

AUTOMATED 3D OBJECT RECOGNITION IN UNDERWATER SCENARIOS FOR MANIPULATION

Khadidja Himri

Per citar o enllaçar aquest document:
Para citar o enlazar este documento:
Use this url to cite or link to this publication:
<http://hdl.handle.net/10803/673811>



<http://creativecommons.org/licenses/by/4.0/deed.ca>

Aquesta obra està subjecta a una llicència Creative Commons Reconeixement

Esta obra está bajo una licencia Creative Commons Reconocimiento

This work is licensed under a Creative Commons Attribution licence



Doctoral Thesis

Automated 3D object recognition in
underwater scenarios for manipulation

KHADIDJA HIMRI

2021



Doctoral Thesis

**Automated 3D object recognition in
underwater scenarios for manipulation**

KHADIDJA HIMRI

2021

Doctoral Program in Technology

Supervised by:

PERE RIDAO and NUNO GRACIAS

Thesis submitted to University of Girona in fulfillment of the requirements for the degree
of

DOCTOR OF PHILOSOPHY

CERTIFICATE OF THESIS DIRECTION

Dr. Pere Ridao and Dr. Nuno Gracias members of the Departament d'Arquitectura i Tecnologia de Computadors of Universitat de Girona,

DECLARE:

That the work entitled *Automated 3D object recognition in underwater scenarios for manipulation* presented by Khadidja Himri to obtain the degree in Doctor of Philosophy has been developed under our supervision.

Therefore, in order to certify the aforesaid statement, we sign this document.

Girona, October 2021

Dr. Pere Ridao

Dr. Nuno Gracias

ACKNOWLEDGMENTS

It was not easy starting a PhD, but I could not have imagined being on a better team with such great research supervisors. First and foremost, I am deeply and sincerely indebted to my supervisors:

- Pere, for his guidance throughout this thesis and his full support. He has opened my mind to new ways of looking at problems, I know he has done his best to make learning easier for me, for that I am very grateful. He has been very dedicated in helping me take my work from concept to completion. I can honestly say that without his support, I would not be where I am today. He had worked his magic to make my dream a reality.

- Nuno because he brought a wealth of knowledge, experience and a supportive attitude that was invaluable and enabled me to feel confident throughout the PhD process. But also the kindness and support for international students like me, who struggle with the difficulty of being far away from their families and home countries, was a bonus that made my journey in Girona a wonderful experience.

I can not tell you everything my outstanding supervisors, but I believe I am very fortunate to learn from exceptionally brilliant leaders. Thank you for your gifts of knowledge and unwavering support. Without your help, this thesis would not have seen the light of day.

I would also like to thank all the team of *CIRS* and the *ViCOROB* research group, especially Marc Carreras, who deserves a special mention for making me feel so welcome and for contributing to one of the most beautiful and memorable experiments of my PhD. I would also like to convey my regards to Pep Forest, Xevi Cufi, Rafael Garcia and Jordi Freixenet.

Thanks are extended to an anonymous referees for careful reading of the thesis and helpful comments.

Thanks a ton to Tali, you have been a wonderful source of endless distraction during this pandemic. Thank you for all the pointless and interesting conversations over the years.

A big thank you to my office mates when I first came to the lab: Albert Ciurana, Quim, Marc Massi, Dina, Eric, Bruno and especially Julia.

A special and inspiring group from the *CIRS* is yet to be mentioned, as I praise their tremendous help and amazing source of knowledge, inspiration and sympathy, thanks to Jep, Eduard, Narcis, Carles, David, Joan, Patrick, Angelos, Juan, Alex and Lluís Petit, of whom Roger, Guillem, Klemen and Albert, especially stand out as they helped me immensely. A special recognition goes to my office mate Miguel. He has been a great friend since we started sharing the office. I could not imagine how to start my day without seeing the smiling face of Lluís Magi. Thank you for making my mornings a good start,

even when I felt down.

The Friday meetings were a wonderful source of knowledge and free food. Thanks to Quintana, Sandra, Ricard.P, Alex, Eduardo, Mariano and special thanks to “our dealer” Ricard.C.

Dozens of people have helped me with the paperwork and bureaucracy. First of all, I am very indebted to Mireia, the first person I met when I arrived in Girona, who embodied the hospitality and warmth of the Catalans very well. Thank you, Mireia, for helping me settle into Girona and for all the bureaucratic formalities. You are the jewel of CIRS. Secondly, I want to sincerely thank Eugènia Paradedà, who always responded with a smile to my many requests. I am also grateful to the Vicorob Girls: Anna, Gemma, Olga and Bego.

