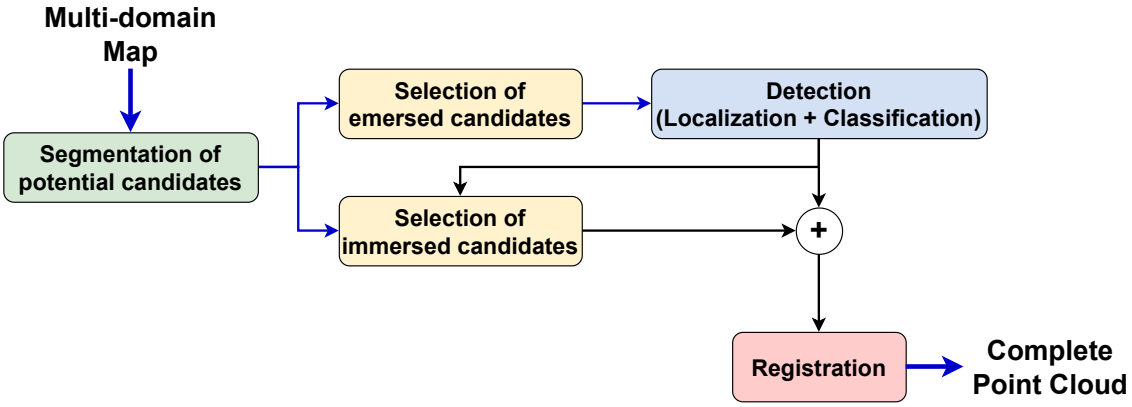


Figure 4.9: Flowchart depicting the FRONTIER pipeline for semantic instance completion of offshore turbine foundations based on multi-domain maps through detection (localization and classification) and alignment.

4.3.1 Segmentation of potential candidates

In maritime environments, the presence of multiple objects, such as support vessels, imposes significant challenges in detecting offshore wind turbine foundations. This challenge is particularly pronounced in coastal locations like the ATLANTIS test centre, where numerous port infrastructures and vessels surround the inspected structure. To address this issue, it becomes essential to segment all elements in the scene to subsequentially verify if it constitutes an offshore wind turbine foundation. The selection of potential candidates is performed using a clustering algorithm that segments the scenario into a maximum number of clusters (M), based on the relative proximity between the cluster centroid - defined by the number of selected keypoints - and each point in the point cloud.

The Bird's-Eye View Adaptive Farthest Point Clustering (BEVA-FPC) is proposed. It adopts a bird's-eye view approach, distinguishing structures in maritime scenarios. The Farthest Point Sampling (FPS) method [195] is applied to the point cloud $P' = \{(x_i, y_i, 0) \mid x_i, y_i \in \mathbb{R}\}_{i=1}^{N}$ projected along the Z-axis, where N is the number of points. Through this projection, the Z-coordinate is effectively ignored in the initial processing, focusing solely on the XY plane. The FPS selects M keypoints, denoted as $\{k_i\}_{i=1}^{M}$, representing the most distanced points within the cloud. The chosen number of keypoints corresponds to the maximum identified clusters, forming the basis for the remaining clustering process. Each point $(x_i, y_i, 0)$ within the point cloud is assigned to its nearest keypoint k_i, effectively partitioning the entire point cloud into M distinct clusters, each labelled with $L = \{1, 2, \ldots, M\}$. The set of centroids for each cluster is established as $C = \{c_i\}_{i=1}^{M}$.

Subsequently, the distances between these centroids are evaluated, and if the distance falls below a user-defined radius r, the label is updated to the one with the minimum value. This step, highlighted in equation 4.7,

$$\{l'_i\}_{i=1}^{M} = \{\min(l_i, \{l_j \mid ||c_j - c_i|| < r : c_j, c_i \in C\}_{j=1}^{M})\}_{i=1}^{M}, \tag{4.7}$$

where l_i is the label for the ith cluster, results in a new label mapping $\{l'_i\}_{i=1}^{M}$. For each point, the assigned label is then remapped according to $\{l'_i\}_{i=1}^{M}$, and the labels values are normalized to $L' = \{1, 2, \ldots, M'\}$, where M' is the number of unique labels within $\{l'_i\}_{i=1}^{M}$ and is either equal to or less than the initial value of M. Finally, the new cluster centroids C' are recalculated. This adaptive approach significantly enhances the algorithm's capacity to capture meaningful structures in maritime scenarios. The pseudo-code for BEVA-FPC is outlined in algorithm 1.

To distinguish between structural components and the seabed in the immersed point cloud

Algorithm 1 BEVA-FPC - Bird's-Eye View Adaptive Farthest Point Clustering

Input:
Point cloud: $P = \{(x_i, y_i, z_i) \mid x_i, y_i, z_i \in \mathbb{R}\}_{i=1}^{N}$
Maximum number of clusters: M
Radius threshold: r

Output:
Number of clusters: M'
Point-wise label assignment: $\{pl_i\}_{i=1}^{N}$, $1 \leq pl_i \leq M'$
Cluster centroids: $C' = \{(x_i^c, y_i^c, z_i^c) \mid x_i^c, y_i^c, z_i^c \in \mathbb{R}\}_{i=1}^{M'}$

$P' = \{(x_i, y_i, 0) \mid (x_i, y_i, z_i) \in P\}_{i=1}^{N}$ // Z-axis projection.
$\{k_i\}_{i=1}^{M} = FPS(P', M)$ // Get M keypoints using Farthest Point Sampling.
for all $p' \in P'$ **do**
  $pl_i \leftarrow \underset{j}{\operatorname{argmin}} \{(k_j - p'_i)^2\}$ // Find nearest keypoint k_j for each point p'_i.
Initialize $C = \{c_i\}_{i=1}^{M} = \{(0,0,0)\}_{i=1}^{M}$
for $(i \leftarrow 1; i \leq M; i++)$ **do**
  $c_i \leftarrow \mu(\{p_j \in P' \mid pl_j = i\})$ // Get the centroid for the ith cluster.
for all $c_i \in C$ **do**
  $\{l'_i\}_{i=1}^{M} = \{\min(l_i, \{l_j \mid ||c_j - c_i|| < r : c_j, c_i \in C\}_{j=1}^{M})\}_{i=1}^{M}$ // Get the minimum label between the original label l_i and the set of labels $\{l_j\}_{j=1}^{M}$ associated with vectors close to the centroid c_i within the specified distance r.
Find unique labels from $\{l'_i\}_{i=1}^{M}$ to get the number of clusters M'
Normalize labels $\{l'_i\}_{i=1}^{M}$ to $L' = \{1, 2, \ldots, M'\}$
for all $l'_i \in \{l'_i\}_{i=1}^{M}$ **do**
  $\{pl_j \in \{pl_j\}_{j=1}^{N} \mid pl_j = i\} \leftarrow l'_i$ // Update point labels with new mapping.
Initialize $C' = \{c'_i\}_{i=1}^{M'} = \{(0,0,0)\}_{i=1}^{M'}$
for $(i \leftarrow 1; i \leq M'; i++)$ **do**
  $c'_i \leftarrow \mu(\{p_j \in P \mid pl_j = i\})$ // Get the centroid for the ith cluster considering the original point cloud.
return M', $\{pl_i\}_{i=1}^{N}$, C'

the depth is crucial. Therefore, it requires a two-stage clustering that initially segments the data according to depth variations along the Z-axis and then each depth cluster is further partitioned using the BEVA-FPC algorithm. For Z-axis clustering, the X and Y components are projected out, retaining only the Z-axis data. This results in a set $P' = \{(0,0,z_i) \mid (0,0,z_i) \in P\}_{i=1}^{N_m}$. Subsequently, the remaining BEVA-FPC process is applied. This segmentation along the Z-axis creates depth clusters containing objects at similar depths. Finally, the BEVA-FPC algorithm is applied to each depth cluster to segment objects along the XY-plane.

After segmentation, the selection of candidates for each domain is crucial to filter out spurious detections and identify suitable clusters for subsequent classification and registration stages. In the emersed domain, small or distant objects are eliminated as they provide sparse information for structure detection. Additionally, objects with small heights, such as docking platforms, are excluded. The remaining emersed candidates are forwarded to the detection stage. For the immersed component, candidate selection considers the foundation detection information. Only candidates falling within the XY-axis limits of the foundation detection are kept. From these, the largest cluster at a higher depth, indicative of the seabed, is removed. All remaining candidates are considered part of the foundation immersed structure.

4.3.2 Learning-based classification of offshore wind turbine foundations

The identification of offshore wind turbine foundations requires the accurate classification of the segmented clusters. Emersed point cloud information from multi-domain maps is employed for this classification task, since it is denser then the immersed information. Each emersed candidate is classified using a deep learning network for 3D point clouds. Fourteen distinct offshore wind turbine foundation types, as illustrated in figure 4.10, are considered. These correspond to subtypes within the most commonly used foundations, including monopile, tripod, jacket, and

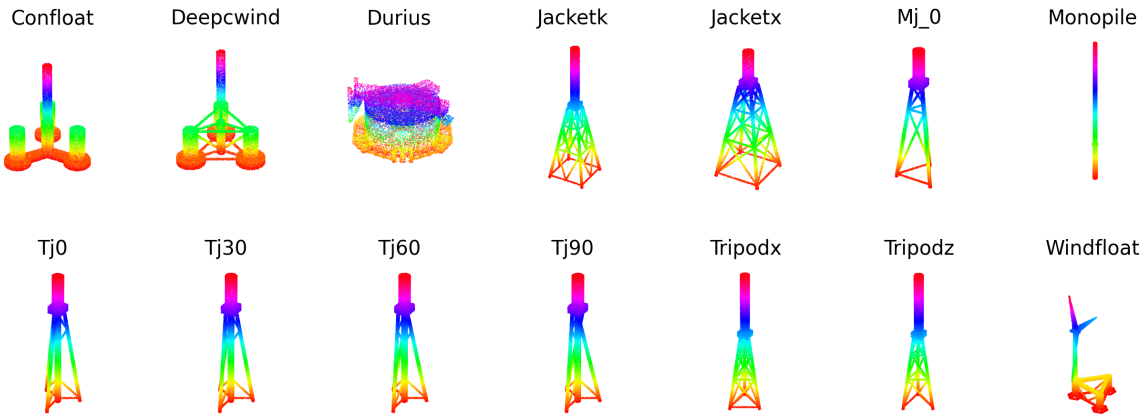


Figure 4.10: Offshore wind turbine foundations included in the classification model.

semi-submersible foundations[1] [196]. An additional class was defined for capturing objects outside the scope of interest (background class), such as vessels, buildings, or port infrastructure.

A RI-PointTransf (Rotation Invariant Point-cloud Transformer) model was designed for this classification by adopting a transformer architecture that comprises five key components: a patchifier, patch embedding, position embedding, a transformer encoder (composed of multiple transformer layers), and a multilayer perceptron (MLP) head. The patchifier initially receives the point cloud and transforms it into a rotation-invariant representation before grouping the points into small patches. Subsequently, a convolution stem inspired by Pointnet [112] is employed to generate the patch embedding by encoding each patch into a feature vector, which serves as the input sequence for the transformer encoder. To facilitate information aggregation from all patches using multi-head self-attention, a classification token (CLS) is prepended to the patch embeddings. Since multi-head attention is permutation-equivariant and cannot distinguish the order of input elements within a sequence [197], transformers require position embedding which directly introduces positional information into the sequence. The transformer encoder then extracts attention-based features, which are further processed by a task-specific head, in this case, a MLP, for classification. The RI-PointTransf architecture is depicted in figure 4.11.

Patchifier: The patchifier preprocesses the point cloud data, enabling it to be more robust to rotations by effectively grouping points into patches. This pre-processing involves four key steps:

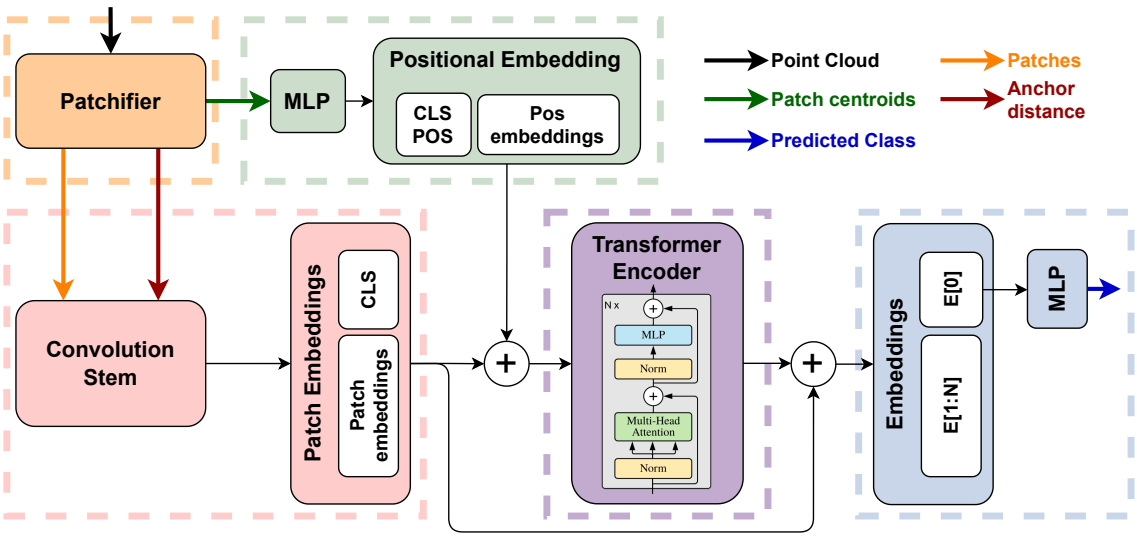
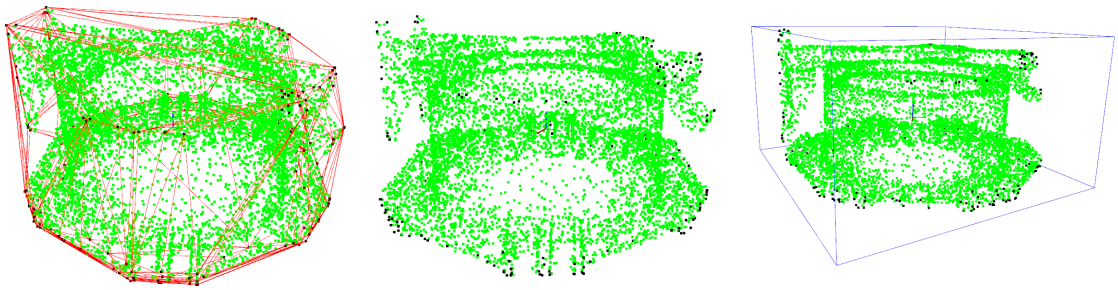


Figure 4.11: RI-PointTransf architecture design. The orange, red, green, purple and blue boxes are the patchifier, patch embedding, position embedding, transformer encoder, and task-specific head, respectively.

[1] Infolink Consulting, 2021. *Floating wind turbine II: Floating structure technology and market trends*. URL: https://www.infolink-group.com/energy-article/Floating-wind-turbine-II-Floating-structure-technology-and-market-trends

rotation and translation normalization, patch creation, anchor point selection, and anchor distance evaluation.

Rotation normalization is accomplished by determining the oriented bounding box of the object and transforming the points to a canonical position by multiplying them with the inverse of the estimated orientation. The quickhull algorithm [198] is used for this normalization, identifying the convex hull of the shape (Figure 4.12a). Next, principal component analysis (PCA) is applied to the convex hull vertices (Figure 4.12b), determining the eigenvalues and corresponding eigenvectors in descending order. The orientation matrix for the bounding box is given by the eigenvector matrix, and the maximum and minimum values along the X, Y, and Z axes are obtained by bringing the mesh into a canonical rotation via multiplication of the vertices with the transpose of the bounding box orientation. An example of an oriented bounding box is shown in Figure 4.12c. Then, a translation normalization is performed by shifting the coordinates of all points by the centroid to ensure that the point cloud is centred around the coordinates $(x,y,z) = (0,0,0)$. A uniform point cloud scaling technique is also employed to address the influence of varying object sizes and facilitate model generalizability. This method adjusts the input data by scaling the normalized point cloud across all axes by a factor equivalent to the maximum value identified in the X, Y, and Z dimensions. Consequently, the point cloud is standardized to a unit size along the axis exhibiting the largest dimension. This procedure preserves the inherent aspect ratio of the object while mitigating the effect of scale on the model's performance.



(a) Convex hull. (b) Selected points for PCA. (c) Oriented bounding box.

Figure 4.12: Oriented bounding box assessment using Durius. Initially computes the convex hull **(a)**, then uses its vertices as input for PCA **(b)**. In **(c)** is the resulting oriented bounding box.

Then, the patch creation is given by grouping the points into M clusters of size K, denoted as $\mathscr{P}_{M\times K\times 3}$. The furthest point clustering (FPC) technique, as introduced by Li *et al.* (2023) [199], is employed for this purpose. The normalized point cloud $P = \{x_i \mid x_i \in \mathbb{R}^3\}_{i=1}^{N}$ is subsampled using farthest point sampling to identify the M keypoints $\{k_i\}_{i=1}^{M}$. Each point x_i is then assigned to its nearest keypoint k_i. Further refinement occurs within each cluster to ensure each patch contains K points. If a cluster exceeds K points, farthest point sampling is again applied to reduce the point set. Conversely, if a cluster contains fewer than K points, oversampling is performed. Oversampling involves duplicating existing points and introducing controlled variability through Gaussian noise until a point set with the desired size is achieved. This strategy ensures a consistent patch size while

preserving local point distribution characteristics. This step will output the patches, $\mathscr{P}_{M\times K\times 3}$, and the patch centroids $\mathscr{P}^{c}_{M\times 3}$.

Furthermore, a set of anchor points are selected within the object for acquiring relative information from the patches to the global point cloud. These anchors are determined by utilizing the vertices derived from the convex hull (e.g. figure 4.12b). The initial steps of FPC are then applied, where the points are subsampled through FPS to identify M_a keypoints $\{k_i\}_{i=1}^{M_a}$. Subsequently, each vertex is assigned to its closest keypoint k_i. The median of each cluster serves as the anchor points, opting for existing measurements from the cluster instead of generating new ones, as using the mean. This methodology ensures that the anchor points are less vulnerable to sensor-induced noise or outlier measurement. These anchor points serves as reference for calculating the relative distances between each patch point and every anchor point, resulting in a distance matrix, denoted as $\mathscr{D}_{M\times K\times M_a}$, where M represents the number of patches, K the number of points per patch, and M_a the number of anchor points. Each element $\mathscr{D}_{ijk}$ within the matrix specifies the Euclidean distance between the ith patch's jth point and the kth anchor point.

At last, to enhance the model's resilience to rotational variations, particularly those originating from the diverse viewpoints during USV inspections, a conversion to spherical coordinates for the patch points and centroids is adopted. Equations 4.8 performs this transformation, yielding radial distance (r), polar angle (ϕ), and azimuthal angle (θ).

$$r=\sqrt{x_i^2+y_i^2+z_i^2};\ \phi=\arctan\left(\frac{\sqrt{x_i^2+y_i^2}}{z_i}\right);\ \theta=\arctan\left(\frac{y_i}{x_i}\right). \tag{4.8}$$

This is based on the assumption that typical rotations encountered during missions primarily affect the Z-axis rotations, thus causing variations of the azimuthal angle, leaving the radial distance and polar angle mainly unaltered. Consequently, a point's representation in spherical coordinates remains invariant under rotations around the Z-axis. Finally, to improve the model generalization capacity, patch points and centroids in spherical coordinates are normalized to an interval of 0 to 1. Normalization involves dividing the radial distance by the maximum observed range and the angles by 2π.

Patch embedding: For the patches ($\mathscr{P}_{M\times K\times 3}$) and anchor distance matrix ($\mathscr{D}_{M\times K\times M_a}$), it is required to encode each patch and anchor distance into a C-dimensional feature vector $f_i\in\mathbb{R}^C$. This process involves passing each patch and respective anchor distance through the convolution stem illustrated in figure 4.13. In this stem, each input passes through its own convolution block ($C1$ and $C2$), composed by a 1D convolutional layer, batch normalization and a Leaky-ReLU, and then are concatenated into a single feature vector.

Due to the inherent variability and abundance of features present in f_{C1} and f_{C2}, incorporating a feature recalibration mechanism is crucial for effective processing. This allows the network to leverage global information and selectively emphasize informative features while suppressing less useful ones. Studies by Qiu et al. (2021) [200] have demonstrated the superiority of channel

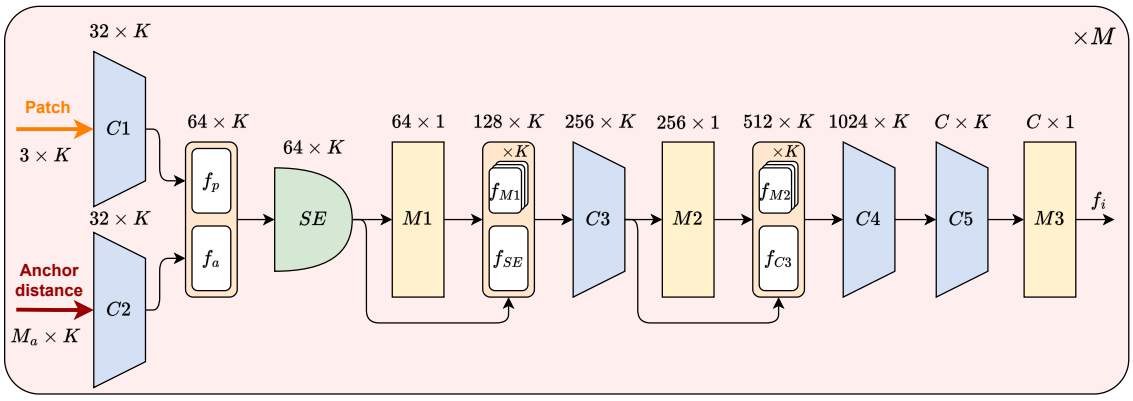


Figure 4.13: Convolution stem architecture. $C1$, $C2$, $C3$, $C4$ and $C5$ are convolution blocks composed by a 1D convolutional layer, batch normalization and a Leaky-ReLU. $M1$, $M2$ and $M3$ are the max pool layers, and SE is a Squeeze-and-Excitation block.

attention modules, such as Squeeze-and-Excitation (SE) [201] and Convolutional Block Attention Module (CBAM) [202], compared to spatial attention mechanisms for this task. Their compact structure and proven effectiveness in refining 3D point cloud features render them well-suited for this work.

Therefore, the SE block (figure 4.14) was selected to refine features based on interdependencies between channels. This attention module, embedded after the feature vectors concatenation, re-weights feature maps by first capturing global channel response distribution via squeeze, generating the channel-wise descriptors, and then applying sample-specific activations learned through a self-gating mechanism, excitation, enabling integration with subsequent layers. At last, the SE block results in the refined feature vector f_{SE} which effectively emphasizes the most informative features from the patch and anchor distance feature vectors.

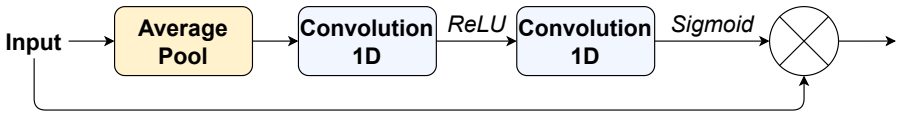


Figure 4.14: Squeeze-and-Excitation architecture, where $\otimes$ represents matrix multiplication.

Following feature recalibration through the SE block, a max-pooling layer ($M1$) is employed to reduce spatial dimensions to 64×1, retaining only the maximum value for each channel. This effectively removes spatial redundancy within channels, preserving the most activated signals. Subsequently, the $M1$ output is expanded to $64 \times K$ through replication and concatenated with f_{SE}. This concatenation enables the model to fuse spatial information from the expanded feature map (indicating "where") with channel-wise attention from the SE block (denoting "what"), thus enhancing its ability to capture both local and global features. Next, a convolutional block ($C3$), comprising a 1D convolution followed by batch normalization and activation function (Leaky-ReLU), extracts hierarchical local features from the data. Spatial dimensionality is further re-

duced through another max-pooling layer ($M2$), resulting in f_{M2}. This is then combined with the extracted features (f_{C3}), generating a more compact representation that highlights the most informative local patterns. Subsequent convolutional blocks further refine and learn deeper features. A final max-pooling layer ($M3$) summarizes the learned features across the entire patch, reducing the spatial dimension. This convolution stem architecture facilitates focusing on important points and channels within the patch points while combining spatial and channel-wise information for a richer feature representation.

At last, a classification token (CLS) is incorporated into the C-dimensional feature vectors, $\{f_i\}_{i=1}^{M}$, describing the sequence representation. Inspired by Dosovitskiy *et al.* (2021) [203], this classification token is defined by a learnable embedding prepended to the sequence of embedded patches, where the model learns to encode information about the whole sequence into this hidden representation, which is then used by the final classification layer to predict the class label.

Position embedding: The position embedding technique similar to that proposed by Yu et al. (2022) [204] was used to leverage spatial relationships within point clouds for classification tasks. It uses a MLP as a mapping function that transforms patches centroid coordinates from $\mathbb{R}^3$ to a C-dimensional feature vector, where $\mathscr{P}^c_{M\times 3}$ is the input. The underlying intuition is that incorporating global positional information can significantly enhance point cloud understanding. By explicitly encoding the relative positions of patches, the model gains the ability to reason about inter-patch relationships and contextualize individual features more effectively. The MLP architecture itself comprises several layers: layer normalization, linear and GeLU layers, a dropout layer for regularization, another layer normalization, and a final linear layer to produce the C-dimensional feature vector. A learnable positional classification token is appended to the positional embeddings. This token is used to captures global scene information and match the patch embedding size after appending the CLS token.

Transformer encoder and task-specific head: The defined transformer encoder architecture is the proposed by Vaswani *et al.* (2017) [197]. It comprises N identical blocks stacked sequentially, each block contains a multi-head self-attention mechanism and a position-wise feed-forward network. Patch embeddings enhanced with positional encodings serve as the input. These undergo layer normalization before entering the multi-head attention block. The resulting attention outputs are then added back to the original input via a residual connection. Once again, layer normalization is applied before feeding the information into the position-wise feed-forward network (FFN). Finally, another residual connection combines the transformed output with the FFN input. The multi-head self-attention mechanism plays a crucial role in uncovering relationships between patch embeddings. This enables the model to selectively focus on relevant patches and accumulate contextual information from across the entire sequence. The position-wise feed-forward networks then process and integrate this contextualized information, further refining the representation.

At last, for the classification head, a multilayer perceptron is employed. This MLP comprises layer normalization, linear and GeLU layers, followed by a dropout layer and a final linear layer

for predicting classification scores across N_c classes. The network leverages the output feature vector of the CLS token to map the model's outcome into class probabilities using softmax, with the class label of the point cloud being assigned based on the score with the highest value.

Every segmented candidate within the RI-PointTransf undergoes classification into predetermined categories (14 foundation types or background). Subsequently, this approach provides the detected points, the oriented bounding box acquired during the patchifier stage, the assigned class, and the associated confidence level determined by the probability score estimated by the network.

4.3.3 Template-based instance registration

Once the emersed and immersed candidates are selected as belonging to the foundation, the alignment with the corresponding 3D model from the database begins. This process is performed by the registration pipeline depicted in figure 4.15, where the foundation detection serves as the basis for selecting the structural shape. Subsequently, pointwise covariances and normals are evaluated for both the structure template and instance observation (i.e., multi-domain representation of the object of interest). The Generalized-ICP (GICP) [205] returns the transformation required to align the structure template with the instance observation data, making possible to complete the gap of 3D information that usually exists in the regions around the splashing zone.

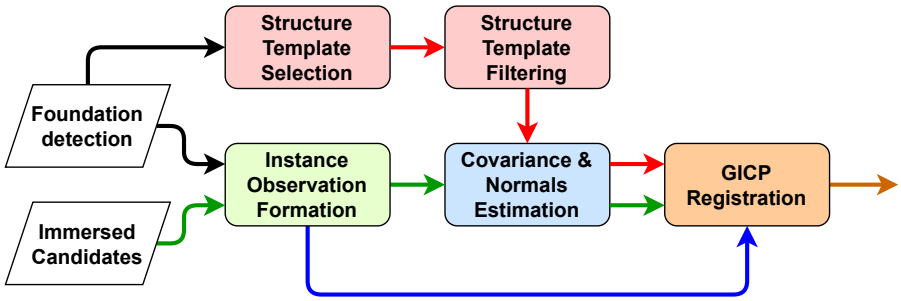


Figure 4.15: Template-based instance registration. Red, green, black, blue and orange arrows represents the structure template point cloud, the instance observation point clouds, the detection output, the target bounding box centre and the output transformation, respectively.

The type of foundation is used to select the structure template from a database. Concurrently, the instance observation point cloud of the multi-domain environment surrounding the robot is formed based on: (i) fixed foundations with observations within the MBES range, then use the LiDAR and MBES data associated with the foundation, fusing them into a single representation; (ii) floating foundations or fixed foundations with observations outside the MBES range, only use the LiDAR data. To expedite convergence in scenario (ii), the structure template point cloud is filtered to remove the immersed structures during registration.

Furthermore, the covariances and normals for each point on the instance observation point clouds are estimated to refine the registration process. This information provides a richer representation of the local surface geometry compared to simple point coordinates. K-Nearest Neighbors

(KNN) search with a KD-tree is employed to estimate the covariance matrix for each point. This 3×3 matrix captures the covariances between the coordinates of neighboring points along different axes, reflecting the local surface shape. Since the covariance matrix is symmetric, it allows for efficient eigenvalue and eigenvector decomposition using the robust algorithm proposed by Eberly (2014) [206], where the eigenvectors represent the principal directions of the local surface, while the eigenvalues describes the spread or curvature along these directions. Finally, the normal vector at each point is calculated by the eigenvector corresponding to the smallest eigenvalue.

The registration is given by the GICP, as described in Segal *et al.* (2009) [205], using the defined source and target inputs. GICP merges the functionalities of standard point-to-point and point-to-plane ICP algorithms within a single probabilistic framework, introducing a plane-to-plane registration. This algorithm outperforms traditional point-to-point and point-to-plane ICP methods when dealing with point clouds exhibiting significant orientation variations and large spatial distances between points. Unlike point-to-point ICP, which focuses solely on minimizing the distance between corresponding points, GICP incorporates additional information like local surface normals or feature descriptors. This allows GICP to capture the complete geometric relationship between points, even when they are far apart or have different surface orientations. Contrarily, the point-to-plane ICP struggle with complex geometries or non-planar surfaces. GICP's ability to leverage additional information makes it more robust to noise, and outliers, leading to more accurate and reliable registration results.

At last, during the registration process, the centre of the detection bounding box (source) is employed as the initial guess, to accelerate the GICP algorithm's convergence. This approach reduces the search space by leveraging the prior knowledge from the detection stage, thereby minimizing the risk of getting locked in erroneous local minima.

4.4 Results

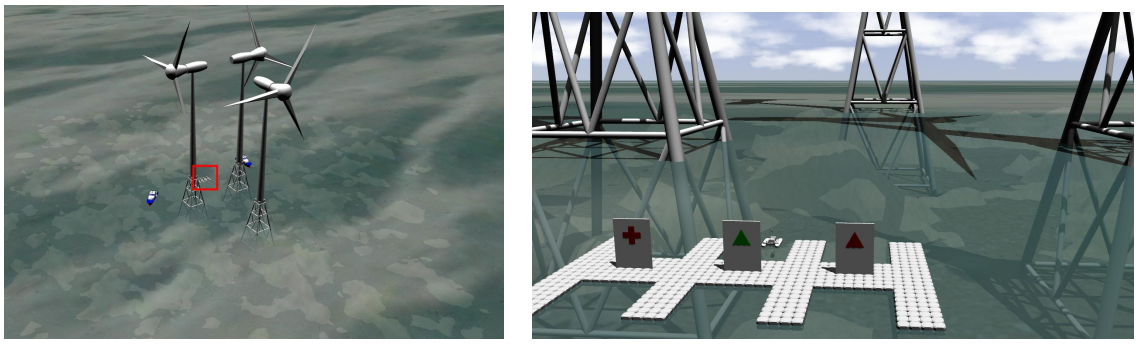
This chapter focuses on enhancing scene understanding capabilities for USVs by creating multi-domain maps augmented with semantic instance completion, thereby extending observability for occluded objects and improving USV navigability. To achieve this, a dual-stage multi-domain architecture was developed, encompassing two methodologies: M2A and FRONTIER.

An extensive set of trials was conducted to evaluate the qualitative and quantitative performance of each method, including simulated and real-world scenarios. Whenever possible, the ATLANTIS technology certification procedures (subsection 3.5.2) were followed. The evaluation of M2A initially used simulation scenarios with a realistic 3D environment of an offshore wind farm created in Gazebo based on the VRX simulator [207], incorporating both emersed and immersed domains. Subsequently, real-world data was used to validate performance with distinct USVs in three different scenarios: the *Leça* harbor, the ATLANTIS coastal test centre, and the WindFloat Atlantic offshore wind farm. The FRONTIER methodology was evaluated using real-world data acquired by Nautilus during the approval in principle and technology qualification trials (subsection 3.5.2). Additionally, some components of FRONTIER, including the classification model

and template-based instance registration, used simulated scenarios to evaluate performance with objects that were not accessible for real-world data acquisition.

4.4.1 M2A - Multi-domain mapping

The ability of M2A to generate multi-domain maps was evaluated in both simulation and real world experiments. The scenario included both emersed and immersed domains and a digital model of the SENSE vehicle with all of its sensors (GPS, IMU, LiDAR, and MBES). All sensors were affected by Gaussian noise, parametrized according to the specifications of each sensor datasheet. The tests were conducted using a compact scenario with multiple objects, namely three jacket type foundations, two vessels, a docking platform and seabed with varying elevation, as shown in figure 4.16. This scenario was used to evaluate the performance of the multi-domain map creation on a scenario with realistic O&M elements. All tests were performed on an Intel Core i7-6700HQ CPU @ 2.6 GHz - 3.5 GHz and 16GB of DDR4 RAM.



(a) Full scenario view.

(b) Close up of the docked SENSE.

Figure 4.16: Implementation and validation offshore wind farm scenario.

The experiments were conducted in zero wind conditions and the waves were modelled as a composition of three ripples with the configuration presented in table 4.1. The ripple components were asynchronous, meaning that at some points in time the maximum peak was given by the sum of the amplitude of all components.

A mission was designed to evaluate the GPS/IMU fusion and multi-domain mapping methodology, performing the trajectory, depicted in figure 4.17, with the following steps: (i) USV starts

Table 4.1: Ripple configuration for simulation.

Component	Amplitude (m)	Period (s)	Direction (x,y)
1	0.06	12.6	(-1.0,0.0)
2	0.04	3.7	(-0.7,0.7)
3	0.03	6.3	(0.7,0.7)

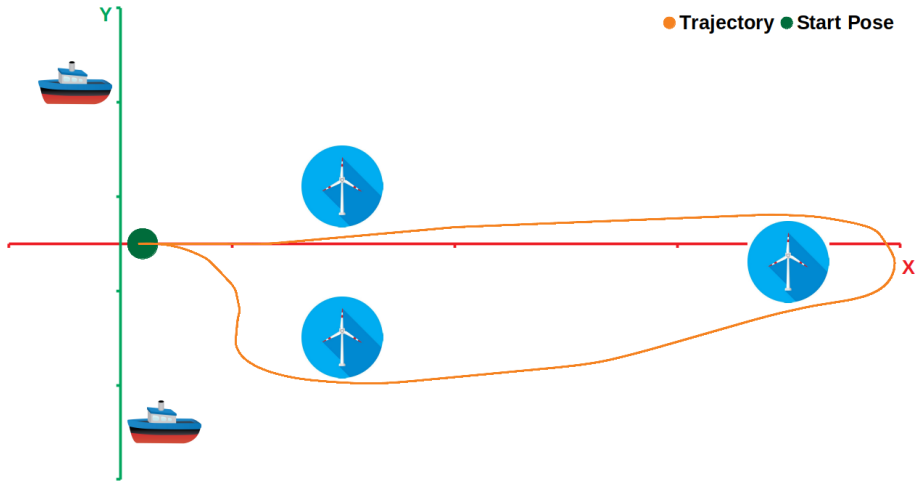
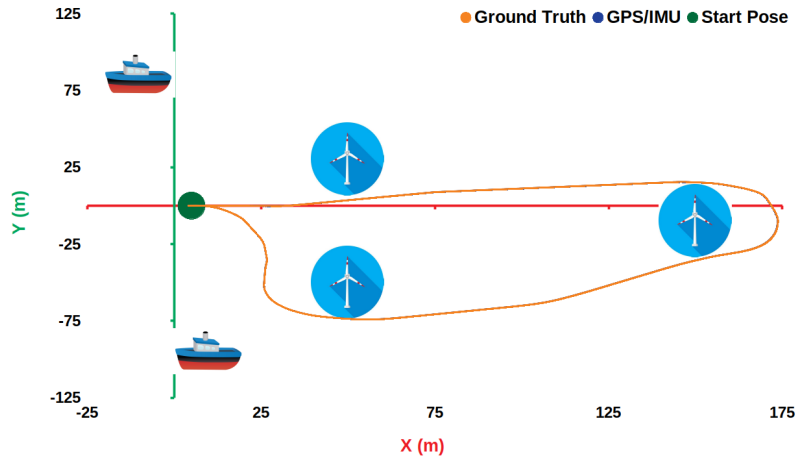


Figure 4.17: Inspection trajectory. The orange and green points depicts the path and the dock, respectively.

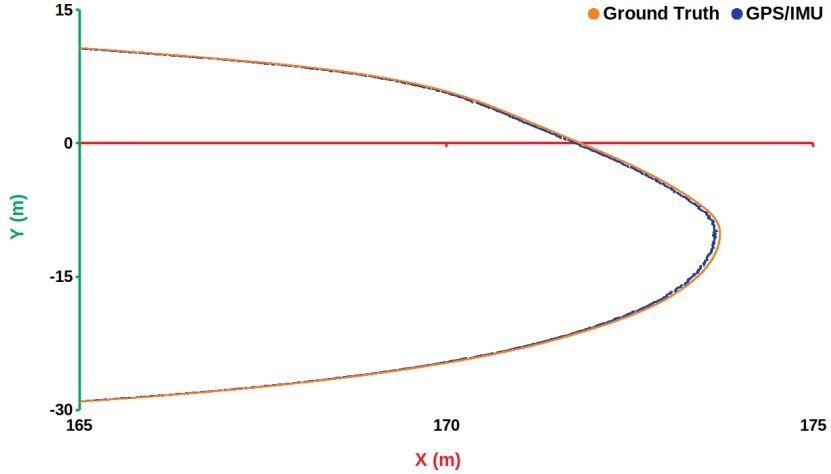
berthed on a dock (green point); (ii) SENSE passes between the two closest wind turbines with an almost linear motion towards the furthest wind turbine; (iii) in the furthest wind turbine performs a close range inspection contouring the jacket foundation; (iv) the vehicle moves outside the wind farm passes through a wind turbine and docks at the starting pose. During the execution of the trajectory, no positional control was applied, resulting in drift in the XY-plane due to the tidal ripples. This drift also affected the USV's altitude.

4.4.1.1 Quality of the localizations

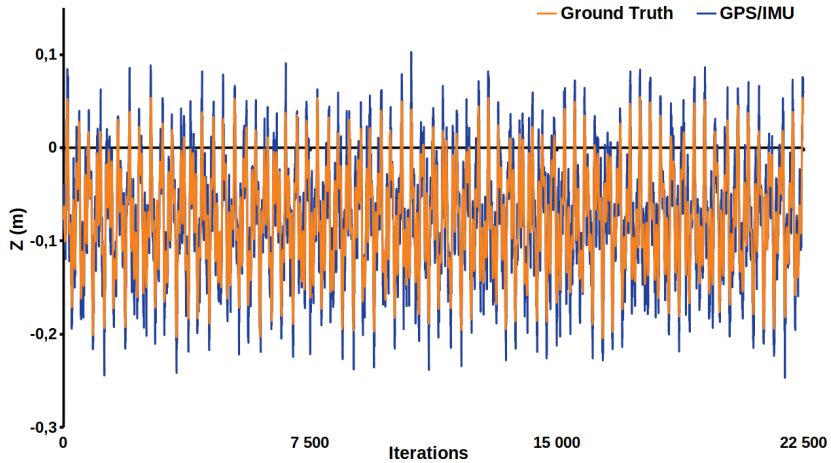
A good odometry estimation must be achieve by using the EKF proposed in section 3.4 to fuse positional information of the GPS with the attitude and angular velocities measurements provided by the IMU. The estimation of the USV localization was first evaluated in a simulator and then in a real operating scenario. In the simulator, the GPS/IMU fusion results were compared against a perfect odometry estimation obtained from the Gazebo simulator. The sensor errors were modelled based on the vendor datasheet specifications, where the IMU experienced deviations of 0.2 °in roll and pitch, 1 °in yaw and 0.15 °/s in angular velocities, and the GPS with RTK had horizontal errors of 0.05 m and vertical errors of 0.075 m. This aims to validate the localization algorithm to provide a reasonable initial estimation for the M2A. In figure 4.18 the XY trajectory and Z-axis variation through time are illustrated. A small error was observed during the route execution, where the orange line is the ground truth and the blue is the GPS-IMU fusion. In figure 4.18b, a small variation between the ground truth and estimation can be seen which are more prominent during movements with angular velocity. Using the metric proposed by Zhang and Scaramuzza in [188], the odometry estimation in this XYZ trajectory presents an average RMSE error of 0.04 m with a standard deviation of 0.01 m, with a travelled distance of 750 m. One explanation for the error is the low acquisition rate of the GPS sensor (5 Hz), where the EKF performs several prediction steps, which does not take into account the drift caused by the tidal ripples. Moreover, the estimated



(a) Full USV trajectory in the XY coordinates.



(b) USV trajectory close-up in the XY coordinates



(c) USV altitude evolution through time.

Figure 4.18: Odometry results during the sample trajectory execution. The orange, blue and green dots are the ground truth, GPS-IMU fusion and starting pose, respectively.

altitude also displays similar errors, figure 4.18c, since it relies in the altitude measure of the GPS which is typically the less accurate.

The odometry estimation was tested using different configurations for tidal ripples and wind considering the SENSE USV sea state constraints, as shown in table 4.2. The USV performs a 250 m route along the centre of the scenario of figure 4.16. The increase of the tidal ripples causes a major impact on the accuracy of the system, where higher amplitudes lead to bigger RMSE and standard deviations, due to the weather conditions that leads to drifts on the EKF prediction model. Moreover, the introduction of wind (trials D, E and F) does not present a large impact on the accuracy, showing a small increase on the standard deviation. Thus, the localization provides an accurate estimation of the vehicle motion through time even with harsher conditions.

Table 4.2: Odometry results with distinct ripple and wind configurations.

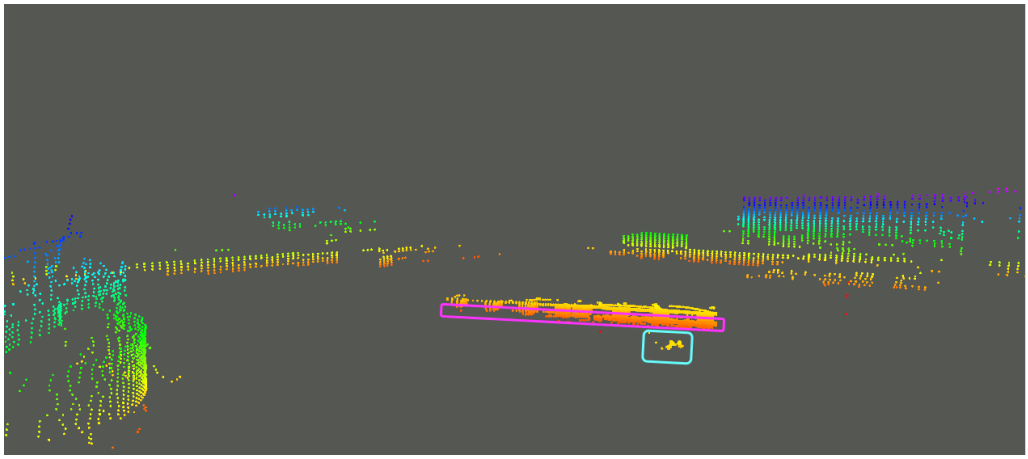
Trials	**Wind**		**Ripples**			**RMSE**	**Standard**
	Speed (m/s)	**Dir**	**Amp (m)**	**Period (s)**	**Dir (x,y)**	**(m)**	**Deviation (m)**
A	0	-	0.12 0.08 0.06	12.6 3.7 6.3	(-1.0,1.0) (-0.7,0.7) (0.7,0.7)	0.031	0.013
B	0	-	0.24 0.16 0.12	12.6 3.7 6.3	(-1.0,1.0) (-0.7,0.7) (0.7,0.7)	0.039	0.017
C	0	-	0.36 0.24 0.18	12.6 3.7 6.3	(-1.0,1.0) (-0.7,0.7) (0.7,0.7)	0.049	0.024
D	1.1	N	0.12 0.08 0.06	12.6 3.7 6.3	(-1.0,1.0) (-0.7,0.7) (0.7,0.7)	0.036	0.014
E	1.1	N	0.24 0.16 0.12	12.6 3.7 6.3	(-1.0,1.0) (-0.7,0.7) (0.7,0.7)	0.039	0.018
F	1.1	N	0.36 0.24 0.18	12.6 3.7 6.3	(-1.0,1.0) (-0.7,0.7) (0.7,0.7)	0.049	0.025

4.4.1.2 Multi-domain mapping

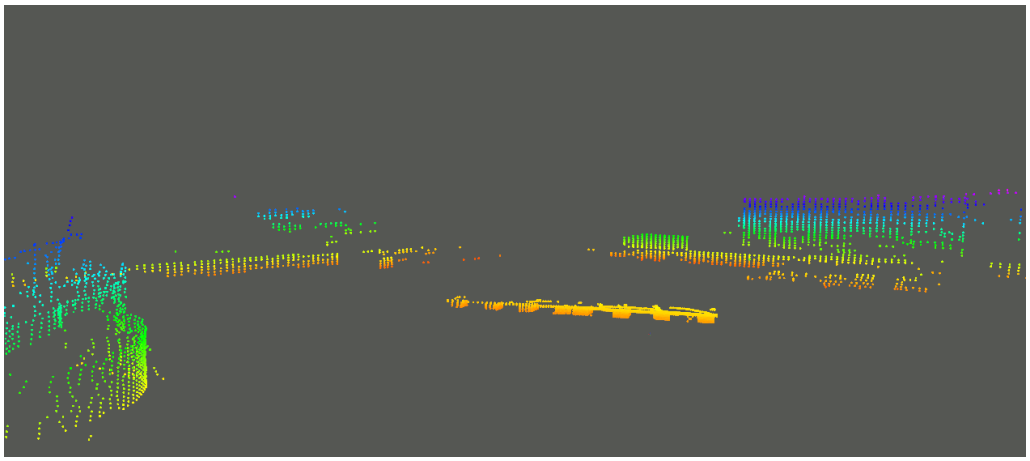
With accurate odometry and sensor data unaffected by noise or outlier measurements, map reconstruction could be achieved through a simple concatenation procedure. However, the drift in USV localization caused by weather conditions, sensor noise, and outliers, along with the inherent noise in perception sensors, demands a filtering and alignment procedure to ensure accurate map reconstruction.

In marine environments, the presence of LiDAR reflections on the water surface is negligible due to its operation wavelength which is mostly absorbed. Nevertheless, the LiDAR can obtain

erroneous observations (pink bounding box) due to visual reflections, as well as readings from the hull (light blue bounding box), figure 4.19a. These outliers contaminate the registration process, leading to inaccurate results. As shown in Figure 4.19b, the emersed data preprocessing filters proposed in section 4.2 were able to remove these spurious information, namely by removing the measurements below sea level and from the hull without losing of structural details.



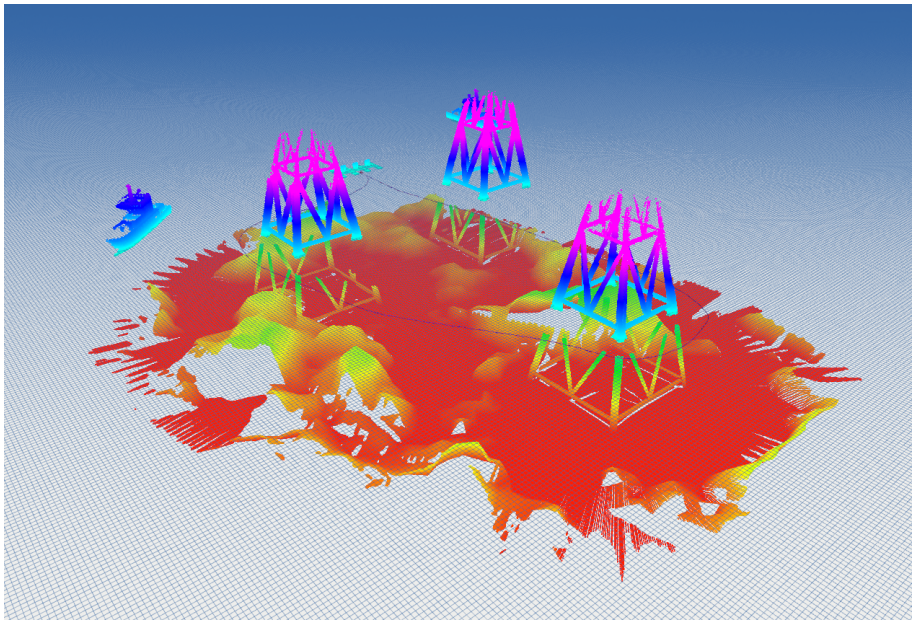
(a) LiDAR raw point cloud. Pink and light blue rectangles highlights the water reflections and the USV hull points, respectively.



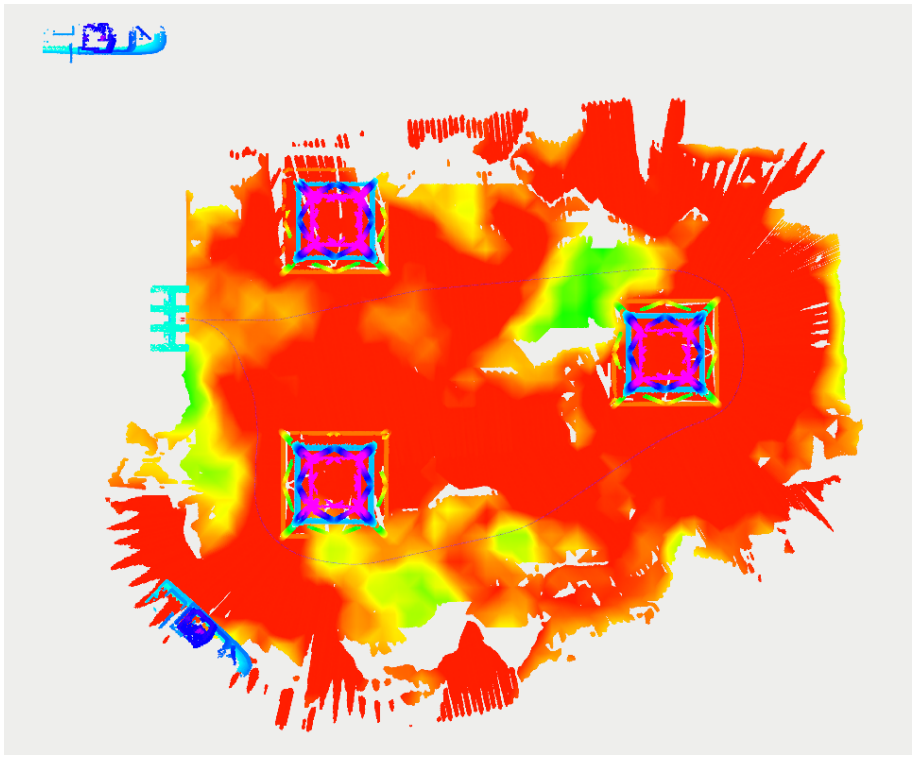
(b) LiDAR point cloud after the preprocessing.

Figure 4.19: LiDAR data acquired on the ATLANTIS coastal test centre.

With the odometry estimates and data preprocessing, the multi-domain map representation can be created by registering the current observations with the global 3D multi-domain map. Using the scenario configuration presented in figure 4.16, the creation of the 3D multi-domain representation with M2A presented accurate results as depicted in figure 4.20, where the diameter for each jacket leg has an error of 0.2 cm between the CAD model and the reconstruction. The common structures in both layers (fixed foundations) are aligned concentrically and with a visible variation in the size up to the seabed. Nevertheless, due to limitations in the sensors FoV there are areas with incomplete data on the transition pieces of the wind turbines fixed foundation. Moreover, in the



(a) Multi-domain map side view.



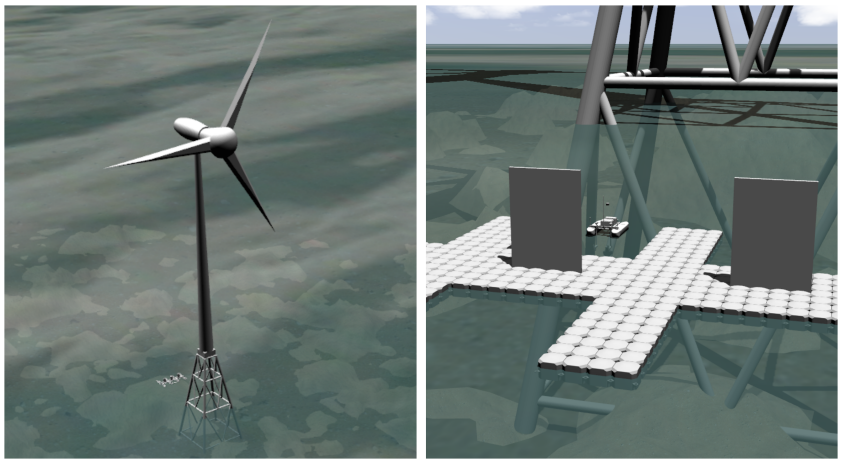
(b) Multi-domain map top view.

Figure 4.20: Multi-domain map of the simulated scenario.

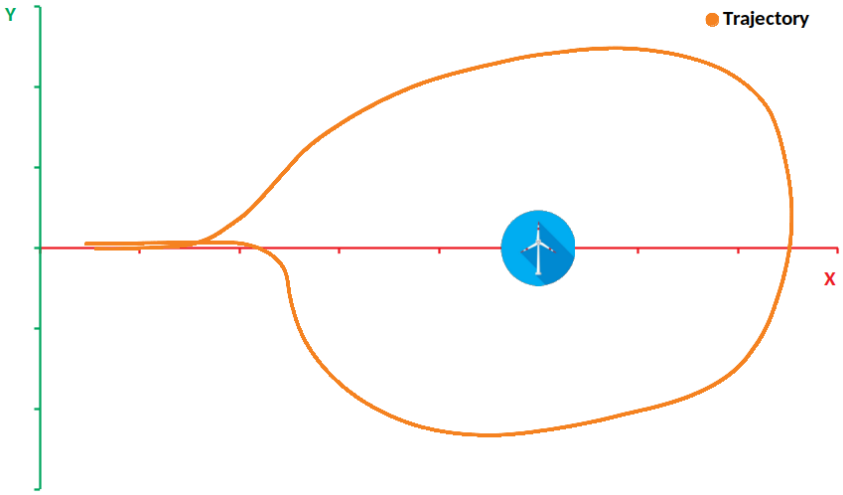
immersed domain, the multi-domain mapping results are denser in areas perpendicular to the USV heading due to the sensor's FoV limitations.

For evaluating the multi-domain map accuracy, the root mean square error was calculated from the 3D representation and the ground-truth point cloud from the Gazebo. This fitness score is a

function of the Euclidean distance of each point in the map to the closest point in the ground-truth. Since the wind intensity shows a low impact on the localization accuracy, two simulated scenarios without wind considering two distinct ripple configurations were performed. The first simulated scenario is composed by a single wind turbine as depicted in figure 4.21a, where the USV performed the trajectory in figure 4.21b. The second simulated scenario represents the scene depicted in figure 4.16 with a trajectory similar to figure 4.17. This aims to assess the level of impact that different environmental and sea state conditions cause to the M2A method proposed in section 4.2.



(a) Single offshore turbine scenario.



(b) Trajectory performed in scenario 1. The orange line depicts the trajectory.

Figure 4.21: Scenario 1: Simulated scene and USV trajectory.

Table 4.3 summarizes the results considering the RMSE fitness score for each trial. As it can be seen, by increasing ripple amplitude a higher error on the 3D scene reconstruction is observed, mainly due to the odometry deviation. The LM-ICP registration was able to reduce the error when compared to the original odometry estimation calculated based on IMU and GPS signals, since the method corrects the data alignment of the sensor data with the 3D global map. Moreover, world

Table 4.3: Multi-domain map RMSE score with distinct ripple configurations.

World	Ripples			RMSE
	Amplitude (m)	Period (s)	Direction (x,y)	(m)
1	0.12	12.6	(-1.0,1.0)	0.028
	0.08	3.7	(-0.7,0.7)	
	0.06	6.3	(0.7,0.7)	
	0.36	12.6	(-1.0,1.0)	0.034
	0.24	3.7	(-0.7,0.7)	
	0.18	6.3	(0.7,0.7)	
2	0.12	12.6	(-1.0,1.0)	0.024
	0.08	3.7	(-0.7,0.7)	
	0.06	6.3	(0.7,0.7)	
	0.36	12.6	(-1.0,1.0)	0.042
	0.24	3.7	(-0.7,0.7)	
	0.18	6.3	(0.7,0.7)	

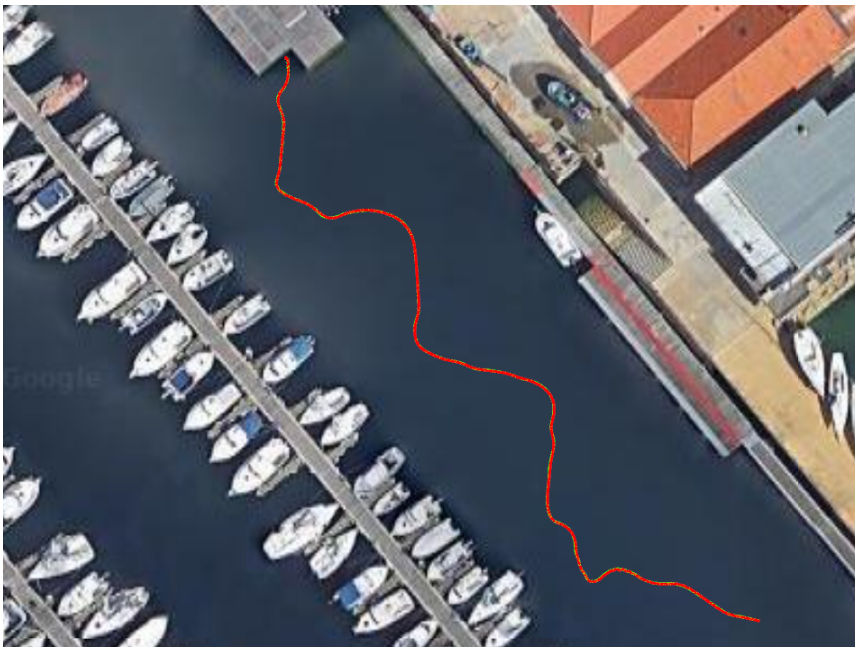
2 present larger errors, mainly due to the increase in the travelled distance which integrates the sensors drift and outliers through time. Nevertheless, the application of the SOR filter reduces the outliers. Also, the octree representation, using only the voxel centroids performs an average of a small neighborhood, thus reducing the sensor noise impact.

The proposed M2A method was able to perform in soft real time requiring an average of 300 ms per iteration to reconstruct the scenario. This is due to the octree representation that allowed for a smaller point cloud and an optimized neighbourhood search process which allowed to reduce the LM-ICP computational complexity.

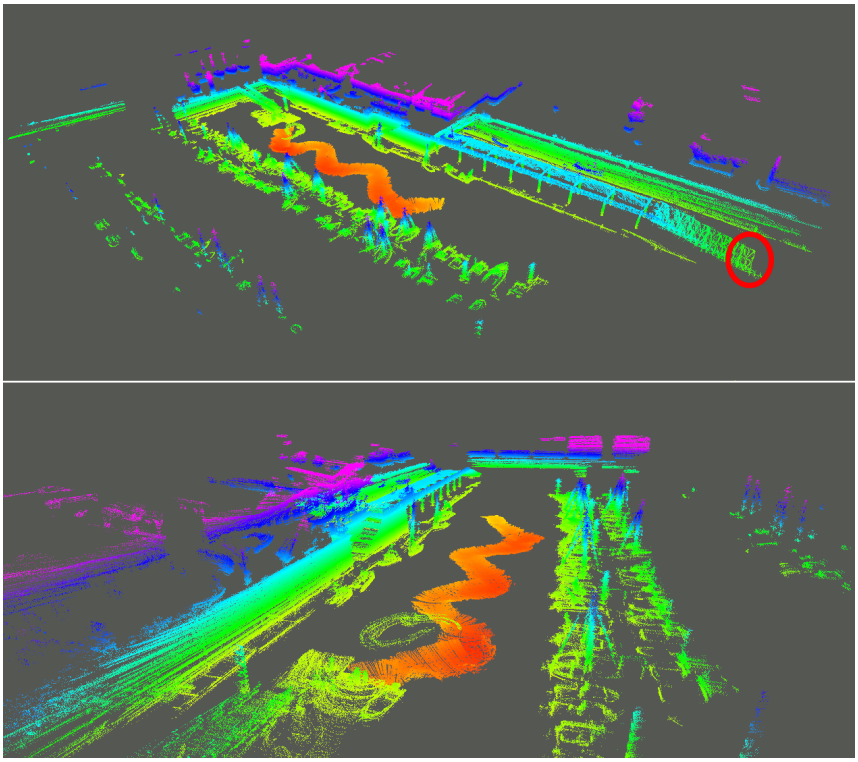
4.4.1.3 Validation on real scenarios

The M2A was also evaluated in real world conditions namely, using the SENSE and Nautilus USV operating in both Leixões harbor and ATLANTIS test centre in Viana do Castelo. These experiments aimed to evaluate the impact of using distinct LiDARs (16 and 128 channels), and the environmental conditions effects on the MBES, namely the water salinity and seabed materials. The M2A was finally evaluated in a real commercial offshore wind farm where Nautilus USV was able to acquire a multi-domain map of a offshore wind farm.