Thank you Joseta for the Casamoner moments and for introducing me to new tastes in music. Speaking of music, I am infinitely grateful to the Zumba group and especially Laura for the unforgettable moments of fun.

I would like to thank my old time friends Carol, Ismahane, Lina. Thank you for the happy distractions that keep me distracted outside of my research project. To the friends and colleagues who crossed my path: Fermin, Nando, Emmanuella, Simona, Enrico, Vincenzo, Gianluca, Andrea Sudo, Andrea Piarulli, Frederique, Sandra and Eric Ziegler.

Last but not least, I would like to thank my most important and fundamental source of my life energy, my family, to whom I always owe everything. They have done everything possible and impossible to support me. My sisters and brothers, you have been a support for me every step of my life, thank you for being part of my life.

A mon idole, ma source de réussite, mon pilier du courage et ma sagesse à toi Papa. Merci pour ton amour et tes encouragements, je te dois ce que je suis aujourd’hui et ce que je serai demain. A mon éternelle flamme Maman. Merci, mille merci pour ton amour inconditionnelle et tes câlins qui apaisent, nourrissent, guérissent mon âme. Aujourd’hui, j’ai réalisé notre rêve. Merci à ma confidente ma soeur Saliha, et à mon frère Youcef qui m’ont insufflé courage, sérénité et foi aux moments difficiles et opportuns. Merci pour tout ce que vous aviez fait pour moi.

A mon idole, ma source de réussite, mon pilier du courage et ma sagesse à toi Papa. Merci pour ton amour et tes encouragements, je te dois ce que je suis aujourd’hui et ce que je serai demain.

A mon éternelle flamme Maman. Merci, mille merci pour ton amour inconditionnelle et tes câlins qui apaisent, nourrissent, guérissent mon âme. Aujourd’hui, j’ai réalisé notre rêve.

Merci à ma confidente ma soeur Saliha, et à mon frère Youcef qui m’ont insufflé courage, sérénité et foi aux moments difficiles et opportuns. Merci pour tout ce que vous aviez fait pour moi.

LIST OF PUBLICATIONS

Publications in the compendium

The presented thesis is a compendium of the following research articles:

- **K. Himri**, P. Ridaou, and N. Gracias. “3D Object Recognition Based on Point Clouds in Underwater Environment with Global Descriptors: A Survey”. In: *Sensors* 19.20 (2019), p. 4451
Quality index: [JCR2019 Instruments & Instrumentation IF 3.275, Q1 (15/64)].
- **K. Himri**, P. Ridaou, and N. Gracias. “Underwater Object Recognition Using Point-Features, Bayesian Estimation and Semantic Information”. In: *Sensors* 21.5 (2021). ISSN: 1424-8220. DOI: 10.3390/s21051807
Quality index: [JCR2019 Instruments & Instrumentation IF 3.275, Q1 (15/64)].
- G. Villacrosa, **K. Himri**, P. Ridaou, and N. Gracias. “Semantic Mapping for Autonomous Subsea Intervention”. In: *Submitted to MDPI Sensors* (2021)
Quality index: [JCR2019 Instruments & Instrumentation IF 3.275, Q1 (15/64)].

Publications derived from this thesis

The work developed in this thesis also led to the following publications:

- **K. Himri**, P. Ridaou, N. Gracias, A. Palomer, N. Palomeras, and R. Pi. “Semantic SLAM for an AUV using object recognition from point clouds”. In: *Proceedings of 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2018*. Vol. 51. 29. 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2018. Croatia, Sept. 2018, pp. 360–365. DOI: <https://doi.org/10.1016/j.ifacol.2018.09.497>
- **K. Himri**, R. Pi, P. Ridaou, N. Gracias, A. Palomer, and N. Palomeras. “Object Recognition and Pose Estimation using Laser scans for Advanced Underwater Manipulation”. In: *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*. IEEE. 2018, pp. 1–6

ACRONYMS

AUV Autonomous Underwater Vehicle.

CIRS Centre de Investigació en Robòtica Submarina.

DR Dead Reckoning.

EKF Extended Kalman Filter.

GRoMI Ground Robot for Mapping Infrastructure.