Figure 4.22a captures the first experiment, comparing the position estimates from the GPS/IMU odometry (green line) with the RTK estimation (red line) similar curves are acquired, where the RMSE error between them is approximately 0.07 m with a standard deviation of 0.04 m. Thus, the odometry provides a good accuracy even when the USV navigates in proximity to metallic structures with ripples caused by larger vehicle crossings nearby. The map obtained through the M2A is depicted in figure 4.22b. No ground truth is available for the 3D scenario and therefore a qualitative evaluation reveals several moored vessels, as well as the pillars, ramps, and walls of



(a) Odometry estimation for the dataset patch. In red is the RTK measure and in green the GPS/IMU odometry.



(b) Multi-domain map.

Figure 4.22: Multi-domain map of the *Marina de Leça* harbor yacht basin.

the real test area, where the access ramp (red circle in figure 4.22b) width is 8.55 m while the estimated width from the M2A result is 8.52 m. Furthermore, the underwater map shows a depth of 1.9-3.0 m, which is consistent with the depth measurements for the harbor, namely the yacht basin has a depth of 2-3.5 m. Some spurious data points can be observed on the multi-domain representations, where the surface reflections is caused by dynamic objects (vessels) that are not currently filtered.

Figure 4.23 shows an aerial view of the multi-domain map of the *Marina de Leça* harbor recovered with SENSE. This illustrates the alignment of the multi-domain map with a satellite image, allowing for tasks such as detecting dock occupation or estimate structural variations.

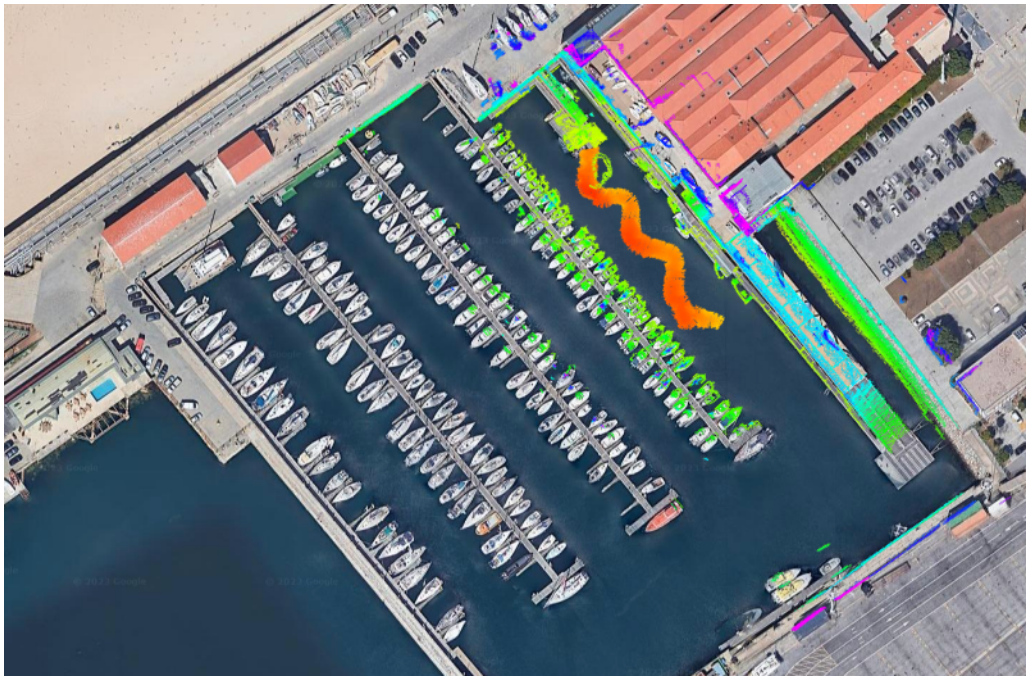
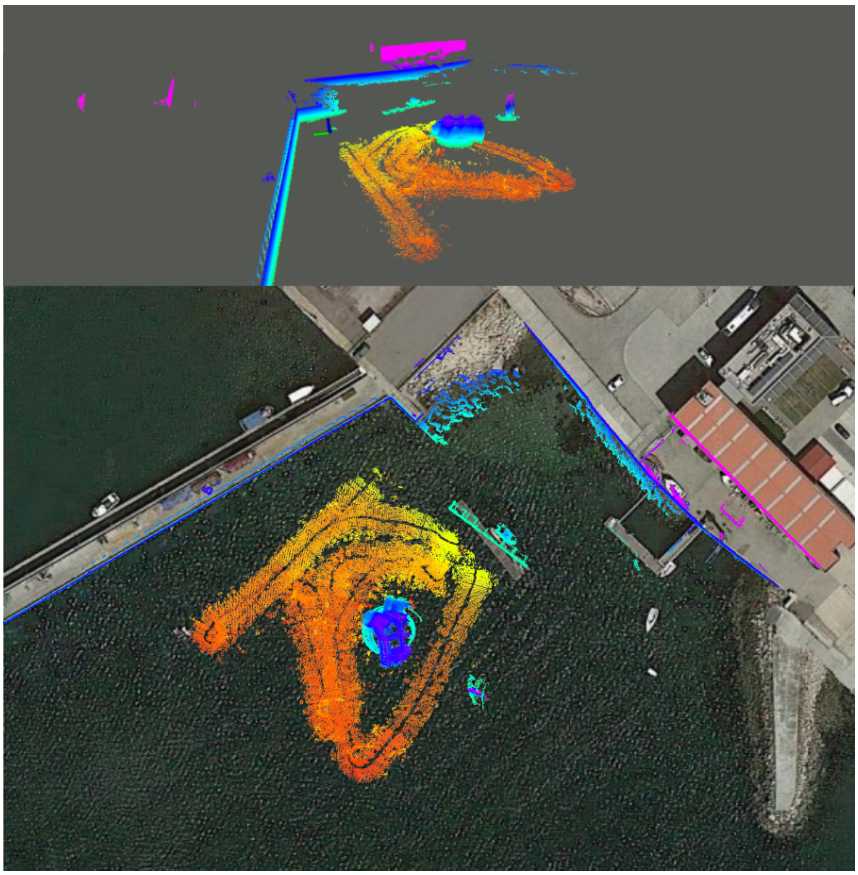


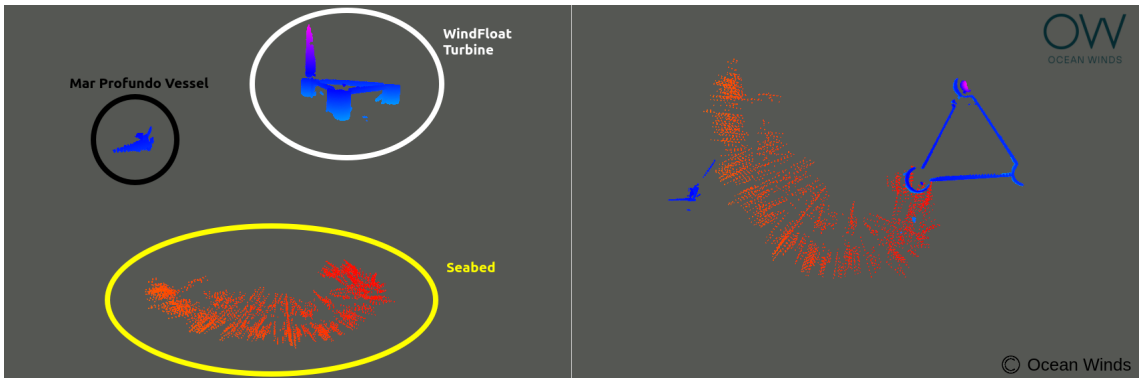
Figure 4.23: Aerial view of the multi-domain map of the *Marina de Leça* harbor yacht basin.

During the trials at the ATLANTIS coastal test centre, multiple multi-domain data acquisition campaigns using the Nautilus USV were performed. This location has water salinity variations that traditionally affect MBES readings as highlighted in subsection 3.5.2, which are highly dependent on the tides. Applying the M2A during low tide can reduce the impact of water mixture on the immersed domain map. This enhanced detail is attributed to the reduced occurrence of spurious data from the MBES sensor. Figure 4.24a shows the multi-domain map of the ATLANTIS coastal test centre. The odometry coordinates were aligned with RTK and IMU information, creating the integration with the satellite view. This alignment is demonstrated by a accurate overlapping of the map representation with the static structures in the scenario, including buildings and port infrastructures.

Finally, the multi-domain map of the Windfloat foundation, shown in Figure 4.24b, was created using the data collected during Nautilus offshore trials. A detailed reconstruction of the emersed



(a) Multi-domain map of the ATLANTIS coastal test centre.



(b) Multi-domain map of the Windfloat Atlantic offshore wind farm.

Figure 4.24: Multi-domain mapping from Nautilus trials in the ATLANTIS coastal test centre and in the Windfloat Atlantic offshore wind farm.

domain of the foundation was obtained using LiDAR information and the immersed component was sparsely reconstructed because the seafloor was close to the maximum range of the MBES (100 m) which is estimated to be deeper than 96 m in the trial zone according to the Portuguese

Navy Hydrographic Institute charts [2]. The CT-ICP LiDAR-based odometry algorithm was used to localize the USV, because the large structure created reflections on the GPS data when the USV navigated nearby. Moreover, the Windfloat turbine was equipped with a RADAR that disrupted the GPS signal lock, leading to intermittent data and even signal loss near the floater.

4.4.2 FRONTIER - Offshore wind farm foundations data completion architecture

The performance of FRONTIER was evaluated through a component-wise validation approach, encompassing the segmentation of potential candidates, the performance of the learning-based classification model for offshore wind turbine foundations, and the precision of the template-based instance registration. Both simulated and real-world data were used for these tests. Moreover, to validate the proposed classification model, a two-stage approach was employed. The first stage involved comparing the model's performance against state-of-the-art algorithms using a benchmark dataset. The second stage used an application-specific synthetic dataset to train and test the model. This dataset was generated using a simulation environment that included the Nautilus model equipped with a multi-domain sensor payload (LiDAR and MBES) affected by Gaussian noise.

Segmentation of potential candidates

Real-world LiDAR and MBES data acquired with Nautilus USV during the approval in principle stage at the ATLANTIS test centre (see section 3.5) was used to evaluate the segmentation of potential candidates. The testing scenario at the ATLANTIS coastal test bed, figure 4.25, features several structures, including the Durius foundation (the primary object of interest), a dock whose

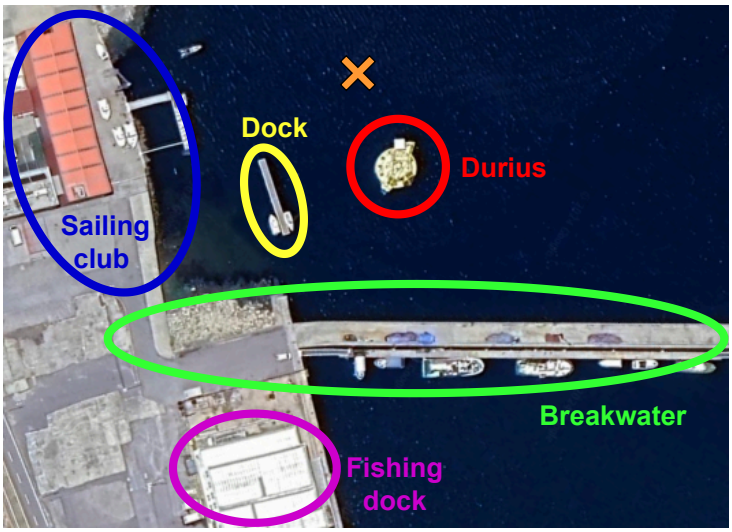
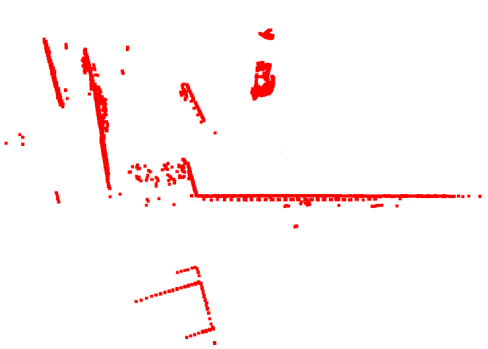


Figure 4.25: Aerial view of the ATLANTIS coastal test bed.

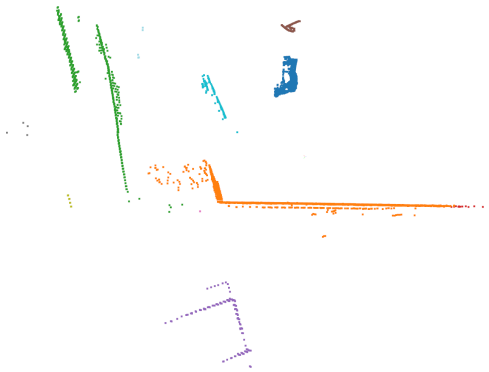
[2]Hydrographic Institute, 2023. *Hidrográfico Mais*. URL:`https://geomar.hidrografico.pt/`

position and orientation vary, a breakwater structure, a fishing dock building, and the sailing club infrastructure with a hangar, crane, and fixed dock. Additionally, the scenario include multiple mooring places (e.g., in the orange cross) and dock occupancy varies according to the sail club usage.

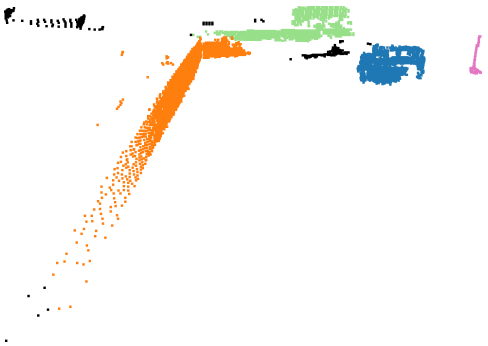
The data acquired by the LiDAR (figure 4.26a) was segmented using BEVA-FPC, with the results visualized in figure 4.26b, being parametrized with 128 initial clusters and a distance radius of 12 meters for grouping labels. These parameters facilitated the consolidation of the initial cluster set into 11 final clusters. BEVA-FPC successfully segmented the points belonging to Durius (blue cluster) even in the presence of nearby elements like a moored sailboat (pink cluster). Subsequently, the filtering process removed spurious clusters, defined as those containing less than 10 points or having a height bellow 2 meters. This filtering resulted in the four objects of interest depicted in figure 4.26c. For the immersed data, the two-stage clustering method from subsection 4.3.1 identified four clusters, as shown in figure 4.26d. The green, dark blue, light blue and grey points correspond to the seafloor, multi-echoes, an object near the seafloor (e.g., a fish), and



(a) Bird's-Eye View of the scenario.



(b) Results of BEVA-FPC clustering on the LiDAR data. The segmentation process identified eleven distinct clusters.



(c) Results of the cluster filtering. This process reduced to four clusters, where the filtered ones are depicted by the black points.

(d) Results of the 1D clustering on the MBES data. The segmentation process identified four distinct clusters.

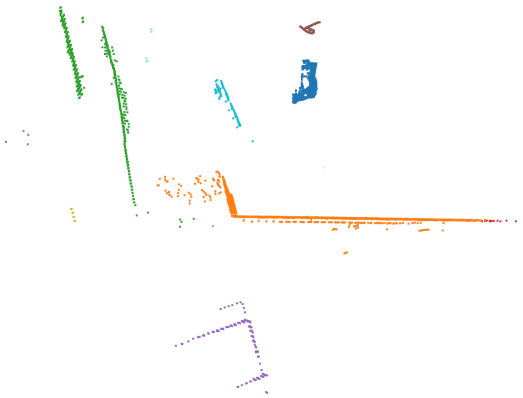
Figure 4.26: Results of the clustering algorithms applied to LiDAR and MBES data acquired on the ATLANTIS coastal test center.

an outlier measurement, respectively.

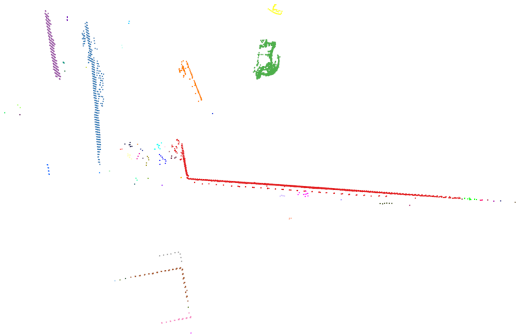
Moreover, to benchmark BEVA-FPC performance against established clustering methodologies from popular open-source libraries for 3D data processing, comparisons were performed with the euclidean clusters from Point Cloud Library (PCL) [208] and the DBSCAN implementation of Open3D [209]. To ensure a consistent testing environment for all methods, no minimum cluster size limitation was imposed (i.e., clusters could contain a single point). Parameter calibration was performed to optimize the execution time while achieving distinct segmentation of points belonging to Durius. The results of these comparisons are presented in table 4.4 and figure 4.27. As shown, the BEVA-FPC clustering procedure demonstrates the capability to segment the scenario into a significantly lower number of clusters while achieving at least a three-fold improvement in processing speed using the LiDAR point cloud.

Table 4.4: Comparison of segmentation results for the scenario depicted in figure 4.26a.

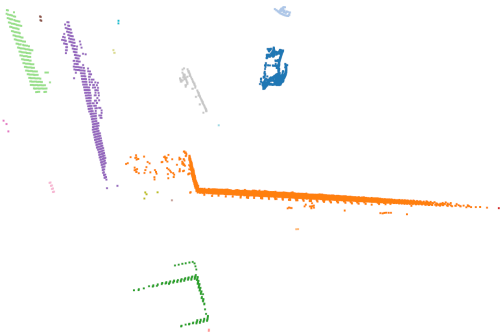
Method	**Number of Clusters**	**Time (s)**
Euclidean Clusters [210]	58	0.28
DBSCAN [211]	19	0.80
BEVA-FPC	**11**	**0.09**



(a) Segmentation results using BEVA-FPC.



(b) Segmentation results using Euclidean clusters.



(c) Segmentation results using DBSCAN.

Figure 4.27: Comparison of the segmentation results.

Learning-based classification of offshore wind turbine foundations

After segmenting the point clouds of the multi-domain map, a list of candidates are elaborated based on the clusters. The point cloud belonging to each emersed cluster is evaluated independently by the RI-PointTransf to characterize the foundation type among 15 different structures that includes a background label (capturing a non-foundation structure). A two-stage approach was adopted to evaluate the performance of the RI-PointTransf composed by:

- *Stage 1*: Evaluation of the performance compared to other methods and an ablation study using a benchmark dataset for 3D data classification tasks. This step assesses the model prediction capabilities and the impact of design choices;
- *Stage 2*: Assessment of the performance of the model when trained and tested using a application-specific synthetic dataset for offshore wind turbines foundation classification. It also validates the model's ability to generalize from synthetic data to real-world scenarios.

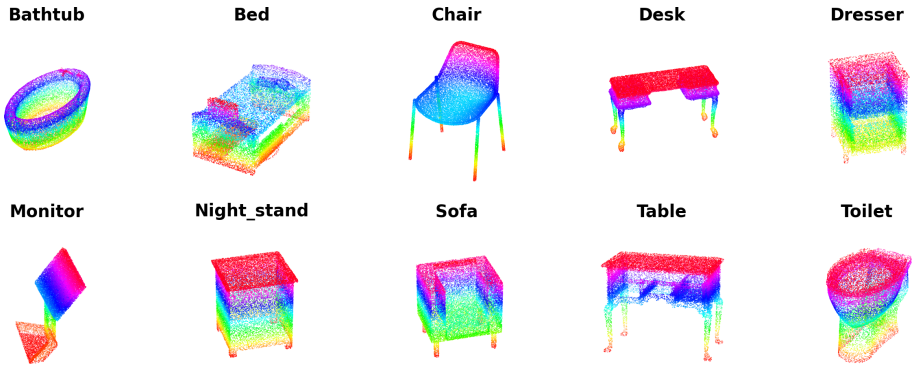
The training process was conducted within a deep learning environment using Ubuntu and Python 3.8. Table 4.5 details the learning framework, corresponding hardware/software environments, and parameters used during training.

Table 4.5: Experiment configurations.

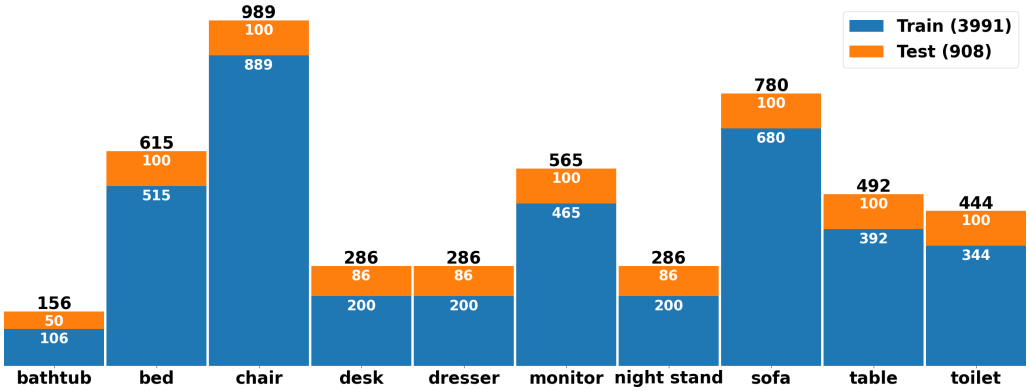
Environment Configurations		**Model Parameters**	
Name	*Configuration*	*Name*	*Value*
CPU	Intel i5-8600k	Batch size	32 (64)*
GPU	Nvidia RTX 2080	Max Epoch	60
RAM	16 Gb	Optimizer	AdamW [212]
Cuda	12.1	Learning Rate	3×10^{-4}
Operating System	Ubuntu 20.04	LR scheduler	ReduceLROnPlateau [213]
Learning Framework	Pytorch 2.2.0	Loss	Focal Loss [214]
		Patches	256 (16)*
		Points per patch	32 (32)*
		Anchor points	5 (5)*

* *ModelNet10 (Foundation dataset)* - Parameters used for ModelNet10 and foundation dataset train and test.

In the first stage, the model was trained and tested using a benchmark dataset for 3D data classification tasks, the ModelNet10 [215]. This dataset comprises ten classes with household objects (as depicted in figure 4.28a), and consists of 4899 CAD models. The data is inherently divided for training and testing purposes, with 3991 models allocated for training and 908 for testing. Some classes exhibit a high degree of inter-class similarity, such as desks and tables, or dressers and nightstands. Moreover, the number of samples per class are imbalanced, as illustrated in figure 4.28b. This class imbalance can potentially hinder the system's accuracy by introducing a bias towards classes with a higher number of samples during the learning and classification processes. To mitigate the potential impact of the class imbalance inherent from the ModelNet10 dataset, several techniques were employed during the training of RI-PointTransf, namely:



(a) Sample data points from each class within the ModelNet10 dataset. Each sample is a 3D point cloud representation of a specific object category.



(b) Distribution of objects across classes within the ModelNet10 training and testing datasets. The black numerals indicate the total number of objects per class, while white numerals represent the number of objects belonging to each class within the corresponding dataset (training or testing).

Figure 4.28: ModelNet10 data samples and distribution.

- *Oversampling*: Using Synthetic Minority Over-sampling Technique (SMOTE) [216], the amount of data from minority classes was increased and Gaussian noise was applied randomly at each epoch to them, aiming to create a more balanced distribution within the training set while providing more variability in the oversampled data;

- *Class Weights*: Employs the method proposed by King and Zeng (2001) [217] to estimate the weights, assigns higher values to under-represented classes during training, encouraging the network to focus on learning these classes more effectively by applying this in the loss function;

- *Focal Loss*: Exploits the focal loss function proposed by Lin *et al.* (2017) [214]. It applies a modulating term to the cross entropy loss to focus the learning process on hard examples. This reduces the contribution of easily classified samples, placing greater emphasis on learning from harder-to-classify examples, particularly those belonging to under-represented classes.

Therefore, the performance of RI-PointTransf compared to other state-of-the-art classification models on the ModelNet10 dataset is presented in table 4.6, with all models evaluated under identical conditions. As shown, NR/NR means that the original, non-rotated data is used for both training and testing. NR/AR assesses the model's ability to generalize to unseen rotations by testing on a dataset that underwent random rotations after training with the original data. The drop value indicates the performance decay observed during rotation experiments. These comparisons demonstrate that the proposed RI-PointTransf method achieves superior accuracy in both testing conditions (NR/NR: 92.03%, NR/AR: 90.81%). However, unlike SRINet [218] and Tao et al. (2021) [219], which exhibit total rotation invariance (0% decay), RI-PointTransf shows a slight performance decrease (1.22% decay) under random rotations. This can be attributed to two factors. First, spherical coordinates employed by RI-PointTransf ensures rotation invariance only for rotations around the Z-axis. Second, the orientation normalization process relies on the convex hull and the oriented bounding box creation, which has potential approximation errors that can introduce a minor sensitivity to angular variations. It is important to note that for the intended application of using USVs to classify foundations, this sensitivity is less critical. USVs primarily experience yaw rotations, while roll and pitch are mainly induced by wave action. Additionally, the observed performance decay in RI-PointTransf remains relatively small, and its overall accuracy surpasses state-of-the-art approaches.

Table 4.6: Result comparison on ModelNet10 test dataset. NR/NR - No rotation; NR/AR - No rotation training, random rotation testing; Drop - Decay rate from NR/NR to NR/NA experiments.

Method	**NR/NR (%)**	**NR/AR (%)**	**Drop (%)**
PointNet [112]	91.47	27.15	64.32
Spherical CNN [220]	90.25	82.13	8.12
PRIN [221]	83.26	75.42	7.84
RICNN [222]	88.92	88.61	0.31
SRINet [218]	90.08	90.08	**0**
Tao *et al.* 2021 [219]	90.63	90.63	**0**
RI-PointTransf	**92.03**	**90.81**	1.22

Figure 4.29 presents the normalized confusion matrix for RI-PointTransf on the NR/NR test data from ModelNet10. The rows represent the true labels, columns the predicted labels, and the values represent the percentage of predictions for each class over the true condition. As shown, the network achieved a strong performance across all classes, with accuracy exceeding 77% for each, where chairs, monitors, and sofas exhibited the highest accuracy, while bathtubs, night stands, and desks showed the lowest performance. The latter corresponds to the under-represented classes, limiting the network's ability to generalize effectively. Additionally, as expected the miss-classification is higher between desks and tables (11%), and night stands and dressers (18%), which can be attributed to the inherent similarity between the data in these classes, as well as to a lack of distinctive features among them. In such cases, the model converged towards a dominant class (e.g., dresser and table).

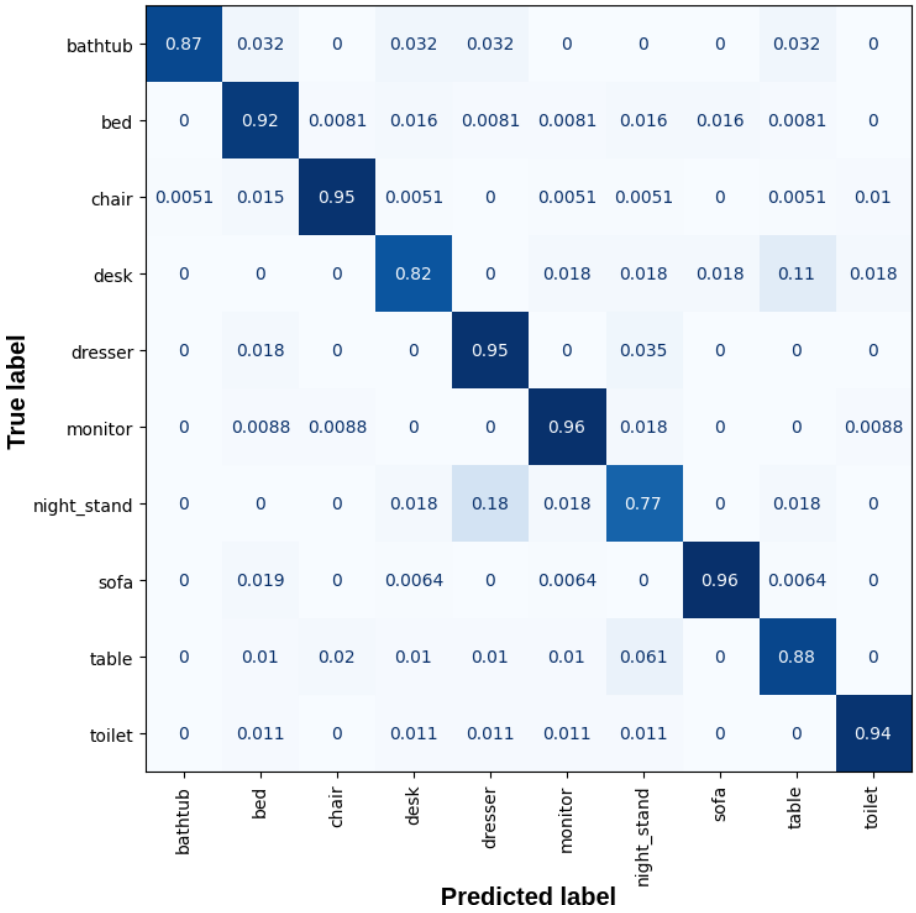


Figure 4.29: Normalized confusion matrix for the ModelNet10 NR/NR test data.

Furthermore, an ablation study was conducted in Stage 1 to analyse the design choices within RI-PointTransf and assess their individual contributions to performance. This involved removing systematically specific components of the model and evaluating the resulting impact on the accuracy. The results are presented in tables 4.7a and 4.7b, where the components that were analysed in the ablation study were:

- *Positional Embedding (No-PE)*: replace the custom positional embedding with the standard learnable parameter approach proposed by Dosovitskiy *et al.* (2021) [203].

- *Squeeze-and-Excitation Block (No-SE)*: delete the SE block using directly the concatenated patch and anchor feature vectors.

- *Convolutional Block Attention Module (CBAM)*: swap the SE block with CBAM to compare the effectiveness of the chosen channel attention module.

- *Residual Connection (No-Res)*: remove the residual connection between the the transformer input and output to understand its influence on the learning process.

- *Spherical Coordinates Transformation (No-Sph)*: exclude the spherical coordinate transformation, feeding the network with the standard Cartesian coordinates of the point cloud data.

Table 4.7a summarizes the impact of design choices within RI-PointTransf on its performance. As observed, the full RI-PointTransf configuration achieves the highest overall accuracy (global accuracy) for both NR/NR and NR/AR testing procedures. It also demonstrates the best average class accuracy and the lowest standard deviation, indicating consistent performance across object categories. These findings suggest that all included components contribute positively to the model's effectiveness. Despite the CBAM exhibiting the lowest decay rate, its overall accuracy and high standard deviation (σ_{acc}) shows less consistent performance across all classes, thus it is not an effective replacement of the SE block. Removing the Squeeze-and-Excitation (SE) block and the positional embedding leads to substantial accuracy drops exceeding 2.5% and 3%, respectively. This indicates the importance of the SE block in focusing on critical features and the positional embedding's role in incorporating spatial information within patch embeddings. The residual connection removal also contributes to an accuracy decrease of over 1.5% for both test scenarios, underlining its influence on the learning process. The removal of the spherical coordinate transformation (No-Sph) results in the largest decay rate compared to other ablations (2.35%). This highlights the crucial role of this component in enhancing rotation robustness alongside the orientation normalization step.

Table 4.7: Ablation study results using ModelNet10. NR/NR - No rotation, NR/AR - No rotation training, random rotation testing; Drop - Decay rate of rotation experiments; μ_{acc} - Class accuracy average for NR/NR; σ_{acc} - Class accuracy standard deviation for NR/NR; σ_{cls} - Standard deviation for each class in NR/NR.

(a) Performance comparison for the component tests.

Method	**NR/NR (%)**	**NR/AR (%)**	**Drop (%)**	μ_{acc} **(%)**	σ_{acc} **(%)**
No-PE	88.97	87.33	1.64	86.49	8.00
No-SE	89.48	87.74	1.74	86.49	8.05
CBAM	88.66	87.54	**1.12**	85.07	9.72
No-Res	90.40	89.07	1.33	88.74	6.65
No-Sph	91.22	88.87	2.35	88.63	8.22
RI-PointTransf	**92.03**	**90.81**	1.22	**90.22**	**6.43**

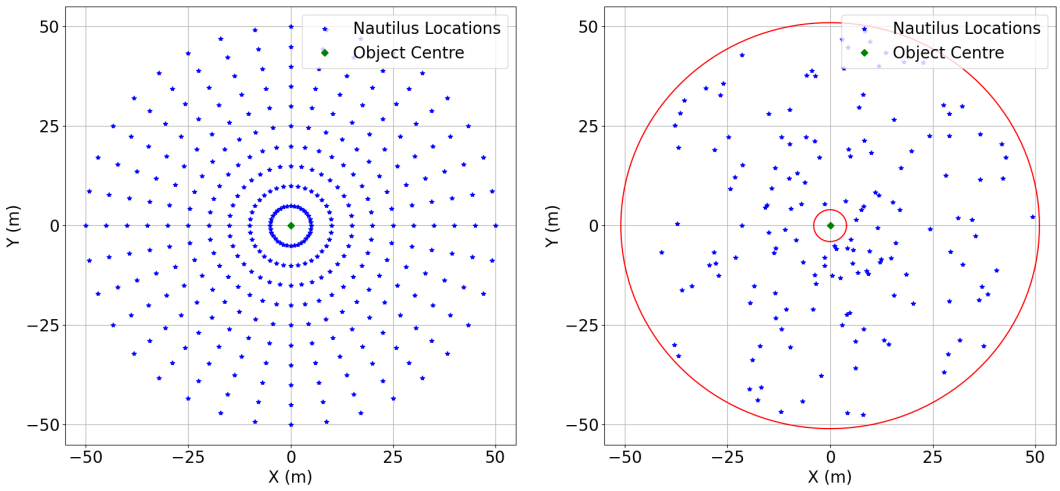
(b) Accuracy per class for NR/NR test.

Model	**Bathtub**	**Bed**	**Chair**	**Desk**	**Dresser**	**Monitor**	**Night stand**	**Sofa**	**Table**	**Toilet**
No-PE	83.87	80.49	95.96	80.70	80.70	96.46	77.19	95.51	79.59	94.38
No-SE	77.42	82.93	95.96	78.95	80.70	96.46	78.95	95.51	83.67	94.38
CBAM	70.98	87.80	**96.46**	71.93	80.70	**97.35**	**85.96**	94.23	76.53	88.76
No-Res	**87.10**	83.74	95.45	**84.21**	85.96	95.58	82.46	95.51	79.59	**97.75**
No-Sph	**87.10**	89.43	**96.46**	75.43	87.72	95.58	77.19	**98.08**	82.65	96.63
RI-PointTransf	**87.10**	**91.87**	94.95	82.46	**94.74**	95.58	77.19	96.15	**87.76**	94.38
σ_{cls} (%)	6.69	4.34	***0.59***	4.58	5.63	*0.72*	3.64	*1.27*	3.93	3.1

Table 4.7b provides a more granular view of performance across individual classes. Notably, classes with high accuracy in the full model (chair, monitor, sofa) exhibit consistent performance across all ablated configurations. This suggests that their good performance stems from being well-represented in the training data. Conversely, under-represented classes (bathtub, desk, dresser, nightstand) show larger variations across different configurations. Despite not achieving the highest class accuracy in all cases, the full model demonstrates the lowest variability across these classes. This suggests a trade-off between achieving a top performance in specific classes and maintaining a more balanced and generalizable classification capability. Overall, the ablation study confirms the design choices implemented in RI-PointTransf effectiveness. Each component contributes significantly to the model's superior performance and generalization capabilities.

Subsequently, a synthetic dataset was created in the second stage using Gazebo and employed to train and test the RI-PointTransf for its intended application. This dataset uses CAD models of real-world structures to represent the simulated environment. Furthermore, realistic environmental parameters were incorporated, including hydrodynamics, Nautilus kinematics, and sensor models, to closely resemble real-world conditions. Subsequently, LiDAR and MBES data were simulated for the 14 foundation types illustrated in figure 4.10. Additionally, a background class encompassing vessels, buildings, and port structures was defined, resulting in a total of 15 classification categories. The primary advantage of employing a simulated environment lies in its ability to automatically generate vast quantities of labelled data encompassing diverse scenarios. This benefit was harnessed through the development of an automated data capture procedure that acquired samples from multiple viewpoints. To introduce variability within the training and testing sets, distinct viewpoints and noise configurations were implemented.

For training data acquisition, the simulated Nautilus model traversed the structure along circular trajectories (figure 4.30a). These trajectories consisted of positions incremented by 10 degrees



(a) Train data acquisition procedure.

(b) Test data acquisition procedure.

Figure 4.30: Data acquisition procedures.

in circumference and progressively increasing radius in increments of 5 meters. At each location Nautilus presented nine different yaw orientations[3]. The test set positions were generated randomly, following a uniform distribution for both angle and distance (figure 4.30b) each with three different yaw orientations between -90°and 90°. Both sets incorporated distances ranging from 5 meters to 50 meters, offset by the radius of the circumscribing circle centred on the structure. This range selection, limited to half the sensor's maximum range of 100 meters, ensured sufficient data points for close-range navigation while maintaining a lower bound corresponding to a safe distance from the structure. Two separate data generation runs were conducted, each incorporating Gaussian noise with standard deviations of 0.03 and 0.1, simulating the Nautilus LiDAR (Ouster OS-128) range accuracy for lambertian targets and retroreflectors, respectively. Additionally, four variations of each foundation CAD model were used, incorporating different levels of detail such as ladders or protective rails. All CAD were approximations of real-world structures designed for this research based on established references in the field [223, 224, 225, 226, 227]. A similar approach was employed for the background classes, leveraging available CAD designs for vessels, buildings, and port infrastructure. In the end, the dataset provides a 70%-30% train-test split, where each class comprised 26640 training point clouds and 11840 testing point clouds.

This dataset was used to train and test the RI-PointTransf network for all 15 classes containing several foundation types and background elements, where the training procedure adopted principles similar to those used for ModelNet10, without any pre-training stage. Since the LiDAR data from the segmented foundations exhibited a smaller volume compared to the full 3D models in ModelNet10, adjustments were made to the training configuration, namely the number of patches considered was reduced to 16, each containing 32 points. The number of anchor points was maintained (5). The reduced data size allowed the use of a larger batch size composed by 64 point clouds, thus speeding up the training process and allowing for a better generalization capacity. This yielded a global accuracy of 98.97% on the test set, with a standard deviation (σ_{acc}) of 2.02%. The detailed class-wise accuracy breakdown is depicted in table 4.8.

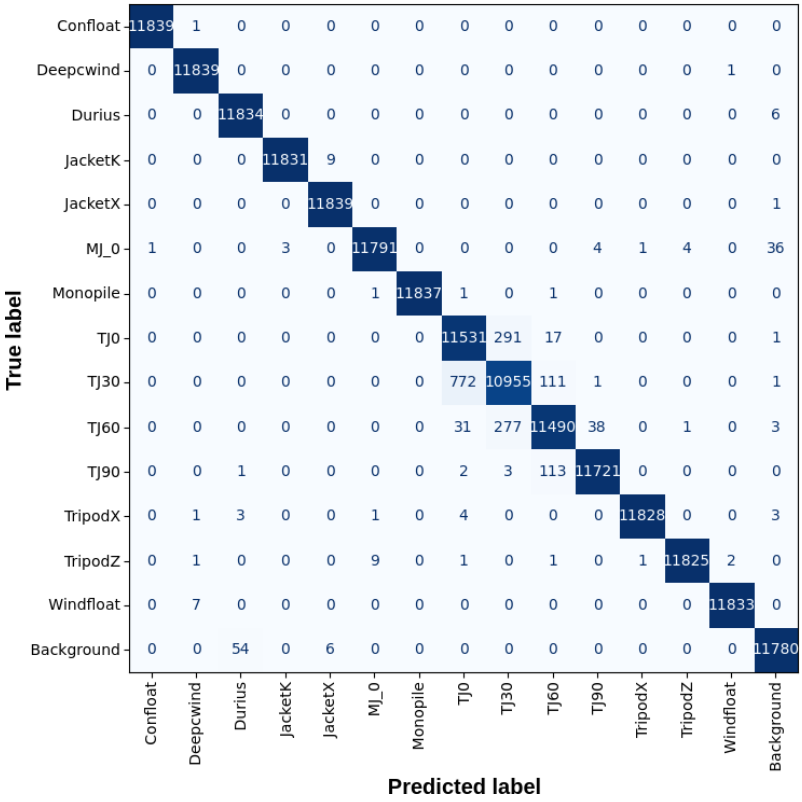
Overall, these results demonstrate the effectiveness of RI-PointTransf in learning from the synthetic dataset, achieving high accuracy in foundation classification. A closer examination of the confusion matrix (figure 4.31a) reveals specific challenges related to the "TJ" foundation classes.

Table 4.8: Accuracy per class using foundation datasets. The global accuracy is 98.97% and the class accuracy standard deviation, σ_{acc}, of 2.02%

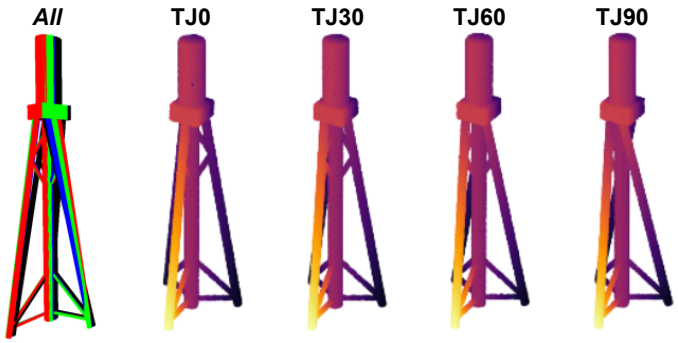
Model	**Accuracy (%)**	**Model**	**Accuracy (%)**	**Model**	**Accuracy (%)**
Confloat	99.99	Deepcwind	99.99	Durius	99.95
JacketK	99.92	JacketX	99.99	MJ_0	99.59
Monopile	99.97	TJ0	97.39	TJ30	92.53
TJ60	97.04	TJ90	98.99	Tripod X	99.89
Tripod Z	99.87	Windfloat	99.94	Background	99.49

[3]The nine training orientation are $-90°, -60°, -45°, -30°, 0°, 30°, 45°, 60°,$ and $90°$.

These classes exhibit the most frequent misclassification between their closest variations, particularly *TJ0*-TJ30, TJ0-*TJ30*-TJ60, TJ30-*TJ60*-TJ90, and TJ60-*TJ90*. As detailed in Chen *et al.* (2016) [224], these variations involve minimal geometric changes, primarily limited to twisting the jacket legs to different angles. Figure 4.31b provides a visual representation of the high similarity between these foundation templates, where *All* represents the overlapped CAD templates. Despite this ambiguity, the model maintains an accuracy exceeding 92% even for these challenging classes,



(a) Confusion matrix for the test set of the foundation dataset.



(b) Different types of TJ foundations. The data in *All* represents the overlapped templates.

Figure 4.31: Foundation dataset performance evaluation: confusion matrix and most ambiguous class comparison.

which demonstrates the performance of RI-PointTransf and its feature extraction capabilities.

Furthermore, to assess the generalization capacity of the model trained solely on synthetic data, its performance was further evaluated using real-world data. This data, collected during the Nautilus approval-in-principle and technology qualification stages, included observations of Durius and WindFloat foundations alongside background classes such as vessels and port infrastructures. This evaluation aimed to quantify the sim-to-real gap, which refers to the difference in performance between the synthetic training domain and the real-world test domain. For robust evaluation, a test dataset was created by selecting fifty point clouds captured by the Nautilus' LiDAR system for each class considered in the experiment. The Durius data included changes due to environmental conditions, as well as its inclined state before maintenance due to weight imbalance and its upright position after maintenance. WindFloat data originated from offshore trial 3, where due to safety requirements all measurements were taken at a distance superior to 20 meters.

The RI-PointTransf model, trained on the previously described synthetic dataset, achieved an accuracy of 96.67% on real-world data, compared to 99.79%, considering only the global accuracy for the evaluated classes using the synthetic dataset. This difference of 3% represents the sim-to-real gap, indicating a successful adaptation from the synthetic training domain to the real-world test domain. As shown in figure 4.32, the model achieved total detection for Durius foundations. However, two false negatives were observed for Windfloat and three for background classes. It

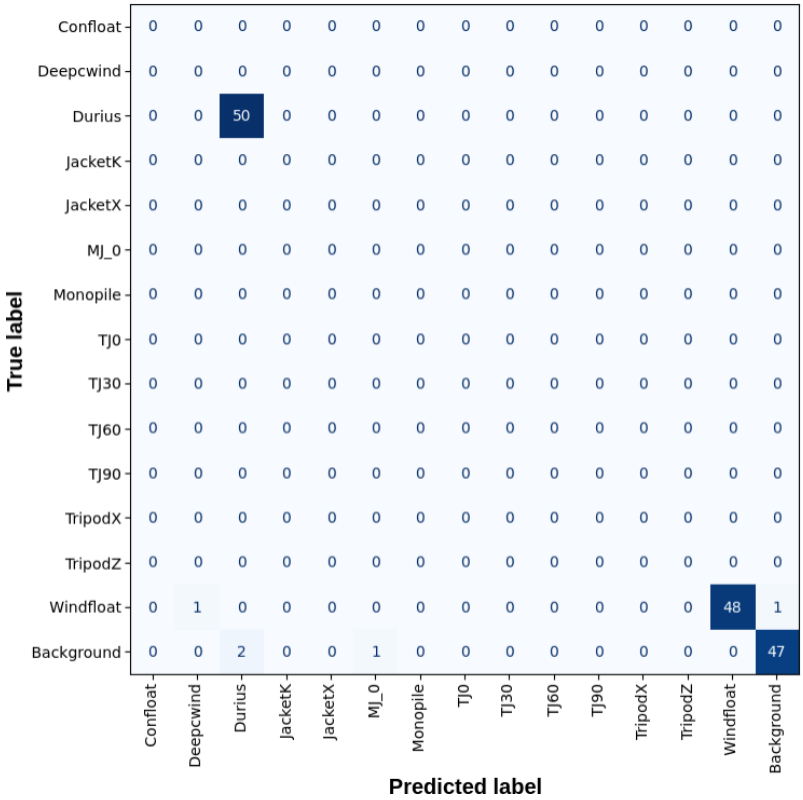


Figure 4.32: Confusion matrix for Durius and Windfloat foundation, and background point clouds captured during the Nautilus trials.

is noteworthy that due to the limited size of the real-world dataset, a single false negative can significantly impact performance metrics (around 2% drop). Consequently, with more real-world data, the sim-to-real gap might be even smaller.

The classification alongside with the oriented bounding box from the patchifier stage and the confidence estimation derived from the softmax of the RI-PointTransf output, provides the 3D detection of the foundation, as illustrated in figure 4.33. Moreover, the model enables real-time application with an average processing time of 0.05 s. The majority of this time is attributed to the patchifier step (approximately 0.045 s), while the model inference itself takes roughly 0.005 s. With this the object for completion is detected, ensuring the selection of the candidate to align with the corresponding structure template.

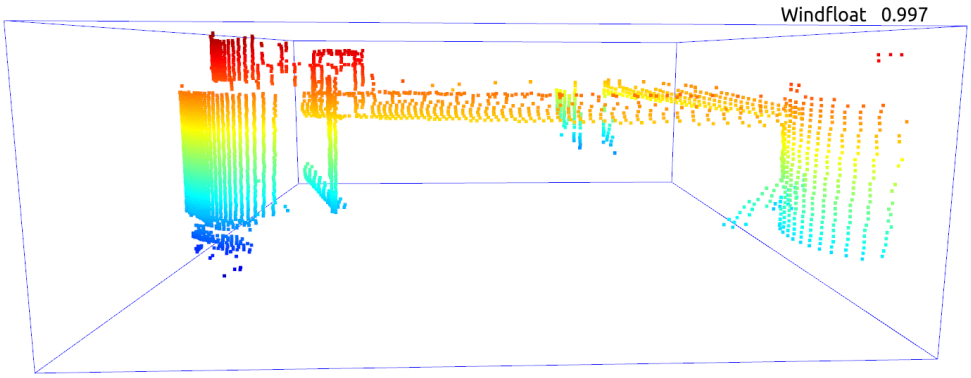


Figure 4.33: Example of the detection output of the WindFloat data with 99.7% confidence.

Template-based instance registration

FRONTIER uses the prior-knowledge of the structures (CAD models) for subsequent template-based instance registration. Several test cases were employed to validate the registration. These included fixed and floating foundations, as well as real-world point clouds registered against the approximated template. The results are depicted in figure 4.34.

For JacketK and Confloat (left and middle clouds), which represent purely synthetic data, the detected foundation (blue points) aligns accurately with the template (yellow points). However, due to the symmetrical nature of these structures, the orientation may not accurately depict the scenario. This potential orientation discrepancy mainly impacts the position of the blades, which fall outside the field of view of the USV and are excluded from the inspection volume. Moreover, Durius (right cloud) demonstrates an accurate alignment between real-world data acquired from the ATLANTIS coastal test centre and the approximated CAD template. It is noteworthy that the Durius CAD design exhibits discrepancies compared to the actual data, namely some elements are absent from the CAD model, and some measurements are slightly offset. The evaluation of the RMSE presents values higher than expected when compared to the visual alignment: 0.389 m for JacketK, 0.559 m for Confloat, and 0.139 m for Durius. This discrepancy arises from the sparsity

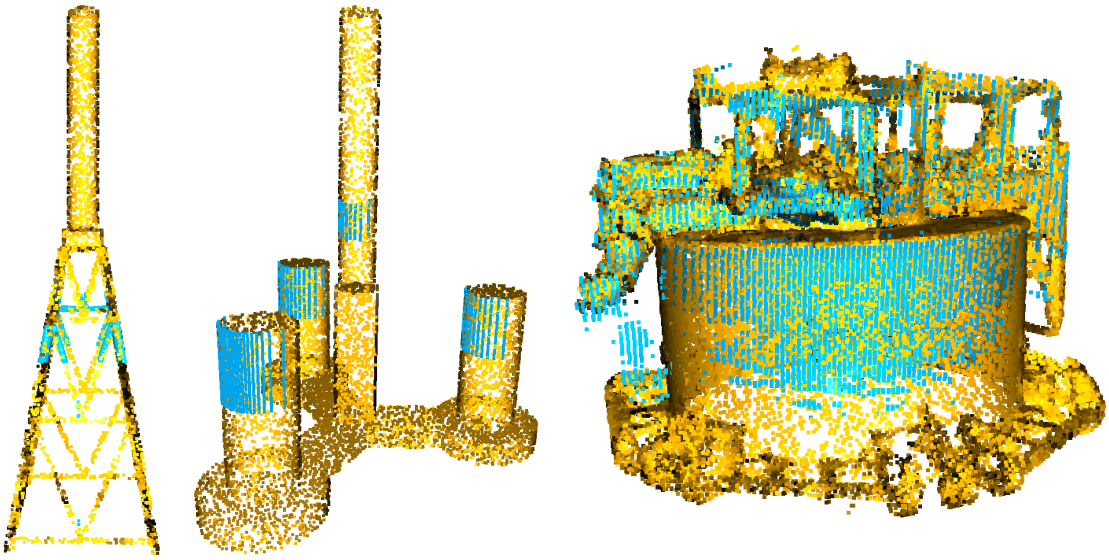


Figure 4.34: Registration of the CAD template (yellow points) to the detection output (blue points) for: fixed foundation JacketK on the left and two floating foundations Confloat in the middle and Durius on the right.

of the LiDAR data relative to the templates, coupled with inherent noise, posing a challenge for the GICP algorithm in establishing precise point correspondences for the evaluation.

At last, figure 4.35 illustrates the outcome of applying the FRONTIER algorithm to the point cloud acquired during the Nautilus approval-in-principle trials (shown in Figure 4.26a). As observed, the semantic instance completion process significantly extends the Durius representation by incorporating previously occluded information into the raw data (such as structural information from the immersed side of Durius that is outside the FoV of MBES). This expanded representation enhances safety by enriching the available data and providing insights into the variability between

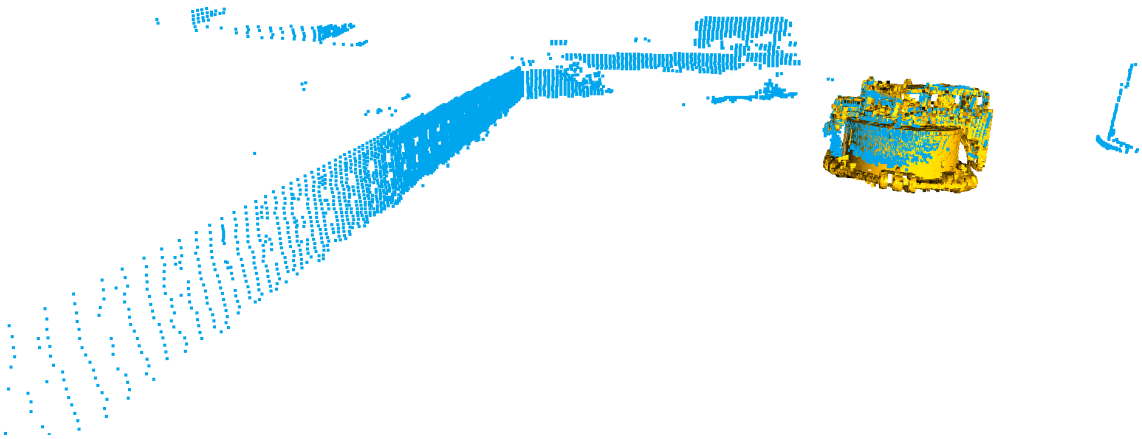


Figure 4.35: Result of the FRONTIER applied to a point cloud acquired during the Nautilus approval in principle trials.

the real and modelled structures. Additionally, this information can be used to update the USV's inspection plan toward specific goals, optimize its navigation during maneuvers by considering the complete representation, enhance safety for close-range inspections, and enable a semantic representation of the scenario for human-in-the-loop strategies.

4.5 Conclusions

This chapter presents two novel multi-domain methodologies, M2A and FRONTIER, specifically designed to tackle the challenges encountered during offshore wind farm inspections. These methodologies enables the creation of multi-domain maps and the completion of missing information through semantic instance completion. Both methodologies offer modular and flexible architectures with unique capabilities, providing enhanced multi-domain maps for various purposes, such as multi-domain inspection of OWFs, increased navigation safety by providing comprehensive scenario information, and semantic representation of the scenario for human-in-the-loop strategies.

The M2A proposes an architecture for multi-domain mapping of offshore wind farms infrastructures using heterogeneous sensors, namely a LiDAR and a MBES, to create a 3D reconstruction of the scenario. The method is capable of reconstructing both emersed and immersed domains into a single coherent representation. An accurate odometry was obtained by fusing the navigation sensors (GPS and IMU), providing a valuable reference to define the underwater sliding window, the LiDAR filtering and the initialization of the LM-ICP registration. Throughout the experiments, the M2A showed promising results towards multi-domain mapping. The odometry process was capable of providing the USV poses with a positional error of at most 0.049 m on simulation and 0.07 m on real data. Moreover, the mapping methodology provided a visually accurate and soft real-time representation of the scenario, requiring an average of 300 ms to reconstruct the environment. In simulation, a RMSE score for the multi-domain map comparison with the ground-truth of at most 0.042 m was obtained, which provides a good accuracy of the representation. Tests performed on real world scenarios demonstrated the capability for creating visually accurate representations using two distinct USVs. Furthermore, comparison of measurements to dimensions of infrastructures demonstrated centimetric-level accuracies.

Furthermore, the FRONTIER provides a semantic instance completion approach designed to operate within the worst case scenario, namely with partial observations. This framework leverages multi-domain data to select an appropriate structure template from a pre-defined database. By doing so, FRONTIER can effectively complete missing information belonging to the foundation that is not being observed, thereby augmenting the available representation. FRONTIER's processing framework performs segmentation using the BEVA-FPC clustering method to define potential candidate. This method efficiently segments the data into objects of interest based on their spatial separation. To further enhance the extraction of information, a 3D classification network (RI-PointTransf) is employed to categorize objects as one of the 14 predefined foundation

types or as a background class. The RI-PointTransf provides valuable detection information, including the object's oriented bounding box, class label, point cloud data, and confidence score, presenting a rotation-invariant model that achieves a high degree of accuracy. Finally, the framework utilizes a GICP approach to register the selected template with the source data, resulting in visually accurate alignments.

Over the course of the experiments, FRONTIER demonstrated promising results in semantic instance completion. The BEVA-FPC method successfully segmented foundations even in close proximity to other elements, reducing processing time by at least a factor of three compared to established clustering methodologies commonly found in popular open-source libraries for 3D data processing. The RI-PointTransf network achieved robust accuracy across all testing conditions. When evaluated on the ModelNet10 dataset, its performance surpassed other state-of-the-art techniques, with an accuracy exceeding 90.81%. Despite not being entirely rotation-invariant, the accuracy drop observed remained minimal, at approximately 1.22%. For a foundation-specific dataset, the model exhibited a high accuracy (98.97%) when trained on synthetic data. This resulted in a small sim-to-real gap of around 3% when evaluated with real-world data, demonstrating the network's strong adaptation capabilities. Furthermore, the registration method, applied to various foundation types for sensor information completion, showed visually accurate alignments even when comparing real-world data with approximated CAD templates. These results collectively demonstrate FRONTIER's effectiveness in completing foundation shapes.

In conclusion, the two multi-domain methodologies, M2A and FRONTIER, address distinct challenges encountered during offshore wind farm inspections with USVs. M2A facilitates simultaneous inspection of both emersed and immersed domains, reducing the need for multiple assets and streamlining mission logistics. FRONTIER uses a semantic instance completion approach to extend missing data, augmenting sensor information and enhancing inspection capabilities. Both M2A and FRONTIER contribute to a comprehensive approach to offshore wind farm inspections. Their combined capabilities have the potential to significantly improve the efficiency and safety of inspections, particularly for critical task like the inspection of the splash zone and scour protection monitoring.

Chapter 5

Safe navigation in offshore scenarios

More than 75% of incidents in the shipping industry are caused by human error and, therefore, the introduction of autonomous vessel technology can be a way of reducing risks [228]. However, that technology presents several challenges to be solved. An example is the ability of an unmanned surface vehicle to avoid unexpected obstacles that could appear along its course. This can be classified as a hard task when considering dynamic and unstructured environments and so, a motion planning capability to anticipate potentially dangerous situations considering multiple arbitrary obstacles, constrains of weather conditions as well as the kinematic motion of the vehicle can easily become a NP-hard problem.

This chapter is organized as: section 5.1 gives a brief introduction to the problem; section 5.2 presents the local knowledge creation architecture and defines the collision avoidance algorithm, namely the AVOA (Adaptive Velocity Obstacle Avoidance); section 5.3 illustrates the results for the local knowledge, and AVOA; and, section 5.4 highlights the main conclusions.

5.1 Introduction

Autonomous navigation of USVs in the maritime environment requires robust obstacle avoidance strategies due to dynamic obstacles, limited sensor range, and unpredictable weather conditions. International regulations like COLREGS [26] establish a framework for safe navigation, primarily focus on manned vessels, where one of the key principles is that collision avoidance maneuvers should be positive, obvious, and made in good time. This requires more preventive actions, i.e. performing evasive actions as soon as possible upon detecting a potential collision threat.

Current obstacle avoidance systems for USVs, such as SENSE and Nautilus, primarily rely on information about the immediate environment. However, for safe and efficient navigation, it is crucial to define a proactive obstacle avoidance architecture that not only considers the state of the surroundings but also potential future risks. To achieve this preventive capability, USVs require a more comprehensive understanding of their environment. This goes beyond just the

position of obstacles since factors like their size, motion state (static or dynamic), and velocity are equally important. This means that contextual information must be gathered from the onboard heterogeneous sensors using distinct detectors. However, integrating this data presents challenges due to the distinct sensor rates, positioning, data formats, and the estimated information provided by each detector. This disparity makes it challenging to maintain a consistent scene representation. A standardized local knowledge representation is needed to provide a comprehensive view of the environment, incorporating details such as the type, position, dynamics, and semantic information of objects. Subsequently, the obstacle avoidance approach can utilize this representation to define a free space configuration. This space considers the dimensions, orientation, and velocity of all objects present, enabling the determination of a safe trajectory for the USV. By leveraging this information, the proactive obstacle avoidance architecture can make informed decisions, ensuring the USV navigates safely even in dynamic environments.

This chapter introduces a novel proactive obstacle avoidance architecture for USVs that combines information potentially obtained by multiple object detection models into a standard scene representation. This information, along with the vehicle motion, is used to create the local knowledge through a topological graph. Subsequently, this contextual representation is interpreted by a real-time method for avoiding static and dynamic obstacles that is efficient in terms of computational power, maneuverability and travel time. The method is called adaptive velocity obstacle avoidance (AVOA) and estimates a collision-free velocity model based on a protective zone concept that represents the scenario combining the position and velocity information for all obstacles that are perceived by the vehicle). No external communication between the agent and the environment is required. The safe velocity is estimated through a Particle Swarm Optimization (PSO) approach that minimizes a multicriteria evaluation metric that characterizes the effort required by the USV to avoid the danger, the positional deviation towards the original course and the crossing distance to static and/or dynamic obstacles. AVOA design provides preventive capabilities, allowing to anticipate potential collisions early on and initiate appropriate maneuvers proactively.

5.2 Proactive obstacle avoidance architecture

The architecture for proactive obstacle avoidance in USV navigation employs a modular framework that addresses the challenges of heterogeneous sensor data and dynamic environments. It handles data from the USV's sensor payload, including LiDAR and cameras, to detect obstacles within the operational scenario (e.g., foundations, vessels, docks). These detections are inherently heterogeneous requiring a standardization process. This step ensures all information is represented in a consistent coordinate frame, format and converted into a specific feature representation. The landmark representation encapsulates essential details for each object, including its timestamp, pose with associated covariance, object class, and velocity. A standardized feature information is used by a local knowledge representation module that represents the information using a topological graph structure. Nodes within the graph represent the USV's odometry data and detected

features, while edges encode the relative constraints between them.