I-AUV intervention Autonomous Underwater Vehicle.

IJCBB Inter-distance Joint Compatibility Branch and Bound.

IMR Inspection, Maintenance and Repair.

LiDAR Light Detection And Ranging.

MLS Mobile Laser Scanning.

NED North East Down.

SLAM Simultaneous Localization and Mapping.

SVM Support Vector Machine.

UAV unmanned aerial vehicle.

UdG Universitat de Girona.

LIST OF FIGURES

1.1	The objective of the thesis is to develop methods to build semantic maps allowing robots to discover objects and the manipulation tasks relevant for them.	9
5.1	Summary of results for all the objects and all the resolutions using synthetic data. The best descriptor is marked in green, while the worst is marked in red. (Extracted from [1])	117
5.2	Summary of results for all the objects using the same resolution in the model and the input scan and for different noise levels. The two best descriptors are marked in green, while the two worst are marked in red. (Extracted from [1])	117
5.3	Confusion Matrix for real and synthetic data, for all descriptors. (Extracted from [1])	118
5.4	Comparison of maps obtained from Dead Reckoning (DR) and Simultaneous Localization and Mapping (SLAM) with the North East Down (NED) reference frame.	120
5.5	3D Object Recognition Pipeline	121
5.6	Processing time required for Plane segmentation and Pipe detection.	124
5.7	Required processing time for Bayesian and semantic based recognition method.	125

CONTENTS

Abstract	1
Resum	3
Resumen	5
1 Introduction	7
1.1 Motivation and aim of research	8
1.2 Objectives	11
1.3 Context	12
1.4 Outline Of The Thesis	12
2 3D Object Recognition Based on Point Cloud in Underwater Environment with Global Descriptors: A Survey	15
3 Underwater Object Recognition Using Point-Features, Bayesian Estimation and Semantic Information	57
4 Semantic Mapping for Autonomous Subsea Intervention	85
5 Results and Discussion	115
5.1 Evaluation of global point feature descriptors for 3D object recognition in non-coloured point clouds	116
5.1.1 Results using Synthetic Data	116
5.1.2 Results Using Real Data	116
5.1.3 Discussion	118
5.2 3D object recognition method using CVFH, Bayesian estimation and semantic information	119
5.2.1 Discussion	120
5.3 Semantic mapping using SLAM and a 3D object recognition system	120
5.3.1 Summary	121
5.4 Computation Time analysis	123
6 Conclusions	127
6.1 Conclusions	128
6.2 Future Work	129

ABSTRACT

IN recent decades, the rapid development of intelligent vehicle and 3D scanning technologies has led to a growing interest in applications based on 3D point data processing, with many applications such as augmented reality or robot manipulation and obstacle avoidance, scene understanding, robot navigation, tracking and assistive technology among others, requiring an accurate solution for the 3D pose of the recognized objects. Thus object recognition is becoming an important topic in computer vision, where machine vision and robotics techniques are becoming key players.

In this thesis work, the main objective is to develop a semantic mapping method by integrating a 3D object recognition pipeline with a feature-based SLAM system, in order to assist autonomous underwater interventions in the near future.

To this end, the work proposed in this paper targets three axes. First, it aims to compare the performance of 3D global descriptors within the state of the art, focusing on those based on point clouds and targeted at real-time object recognition applications. For this purpose, we selected a set of test objects representative of Inspection, Maintenance and Repair (IMR) applications and whose shape is usually known *a priori*. Their CAD models were used to: 1) create a data base of synthetic object views used as *a priori* knowledge, and 2) simulate the point clouds that would be gathered during the scanning under realistic conditions, with added noise and varying resolution. Extensive experiments were performed with both virtual scans and real data collected with an AUV equipped with a fast laser scanner developed at our research centre.

The second goal of our work was to use a real-time laser scanner mounted on an AUV to detect, identify, and locate objects in the robot's environment, with the aim of allowing an intervention Autonomous Underwater Vehicle (I-AUV) to know what manipulation actions could be performed on each object. This goal was tackled by the design and development of a 3D object recognition method for uncolored point clouds (laser scans) using point features. The algorithm uses a database of partial views of the objects stored as point clouds. The recognition pipeline includes 5 stages: 1) Plane segmentation, 2) Pipe detection, 3) Semantic Object-segmentation, 4) Feature-based Object Recognition and 5) Bayesian estimation. To apply Bayesian estimation, it is necessary to track objects across scans. For this purpose, the Inter-distance Joint Compatibility Branch and Bound (IJCBB) data association algorithm was proposed based on the distances between objects. The performance of the method was tested using a dataset of the inspection of a pipe infrastructure made of PVC objects connected by pipes. The structure is representative of those commonly used by the offshore industry. Experimental results show that Bayesian estimation improves the recognition performance with respect to the case where only the

descriptor is used. The inclusion of semantic information about object pipe connectivity further improves recognition performance.

The final goal of the thesis, consists of integrating the 3D object recognition system with a feature-based SLAM system to implement a semantic map providing the robot with information about the location and the type of objects in its surroundings. The SLAM improved both the accuracy and reliability of pose estimates of the robot and the objects. This is especially important in challenging scenarios where significant changes in viewpoint and appearance arise.

RESUM

A les darreres dècades, el ràpid desenvolupament de vehicles intel·ligents i de les tecnologies d'escaneig 3D han contribuït a augmentar l'interès en les aplicacions basades en processament de núvols de punts 3D, amb aplicacions com la realitat augmentada, la manipulació robòtica, l'evasió d'obstacles, la comprensió d'escenes, la navegació robòtica, el seguiment d'objectes i la tecnologia d'assistència, etc., que requereixen una solució precisa de la posició 3D i l'orientació d'un objecte. Per tant, el reconeixement d'objectes s'està convertint en un tema, on la visió per computador i les tècniques robòtiques esdevenen protagonistes clau. En aquest treball de tesi, l'objectiu principal és desenvolupar un mètode per a la construcció de mapes semàntic mitjançant la integració d'una cadena de processament per al reconeixement d'objectes 3D, amb un sistema de SLAM basat en característiques, amb l'objectiu d'ajudar a les futures intervencions submarines. Per això, el treball proposat en aquesta tesi es divideix en tres eixos principals. El primer té com a objectiu comparar el rendiment de descriptors globals d'última generació, centrant-se en els basats en núvols de punts 3D i destinats a aplicacions de reconeixement d'objectes en temps real. Per a aquest objectiu, s'ha seleccionat un conjunt d'objectes de prova representatius d'aplicacions d'inspecció, manteniment i reparació (IMR), la forma dels quals es coneix a priori. Els seus models CAD s'han utilitzat per a: 1) crear una base de dades amb les vistes sintètiques dels objectes, i 2) simular els núvols de punts que adquiriria, en condicions realistes, un escàner làser incloent soroll sintètic i simulant diferents resolucions. S'han dut a terme experiments tant a partir d'escaneigs virtuals com de dades reals recopilades amb un AUV equipat amb un escàner làser de temps real desenvolupat al nostre centre de recerca. El segon objectiu del nostre treball va consistir en utilitzar aquest escàner làser, muntat a un AUV per detectar, reconèixer i localitzar objectes a l'entorn del robot, per tal de permetre, a un Vehicle Submarí Autònom d'Intervenció (I-AUV), saber què accions de manipulació podria fer amb cada objecte. Aquest objectiu es va abordar amb el disseny i el desenvolupament d'un mètode de reconeixement d'objectes 3D en núvols de punts incolors (escanejos làser) utilitzant descriptors dels punts 3D. L'algorisme utilitza una base de dades de vistes parcials dels objectes emmagatzemats en forma de núvols de punts. El procés de reconeixement consta de 5 passos: 1) Segmentació de plànols, 2) Detecció de canonades, 3) Segmentació semàntica d'objectes, 4) Reconeixement d'objectes a partir dels descriptors de punts 3D i 5) Estimació bayesiana. Per aplicar l'estimació bayesiana, cal ser capaços de fer un seguiment dels objectes en escans successius. Per fer-ho, s'ha proposat l'algorisme *Inter-distance Joint-Compatibility Branch and Bound* (IJCBB) d'associació de dades basada en les distàncies entre objectes dins del núvol de punts. El rendiment del mètode es va avaluar fent servir dades experimentals

relatives a la inspecció d'una infraestructura composta de canonades interconnectades per objectes de PVC. L'estructura és representativa de les comunament utilitzades per la indústria *offshore*. Els resultats experimentals mostren que l'estimació bayesiana millora el rendiment del reconeixement en comparació de l'ús únic del descriptor. La inclusió d'informació semàntica sobre la connectivitat d'objectes a canonades millora encara més el rendiment del reconeixement. L'objectiu final de la tesi va abordar la integració del sistema de reconeixement d'objectes 3D basat en descriptors amb un sistema de SLAM basat en característiques, per implementar un mapa semàntic que proporciona al robot informació sobre la ubicació i el tipus d'objectes a l'entorn. La utilització de tècniques de SLAM ha millorat la precisió i la fiabilitat de les estimacions de la postura del robot i els objectes. Això és especialment important en escenaris difícils on es produeixen canvis significatius de perspectiva i aparença.