Afterwards, the processed data is provided as input to the AVOA technique. It defines a protective zone for each object, accounting for its relative pose and velocity. This protective zone information is used to describe the configuration space, which represents the set of feasible velocities that guarantee collision-free motion for the USV. To select the optimal velocity within the configuration space, AVOA employs a multicriteria heuristic. This heuristic prioritizes velocities that minimize divergence from the desired route, considering the deviation from the desired velocity, heading error, and a risk cost associated with proximity to infeasible solutions. A visual representation of this architecture is depicted in figure 5.1.

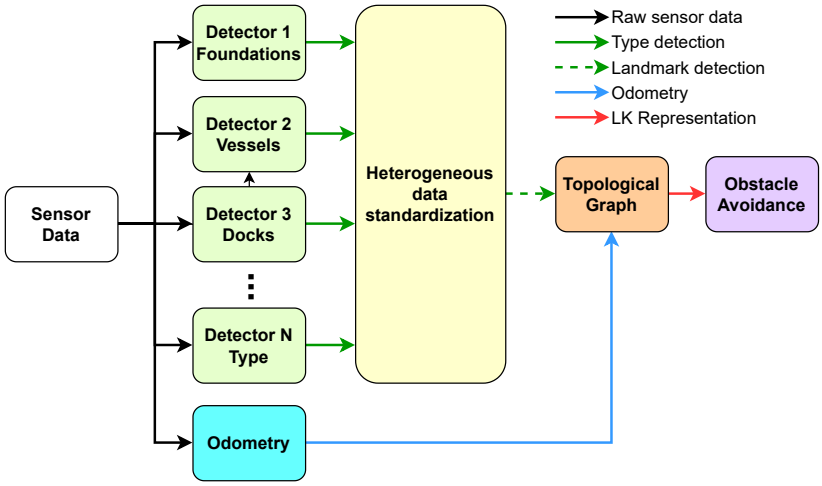


Figure 5.1: Proactive obstacle avoidance architecture.

5.2.1 Local Knowledge Representation

For USVs to effectively navigate autonomously, a robust representation of their surroundings is essential. Obstacle avoidance algorithms rely on this representation to identify obstacles, their boundaries, and dynamic information. This local knowledge represents the contextual awareness of the environment obtained by integrating sensor information into a format that correlates the measurement with the USV's current pose.

Given the diverse nature of maritime environments, which can range from sparse and featureless areas to cluttered regions with numerous features (e.g. ports), a topological graph representation is employed. This graph structure creates nodes that correspond to two key elements: vehicle pose, typically obtained from the USV's odometry system (such as the GPS/IMU fusion presented in section 3.4); and landmarks, depicting features identified by various detection algorithms operating on the vehicle sensor data. The nodes are connected by edges that encode the relative observations between them. These edges store the displacement between the connected nodes and the associated detection measurement. Additionally, prior information can be incorporated to initialize the graph. Figure 5.2 illustrates an example of a topological graph.

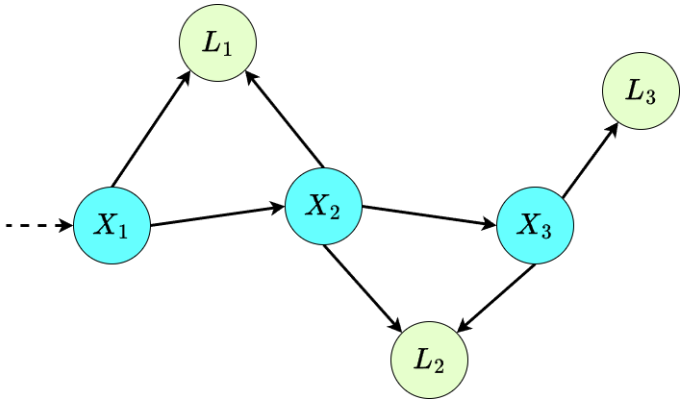


Figure 5.2: Example of a topological graph. Blue and green circles are the pose and landmark nodes, respectively. Dashed and solid arrows are prior and constraints edges, respectively.

The inclusion of odometry nodes in the topological map follows a set of pre-defined criteria to ensure efficient map representation and computational resource management. These criteria determine the frequency and distribution of odometry nodes within the map, including:

- *Minimum Time Threshold*: A minimum time interval must elapse since the inclusion of the previous node. This prevents the addition of superfluous nodes, reducing computational complexity and data redundancy;

- *Minimum Displacement Threshold*: New nodes are only added when the USV travels a certain minimum distance. This ensures the map captures only significant movements and avoids unnecessary data points;

- *Maximum Travel Distance*: To guarantee a uniform distribution of odometry nodes, a maximum travel distance is established. If the USV travels more then a maximum distance, a node is added regardless of the time criteria. This safeguards against sparse node distribution in specific areas while travelling in high speed.

These criteria provide precise control over the map's resolution, directly impacting the computational demands of map processing and updates.

The landmark nodes uses a pre-defined representation to integrate information from various obstacle detection methods employed within the USV's sensor payload. These detectors, such as the FRONTIER foundation detections, LiDAR-based dock detectors [24] or camera-LiDAR vessel detectors [82], identify relevant environmental elements. However, the use of heterogeneous detectors present a key challenge. Each detector operates independently and produces data with potentially different structures. To address this heterogeneity, a crucial step involves standardizing the landmark representation. This standardization process ensures that all landmarks, regardless of their originating detector, are interpreted uniformly within the framework, fostering adaptability and scalability, allowing for the integration of new detectors with potentially diverse output data structures. This feature representation encompasses several information for each object, namely:

- *ID*: Unique identifier for the object. It is relevant for detecting and tracking moving objects (e.g., vessels). This ID allows to maintain temporal awareness of the dynamic elements within the environment;
- *Timestamp*: Depicts the time in which the data was captured. It ensures proper synchronization between data points acquired at different detection times. It also facilitates temporal interpolation and the estimation of object velocities;
- *Reference Frame*: Defines the coordinate frame where the information is represented;
- *Object class*: Identifies the type of obstacle detected (e.g., dock, vessel, foundation);
- *Size*: Captures the dimensions of the detected object, e.g. using the bounding box;
- *Pose with covariance*: Captures the object's position, orientation, and the associated uncertainty in the detection process. The covariance reflects the variability of the data and serves as a measure of feature quality within the topological map;
- *Velocity*: Defines the object's estimated velocity, calculated using timestamps and historical pose information or tracking methodologies.

With this standard representation, landmarks can be integrated into the local knowledge based on a set of defined rules (figure 5.3), that govern how the framework handles potential redundancies and leverages information. First, any new landmark farther then a distance threshold (dashed circle) of existing landmarks is added to the map. This ensures that new obstacles are represented. Otherwise, landmarks within the distance range are considered as belonging to an existing landmark. When a detected feature deviates from an existing landmark but lands within its associated covariance ellipse, the feature's position is recalculated as the weighted average between the last two estimates. This approach refines the landmark's position within the map. For closely spaced

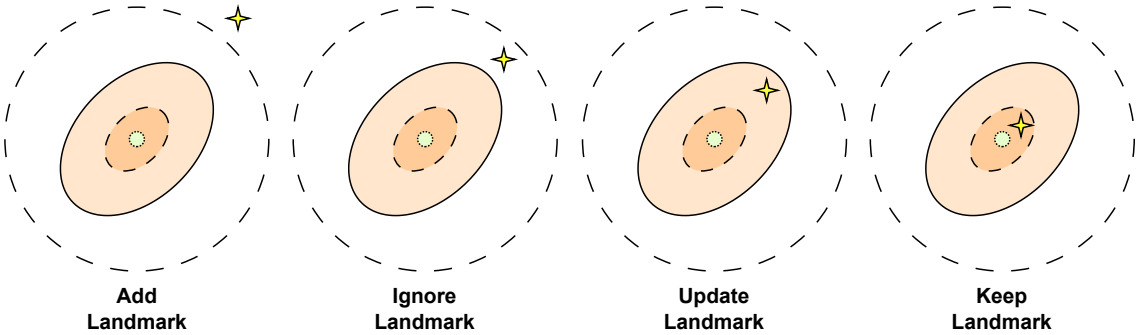


Figure 5.3: Landmark node association rules. The yellow circle is the estimated landmark node pose, the light orange ellipse is the node covariance, the darker ellipse is the area for the covariance distance threshold, and the yellow star is the estimated feature. The dashed circle represents the distance threshold employed to determine if the estimated feature belongs to an existing landmark node.

landmarks, their position are assumed to be identical keeping them unchanged. This avoids minimal variations of the pose, reducing the computational complexity.

To analyse the topological map representation consider the simulated scenario depicted in figure 5.4, comprising three docks and wind turbines supported by Jacket foundations, where the evaluation trajectory started with the USV near the OWT, then performed an exploratory sweep around the entire wind farm perimeter, and docked at the end.

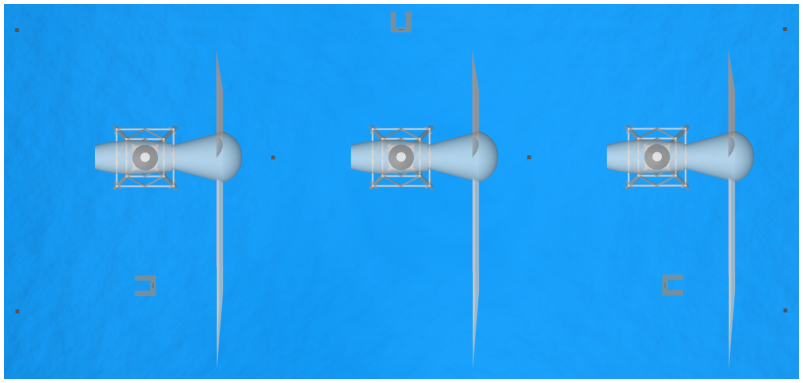


Figure 5.4: Gazebo scenario for testing the topological graph.

Figure 5.5 visually depicts the topological graph result. The USV's trajectory is represented by the yellow lines, signifying pose-pose edges connecting consecutive pose nodes. The local knowledge representation was constructed by using section 3.4 GPS/IMU odometry, and incorporating detections of the U-shaped docks, identified using the method proposed by Pereira et al. (2021) [24], while the Jacket foundations were detected using FRONTIER detections. As the USV navigates the environment, newly detected landmarks trigger the creation of pose-landmark edges (shown in orange lines) that connect the USV's pose to the identified landmark. A key

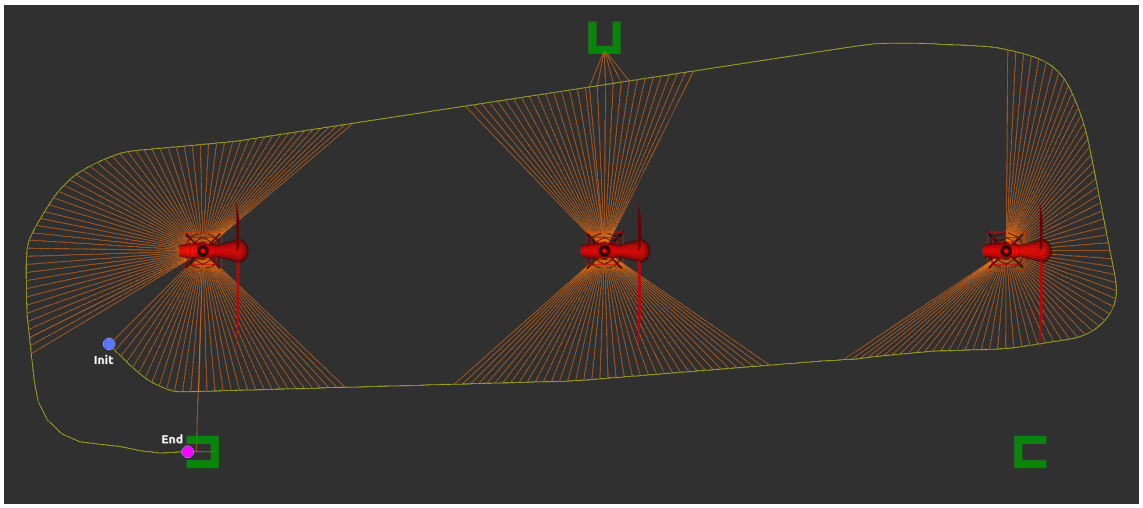


Figure 5.5: Local knowledge representation of the scenario. Yellow and orange lines depict the pose-pose and pose-landmark edges. The blue and pink circles are the initial and end poses of the USV, respectively.

consideration during landmark addition was the sparse nature of the offshore wind farm environment. This sparsity allowed for a larger distance threshold between odometry nodes before adding a new landmark. This approach ensured accurate capture of the relevant features within the scenario. Furthermore, occasional detection gaps due to limitations of the detection algorithms are observed. These gaps may represent missed detections or outliers. Conversely, sections with frequent landmark detections highlight the spacing between odometry nodes, as these edges originate from consecutive USV poses.

A real-world experiment was also conducted using the Nautilus USV at the ATLANTIS coastal test centre. Figure 5.6 provides a visual representation of the operational environment taken before initiating the test. The scenario included several small boats docked. However, it's important to note that there were time elapsed between the instant captured by figure 5.6 and the inspection mission execution. This resulted in some changes in vessel positions. Specifically, the vessel marked by the purple circle went outside the inspected area, while a sailboat arrived and docked in the green circle location. During the mission, the Nautilus USV (represented by the red ellipse) was remotely operated, navigating around the central dock (yellow ellipse). This real-world experiment provided valuable insights into the performance of the local knowledge representation in a dynamic maritime environment.

Figure 5.6: Approximate conditions of the scenario to be reconstructed, namely a port environment in the ATLANTIS coastal test centre with six docked vessels.

This real-world validation experiment at ATLANTIS used the vessel detection and tracking methodology proposed by Duarte *et al.* (2021) [25]. Figure 5.7 presents both the trajectory taken by Nautilus and the constructed topological representation of the environment. The USV navigated around the central dock, and the framework successfully detected most of the vessels within the scenario. However, some limitations were observed, namely an outlier measurement, repre-

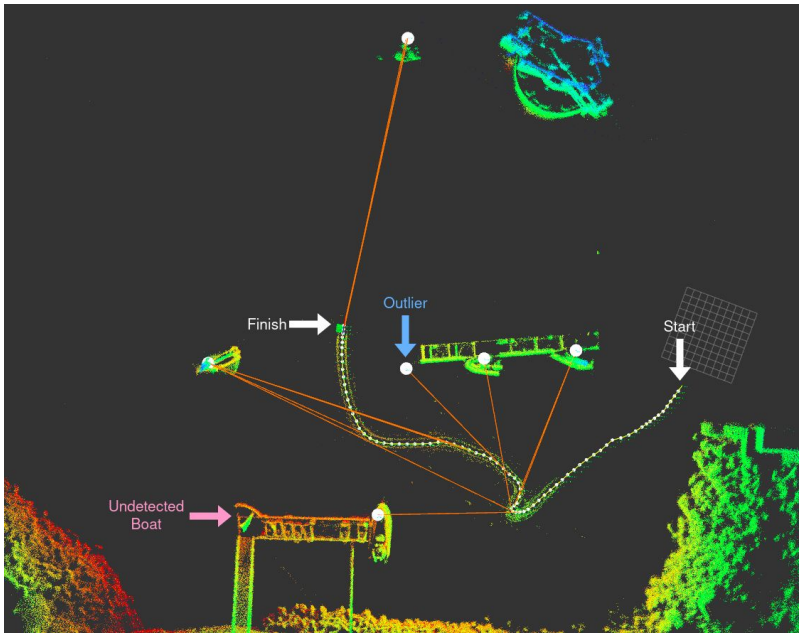


Figure 5.7: Topological representation of the scenario overlapped with the emersed domain 3D map.

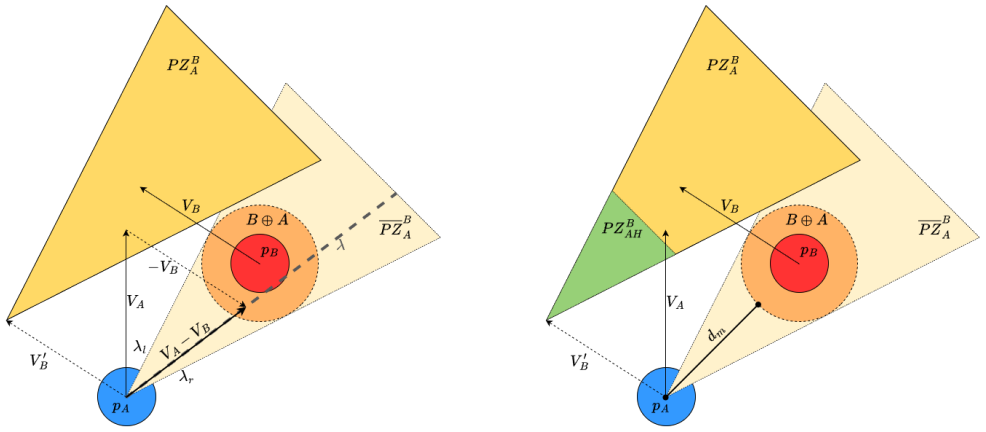
sented by the blue arrow, resulted from a false positive detection by the vessel detector, and a vessel (pointed by the pink arrow) remained undetected due to occlusion during the USV's trajectory. Despite the limitations, the real-world experiment demonstrated the framework's capability to represent the USV local knowledge through the topological map in a dynamic maritime environment, successfully detecting the majority of the surrounding vessels. This validation provides valuable insights for further development and refinement of the overall system.

Therefore, the aforementioned rules enables a topological map capturing semantic information. The importance of a standardized representation is further emphasized by the simulated and real-world validations. This map serves as local knowledge for USVs, providing a comprehensive understanding of its surroundings. By leveraging data using onboard sensors, the framework interprets the scenario and constructs this informative representation. This local knowledge representation serves as input for the proposed obstacle avoidance technique, providing essential information about the surrounding objects, including their size, pose, and velocity.

5.2.2 AVOA - Adaptive velocity obstacle avoidance

AVOA leverages the information provided by the local knowledge representation to construct the configuration space. This is defined by the protective zones, which are a set of all velocities that result in a collision [229]. Additionally, it incorporates the kinematic constraints of the vehicle to filter out unfeasible velocity estimates, considering both magnitude and heading. Furthermore, the framework introduces a risk cost to classify the safety level associated with feasible velocities.

Assuming **A** is an agent moving at a velocity v_A in a plane, with the reference pose p_A, and **B** an obstacle moving in the same plane with the pose p_B and velocity v_B, as shown in figure 5.8a,



(a) Considering all collision time windows.

(b) Considering a non-imminent collision time window (PZ_{AH}^B).

Figure 5.8: Protective zone PZ_A^B of an obstacle B relative to agent A.

the protective zone represented by PZ_A^B (yellow cone in figure 5.8a) can be defined by the set of all velocities v_A that lead to a collision with the moving obstacle within a given time interval. Both agent and obstacles are defined by a circular region delimiting the object areas, represented in blue and red, respectively. For objects with complex shapes an approximation with several circular regions can be performed.

Initially, the definition of relative velocities protective zone $\overline{PZ}_A^B$ can be achieved by equation 5.1,

$$\overline{PZ}_A^B = \{[v_A, \theta_A] : \lambda \cap (B \oplus A) \neq \varnothing\}, \tag{5.1}$$

where θ is the heading, $\lambda = p_A + t * (v_A - v_B)$, $t \geq 0$, and $B \oplus A$ is the Minkowski sum [230] for an agent **A** and for the obstacle **B** (depicted as the orange circle in figure 5.8a). The Minkowski sum reduces agent **A** to a point and increases **B** by the radius of **A**. The state for each object is described by its position and a velocity vector attached to its centre. Then, the $\overline{PZ}_A^B$ cone is defined by a planar sector with the apex in **A**, bounded by the tangent to each side of $\mathbf{B} \oplus \mathbf{A}$ (λ_l and λ_r). Therefore, if any relative velocity lies within the $\overline{PZ}_A^B$, a collision between **A** and **B** is inevitable.

The relative protective zone is specific to a particular agent/obstacle pair. To consider multiple obstacles, it is useful to establish an equivalent condition on the absolute velocities of **A**. This can be achieved by adding the velocity of **B**, v_B, to each velocity in $\overline{PZ}_A^B$, or by translating the cone $\overline{PZ}_A^B$ by v_B, defining the protective zone PZ_A^B. This relationship is formally defined in equation 5.2,

$$PZ_A^B = \overline{PZ}_A^B \oplus v_B. \tag{5.2}$$

PZ_A^B divides the absolute velocities of **A** into collision-free and colliding velocities. In other words, selecting v_A outside the protective zone results in a collision-free motion; conversely when v_A intersects PZ_A^B, **A** and **B** will collide with each other at some moment in time. Velocities on the

boundaries indicate that elements **A** and **B** will graze each other's surfaces.

Given that protective zones rely on a linear approximation of an obstacle's trajectory, their application to predict distant collisions may give inaccuracies when the obstacle's movement does not follow a straight line. Therefore, to avoid an early route change, a simplification to the protective zone can be made, focusing solely on imminent collisions. Specifically, velocities leading to collisions within a predefined time horizon (t_H) are maintained, while non-imminent collision velocities are excluded. This can be done by removing velocities outside the imminent collision zone, denoted as PZ^B_{AH}, depicted in the green area of Figure 5.8b, from the overall protective zone PZ^B_A. The PZ^B_{AH} region is mathematically defined in equation 5.3,

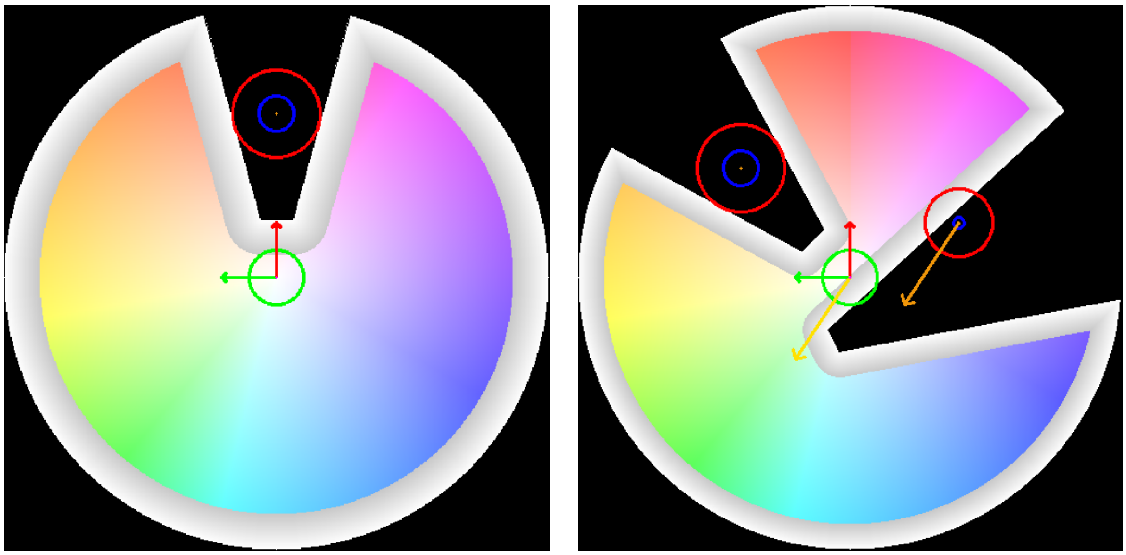
$$PZ^B_{AH} = \left\{ [v_A, \theta_A] : v_A \in PZ^B_A, |v_A - v_B| \leq \frac{d_m}{t_H} \right\}, \tag{5.3}$$

where d_m represents the minimum distance between p_A and $B \oplus A$. With the subtraction of PZ^B_{AH} from PZ^B_A, only velocities that would result in imminent collisions are considered unfeasible.

The position and velocities of the obstacles are referenced to the agent frame, and the velocity space is described by the linear velocity (v) and heading (θ). Hence, the tracking velocity information is transformed to polar coordinates from $[v_x, v_y] \rightarrow [v, \theta]$. Additionally, all velocities surpassing a maximum velocity are set as unfeasible, i.e. $v > v_{max} \Rightarrow v \in PZ^K_R$, where PZ^K_R is the protective zone due to the kinematic constraints. In short, the protective zone defines all velocities that are unfeasible and jeopardize the safety of the vehicle.

This concept is further improved in AVOA by introducing a safety metric based on the distance to the nearest PZ^B_A point, for instance, using Euclidean distance estimation [231]. This metric helps distinguish between tangent crossings and movements in the opposite direction, resulting in a decreasing distance gradient from the farther velocities to PZ^B_A. Two scenarios illustrating this approach are depicted in figures 5.9a and 5.9b, representing a scenario with a single static obstacle and another with two obstacles (one static and one dynamic), respectively. The figures illustrates various elements: the red and green arrows denote the X-axis and Y-axis, respectively. The blue, green, and red circles represent the obstacle and agent shapes, along with their respective Minkowski sums. The orange and yellow arrows indicate the obstacle velocity in its position and projected to the agent. The black region indicates the unfeasible areas, where the cone denotes $PZ^{O_j}_R$, the protective zone relative to the j^{th} obstacle, while the image margins depict velocities exceeding the maximum speed allowed in PZ^K_R. The color scheme in the HSV color space is used to denote velocity (v) and heading (θ) for all non-colliding velocities [232]. Softer colors indicate velocities with smaller magnitudes, while the transition in color represents a change in heading relative to the X-axis. Furthermore, the white-to-dark grey gradient illustrates the distance to the nearest $PZ^{O_j}_R$ within a predefined threshold, indicating the risk level associated with selecting such velocities. As the distance to the protective zone decreases, the cost for selecting these velocities increases.

In the case of USVs, such as SENSE or Nautilus, which have a nonholonomic system, namely differential drive kinematics, the motion of the vehicle is constrained by the limitations of its



(a) Single static obstacle.

(b) Static and dynamic obstacles.

Figure 5.9: Protective zone representation, where the red, green, orange and yellow arrows denote the X-axis, Y-axis, obstacle velocity in its position and projected to the agent, respectively. The blue, green, and red circles represent the obstacle and agent shapes, and the respective Minkowski sums. The black region indicates the unfeasible areas and the HSV color space is used to denote velocity and heading for all non-colliding velocities.

thrusters. These limitations must be explicitly incorporated into the space model defined previously. With two thrusters positioned side by side, the motion of the USV is restricted to linear (v) and angular (ω) components, and lateral movement is not feasible. As so, the speed assigned to each thruster ($[v_1^t, v_2^t]$) can be determined using equations 5.4 and 5.5, where α and β are constants for thrusters placement and configuration within the USV, and the angular velocity is considered to be proportional to the heading ($\omega \propto \theta$) in the protective zones.

$$v_1^t = \alpha \cdot v + \beta \cdot \omega, \tag{5.4}$$

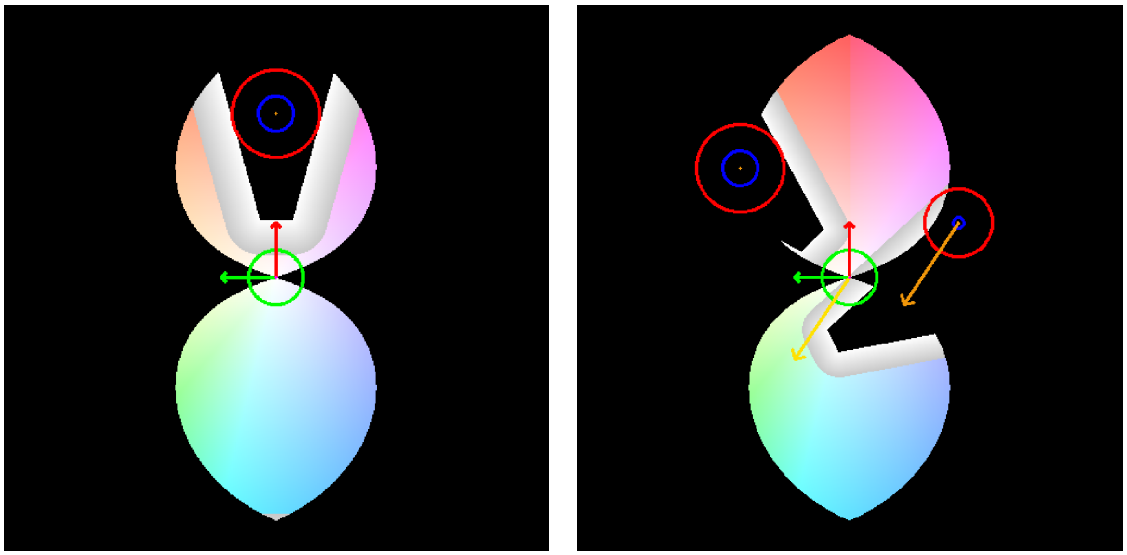
$$v_2^t = \alpha \cdot v - \beta \cdot \omega. \tag{5.5}$$

Assuming the maximum velocity of thrusters (v_{max}^t), the kinematic protective zone, PZ_R^K, can be updated by further excluding the set of unreachable velocities. Thus, the kinematic protective zone (PZ_R^K) will be defined as equation 5.6.

$$PZ_R^K = \left\{ [v_R, \theta_R] : v_R > v_{max} \cup v_1^t > v_{max}^t \cup v_2^t > v_{max}^t \right\}. \tag{5.6}$$

Figures 5.10a and 5.10b demonstrate the visual representation of the protective zones after applying the constraint of equation 5.6, to the scenario depicted in figure 5.9.

Once the protective zones have been fully represented, the next step is to estimate a velocity



(a) Single static obstacle. (b) Static and dynamic obstacles.

Figure 5.10: Protective zone representation with kinematics, where the red, green, orange and yellow arrows denote the X-axis, Y-axis, obstacle velocity in its position and projected to the agent, respectively. The blue, green, and red circles represent the obstacle and agent shapes, and the respective Minkowski sums. The black region indicates the unfeasible areas and the HSV color space is used to denote velocity and heading for all non-colliding velocities.

that minimizes errors based on a multicriteria problem formulation. An optimization approach selects a feasible velocity, denoted as $V_i = [v_i, \theta_i]$, from the feasible set $F = \{V : V \notin PZ_R^K \cap V \notin PZ_R^{O_j}, \forall j = 0, ..., n\}$, where n represents the number of obstacles. This selected solution aims to minimize deviation from the predefined speed while ensuring a safe trajectory. For such, the Particle Swarm Optimization (PSO) [233] is employed to search the feasible set F for an optimal solution, using a multicriteria objective function to assess the performance of particles. Throughout this section, the following notation is used: the subscript d denotes the desired attributes under consideration, while i signifies a solution within the set of all feasible velocities in the protective zones space.

The optimization formulation aims to minimize the multicriteria objective function defined in equation 5.7:

$$\min_{\Delta v_d, \Delta\theta_d, \nabla_i} h(V_i), \tag{5.7}$$

where:

- Δv_d is the velocity deviation relative to the expected velocity, $\Delta v_d = v_d - v_i$;
- $\Delta\theta_d$ is the heading error, $\Delta\theta_d = \theta_d - \theta_i$;
- ∇_i is the risk cost defined by the distance to the nearest protective zone for a specific solution V_i. It penalizes the proximity to the protective zone up to a threshold;

- $h(V_i)$ is a multicriteria heuristic that considers the error of the velocity $V_i = [v_i, \theta_i]$, defined by ε.

The multicriteria objective function is formed by a weighted function, equation 5.8,

$$h(V_i) = \varepsilon^{\intercal} \cdot \psi, \tag{5.8}$$

where ε is the error vector defined by equation 5.9 and ψ is a weight vector with $\sum_{n=0}^{N} \psi_n = 1$. In ε all features are normalized to the range $[0,1]$. This ensures that the contribution of each error term is determined by the corresponding weight in the ψ vector.

$$\varepsilon = \begin{bmatrix} |\Delta v_d| \\ |\Delta \theta_d| \\ \nabla_i \end{bmatrix}, \ \Delta v_d, \Delta \theta_d \in [-1,1], \ \nabla_i \in [0,1]. \tag{5.9}$$

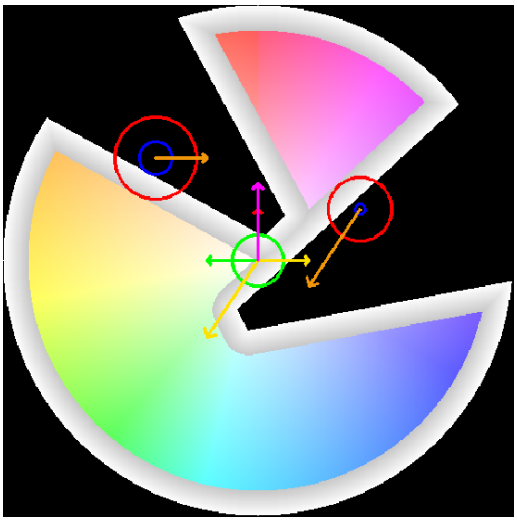
Therefore, the metric $h(V_i)$ serves as a quality indicator for particle fitness evaluation within the PSO algorithm.

Moreover, the combined nature of the optimization problem and the multicriteria evaluation metric can lead to the presence of several local minima, potentially resulting in suboptimal solutions. Figures 5.11a and 5.11b illustrate an example scenario with two dynamic obstacles. The desired velocity (V_d), represented by the purple arrow, lies within a protective zone, rendering it infeasible. Thus, the optimal solution with a metric value of $h(V_i) = 0$ becomes unreachable. This is reflected in the multicriteria metric representations shown in figures 5.11c and 5.11d, where lighter colors depict better solutions.

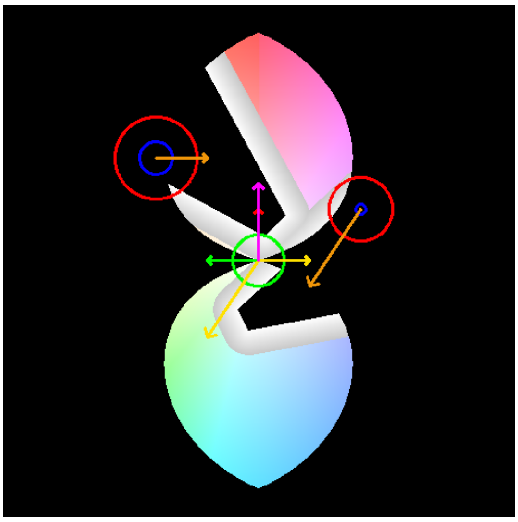
Independently of considering kinematic constraints, the problem inherently exhibits local minima due to the multicriteria nature of the evaluation metric, where ∇_i prioritizes solutions with no risk cost, $|\Delta v_d|$ minimize the deviation from a circle centred at the origin with radius v_d and $\Delta \theta_d$ reduces the deviation from a line starting at the origin with a slope of θ_d. These competing objectives can create high-intensity regions where multiple criteria are satisfied simultaneously. For instance, a bottom-up search algorithm might become trapped in a local minimum with a heading error exceeding 90 degrees. This highlights the importance of selecting an appropriate optimization algorithm capable of effectively navigating these local minima landscapes and converging towards high-quality solutions.

Therefore, a population-based metaheuristic algorithm, Particle Swarm Optimization (PSO), was employed to address the challenge of local minima. Originally proposed by Kennedy *et al.* (1995) in [234], PSO leverages the concept of a swarm of particles to explore the solution space. This approach allows for the diversification, enabling the exploration beyond local minima. By allowing the pursuing of some suboptimal solutions, PSO fosters variability within the search pool, ultimately increasing the probability of converging towards a globally optimal solution.

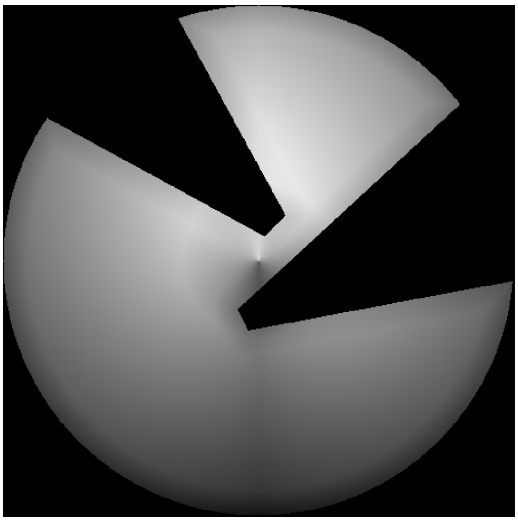
PSO leverages a swarm of particles to explore the solution space. To each particle is assigned a velocity, $V = [v_l, v_h]$, and a pose, $P = [l, h]$, where l is the linear velocity and h the heading. During



(a) Protective zones representation for local minima analysis without kinematics.



(b) Protective zones representation for local minima analysis with kinematics.



(c) Local minima map without kinematics.

(d) Local minima map with kinematics.

Figure 5.11: Protective zones representation and local minima analysis.

each iteration, particles move throughout the feasible solution space, creating a neighborhood where all particles are neighbors.

Within an iteration, each particle is attracted towards two key positions: its personal best solution so far ($pBest = [l_{ibest}, h_{ibest}]$) and to the best solution found by any particle within the neighborhood ($gBest = [l_{gbest}, h_{gbest}]$). Additionally, the particle considers the previous velocity ($V_{i-1} = [v_l, v_h]$) to maintain some momentum during updates [235]. Equations 5.10 and 5.11 present the update rules for a particle's velocity and position at each iteration [233].

$$V_i = \varphi_1 \cdot V_{i-1} + \varphi_2 \cdot r_1 \cdot (pBest - P_{i-1}) + \varphi_3 \cdot r_2 \cdot (gBest - P_{i-1}) + \varphi_4 \cdot \tau, \tag{5.10}$$

$$P_i = P_{i-1} + V, \tag{5.11}$$

where:

- $\varphi_1 = [\varphi_{1l}, \varphi_{1h}]$ controls the inertia weight. Influences the particle update tendency, values lesser than one promote convergence;
- $\varphi_2 = [\varphi_{2l}, \varphi_{2h}]$ represents the confidence in the personal best solution;
- $\varphi_3 = [\varphi_{3l}, \varphi_{3h}]$ represents the confidence in the neighborhood's best solution;
- r_1, r_2, and r_3 are random uniform numbers between $[0,1]$, introducing a stochastic element to the search process;
- $\varphi_4 = [\varphi_{4l}, \varphi_{4h}]$ sets a shacking value for a static particle ($V = [0,0]$):
  - $\tau = [\tau_l, \tau_h]$ is a vector of random uniform numbers between $[-1,1]$ when a particle is static. This injects variability around the best minima, promoting diversification and preventing stagnation.
  - Otherwise, $\tau = [0,0]$ for particles with non-zero velocity.

Algorithm 2 presents the pseudocode for the implemented PSO. During the initialization phase of the PSO algorithm each particle is randomly assigned a position within the feasible solution space, F. The initial velocity of each particle is also randomly chosen from a uniform distribution between $-v_{ini}$ and v_{ini}. This initialization is achieved by employing a uniform random number vector κ generated between $[-1,1]$. Following the initialization of all particles, the PSO loop iterates until a termination criterion is met, namely the multicriteria metric value falls below a predefined threshold, indicating an acceptable solution has been found, or a maximum number of iterations is reached. At each iteration, the multicriteria metric value is evaluated for the particles, then their personal best and global best positions are updated. The particle velocities and positions are reevaluated based on equations 5.10 and 5.11. This ensures that particles are continuously drawn towards different solutions encountered during the search process. If a particle update results in an infeasible solution (i.e., a solution outside the valid range or violating constraints), it is respawned within the feasible space, F. This mechanism prevents particles from getting stuck in invalid regions and guides them back towards valid solutions.

Finally, the PSO algorithm returns the global best solution, which represents the optimal velocity (including both linear velocity and heading) for the USV to navigate the environment safely while avoiding collisions. This optimization process, by searching within the protective zone representation, effectively introduces variability in the solution space. This variability ensures that the USV can identify the most suitable velocity to navigate the specific scenario at hand.

In essence, the AVOA framework, through its exploration of the representation space and its PSO-driven velocity selection, ensures the USV ability to anticipate and adapt to both static and dynamic obstacles. This capability fosters safe navigation during autonomous inspection missions.

Algorithm 2 Particle Swarm Optimization - Solution evaluation method

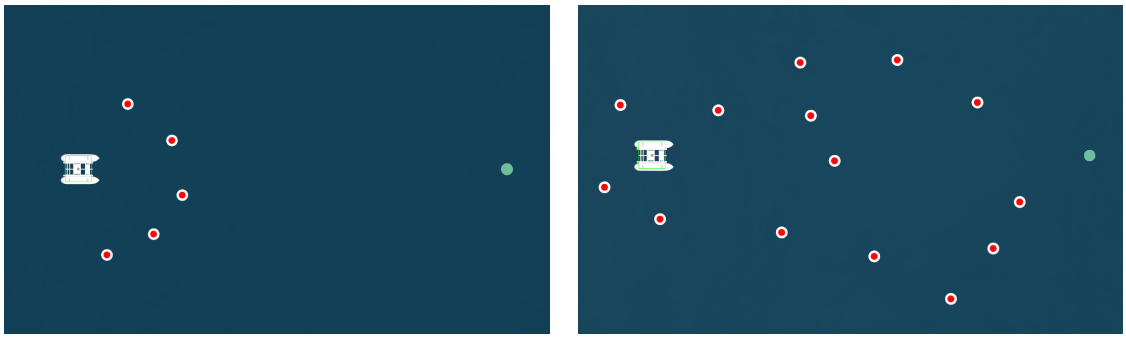
Input:
Number of particles ($npart$)
Initial particle velocity (v_{ini})
Maximum number of iterations ($maxit$)
Acceptable error for early stop (a_e)
Desired velocity (v_d)
List of feasible velocities (f)
Particle current solution vector (p)
Particle velocity vector (v)
Output:
Global best solution ($gBest$)

```
for all npart do
    r ← rand(0, size(f))
    p_i ← f_r
    v_i ← v_ini · κ
while (iter < maxit and h(v_d, gBest) > a_e) do
    for all npart do
        objf ← h(v_d, p_i)
        if (objf < h(v_d, pBest)) then
            pBest ← p_i
        if (objf < h(v_d, gBest)) then
            gBest ← p_i
    for all npart do
        Solve equations 5.10 and 5.11
        if (p_i == unfeasible) then
            r ← rand(0, size(f))
            p_i ← f_r
            v_i ← v_ini · κ
return gBest
```

5.3 Results

Towards the analysis of the AVOA framework's performance, various configuration spaces were defined within the Gazebo simulation environment. These scenarios involved both static obstacles (buoys) and dynamic object (USV) moving at a constant speed. The USV model employed for the simulations was based on the SENSE USV. For the performance evaluation the detections of the obstacle were provided by the simulator and used to obtain the local knowledge representation. Figure 5.12 depicts the two test scenarios considered, where the green dot represents the desired goal location for the USV, including:

- U-shaped wall of buoys placed directly in front of the vessel used only for static scenario tests (figure 5.12a);



(a) U-shaped wall. (b) Cluttered environment.

Figure 5.12: Obstacle avoidance with static obstacles, where the green and red dots represents the desired goal location and the static obstacles, respectively.

- Cluttered environment, a more complex scenario with a high density of buoys used for static and dynamic obstacles validation (figure 5.12b).

Moreover, the test set with static and dynamic obstacles, depicted in figure 5.13, presents different scenarios with a crossing vessel moving at a fixed velocity ($0.55\,\mathrm{m\,s^{-1}}$). In scenario A, the crossing vessel approaches the USV head-on; in B the crossing vessel moves laterally across the path between the USV and the goal location; in C the crossing vessel travels diagonally relative to the USV and the goal.

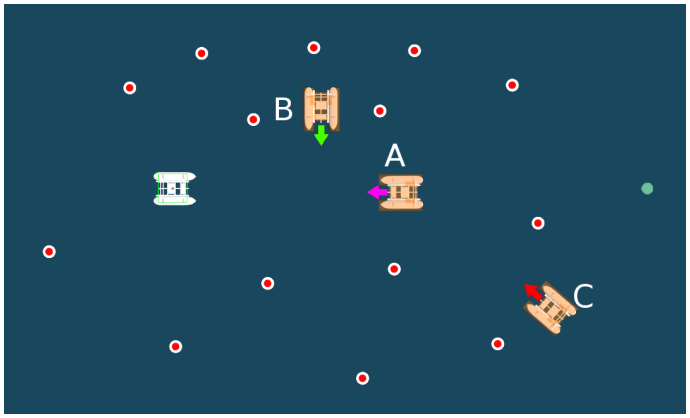


Figure 5.13: Obstacle avoidance with a moving dynamic obstacle simulator, where the green and red dots represents the desired goal location and the static obstacles, respectively. The orange USV is the crossing vessel.

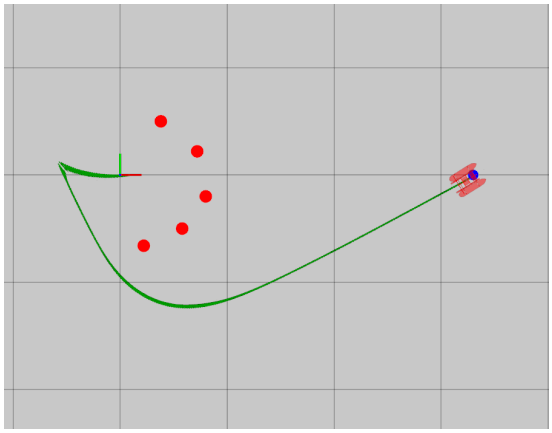
All experiments assumed zero wind velocity and waves composed of three components with the configurations detailed in table 5.1. The performance of AVOA was compared against the Timed-Elastic-Band (TEB) method [236] implemented within the ROS using default parameters. Both AVOA and TEB were configured with identical robot-specific parameters, including a maximum linear velocity of $1.5\,\mathrm{m\,s^{-1}}$ and a maximum angular velocity of $0.3751\,\mathrm{rad\,s^{-1}}$.

Table 5.1: Wave configuration for simulation.

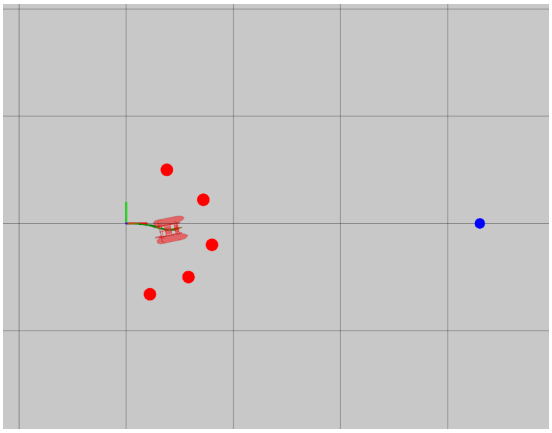
Component	Amplitude	Period	Direction (x,y)
1	0.06	12.6	(-1.0,0.0)
2	0.04	3.7	(-0.7,0.7)
3	0.03	6.3	(0.7,0.7)

5.3.1 AVOA - Adaptive velocity obstacle avoidance

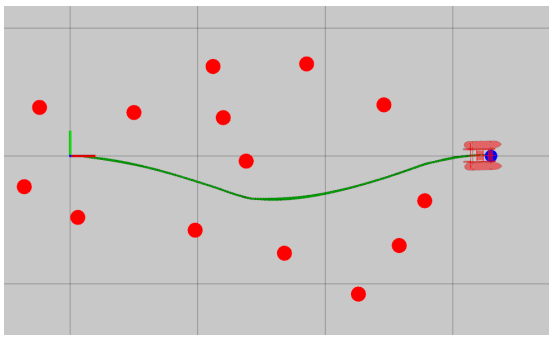
Starting with the static environments, figures 5.14a and 5.14b depict the trajectories generated by the proposed AVOA method and the TEB method for the Uu-shaped wall scenario. In this scenario, TEB becomes locked in a local minimum caused by the obstacle placement. In contrast, AVOA successfully avoids this entrapment. The method enables the USV to retreat and find a viable path around the obstacles, ultimately reaching the goal location (represented by the blue dot). This maneuver is achieved by contouring the obstacles while adhering to the kinematic constraints, resulting in a smooth path with minimal heading variations. The travel time using



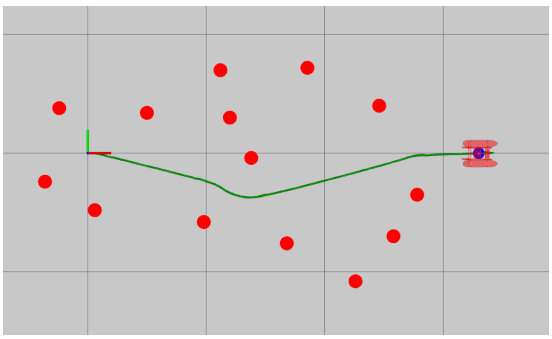
(a) AVOA u-shaped wall.



(b) TEB u-shaped wall.



(c) AVOA cluttered environment.

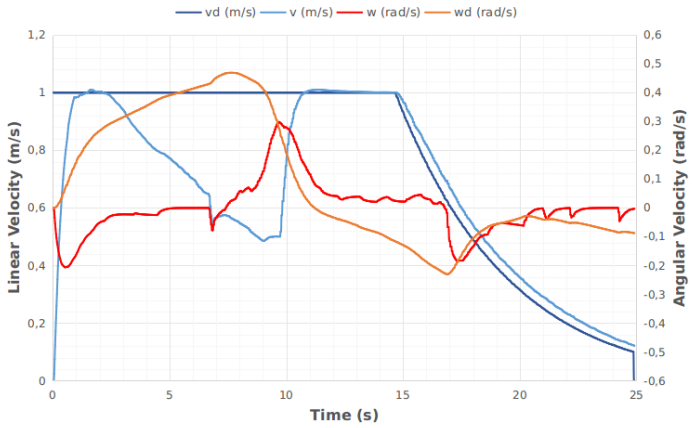


(d) TEB cluttered environment.

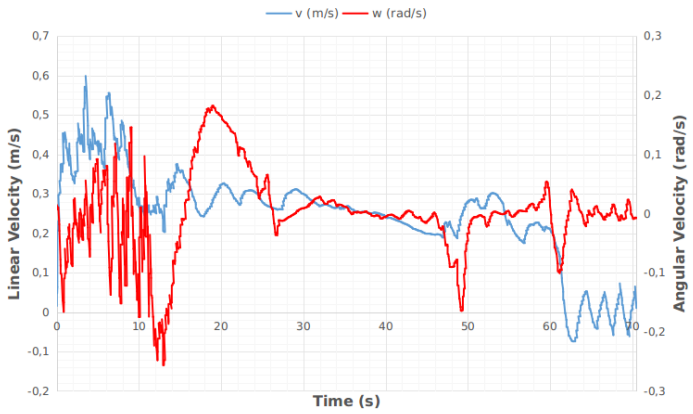
Figure 5.14: Obstacle avoidance with static obstacles.

the proposed method was approximately 33 s. On the cluttered scenario, figures 5.14c and 5.14d, despite almost all the headings relative to the starting pose presenting a collision risk at some point in time, both approaches performed a collision-free trajectory. This observation demonstrates the impact of removing the non-imminent collision window (PO^{O}_{RH}) on the proposed method. By excluding this window, AVOA can select velocities that might initially appear risky but ultimately avoid immediate obstacles while still considering potential future hazards. This enables AVOA to prioritize the avoidance of the most critical threats while maintaining situational awareness for informed decision-making throughout the navigation process.

The AVOA method's trajectory for the cluttered environment exhibits smoother movement compared to TEB. Additionally, AVOA achieves a faster travel time to the goal, reaching it in approximately 25 s compared to TEB's 71 s, representing a 65% improvement. This performance is attributed to AVOA's ability to minimize deviations from the desired velocity ($[v_d, \omega_d]$). This translates to smoother velocity transitions for the USV. Conversely, TEB estimates velocity based on the planned path, resulting in higher variability in the USV's actual velocity, figure 5.15. The figure depicts the robot's linear (v in ms^{-1}) and angular (ω in $rads^{-1}$) velocities (light blue and red lines) alongside the desired linear (v_d in ms^{-1}) and angular (ω_d in $rads^{-1}$) velocities (dark blue



(a) AVOA.

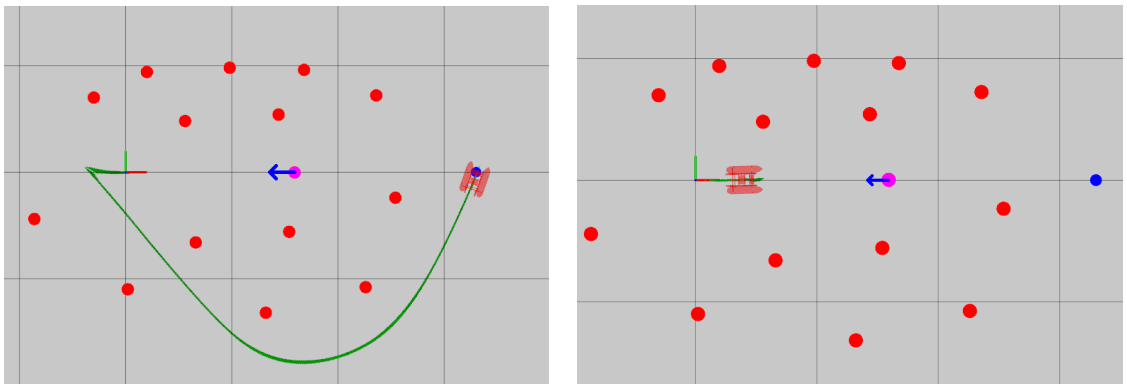


(b) TEB.

Figure 5.15: Robot velocities for the cluttered environment.

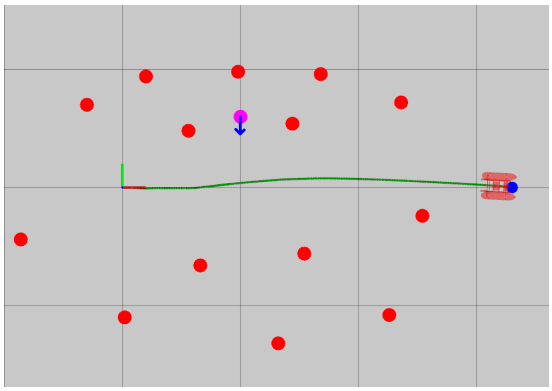
and orange lines). Moreover, AVOA achieves higher linear velocities compared to TEB.

Considering the test sets with both static and dynamic objects (figure 5.13) the results for each scenario using both AVOA and TEB are presented in figures 5.16 for scenario A (figures 5.16a and 5.16b), scenario B (figures 5.16c and 5.16d) and scenario C (figures 5.16e and 5.16f).

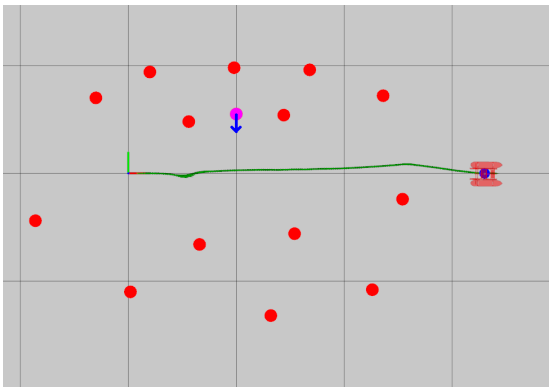


(a) AVOA (Scenario A).

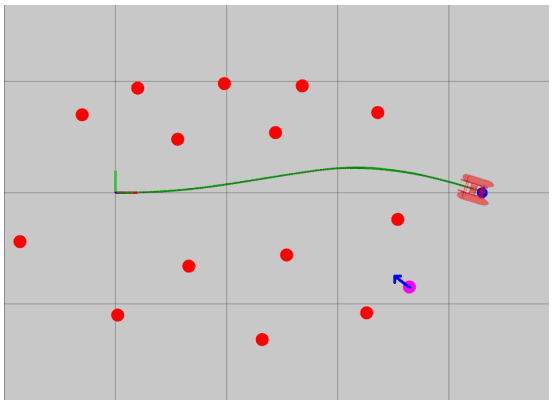
(b) TEB (Scenario A).



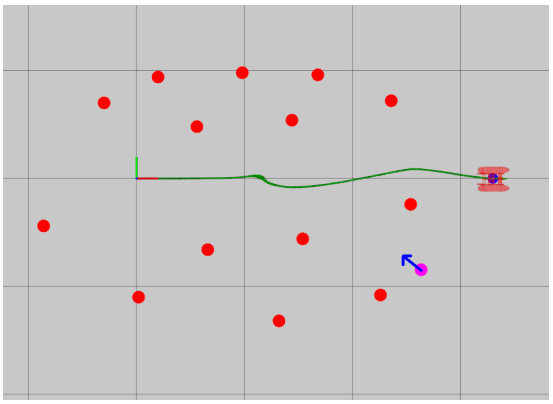
(c) AVOA (Scenario B).



(d) TEB (Scenario B).



(e) AVOA (Scenario C).

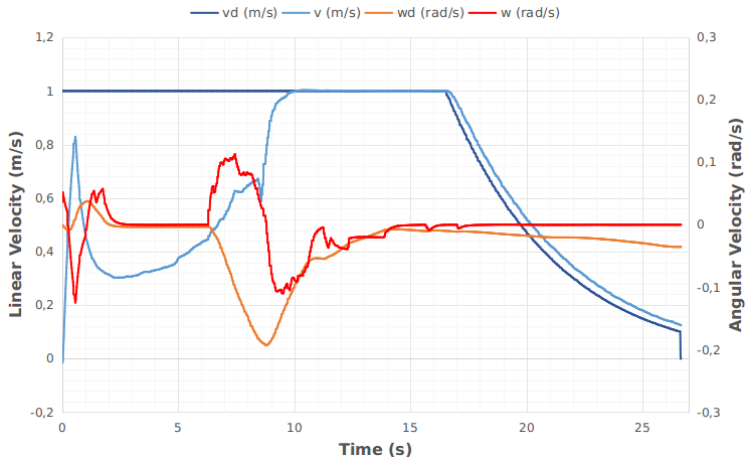


(f) TEB (Scenario C).

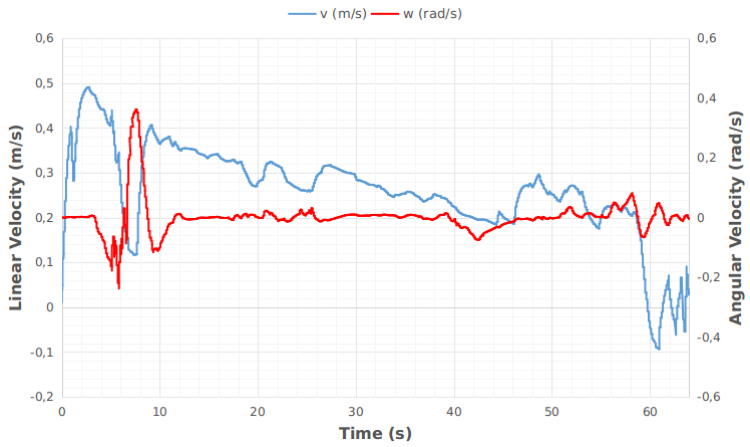
Figure 5.16: Obstacle avoidance with a moving dynamic obstacle. The pink dot depicts the crossing vessel and the blue arrow its direction.

Scenario A evaluates the performance of AVOA and TEB when encountering a head-on moving obstacle. The AVOA method successfully predicted the incoming collision, figure 5.16a, where the initial position of the obstacle is depicted by the pink circle with a blue arrow indicating its heading. Due to the obstacle placement, a direct path towards the goal becomes infeasible. Accordingly, AVOA re-routes the USV to a collision-free trajectory, retreating and, then, navigating around the static obstacles, reaching the goal in approximately 35 s. Conversely, TEB fails to generate a collision-free path in this scenario, figure 5.16b.

Scenario B illustrates the behavior of AVOA and TEB when a dynamic obstacle moves laterally across the USV's path towards the goal. Both methods achieve collision-free trajectories in this scenario, figures 5.16c and 5.16d. However, AVOA reaches the goal faster, taking approximately 27 s compared to TEB's 64 s, representing a 58% improvement in efficiency. The proposed AVOA method allows the USV to remain close to the direct path by initially accelerating and then slowing down to allow the obstacle vessel to pass. The USV then resumes its navigation towards the goal, converging to the desired trajectory, figure 5.17a. TEB generates a slower initial velocity for the USV. While this allows the obstacle to pass first, TEB exhibits a indecision before the rup-



(a) AVOA.



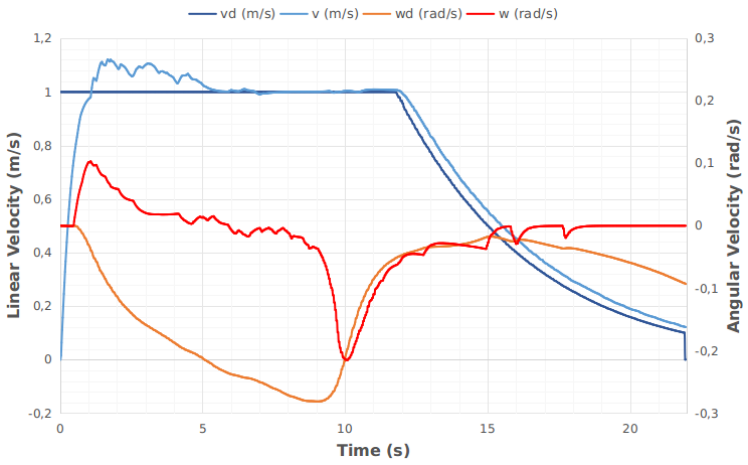
(b) TEB.

Figure 5.17: Robot velocities for the avoidance scenario B.

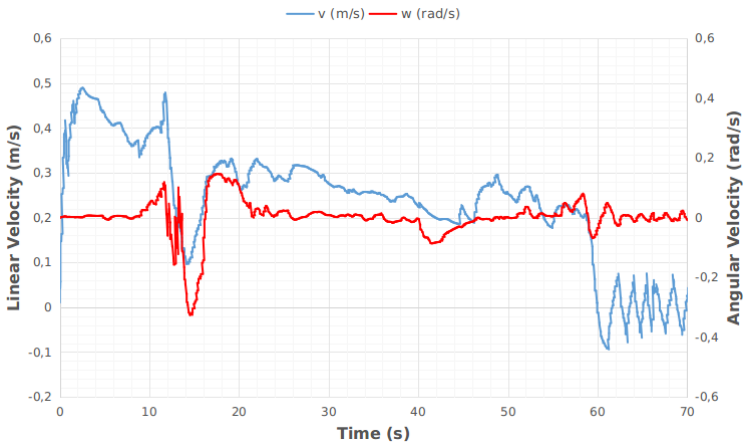
ture of its initially defined elastic band. This behavior manifests as a significant reduction in linear velocity, nearly coming to a complete stop, accompanied by a intermittent changes in heading direction, figure 5.17b.