RESUMEN

En las últimas décadas, el rápido desarrollo de vehículos inteligentes y tecnologías de escaneo 3D ha llevado a un creciente interés en aplicaciones basadas en procesamiento de nubes de puntos 3D, con muchas aplicaciones, como la realidad aumentada, la manipulación robótica, la evasión de obstáculos, la comprensión de escenas, la navegación robótica, el seguimiento de objetos y la tecnología de asistencia, etc., que requieren una solución precisa para que se reconozca la posición 3D y la orientación de un objeto. Por lo tanto, el reconocimiento de objetos se está convirtiendo en un tema, donde la visión por computador y las técnicas robóticas son protagonistas clave. En este trabajo de tesis, el objetivo principal es desarrollar un método de mapeo semántico mediante la integración de una cadena de procesado para el reconocimiento de objetos 3D con un sistema SLAM basado en características cuyo objetivo es ayudar a las intervenciones submarinas en un futuro próximo. Para ello, el trabajo propuesto en esta tesis se divide en tres ejes principales. El primero, tiene como objetivo comparar el rendimiento de descriptores globales de última generación, centrándose en aquellos basados en nubes de puntos 3D y destinados a aplicaciones de reconocimiento de objetos en tiempo real. Para este objetivo, se ha seleccionado un conjunto de objetos de prueba representativos de aplicaciones de inspección, mantenimiento y reparación (IMR), cuya forma se conoce generalmente a priori. Sus modelos CAD se han utilizado para: 1) crear una base de datos con las vistas intéticas de los objetos, y 2) simular las nubes de puntos que adquiriría, en condiciones realistas, un escáner láser incluyendo ruido sintético y simulando diferentes resoluciones. Se han llevado a cabo experimentos tanto a partir de escaneos virtuales como de datos reales recopilados con un AUV equipado con un escáner láser de tiempo real desarrollado en nuestro centro de investigación. El segundo objetivo de nuestro trabajo fue utilizar dicho escáner láser, montado en un AUV para detectar, reconocer y localizar objetos en el entorno del robot, con el fin de permitir, a un Vehículo Submarino Autónomo de Intervención (I-AUV), saber qué acciones de manipulación podría realizar con cada objeto. Este objetivo se abordó con el diseño y desarrollo de un método de reconocimiento de objetos 3D en nubes de puntos incoloras (escaneos láser) utilizando descriptores de los puntos 3D. El algoritmo utiliza una base de datos de vistas parciales de los objetos almacenados en forma de nubes de puntos. El proceso de reconocimiento consta de 5 pasos: 1) Segmentación de planos, 2) Detección de tuberías, 3) Segmentación de objetos semánticos, 4) Reconocimiento de objetos a partir de los descriptores de puntos 3D y 5) Estimación bayesiana. Para aplicar la estimación bayesiana, es necesario ser capaces de hacer un seguimiento de los objetos en escans sucesivos. Para ello, se ha propuesto el algoritmo *Inter-distance Joint-Compatibility Branch and Bound* (IJCBB) de asociación de datos basado en las distancias entre objetos dentro

del escan. El rendimiento del método se evaluó utilizando un datos experimentals relativos a la inspección de una infraestructura compuesta de tuberías interconectadas por objetos de PVC. La estructura es representativa de las comúnmente utilizadas por la industria *offshore*. Los resultados experimentales muestran que la estimación bayesiana mejora el rendimiento del reconocimiento en comparación con el uso único del descriptor. La inclusión de información semántica sobre la conectividad de objetos a tuberías mejora aún más el rendimiento del reconocimiento. El objetivo final de la tesis abordó la integración del sistema de reconocimiento de objetos 3D basado en descriptores con un sistema de SLAM basado en características para implementar un mapa semántico que proporciona al robot información sobre la ubicación y tipo de objetos en el entorno. La utilización de técnicas de SLAM ha mejorado la precisión y la fiabilidad de las estimaciones de la postura del robot y el objeto. Esto es especialmente importante en escenarios difíciles donde ocurren cambios significativos de perspectiva y apariencia.

1

INTRODUCTION

THIS chapter presents the motivation behind this Ph.D. thesis in Section 1.1, where the reader is introduced to the underwater object recognition problem. The main objectives of this work are presented in Section 1.2. Section 1.3 describes the context in which this work has been developed, and Section 1.4 concludes with a summary of the organization of this document.

1.1 Motivation and aim of research

Visual perception is the natural sensing modality for scene understanding. It can be used for the recognition of the objects surrounding the robot. Since the early days of computer engineering, the input information used for scene interpretation has evolved from images to more complete and representative data such as depth images and 3D point clouds. Recently, with the development of new sensing technologies, a significant change has taken place, allowing machines to interpret their environment in a more accurate and efficient way.

Many mobile robots incorporate state-of-the-art algorithms in scene understanding to perform various tasks, where the behavior of the autonomous agent depends on its contextual environment. Having the capability of detecting and recognising the surrounding objects, the robot can build semantic maps which can be used for long term robot localization, for instance. It also makes the robot aware of where the relevant objects are. On the other hand, the objects' semantics, in the sense of which class or classes it belongs to, defines what can be done with them. For instance, a mobile robot (fig. 1.1a) which discovers a cup on a table may decide to "Grasp" it. If it instead detects a calculator, it may decide to "Grasp" it or to "Push a Button". In our case, when an I-AUV detects a valve (fig. 1.1b), it will understand that the relevant action to apply is "Valve Turning". In case a connector object ("Hot Stab") is detected, then, the relevant actions are "Plug" or "Unplug". Moreover, knowing the object class enables task planning. For instance, if we have 2 objects (A and B) of the class "Pipe Stand" and one object (P) of the "Pipe" class, we may discover that P stands on A while B is free. Then we can plan, for instance, the pipe transportation from A to B. Therefore, object detection and recognition plays an important role in achieving robot autonomy.

Object Recognition is the problem of discovering the membership class of queried objects. This is achieved by using approaches and methods originally from 2D image processing which have been later extended by various researchers to deal with 2.5D images and 3D point clouds. They are applied to different parts of the information contained on the image or point cloud (descriptors, segments, semantic information, etc.). The whole process is implemented as a pipeline through a sequence of processing steps.

Recent research has focused on object recognition for indoor and outdoor applications, from air and land vehicles, using 3D point clouds or 2.5D data from range cameras or stereo rigs. In contrast, to our knowledge, there have been no attempts to perform object recognition from point clouds with underwater vehicles.

In terms of object recognition and segmentation, autonomous aerial and land robots have provided promising results. A diverse range of applications has been developed, from which we select, as motivation examples, the following:

- **Road and railway monitoring**

Che *et.al* [6] summarise data processing strategies for extraction, segmentation and object recognition and classification as well as the available benchmark datasets based on Mobile Laser Scanning (MLS) data. The object recognition and point cloud classification methods are reviewed and summarised in terms of both general ideas and technical details by further discussing the limitations and challenges of the existing methods. These authors also highlight the influence of scene types and summarise the publicly available benchmark data.