The final scenario, C, shows the the behavior of AVOA and TEB with a diagonally moving obstacle vessel relative to the USV and the goal. AVOA successfully reaches the goal in approximately 22 s, as shown in figure 5.16e. The AVOA method initially estimates a higher velocity than desired with a positive heading, figure 5.18a. This strategy ensures that the USV reaches a safe position before the obstacle vessel arrives to the critical intersection point. By doing so, the USV avoids the potential collision and proceeds directly towards the goal. TEB also generates a complete solution in this scenario, figure 5.16f. Nevertheless, TEB requires a longer travel time of around 70 s, translating to a 69% performance improvement in favor of AVOA. While TEB's path allows the obstacle vessel to pass first, it exhibits the same indecisive behavior observed in scenario B, figure 5.18b. This manifests as a sudden reduction in velocity and changes in heading direction.

The simulation results demonstrate AVOA's promising capabilities in a maritime environment.



(a) AVOA.



(b) TEB.

Figure 5.18: Robot velocities for the avoidance scenario C.

AVOA excels at anticipating obstacles, leading to the generation of faster and smoother trajectories compared to TEB. AVOA successfully completed all five experimental scenarios, whereas TEB only achieved collision-free paths in three out of five. Additionally, AVOA consistently outperformed TEB in terms of travel time, achieving at least a 58% improvement in efficiency across all scenarios. A video with the experimental scenarios presented can be found on the footnote[1].

5.4 Conclusion

This chapter presents the design of a safe navigation procedure for USVs conducting inspections near structures. The proposed framework leverages two key components: (i) Local knowledge representation, uses topological maps enriched with semantic information to provide a standard input to AVOA. This representation captures not only the spatial layout of the environment but also incorporates high-level details such as pose, type, velocity and size; (ii) Adaptive obstacle avoidance method (AVOA), which dynamically plans and predicts safe and efficient motion for the USVs navigating the inspection area, defining a collision-free trajectory. The inclusion of semantic information contributes to a richer understanding of the environment. Additionally, AVOA enhances mission safety by enabling real-time obstacle avoidance. Ultimately, this framework aims to increase the autonomy of O&M activities, thereby mitigating risks associated with human intervention and potential structural damage.

The local knowledge representation improves the contextual awareness through the standardization of distinct detectors information and serves as the input for AVOA's decision-making process by transforming diverse detection data estimated from the available sensor payload into a unified feature representation. Experiments conducted in both simulated and real-world scenarios demonstrated the framework's ability to generalize this approach across various detectors. This ensures the integration of data regardless of the specific sensor and detector used, such as docks, foundations, or vessels detectors. By capturing this comprehensive information, the local knowledge representation effectively translates the environment's context into a format easily interpreted by AVOA. This endows AVOA with the full capacity to consider both static and dynamic obstacles within the environment, enabling the selection of safe and efficient motion for the USV. AVOA is a novel obstacle avoidance technique that prioritizes both collision-free navigation and smooth USV motion. This is achieved through the creation of a protective zone representation that defines the zones around obstacles within the environment, establishing safe regions for the USV to travel. It accounts for the velocities that could lead to a collision according to the obstacle dynamics. Additionally, the algorithm incorporates the USV's kinematic constraints, ensuring that the estimated collision-free velocities are physically achievable for the vehicle. To further enhance decision-making, AVOA employs a formally defined risk cost function using a multicriteria metric. This function minimizes the potential risks within the environment, promoting the smallest variation possible to the current motion of the USV.

[1] Daniel Campos, 2019. *Avoa - Adaptive Velocity Obstacle Avoidance*. URL:`https://youtu.be/jOTsiBlszuA`

Throughout extensive experiments, the AVOA method showed promising results towards a collision-free approach for USVs when considering tidal waves constraints. When compared with an existing state-of-the-art algorithm, AVOA showed an improvement in the travel time of at least 58%. Contrary to the compared method, AVOA finished all five courses without collisions or lock situations. Moreover, the proposed algorithm demonstrates a proactive behavior capable of undertaking preemptive actions to maintain the predefined course. Overall, the results highlight AVOA's effectiveness in enhancing both safety and efficiency for USV navigation in complex maritime environments.

In conclusion, the safe navigation architecture enables USVs to navigate safely near structures by integrating heterogeneous information and selecting the safest motion. This pipeline has the potential to substantially enhance the efficiency and safety of inspections, mitigating risks associated with operations and facilitating close-range navigation.

Chapter 6

Conclusion

Current offshore wind farm inspection practices rely heavily on human operators, exposing them to inherent risks associated with the harsh maritime scenarios. This environment presents unique challenges for each operation domain, where the sensor performance and availability for the autonomous vehicles varies. Existing solutions often uses dedicated vehicle with specific sensor payload, hindering efficient multi-domain inspections. Furthermore, ensuring safe navigation near critical infrastructure in these conditions remains essential. As a result, achieving comprehensive and safe multi-domain inspection of offshore wind farms poses a significant challenge.

This research presents novel multi-domain USV systems designed to overcome the limitations of traditional single-domain USVs for offshore inspections. These vehicles are endowed with a multi-domain sensor payload that enables data acquisition from both surface and underwater environments. Furthermore, the multi-domain mapping architecture (M2A) uses this data to create a comprehensive 3D representation. FRONTIER, a data completion and semantic understanding approach, augments the representation by identifying missing data and extracting high-level features. Then a local knowledge representation is created to be used as input for the collision avoidance methodology by leveraging landmark detections. Finally, AVOA ensures safe navigation with minimal deviation from the planned route. This innovative multi-domain USV system enhances offshore inspections by eliminating the need for multiple operations and dedicated assets, leading to reductions in cost, time, and mission-related risks.

The multi-domain platforms, SENSE and Nautilus, provided modular and adaptable architectures, each designed to unique operational requirements for multi-domain data acquisition, navigation, and perception. SENSE's modular framework facilitates simple transitions between configurations, enabling swift adjustments to sensor payloads, navigation systems, and energy sources. Its catamaran design ensures operational efficiency under calm maritime conditions, making it a versatile option for both data acquisition and methodology validation. In contrast, Nautilus presents a compact monohull design optimized for offshore deployment, with enhanced operational capacity and resilience in moderate sea states. Its ability to conduct beyond visual line-of-sight operations and transmit data to shore control centers underscores its suitability for demanding offshore environments, adhering to industry safety standards and enabling ground-

breaking advancements in autonomous maritime operations. Both platforms share their software architecture through MUSA, a multi-layer software architecture designed to standardize development across different vehicle types. MUSA defines a four-layer structure (mission, task, skill, action) to provide a flexible framework for describing missions, tasks, and controlling the USVs at various complexity levels. Extensive field tests conducted at *Marina de Leça* harbor and the ATLANTIS coastal test bed provided the validation of navigation and data acquisition capabilities of both SENSE and Nautilus. These tests culminated in the creation of the ROAM@CRAS dataset, offering insights into challenges such as sensor fusion, tracking, and obstacle detection. Nautilus underwent rigorous offshore testing, demonstrating its resilience in harsh conditions, and achieving a TRL 8. The ATLANTIS testing process demonstrated the reliability of Nautilus under varying environmental conditions, including significant wave heights and wind speeds within industry safety standards, namely the maximum significant weight of 1.5 m and wind speeds of $10\,\mathrm{m\,s^{-1}}$. These tests and validations lead to Nautilus's becoming the first USV to autonomously operate on a floating offshore wind farm in the Atlantic.

To capitalize on the USVs' multi-domain capabilities, this research developed two distinct architectures: M2A and FRONTIER. M2A focuses on multi-domain inspection, enabling the simultaneous reconstruction of above-water (emersed) and underwater (immersed) environments. This fosters comprehensive inspections of both the transition piece and seabed reconstruction. FRONTIER addresses incomplete data within the vehicles FOV by filling occluded regions. This information can also be used as semantic features to enrich local knowledge information. The M2A employs an innovative approach for multi-domain mapping of offshore wind farm infrastructure using various sensors (e.g., LiDAR and MBES). It integrates navigation data from GPS and IMU sensors to estimate the odometry used in the 3D reconstruction of the environment. The methodology effectively filters out outlier data to minimize bias during registration. M2A demonstrates promising results with positional errors of 0.049 meters in simulations and 0.07 meters in real-world data, while achieving real-time reconstruction at an average of 300 milliseconds. Tests conducted with two USVs validate its accuracy and efficiency in multi-domain mapping across diverse scenarios, ranging from harbors to offshore wind farms. FRONTIER introduces a template-based framework for semantic instance segmentation. It begins with data preprocessing and segmentation using the BEVA-FPC clustering method, followed by object classification with the RI-PointTransf network. This network achieves state-of-the-art accuracy in classifying foundation types, enabling precise detection of the object of interest with the 3D bounding box. The GICP registration method ensures visually accurate alignments, ensuring the completion of missing foundation data. Experiments demonstrate significant improvements from FRONTIER in processing time and accuracy. The BEVA-FPC method reduces processing time by at least threefold compared to established clustering methods, and the RI-PointTransf network achieves high accuracy exceeding 90.81% on the ModelNet10 dataset, surpassing other state-of-the-art techniques. For a foundation-specific dataset, the classification network exhibited high accuracy (98.97%) when trained on synthetic data. This resulted in a small sim-to-real gap (around 3%) when evaluated with real-world data, demonstrating the network's strong adaptation capabilities.

Furthermore, the registration method applied to various foundation types for sensor information completion showed visually accurate alignments even when comparing real-world data with approximated CAD templates. These results solidify FRONTIER's effectiveness in completing foundation shapes, enhancing multi-domain perception in offshore environments, and establishing a foundation for high-level semantic features.

Leveraging the multi-domain frameworks and existing autopiloting systems within the USVs, this research introduces a novel proactive obstacle avoidance architecture for safe USV navigation. This architecture relies on a semantic-based topological representation, which generates contextual awareness using data from detections, like the ones available in FRONTIER. This representation provides a standard feature set describing the environment's context and serves as input for the AVOA algorithm. AVOA prioritizes collision-free navigation while ensuring smooth and efficient USV motion by defining protective zones around obstacles and employing a multicriteria heuristic function. This approach minimizes potential risks and promotes minimal deviation from the USV's planned route. The framework's proactive nature allows for preemptive actions, ensuring safe navigation even in cluttered environments. Comprehensive trials validate the effectiveness of this safe navigation pipeline. Experiments conducted in both simulated and real-world scenarios demonstrate the local knowledge representation's ability to generalize across various detectors (e.g., docks, foundations, vessels). This ensures seamless integration of data regardless of the specific sensor and method used. By capturing this comprehensive information, the local knowledge representation effectively defines the scenario into an unified representation to be used in AVOA. Furthermore, AVOA exhibits significant improvements in travel time (at least 58% reduction) and a safe behavior (no collisions) across various courses compared to an existing state-of-the-art algorithm. AVOA's proactive behavior enables preemptive actions to maintain predefined courses and eliminate non-imminent collision risks, ensuring safe navigation in cluttered scenarios. These findings solidify the framework's effectiveness in enhancing USV navigation capabilities, clearing the way for advancements in autonomous maritime operations.

In conclusion, this research addressed the limitations of autonomous multi-domain inspections using USVs in offshore wind farms. It proposes a novel pipeline for safe and comprehensive inspections through the development of modular hardware and software for multi-domain USVs. This approach reduces reliance on human operators and necessary assets, and facilitates BVLOS missions, leading to cost reductions and increased efficiency. This allows the execution safe and comprehensive 3D reconstructions of offshore wind farms with a multi-domain USV. The vast amount of experiments in simulated, near-real, and real-world scenarios validated the accuracy and robustness of the proposed technologies for multi-domain USV operations, particularly where safe autonomous solutions are limited. The combined multi-domain hardware, software and collision avoidance technology offers a novel solution for highly effective, safe and thorough inspections. This advancement tackles key challenges in multi-domain inspections and enhances operational efficiency and reliability, especially in demanding environments, such as offshore wind farms.

6.1 Future work

The next steps of this research work includes several key aspects namely:

- Establish automated deployment and recovery mechanisms for small USVs to facilitate fully uncrewed missions and to extend the wave height limitation for offshore wind farm inspection and maintenance operations;
- Incorporate visual and thermal images into the multi-domain map representation to enhance the completeness of information, providing comprehensive data for expert inspection purposes;
- Develop automated fault detection procedures tailored specifically for identifying issues in the transition piece and scour protection;
- Deploy RI-PointTransf on an embedded system onboard the USV for real-time 3D classification task;
- Investigate novel registration techniques to improve the real-time capabilities of the system, aiming to streamline the process of aligning multi-domain sensor data for enhanced perception and navigation;
- Incorporate multi-object tracking algorithms to improve the standard feature representation, particularly focusing on tracking object velocities and enhancing node association;
- Upgrade the local knowledge representation to improve the topological graph by incorporating graph optimization techniques, enhancing the accuracy and completeness of the environmental model;
- Establish a global knowledge base by associating information from multiple agents operating within the environment, fostering cooperation between UAVs and USVs to enhance situational awareness and decision-making capabilities;
- Conduct real-world validation experiments to assess the performance and effectiveness of the AVOA algorithm under diverse and dynamic maritime conditions;
- Extend AVOA by integrating reinforcement learning techniques to enhance the USV's ability to respond to external environmental factors like wind and waves, thereby improving its adaptive navigation capabilities.

Appendix A

Supplementary information

A.1 ROAM@CRAS dataset - Sensor placement and topics

Table A.1 the relation between the sensors and the coordinate frames depicted in figure 3.2 are listed, as well as the position in meters from each in relation to *base_link*.

Table A.1: Sensors available in SENSE, respective frames and position in meters from figure 3.2.

Sensor	**Frame**	**Position (x,y,z) m**
Sparkfun Razor IMU 9DOF	imu_nav_box	(0.48,-0.17,0.09)
Xsens MTi-30	imu	(0.98,0.17,0.09)
Vk-162 Glonass GPS	ublox_gps	(0.27,-0.20,0.12)
Swift Navigation Piksi Multi	gps	(0.80,0.10,1.09)
Velodyne VLP-16	velodyne	(0.93,0.0,0.70)
Mynt Eye D	mynteye_left_color_frame mynteye_right_color_frame	(0.99,0.07,0.68)
Imagenex "Delta T" 837B	sonar_link	(0.45,0.0,-0.50)

Table A.2 lists the data available in the ROAM@CRAS and the respective ROS topics, where *VK-GPS* is the Vk-162 GPS, *SN-GPS* is the Swift Navigation GPS, *R9-IMU* is the Razor 9DOF IMU and *XS-IMU* is the Xsens IMU.

Table A.2: List of the data available in the ROAM@CRAS and the respective ROS topics.

Sensor	**ROS topic**	**Summary**
Internal	/cmd_vel /left/cmd /right/cmd /roboteq_driver/status /safety_stop	- Velocity commands - Thruster commands - Vehicle diagnostics
R9-IMU	/imu_nav/data	- Angular velocity - Linear acceleration - Orientation
XS-IMU	/imu/data /velocity	- Angular velocity - Linear acceleration - Orientation
VK-GPS	/ublox_nav/fix /ublox_nav/fix_velocity	- Latitude - Longitude - Altitude - Linear velocity
SN-GPS	/gps/fix /gps/rtkfix /gps/time /gps/basestation_ecef	- Latitude - Longitude - Altitude - Odometry - Linear velocity - Time reference - Base station location
LiDAR	/velodyne_points	- 3D points
Camera*	/depth/camera_info /depth/image_raw /left/image_color** /right/image_color**	- Depth - Compressed Raw Images - Calibration
MBES	/mbes/scan	- 2D Scan - Polar data

* All camera topics includes the prefix */mynteye*, e.g. */mynteye/depth/camera_info*.

** Compressed images, includes suffix */compressed*, e.g. */mynteye/left/image_color/compressed*.

Bibliography

[1] X. Wang, X. Zeng, J. Li, X. Yang, and H. Wang, "A review on recent advancements of substructures for offshore wind turbines," *Energy Conversion and Management*, vol. 158, pp. 103–119, 2018.

[2] Z. Ren, A. S. Verma, Y. Li, J. J. Teuwen, and Z. Jiang, "Offshore wind turbine operations and maintenance: A state-of-the-art review," *Renewable and Sustainable Energy Reviews*, vol. 144, p. 110886, 2021.

[3] F. Jovan and S. Bernardini, "Adaptive temporal planning for multi-robot systems in operations and maintenance of offshore wind farms," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 15782–15788, Sep. 2023.

[4] G+ Global Offshore Wind and Energy Institute London, "G+ Global Offshore Wind Health & Safety Organisation - 2022 incident data report," tech. rep., 2022.

[5] O. Khalid, G. Hao, C. Desmond, H. Macdonald, F. D. McAuliffe, G. Dooly, and W. Hu, "Applications of robotics in floating offshore wind farm operations and maintenance: Literature review and trends," *Wind Energy*, vol. 25, no. 11, pp. 1880–1899, 2022.

[6] N. Silva and A. Estanqueiro, "Impact of weather conditions on the windows of opportunity for operation of offshore wind farms in portugal," *Wind Engineering*, vol. 37, no. 3, pp. 257–268, 2013.

[7] Y. Gu, P. Wang, Z. Rong, H. Wei, S. Yang, K. Zhang, Z. Tang, T. Han, and Y. Si, "Vessel intrusion interception utilising unmanned surface vehicles for offshore wind farm asset protection," *Ocean Engineering*, vol. 299, p. 117395, 2024.

[8] H. Fox, A. C. Pillai, D. Friedrich, M. Collu, T. Dawood, and L. Johanning, "A review of predictive and prescriptive offshore wind farm operation and maintenance," *Energies*, vol. 15, no. 2, 2022.

[9] E. E. Ambarita, A. Karlsen, O. Osen, and A. Hasan, "Towards fully autonomous floating offshore wind farm operation & maintenance," *Energy Reports*, vol. 9, pp. 103–108, 2023. The 8th International Conference on Sustainable and Renewable Energy Engineering.

[10] M. F. A. Bakar and M. R. Arshad, "ASV data logger for bathymetry mapping system," in *2017 IEEE 7th International Conference on Underwater System Technology: Theory and Applications (USYS)*, vol. 2018-Janua, pp. 1–5, IEEE, 2017.

[11] H. Kurniawati, J. C. Schulmeister, and T. Bandyopadhyay, "Infrastructure for 3D Model Reconstruction of Marine Structures," *International Society of Offshore and Polar Engineers (ISOPE)*, vol. 8, pp. 359–364, 2011.

[12] H. Mousazadeh, J. Hamid, O. Elham, M. Farshid, K. Ali, S.-Z. Yousef, and M. Ashkan, "Experimental evaluation of a hydrography surface vehicle in four navigation modes," *Journal of Ocean Engineering and Science*, vol. 2, no. 2, pp. 127–136, 2017.

[13] W. Wang, L. A. Mateos, S. Park, P. Leoni, B. Gheneti, F. Duarte, C. Ratti, and D. Rus, "Design, modeling, and nonlinear model predictive tracking control of a novel autonomous surface vehicle," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6189–6196, 2018.

[14] E. Pairet, S. Spanò, N. Mankovskii, P. Pellegrino, I. Zhilin, J. Nicola, F. La Gala, and G. De Masi, "Nukhada usv: a robot for autonomous surveying and support to underwater operations," in *OCEANS 2022 - Chennai*, pp. 1–6, 2022.

[15] J. Jung, J. Park, Y. Lee, T.-K. Yeu, J. Choi, and J. bu Han, "Consistent mapping of marine structures with an autonomous surface vehicle using motion compensation and submap-based filtering," *Ocean Engineering*, vol. 290, p. 116418, 2023.

[16] D. F. Campos, M. Pereira, A. Matos, and A. M. Pinto, "Diius - distributed perception for inspection of aquatic structures," in *OCEANS 2021: San Diego – Porto*, pp. 1–5, 2021.

[17] A. M. Pinto, J. V. A. Marques, D. F. Campos, N. Abreu, A. Matos, M. Jussi, R. Berglund, J. Halme, P. Tikka, J. Formiga, C. Verrecchia, S. Langiano, C. Santos, N. Sá, J.-J. Stoker, F. Calderoni, S. Govindaraj, A. But, L. Gale, D. Ribas, N. Hurtós, E. Vidal, P. Ridao, P. Chieslak, N. Palomeras, S. Barberis, and L. Aceto, "Atlantis - the atlantic testing platform for maritime robotics," in *OCEANS 2021: San Diego – Porto*, pp. 1–5, 2021.

[18] M. Martinez-Luengo and M. Shafiee, "Guidelines and cost-benefit analysis of the structural health monitoring implementation in offshore wind turbine support structures," *Energies*, vol. 12, no. 6, 2019.

[19] A. Gray, "Setting a benchmark for decarbonising o&m vessels of offshore wind farms," 2 2021.

[20] G. A. Hollinger, B. Englot, F. Hover, U. Mitra, and G. S. Sukhatme, "Uncertainty-driven view planning for underwater inspection," in *2012 IEEE International Conference on Robotics and Automation*, pp. 4884–4891, IEEE, 2012.

[21] R. B. Wynn, V. A. Huvenne, T. P. Le Bas, B. J. Murton, D. P. Connelly, B. J. Bett, H. A. Ruhl, K. J. Morris, J. Peakall, D. R. Parsons, E. J. Sumner, S. E. Darby, R. M. Dorrell, and J. E. Hunt, "Autonomous Underwater Vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience," *Marine Geology*, vol. 352, pp. 451–468, 2014.

[22] G. Papadopoulos, H. Kurniawati, A. S. B. M. Shariff, L. J. Wong, and N. M. Patrikalakis, "Experiments on Surface Reconstruction for Partially Submerged Marine Structures," *Journal of Field Robotics*, vol. 31, no. 2, pp. 225–244, 2014.

[23] J. C. Leedekerken, M. F. Fallon, and J. J. Leonard, *Mapping Complex Marine Environments with Autonomous Surface Craft*, pp. 525–539. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[24] M. I. Pereira, R. M. Claro, P. N. Leite, and A. M. Pinto, "Advancing autonomous surface vehicles: A 3d perception system for the recognition and assessment of docking-based structures," *IEEE Access*, vol. 9, pp. 53030–53045, 2021.

[25] D. F. Duarte, M. I. Pereira, and A. M. Pinto, "Multiple vessel detection and tracking in harsh maritime environments," in *OCEANS 2021: San Diego – Porto*, pp. 1–5, 2021.

[26] IMO, "COLREGS - International Regulations for Preventing Collisions at Sea," *Convention on the International Regulations for Preventing Collisions at Sea, 1972*, 2012.

[27] D. F. Campos, A. Matos, and A. M. Pinto, "Modular multi-domain aware autonomous surface vehicle for inspection," *IEEE Access*, vol. 10, pp. 113355–113375, 2022.

[28] D. F. Campos, E. P. Gonçalves, H. J. Campos, M. I. Pereira, and A. M. Pinto, "Nautilus: An autonomous surface vehicle with a multilayer software architecture for offshore inspection," *Journal of Field Robotics*, vol. 41, no. 4, pp. 966–990, 2024.

[29] D. F. Campos, A. Matos, and A. M. Pinto, "Multi-domain Mapping for Offshore Asset Inspection using an Autonomous Surface Vehicle," in *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 221–226, IEEE, apr 2020.

[30] D. F. Campos, A. Matos, and A. M. Pinto, "Multi-domain inspection of offshore wind farms using an autonomous surface vehicle," *SN Applied Sciences*, vol. 3, p. 455, 4 2021.

[31] D. F. Campos, A. Matos, and A. M. Pinto, "An Adaptive Velocity Obstacle Avoidance Algorithm for Autonomous Surface Vehicles," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8089–8096, IEEE, nov 2019.

[32] D. F. Campos, P. N. Leite, S. Renato, M. I. Pereira, R. Claro, and A. M. Pinto, "ROAM@CRAS - A haRbor multidOmAin Mapping dataset," INESC TEC research data repository, 2020.

[33] R. Silva, P. Leite, D. Campos, and A. M. Pinto, "Hybrid approach to estimate a collision-free velocity for autonomous surface vehicles," in *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 1–6, 2019.

[34] P. N. Leite, R. J. Silva, D. F. Campos, and A. M. Pinto, "Dense disparity maps from rgb and sparse depth information using deep regression models," in *Image Analysis and Recognition* (A. Campilho, F. Karray, and Z. Wang, eds.), (Cham), pp. 379–392, Springer International Publishing, 2020.

[35] F. S. Neves, H. J. Campos, D. F. Campos, R. M. Claro, P. N. Almeida, J. V. Marques, and A. M. Pinto, "Shore control centre for multi-domain heterogeneous robotic vehicles," in *OCEANS 2023 - Limerick*, pp. 1–4, 2023.

[36] A. M. Pinto, J. V. A. Marques, N. Abreu, D. F. Campos, M. I. Pereira, E. Gonçalves, H. J. Campos, P. Pereira, F. Neves, A. Matos, S. Govindaraj, and L. Durand, "Atlantis coastal testbed: A near-real playground for the testing and validation of robotics for o&m," in *OCEANS 2023 - Limerick*, pp. 1–5, 2023.

[37] A. M. Pinto, A. Matos, J. V. A. Marques, D. F. Campos, M. I. Pereira, R. Claro, E. Mikola, J. Formiga, M. E. Mobachi, J.-J. Stoker, J. Prevosto, S. Govindaraj, D. Ribas, P. Ridao, and L. Aceto, "Promoting the use of robotics in the inspection and maintenance of offshore wind," in *Robotics and Automation Solutions for Inspection and Maintenance in Critical Infrastructures* (K. Loupos, ed.), pp. 202–240, Now Publishers, 2024.

[38] ATLANTIS, "Survey on legal and regulatory specifications for IMR robots operating in offshore wind farms," July 2021.

[39] M. Fun Sang Cepeda, M. d. S. Freitas Machado, F. H. Sousa Barbosa, D. Santana Souza Moreira, M. J. Legaz Almansa, M. I. Lourenço de Souza, and J.-D. Caprace, "Exploring autonomous and remotely operated vehicles in offshore structure inspections," *Journal of Marine Science and Engineering*, vol. 11, no. 11, 2023.

[40] X. Wu, Y. Hu, Y. Li, J. Yang, L. Duan, T. Wang, T. Adcock, Z. Jiang, Z. Gao, Z. Lin, A. Borthwick, and S. Liao, "Foundations of offshore wind turbines: A review," *Renewable and Sustainable Energy Reviews*, vol. 104, pp. 379–393, 2019.

[41] S. Bernardini, F. Jovan, Z. Jiang, S. Watson, A. Weightman, P. Moradi, T. Richardson, R. Sadeghian, and S. Sareh, "A multi-robot platform for the autonomous operation and maintenance of offshore wind farms," in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, (Richland, SC), p. 1696–1700, International Foundation for Autonomous Agents and Multiagent Systems, 2020.

[42] M. Civera and C. Surace, "Non-destructive techniques for the condition and structural health monitoring of wind turbines: A literature review of the last 20 years," *Sensors*, vol. 22, no. 4, 2022.

[43] K. Vázquez, R. Rodríguez, and M. Esteban, "Proposal for a programme of periodic inspections for preventive maintenance of offshore monopiles," *Ocean Engineering*, vol. 277, p. 114262, 2023.

[44] M. Esteban, J.-S. López-Gutiérrez, V. Negro, and L. Sanz, "Riprap scour protection for monopiles in offshore wind farms," *Journal of Marine Science and Engineering*, vol. 7, p. 440, Dec. 2019.

[45] W.-G. Qi and F.-P. Gao, *Local Scour around a Monopile Foundation for Offshore Wind Turbines and Scour Effects on Structural Responses*. IntechOpen, July 2020.

[46] X. Bai, B. Li, X. Xu, and Y. Xiao, "A review of current research and advances in unmanned surface vehicles," *Journal of Marine Science and Application*, vol. 21, p. 47–58, June 2022.

[47] N. Amiri, M. Shaterabadi, K. Reza Kashyzadeh, and M. Chizari, "A comprehensive review on design, monitoring, and failure in fixed offshore platforms," *Journal of Marine Science and Engineering*, vol. 9, no. 12, p. 1349, 2021.

[48] T. Smith, S. Mukhopadhyay, R. R. Murphy, T. Manzini, and I. Rodriguez, "Path coverage optimization for usv with side scan sonar for victim recovery," in *2022 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 160–165, 2022.

[49] C. V. Amaechi, A. Reda, I. M. Kgosiemang, I. A. Ja'e, A. K. Oyetunji, M. A. Olukolajo, and I. B. Igwe, "Guidelines on asset management of offshore facilities for monitoring, sustainable maintenance, and safety practices," *Sensors*, vol. 22, no. 19, p. 7270, 2022.

[50] C. Zhou, S. Gu, Y. Wen, Z. Du, C. Xiao, L. Huang, and M. Zhu, "The review unmanned surface vehicle path planning: Based on multi-modality constraint," *Ocean Engineering*, vol. 200, p. 107043, 2020.

[51] M. Wahlström, J. Hakulinen, H. Karvonen, and I. Lindborg, "Human factors challenges in unmanned ship operations – insights from other domains," *Procedia Manufacturing*, vol. 3, pp. 1038–1045, 2015. 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015.

[52] Z. Liu, Y. Zhang, X. Yu, and C. Yuan, "Unmanned surface vehicles: An overview of developments and challenges," *Annual Reviews in Control*, vol. 41, pp. 71–93, 2016.

[53] N. Cruz, A. Matos, S. Cunha, and S. Silva, "Zarco – an Autonomous Craft for Underwater Surveys," in *Proceedings of the 7th Geomatic Week*, 2007.

[54] J. Han, J. Park, T. Kim, and J. Kim, "Precision navigation and mapping under bridges with an unmanned surface vehicle," *Autonomous Robots*, vol. 38, no. 4, pp. 349–362, 2015.

[55] L. Garberoglio, P. Moreno, I. Mas, and J. I. Giribet, "Autonomous vehicles for outdoor multidomain mapping," in *2018 IEEE Biennial Congress of Argentina (ARGENCON)*, pp. 1–8, IEEE, 2018.

[56] S. Barkby, S. B. Williams, O. Pizarro, and M. V. Jakuba, "A featureless approach to efficient bathymetric SLAM using distributed particle mapping," *Journal of Field Robotics*, vol. 28, no. 1, pp. 19–39, 2011.

[57] D. Thompson, D. Caress, C. Paull, D. Clague, H. Thomas, and D. Conlin, "Mbari mapping auv operations: In the gulf of california," in *2012 Oceans*, pp. 1–5, IEEE, 2012.

[58] A. Martins, J. Almeida, C. Almeida, A. Dias, N. Dias, J. Aaltonen, A. Heininen, K. T. Koskinen, C. Rossi, S. Dominguez, C. Vörös, S. Henley, M. McLoughlin, H. van Moerkerk, J. Tweedie, B. Bodo, N. Zajzon, and E. Silva, "Ux 1 system design - a robotic system for underwater mining exploration," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1494–1500, 2018.

[59] A. Martins, J. Almeida, C. Almeida, B. Matias, S. Kapusniak, and E. Silva, "Eva a hybrid rov/auv for underwater mining operations support," in *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*, pp. 1–7, 2018.

[60] E. Albrechtsen, "Occupational Safety Management in the Offshore Windindustry–Status and Challenges," *Energy Procedia*, vol. 24, no. January, pp. 313–321, 2012.