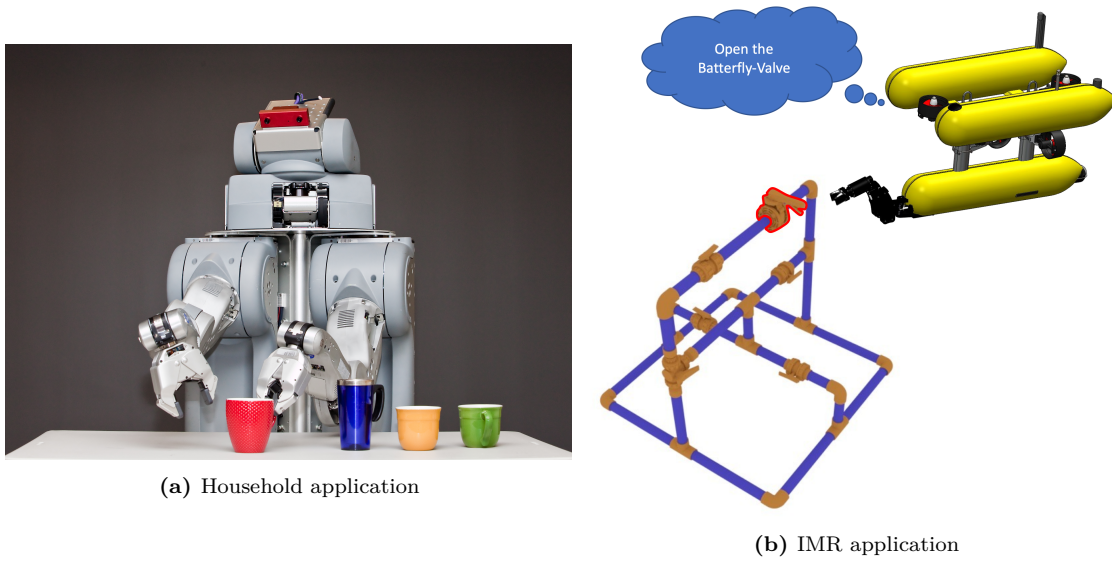


Figure 1.1: The objective of the thesis is to develop methods to build semantic maps allowing robots to discover objects and the manipulation tasks relevant for them.

In [7] the authors present a literature review describing various methods and applications for monitoring terrestrial transportation networks using data collected by Mobile Mapping Systems equipped with Light Detection And Ranging (LiDAR) sensors. Various commercial LiDAR based terrestrial systems are described and compared to provide a broad scope of available sensors and tools for remote monitoring of infrastructures based on terrestrial systems. The paper includes a summary of the main applications of LiDAR data in relation to the railway network which have been elaborated, including road surface monitoring, rails, power lines, signalization, and inspection.

In another study, an autonomous on-site 3D spatial data acquisition and sensing method for mobile robot with a hybrid LiDAR system was presented in [8]. The proposed Simultaneous Localization and Mapping (SLAM) based navigation and object recognition method was implemented and tested by a specially designed mobile robot platform, Ground Robot for Mapping Infrastructure (GRoMI), which uses multiple laser scanners and cameras to sense and create a 3D environment map. Overall, the three-dimensional color map point cloud generated by GRoMI on the construction site is of sufficient quality to be used in many construction management applications, such as monitoring construction schedules, identifying safety hazards, and detecting defects.

- **Surface models and landslide scarp identification using point clouds**

In [9], the authors presented a method based on using unmanned aerial vehicle (UAV) photo-grammetric point clouds to generate surface models of structural elements of historical bridges in China. The proposed methods deal with point cloud segmentation and recognition. The segmentation was based on super voxel structure and global graph optimization. For the recognition of structural elements from the segments, the authors introduced a rule-based classification method that used

the salience of segments to determine the type of object associated with it (label of segments), where the salience of the segment was defined by its geometric properties.

Al-Rawabdeh *et al.* [10] presented a practical approach for landslide scarp identification using point clouds. The landslide scarps were identified using a dense 3D point cloud with topographic information generated from high-resolution images acquired by low-cost UAVs. Three methods were used to extract scarps using the morphometric features of the derived point cloud: PCA-based eigenvalues in local neighborhoods, slope variability within local neighborhoods, and evaluation of the surface roughness index for the derived point cloud in local neighborhoods. Their proposed method enables the derivation of accurate information for landslide characterization while mitigating the inherent risk in surveying hazardous landslide-prone areas and reducing the costs incurred.

- **Object recognition from thermal images and point clouds**

In [11], the authors proposed a hybrid recognition system of thermally mapped point clouds implemented using the in-house developed GROMI. The system used integrates four 2D scanners, a LiDAR and an infrared camera. The authors computed a complete thermally mapped point cloud and segmented it into point clusters representing individual objects. This was done by thresholding with a lower bound on the temperature value of interest for the heat-radiating objects. This separates the point cloud into foreground (high temperature) and background (low temperature) points and creates object clusters by applying a region-growing technique. For each segment, a Principal Component Analysis was used to identify the eigenvalues and eigenvectors corresponding to the main directions of the object geometry. Each object cluster is parameterized by a four-dimensional feature vector that includes the length, width, height, and altitude of its bounding box. These features were fed into a decision tree classifier to assign a class label to each object. Another investigation based on thermally mapped point clouds was presented in [12]. These authors proposed a method to detect building elements. The region containing the building elements of interest was identified and extracted from the point cloud. Segmentation of maximum and minimum thermal intensity regions was performed, based on absolute and relative temperature thresholds. Each of the identified point cloud clusters could thus be assigned to a building element and located based on the cluster center point.