[61] O. Khalid, G. Hao, C. Desmond, H. Macdonald, F. D. McAuliffe, G. Dooly, and W. Hu, "Applications of robotics in floating offshore wind farm operations and maintenance: Literature review and trends," *Wind Energy*, vol. 25, no. 11, pp. 1880–1899, 2022.

[62] D. Curto, V. Franzitta, and A. Guercio, "Sea wave energy. a review of the current technologies and perspectives," *Energies*, vol. 14, no. 20, 2021.

[63] F. Plumet, C. Pêtrès, M.-A. Romero-Ramirez, B. Gas, and S.-H. Ieng, "Toward an autonomous sailing boat," *IEEE Journal of Oceanic Engineering*, vol. 40, no. 2, pp. 397–407, 2015.

[64] G. Eudeline, "Drix USV - Improving Safety and Margins Through Efficient Data Acquisition," in *Abu Dhabi International Petroleum Exhibition & Conference*, Society of Petroleum Engineers, nov 2019.

[65] H. Ferreira, C. Almeida, A. Martins, J. Almeida, N. Dias, A. Dias, and E. Silva, "Autonomous bathymetry for risk assessment with ROAZ robotic surface vehicle," in *OCEANS 2009-EUROPE*, pp. 1–6, IEEE, may 2009.

[66] J. Han, Y. Cho, J. Kim, J. Kim, N.-s. Son, and S. Y. Kim, "Autonomous collision detection and avoidance for aragon usv: Development and field tests," *Journal of Field Robotics*, vol. 37, no. 6, pp. 987–1002, 2020.

[67] W. Wang, B. Gheneti, L. A. Mateos, F. Duarte, C. Ratti, and D. Rus, "Roboat: An autonomous surface vehicle for urban waterways," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6340–6347, 2019.

[68] W. Wang, D. Fernández-Gutiérrez, R. Doornbusch, J. Jordan, T. Shan, P. Leoni, N. Hagemann, J. K. Schiphorst, F. Duarte, C. Ratti, and D. Rus, "Roboat iii: An autonomous surface vessel for urban transportation," *Journal of Field Robotics*, vol. 40, no. 8, pp. 1996–2009, 2023.

[69] R. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal on Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.

[70] T. Hasegawa, T. Suehiro, and K. Takase, "A model-based manipulation system with skill-based execution," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 535–544, 1992.

[71] M. R. Pedersen, L. Nalpantidis, R. S. Andersen, C. S., S. Bøgh, V. Krüger, and O. Madsen, "Robot skills for manufacturing: From concept to industrial deployment," *Robotics and Computer-Integrated Manufacturing*, vol. 37, pp. 282–291, 2016.

[72] C. Schou, R. S. Andersen, D. Chrysostomou, S. Bøgh, and O. Madsen, "Skill-based instruction of collaborative robots in industrial settings," *Robotics and Computer-Integrated Manufacturing*, vol. 53, pp. 72–80, 2018.

[73] M. Kyrarini, M. A. Haseeb, D. Ristić-Durrant, and A. Gräser, "Robot learning of industrial assembly task via human demonstrations," *Autonomous Robots*, vol. 43, no. 1, p. 239 – 257, 2019.

[74] P.-C. Huang, Y.-H. Hsieh, and A. K. Mok, "A skill-based programming system for robotic furniture assembly," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, pp. 355–361, 2018.

[75] A. Leite, A. M. Pinto, and A. Matos, "A safety monitoring model for a faulty mobile robot," *Robotics*, vol. 7, no. 3, 2018.

[76] A. Albore, D. Doose, C. Grand, C. Lesire, and A. Manecy, "Skill-based architecture development for online mission reconfiguration and failure management," in *2021 IEEE/ACM 3rd International Workshop on Robotics Software Engineering (RoSE)*, pp. 47–54, 2021.

[77] A. Albore, D. Doose, C. Grand, J. Guiochet, C. Lesire, and A. Manecy, "Skill-based design of dependable robotic architectures," *Robotics and Autonomous Systems*, vol. 160, p. 104318, 2023.

[78] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3d point clouds: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4338–4364, 2021.

[79] S. Garg, N. Sünderhauf, F. Dayoub, D. Morrison, A. Cosgun, G. Carneiro, Q. Wu, T.-J. Chin, I. Reid, S. Gould, P. Corke, and M. Milford, "Semantics for robotic mapping, perception and interaction: A survey," *Foundations and Trends® in Robotics*, vol. 8, no. 1–2, pp. 1–224, 2020.

[80] H. Zhang, Z.-Q. Liu, and Y.-L. Wang, "U-loam: A real-time 3d lidar slam system for water-surface scene applications," in *2022 IEEE International Conference on Unmanned Systems (ICUS)*, pp. 653–657, 2022.

[81] Y. Cai, X. Chen, C. Zhang, K.-Y. Lin, X. Wang, and H. Li, "Semantic scene completion via integrating instances and scene in-the-loop," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 324–333, June 2021.

[82] D. F. Duarte, M. I. Pereira, and A. M. Pinto, "Multiple vessel detection in harsh maritime environments," *Marine Technology Society Journal*, vol. 56, pp. 58–67, 10 2022.

[83] L. Fahrni, P. Thies, L. Johanning, and J. Cowles, "Scope and feasibility of autonomous robotic subsea intervention systems for offshore inspection, maintenance and repair," 2018.

[84] P. N. Leite and A. M. Pinto, "Fusing heterogeneous tri-dimensional information for reconstructing submerged structures in harsh sub-sea environments," *Information Fusion*, vol. 103, p. 102126, 2024.

[85] Y. Singh, S. Sharma, R. Sutton, and D. Hatton, "Towards use of Dijkstra Algorithm for Optimal Navigation of an Unmanned Surface Vehicle in a Real-Time Marine Environment with results from Artificial Potential Field," *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 12, no. 1, pp. 125–131, 2018.

[86] D. K. M. Kufoalor, T. A. Johansen, E. F. Brekke, A. Hepsø, and K. Trnka, "Autonomous maritime collision avoidance: Field verification of autonomous surface vehicle behavior in challenging scenarios," *Journal of Field Robotics*, vol. 37, no. 3, pp. 387–403, 2020.

[87] P. Leite, R. Silva, A. Matos, and A. M. Pinto, "An hierarchical architecture for docking autonomous surface vehicles," in *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 1–6, April 2019.

[88] M. I. Pereira and A. M. Pinto, "Reinforcement learning based robot navigation using illegal actions for autonomous docking of surface vehicles in unknown environments," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108506, 2024.

[89] L. Yang, B. Li, W. Li, B. Jiang, and J. Xiao, "Semantic metric 3d reconstruction for concrete inspection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[90] L. Roldão, R. de Charette, and A. Verroust-Blondet, "3d semantic scene completion: A survey," *International Journal of Computer Vision*, vol. 130, p. 1978–2005, June 2022.

[91] J. Hong, K. de Langis, C. Wyethv, C. Walaszek, and J. Sattar, "Semantically-aware strategies for stereo-visual robotic obstacle avoidance," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2450–2456, IEEE, 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)*, pp. 14112–14119, IEEE, 2021.

[93] J. Han, Y. Cho, and J. Kim, "Coastal slam with marine radar for usv operation in gps-restricted situations," *IEEE Journal of Oceanic Engineering*, vol. 44, no. 2, pp. 300–309, 2019.

[94] W. Wang, T. Shan, P. Leoni, D. Fernández-Gutiérrez, D. Meyers, C. Ratti, and D. Rus, "Roboat ii: A novel autonomous surface vessel for urban environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1740–1747, 2020.

[95] H. Wang, Y. Yin, and Q. Jing, "Comparative analysis of 3d lidar scan-matching methods for state estimation of autonomous surface vessel," *Journal of Marine Science and Engineering*, vol. 11, no. 4, 2023.

[96] A. W. Fitzgibbon, "Robust registration of 2D and 3D point sets," *Image and Vision Computing*, vol. 21, no. 13-14, pp. 1145–1153, 2003.

[97] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized gicp for fast and accurate 3d point cloud registration," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11054–11059, 2021.

[98] L. Zhao and Y. Bai, "Energy efficient coverage path planning for usv-assisted inland bathymetry under current effects: An analysis on sweep direction," *Ocean Engineering*, vol. 305, p. 117910, 2024.

[99] J. Jung, J. Park, J. Choi, and H.-T. Choi, "Bathymetric pose graph optimization with regularized submap matching," *IEEE Access*, vol. 10, pp. 31155–31164, 2022.

[100] J. Jung, Y. Lee, J. Park, and T.-K. Yeu, "Multi-modal sonar mapping of offshore cable lines with an autonomous surface vehicle," *Journal of Marine Science and Engineering*, vol. 10, no. 3, 2022.

[101] J. Han and J. Kim, "Three-dimensional reconstruction of a marine floating structure with an unmanned surface vessel," *IEEE Journal of Oceanic Engineering*, vol. 44, no. 4, pp. 984–996, 2019.

[102] L. Nan, K. Xie, and A. Sharf, "A search-classify approach for cluttered indoor scene understanding," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, pp. 1–10, 2012.

[103] T. Shao, W. Xu, K. Zhou, J. Wang, D. Li, and B. Guo, "An interactive approach to semantic modeling of indoor scenes with an rgbd camera," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, pp. 1–11, 2012.

[104] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "Pcn: Point completion network," in *2018 international conference on 3D vision (3DV)*, pp. 728–737, IEEE, 2018.

[105] J. Hou, A. Dai, and M. Nießner, "Revealnet: Seeing behind objects in rgb-d scans," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2095–2104, 2020.

[106] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik, "Aligning 3d models to rgb-d images of cluttered scenes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4731–4740, 2015.

[107] D. Li, T. Shao, H. Wu, and K. Zhou, "Shape completion from a single rgbd image," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 7, pp. 1809–1822, 2017.

[108] H. Liu, Y. Cong, C. Yang, and Y. Tang, "Efficient 3d object recognition via geometric information preservation," *Pattern Recognition*, vol. 92, pp. 135–145, 2019.

[109] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone, "Kimera: From slam to spatial perception with 3d dynamic scene graphs," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1510–1546, 2021.

[110] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, 2021.

[111] Y. Nie, J. Hou, X. Han, and M. Nießner, "Rfd-net: Point scene understanding by semantic instance reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4608–4618, 2021.

[112] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[113] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, (New York, NY, USA), p. 163–169, Association for Computing Machinery, 1987.

[114] J. Wan, Z. Fu, M. Chen, P. Zhang, H. Wang, and Y. Guo, "Pmnet: A point-to-mesh network for 3-d semantic instance reconstruction," *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1–5, 2023.

[115] D. Fernandes, A. Silva, R. Névoa, C. Simões, D. Gonzalez, M. Guevara, P. Novais, J. Monteiro, and P. Melo-Pinto, "Point-cloud based 3d object detection and classification methods for self-driving applications: A survey and taxonomy," *Information Fusion*, vol. 68, pp. 161–191, 2021.

[116] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Trans. Graph.*, vol. 38, oct 2019.

[117] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, "3d object detection with pointformer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7463–7472, 2021.

[118] Y. Wang, J. Yin, W. Li, P. Frossard, R. Yang, and J. Shen, "Ssda3d: Semi-supervised domain adaptation for 3d object detection from point cloud," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 2707–2715, 2023.

[119] X. Feng, H. Du, H. Fan, Y. Duan, and Y. Liu, "Seformer: Structure embedding transformer for 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 632–640, 2023.

[120] Y. Pei, X. Zhao, H. Li, J. Ma, J. Zhang, and S. Pu, "Clusterformer: Cluster-based transformer for 3d object detection in point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6664–6673, 2023.

[121] Y. Huang, S. Zhou, J. Zhang, J. Dong, and N. Zheng, "Voxel or pillar: Exploring efficient point cloud representation for 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 2426–2435, 2024.

[122] Y. Liu, Z. Yang, J. Tong, J. Yang, J. Peng, L. Zhang, and W. Cheng, "ET-PointPillars: improved PointPillars for 3D object detection based on optimized voxel downsampling," *Mach. Vis. Appl.*, vol. 35, May 2024.

[123] P. Sun, W. Wang, Y. Chai, G. Elsayed, A. Bewley, X. Zhang, C. Sminchisescu, and D. Anguelov, "Rsn: Range sparse net for efficient, accurate lidar 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5725–5734, 2021.

[124] Z. Liang, Z. Zhang, M. Zhang, X. Zhao, and S. Pu, "Rangeioudet: Range image based real-time 3d object detector optimized by intersection over union," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7140–7149, 2021.

[125] Z. Tian, X. Chu, X. Wang, X. Wei, and C. Shen, "Fully convolutional one-stage 3d object detection on lidar range images," *Advances in Neural Information Processing Systems*, vol. 35, pp. 34899–34911, 2022.

[126] J. Mao, S. Shi, X. Wang, and H. Li, "3D object detection for autonomous driving: A comprehensive survey," *Int. J. Comput. Vis.*, vol. 131, pp. 1909–1963, Aug. 2023.

[127] X. Li, J. Guivant, and S. Khan, "Real-time 3d object proposal generation and classification using limited processing resources," *Robotics and Autonomous Systems*, vol. 130, p. 103557, 2020.

[128] S. Wang, Z. Chen, S. Sun, J. Li, and R. Sun, "3d object detection method based on lidar point cloud data," in *Computer Supported Cooperative Work and Social Computing* (Y. Sun, T. Lu, T. Wang, H. Fan, D. Liu, and B. Du, eds.), (Singapore), pp. 496–506, Springer Nature Singapore, 2024.

[129] S. Garg, K. Rana, M. Hosseinzadeh, L. Mares, N. Sünderhauf, F. Dayoub, and I. Reid, "Robohop: Segment-based topological map representation for open-world visual navigation," 2024.

[130] E. Sandström, Y. Li, L. Van Gool, and M. R. Oswald, "Point-slam: Dense neural point cloud-based slam," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18433–18444, 2023.

[131] J. Sun, Y. Xie, L. Chen, X. Zhou, and H. Bao, "Neuralrecon: Real-time coherent 3d reconstruction from monocular video," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15598–15607, 2021.

[132] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12786–12796, 2022.

[133] Z.-X. Zou, S.-S. Huang, Y.-P. Cao, T.-J. Mu, Y. Shan, and H. Fu, "Mononeuralfusion: Online monocular neural 3d reconstruction with geometric priors," *arXiv preprint arXiv:2209.15153*, 2022.

[134] N. Marniok, O. Johannsen, and B. Goldluecke, "An efficient octree design for local variational range image fusion," in *Pattern Recognition: 39th German Conference, GCPR 2017, Basel, Switzerland, September 12–15, 2017, Proceedings 39*, pp. 401–412, Springer, 2017.

[135] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15651–15663, 2020.

[136] X. Yang, H. Li, H. Zhai, Y. Ming, Y. Liu, and G. Zhang, "Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation," in *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 499–507, IEEE, 2022.

[137] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 1, 2017.

[138] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.

[139] Y. Wang, Q. Han, M. Habermann, K. Daniilidis, C. Theobalt, and L. Liu, "Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3295–3306, 2023.

[140] T. Schops, T. Sattler, and M. Pollefeys, "Bad slam: Bundle adjusted direct rgb-d slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 134–144, 2019.

[141] H. M. Cho, H. Jo, and E. Kim, "Sp-slam: Surfel-point simultaneous localization and mapping," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 5, pp. 2568–2579, 2021.

[142] C.-M. Chung, Y.-C. Tseng, Y.-C. Hsu, X.-Q. Shi, Y.-H. Hua, J.-F. Yeh, W.-C. Chen, Y.-T. Chen, and W. H. Hsu, "Orbeez-slam: A real-time monocular visual slam with orb features and nerf-realized mapping," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9400–9406, IEEE, 2023.

[143] X. Meng, N. Ratliff, Y. Xiang, and D. Fox, "Scaling local control to large-scale topological navigation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 672–678, 2020.

[144] W. Xue, R. Ying, Z. Gong, R. Miao, F. Wen, and P. Liu, "Slam based topological mapping and navigation," in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pp. 1336–1341, 2020.

[145] A. S. Aguiar, F. N. d. Santos, L. C. Santos, A. J. Sousa, and J. Boaventura-Cunha, "Topological map-based approach for localization and mapping memory optimization," *Journal of Field Robotics*, vol. 40, no. 3, pp. 447–466, 2023.

[146] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard, "Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, p. 1309–1332, 2016.

[147] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone, "3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans," *arXiv preprint arXiv:2002.06289*, 2020.

[148] S. Zhang, A. Hao, H. Qin, *et al.*, "Knowledge-inspired 3d scene graph prediction in point cloud," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18620–18632, 2021.

[149] X. Li, D. Guo, H. Liu, and F. Sun, "Embodied semantic scene graph generation," in *Conference on robot learning*, pp. 1585–1594, PMLR, 2022.

[150] J. Bae, D. Shin, K. Ko, J. Lee, and U.-H. Kim, "A survey on 3d scene graphs: Definition, generation and application," in *Robot Intelligence Technology and Applications 7* (J. Jo, H.-L. Choi, M. Helbig, H. Oh, J. Hwangbo, C.-H. Lee, and B. Stantic, eds.), (Cham), pp. 136–147, Springer International Publishing, 2023.

[151] G. Wu, D. Li, H. Ding, D. Shi, and B. Han, "An overview of developments and challenges for unmanned surface vehicle autonomous berthing," *Complex & Intelligent Systems*, vol. 10, p. 981–1003, Aug. 2023.

[152] Y. Fan, Z. Sun, and G. Wang, "A novel intelligent collision avoidance algorithm based on deep reinforcement learning approach for usv," *Ocean Engineering*, vol. 287, p. 115649, 2023.

[153] M. Xu, Y. Liu, Q. Huang, Y. Zhang, and G. Luan, "An improved dijkstra's shortest path algorithm for sparse network," *Applied Mathematics and Computation*, vol. 185, no. 1, pp. 247–254, 2007.

[154] B. Shi, Y. Su, C. Wang, L. Wan, and Y. Luo, "Study on intelligent collision avoidance and recovery path planning system for the waterjet-propelled unmanned surface vehicle," *Ocean Engineering*, vol. 182, pp. 489–498, 2019.

[155] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[156] Y. Huang, L. Chen, and P. van Gelder, "Generalized velocity obstacle algorithm for preventing ship collisions at sea," *Ocean Engineering*, vol. 173, pp. 142–156, 2019.

[157] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-Body Collision Avoidance," in *Robotics Research* (C. Pradalier, R. Siegwart, and G. Hirzinger, eds.), (Berlin, Heidelberg), pp. 3–19, Springer Berlin Heidelberg, 2011.

[158] W. Shaobo, Z. Yingjun, and L. Lianbo, "A collision avoidance decision-making system for autonomous ship based on modified velocity obstacle method," *Ocean Engineering*, vol. 215, p. 107910, 2020.

[159] J. Zhuang, Y. Zhang, P. Xu, Y. Zhao, J. Luo, and S. Song, "Multiple moving obstacles avoidance for usv using velocity obstacle method," in *2021 IEEE International Conference on Unmanned Systems (ICUS)*, pp. 140–145, 2021.

[160] Y. Cho, J. Han, and J. Kim, "Efficient colreg-compliant collision avoidance in multi-ship encounter situations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 1899–1911, 2022.

[161] C. Zhou, Y. Wang, L. Wang, and H. He, "Obstacle avoidance strategy for an autonomous surface vessel based on modified deep deterministic policy gradient," *Ocean Engineering*, vol. 243, p. 110166, 2022.

[162] R. Han, S. Chen, S. Wang, Z. Zhang, R. Gao, Q. Hao, and J. Pan, "Reinforcement learned distributed multi-robot navigation with reciprocal velocity obstacle shaped rewards," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 5896–5903, 2022.

[163] J. D. Van Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1928–1935, 2008.

[164] D. Xue, D. Wu, A. S. Yamashita, and Z. Li, "Proximal policy optimization with reciprocal velocity obstacle based collision avoidance path planning for multi-unmanned surface vehicles," *Ocean Engineering*, vol. 273, p. 114005, 2023.

[165] O. Litany, A. Bronstein, M. Bronstein, and A. Makadia, "Deformable shape completion with graph convolutional autoencoders," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1886–1895, 2018.

[166] C.-J. Si, Z.-B. Yin, Z.-Q. Fan, F.-Y. Liu, R. Niu, N. Yao, S.-Q. Shen, M.-D. Shi, and Y.-J. Xi, "Point cloud completion network for 3d shapes with morphologically diverse structures," *Complex & Intelligent Systems*, vol. 10, p. 3389–3409, Feb. 2024.

[167] A. Yazdani, S. MahmoudZadeh, O. Yakimenko, and H. Wang, "Perception-aware online trajectory generation for a prescribed manoeuvre of unmanned surface vehicle in cluttered unstructured environment," *Robotics and Autonomous Systems*, vol. 169, p. 104508, 2023.

[168] M. Teng, L. Ye, Z. Yuxin, J. Yanqing, Z. Qianyi, and A. M. Pascoal, "Efficient bathymetric slam with invalid loop closure identification," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 5, pp. 2570–2580, 2021.

[169] R. Silva, A. Matos, and A. M. Pinto, "Multi-criteria metric to evaluate motion planners for underwater intervention," *Autonomous Robots*, vol. 46, pp. 971–983, 12 2022.

[170] A. M. Pinto and A. C. Matos, "Maresye: A hybrid imaging system for underwater robotic applications," *Information Fusion*, vol. 55, pp. 16 – 29, 2020.

[171] P. P. V. Tecchio, M. Vieira, L. Batista, J. Cerqueira, R. Lima, O. Pinheiro, L. Nardy, M. A. Dos Reis, H. Lepikson, and L. F. M. Lopes, "Overview of robotic applications on offshore ogi imr: Splash zone," in *OCEANS 2021: San Diego – Porto*, pp. 1–10, 2021.

[172] F. P. Santos, Â. P. Teixeira, and C. Guedes Soares, *Operation and Maintenance of Floating Offshore Wind Turbines*, pp. 181–193. Cham: Springer International Publishing, 2016.

[173] R. J. Silva, P. Nuno Leite, and A. M. Pinto, "Multi-agent optimization for offshore wind farm inspection using an improved population-based metaheuristic," in *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 53–60, 2020.

[174] R. Claro, R. Silva, and A. Pinto, "Detection and mapping of monopiles in offshore wind farms using autonomous surface vehicles," in *Global Oceans 2020: Singapore – U.S. Gulf Coast*, pp. 1–4, IEEE, 10 2020.

[175] M. T. Wolf, C. Assad, Y. Kuwata, A. Howard, H. Aghazarian, D. Zhu, T. Lu, A. Trebi-Ollennu, and T. Huntsberger, "360-degree visual detection and target tracking on an autonomous surface vehicle," *Journal of Field Robotics*, vol. 27, no. 6, pp. 819–833, 2010.

[176] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, vol. 3, 01 2009.

[177] T. Moore and D. Stouch, "A Generalized Extended Kalman Filter Implementation for the Robot Operating System," in *Intelligent Autonomous Systems 13* (E. Menegatti, N. Michael, K. Berns, and H. Yamaguchi, eds.), (Cham), pp. 335–348, Springer International Publishing, 2016.

[178] J. P. Snyder, "Map projections used by the U.S. Geological Survey," tech. rep., 1982.

[179] M. I. Pereira, P. N. Leite, and A. M. Pinto, "Detecting Docking-based Structures for Persistent ASVs using a Volumetric Neural Network," in *Global OCEANS 2020: Singapore - U.S. Gulf Coast*, pp. 1–6, 2020.

[180] N. Wawrzyniak, T. Hyla, and A. Popik, "Vessel Detection and Tracking Method Based on Video Surveillance," *Sensors*, vol. 19, p. 5230, nov 2019.

[181] L. Steccanella, D. Bloisi, A. Castellini, and A. Farinelli, "Waterline and obstacle detection in images from low-cost autonomous boats for environmental monitoring," *Robotics and Autonomous Systems*, vol. 124, p. 103346, feb 2020.

[182] A. R. Gaspar, A. Nunes, A. M. Pinto, and A. Matos, "Urban@CRAS dataset: Benchmarking of visual odometry and SLAM techniques," *Robotics and Autonomous Systems*, vol. 109, pp. 59–67, 2018.

[183] Y. M. Wang, Y. Li, and J. B. Zheng, "A camera calibration technique based on OpenCV," in *The 3rd International Conference on Information Sciences and Interaction Sciences*, pp. 403–406, IEEE, jun 2010.

[184] A. Dhall, K. Chelani, V. Radhakrishnan, and K. M. Krishna, "LiDAR-Camera Calibration using 3D-3D Point correspondences," may 2017.

[185] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15–22, IEEE, may 2014.

[186] B. Teixeira, H. Silva, A. Matos, and E. Silva, "Deep Learning for Underwater Visual Odometry Estimation," *IEEE Access*, vol. 8, pp. 44687–44701, 2020.

[187] E. Skjellaug, E. F. Brekke, and A. Stahl, "Feature-Based Laser Odometry for Autonomous Surface Vehicles utilizing the Point Cloud Library," in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pp. 1–8, IEEE, jul 2020.

[188] Z. Zhang and D. Scaramuzza, "A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7244–7251, IEEE, oct 2018.

[189] P. Dellenbach, J. Deschaud, B. Jacquet, and F. Goulette, "CT-ICP: real-time elastic lidar odometry with loop closure," *CoRR*, vol. abs/2109.12979, 2021.

[190] J. McConnell, I. Collado-Gonzalez, and B. Englot, "Perception for underwater robots," *Current Robotics Reports*, vol. 3, p. 177–186, Oct. 2022.

[191] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[192] X. Zhao and S. Schwertfeger, "3dref: 3d dataset and benchmark for reflection detection in rgb and lidar data," *arXiv preprint arXiv:2403.06538*, 2024.

[193] "Towards 3D object maps for autonomous household robots," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3191–3198, IEEE, 2007.

[194] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: an efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[195] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 5105–5114, Curran Associates Inc., 2017.

[196] S. Sánchez, J.-S. López-Gutiérrez, V. Negro, and M. D. Esteban, "Foundations in offshore wind farms: Evolution, characteristics and range of use. analysis of main dimensional parameters in monopile foundations," *Journal of Marine Science and Engineering*, vol. 7, no. 12, 2019.

[197] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[198] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, p. 469–483, dec 1996.

[199] L. Li and M. Heizmann, "Applying plain transformers to real-world point clouds," *arXiv preprint arXiv:2303.00086*, 2023.

[200] S. Qiu, Y. Wu, S. Anwar, and C. Li, "Investigating attention mechanism in 3d point cloud object detection," in *2021 International Conference on 3D Vision (3DV)*, pp. 403–412, 2021.

[201] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[202] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[203] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[204] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-bert: Pre-training 3d point cloud transformers with masked point modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19313–19322, June 2022.

[205] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Robotics: Science and Systems V*, RSS2009, Robotics: Science and Systems Foundation, June 2009.

[206] D. Eberly, "A robust eigensolver for 3×3 symmetric matrices."

[207] B. Bingham, C. Aguero, M. McCarrin, J. Klamo, J. Malia, K. Allen, T. Lum, M. Rawson, and R. Waqar, "Toward maritime robotic simulation in gazebo," in *Proceedings of MTS/IEEE OCEANS Conference*, (Seattle, WA), October 2019.

[208] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *2011 IEEE International Conference on Robotics and Automation*, pp. 1–4, 2011.

[209] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[210] R. B. Rusu, *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Technische Universität München, 2009.

[211] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, p. 226–231, AAAI Press, 1996.

[212] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *CoRR*, vol. abs/1711.05101, 2017.

[213] A. Al-Kababji, F. Bensaali, and S. P. Dakua, "Scheduling techniques for liver segmentation: Reducelronplateau vs onecyclelr," in *Intelligent Systems and Pattern Recognition* (A. Bennour, T. Ensari, Y. Kessentini, and S. Eom, eds.), (Cham), pp. 204–212, Springer International Publishing, 2022.

[214] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, Oct 2017.

[215] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1912–1920, 2015.

[216] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, p. 321–357, June 2002.

[217] G. King and L. Zeng, "Logistic regression in rare events data," *Political Analysis*, vol. 9, no. 2, p. 137–163, 2001.

[218] X. Sun, Z. Lian, and J. Xiao, "Srinet: Learning strictly rotation-invariant representations for point cloud classification and segmentation," in *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, (New York, NY, USA), p. 980–988, Association for Computing Machinery, 2019.

[219] Z. Tao, Y. Zhu, T. Wei, and S. Lin, "Multi-head attentional point cloud classification and segmentation using strictly rotation-invariant representations," *IEEE Access*, vol. 9, p. 71133–71144, 2021.

[220] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis, "Learning so(3) equivariant representations with spherical cnns," *International Journal of Computer Vision*, vol. 128, p. 588–600, Sept. 2019.

[221] Y. You, Y. Lou, Q. Liu, Y.-W. Tai, L. Ma, C. Lu, and W. Wang, "Pointwise rotation-invariant network with adaptive sampling and 3d spherical voxel convolution," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 12717–12724, 2020.

[222] Z. Zhang, B.-S. Hua, D. W. Rosen, and S.-K. Yeung, "Rotation invariant convolutions for 3d point clouds deep learning," in *2019 International Conference on 3D Vision (3DV)*, pp. 204–213, 2019.

[223] M. Wittingen, "Offshore wind turbine monopile foundation installation with a dynamic positioned vessel," 2018.

[224] I.-W. Chen, B.-L. Wong, Y.-H. Lin, S.-W. Chau, and H.-H. Huang, "Design and analysis of jacket substructures for offshore wind turbines," *Energies*, vol. 9, no. 4, 2016.

[225] K. R. Hussein, A. W. Hussein, E. Hegazy, and A. A. Amin, "Structural design of a floating foundation for offshore wind turbines in red sea," *Analysis and Design of Marine Structures*, pp. 575–583, 2013.

[226] J. George, "Windfloat design for different turbine sizes," 2014.

[227] S. Zhou, B. Shan, Y. Xiao, C. Li, G. Hu, X. Song, Y. Liu, and Y. Hu, "Directionality effects of aligned wind and wave loads on a y-shape semi-submersible floating wind turbine under rated operational conditions," *Energies*, vol. 10, no. 12, 2017.

[228] Allianz Global Corporate & Specialty, *Safety and Shipping Review 2018*, 2018.

[229] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *International Journal of Robotics Research*, 1998.

[230] M. d. Berg, O. Cheong, M. v. Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*. Santa Clara, CA, USA: Springer-Verlag TELOS, 3rd ed. ed., 2008.

[231] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory of computing*, vol. 8, no. 1, pp. 415–428, 2012.

[232] A. M. Pinto, P. G. Costa, M. V. Correia, A. C. Matos, and A. P. Moreira, "Visual motion perception for mobile robots through dense optical flow fields," *Robotics and Autonomous Systems*, vol. 87, pp. 1–14, 2017.

[233] A. M. Pinto, A. P. Moreira, and P. G. Costa, "A Localization Method Based on Map-Matching and Particle Swarm Optimization," *Journal of Intelligent & Robotic Systems*, vol. 77, no. 2, pp. 313–326, 2015.

[234] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948 vol.4, Nov 1995.

[235] L. Zhang, Y. Zhang, and Y. Li, "Mobile robot path planning based on improved localized particle swarm optimization," *IEEE Sensors Journal*, vol. 21, no. 5, pp. 6962–6972, 2021.

[236] C. Rösmann, F. Hoffmann, and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, 2017.