- **Object recognition based on semantic knowledge for Robotic Housework**

On the other hand, robots are equipped with manipulators dedicated to specific tasks, such as recognizing objects and manipulating them. In this context, there have been significant advances, for example, in the use of robots in kitchen environments, cited in [13, 14, 15], where the robot was able to identify everyday objects such as a bowl, plate or glass, using RGB-D cameras (such as the Kinect sensor) to locate and grasp them in an automated manner. NVIDIA recently opened a new artificial intelligence robotics research lab focused on teaching a robotic arm to navigate an Ikea kitchen.

Günther *et al.*[11] presented an anchoring system that continuously integrates new observations from a 3D object recognition algorithm into a probabilistic world model. To approximate the similarity between two objects, the authors trained an Support

Vector Machine (SVM) on samples of object pairs manually labeled as "same object"/"different object". To evaluate the performance of their framework, the authors collected a dataset of 15 scenes with table-top settings that produced positive results in both data association and object recognition.

The works summarized above highlight the need for object recognition for wide variety of applications of robot monitoring and intervention. They also illustrate how such application scenarios have very different challenges and requirements from the perception and recognition points-of-view. In particular, the work of Rusu *et al.* [13, 14] and Blodow *et al.* [15] provide useful insights of the different stages of recognition, and influenced the decision on which approaches to use in this thesis.

Having described the main motivations behind our work, the goal of this thesis is presented in the next section.

1.2 Objectives

The goal of this thesis is to propose a 3D object recognition method, operating on colour-less point clouds gathered with a laser scanner, to recognize industrial infrastructures made of pipes and objects relevant for IMR operations, using shape information known *a priori*. This goal can be broken down into the following sub-goals:

1. **To propose a 3D object recognition system using point-features and a data-base containing the 3D models of the objects to be recognized**

This includes:

- (a) To propose an object recognition pipeline based on the matching of the current object view, described with point-features, against the object views stored in the database corresponding to a complete set of views of all the *a priori* known objects.
- (b) To compare the performance of state-of-the-art global point-feature descriptors. Global descriptors are chosen over local ones since they better represent the whole object in a compact way, exploiting the relationships between points on different parts of the object. The comparison will be carried out using synthetic and real data. The confusion matrices for each descriptor will be computed to conclude which is the best descriptor for our problem.

2. **To improve the performance of the recognition using Bayesian inference, semantic segmentation and semantic object information**

Most global 3D descriptor methods assume that the point clouds are de-noised, complete, and consistent. This is not always the case. Partial occlusion, un-modeled deformations due to the robot motion during the scan acquisition or just the similarity between certain objects views may lead to the failure of recognizing an object in a single observation. Nevertheless, it is our hypothesis that we can achieve a good object recognition rate by considering successive observations of the same object across the scans. To achieve this, the following sub-goals were considered:

- (a) To propose a semantic segmentation method, based on geometric constraints together with rules for decomposing connected pipe structures. The aim of this

method is to separate and distinguish, at the point-cloud level, the points that belong to objects and those that belong to connecting pipes.

- (b) To propose a Bayesian model to iteratively estimate, for each detected object and every object class, its membership probability, selecting, the most probable as the recognized class. This method can be refined using semantic information about the object connectivity (number of connected pipes). Therefore, given an object observation, the Bayesian estimation will only take into account those classes which are compatible with the actual object connectivity.
- (c) To feed the Bayesian estimation model, observations of the same object across multiple scans are required. Therefore it is necessary to propose a multi-object tracking method to follow the objects across the scans. The method must be robust against sudden position jumps due to loss of DVL bottom-lock which may happen when moving close to 3D structures.

3. To implement a semantic map integrating the object recognition system with an state-of-the-art SLAM

Having the capability to recognize pipes and objects in place, our final goal is to set up a consistent semantic map exploiting the state-of-the-art SLAM techniques in order to provide a long-term localization of the robot and its surrounding objects. This makes the robot aware of the objects around it, but also simplifies the object tracking across scans, solving the position jumps, and helping with the semantic Bayesian estimation.

The survey of the state of the art and the validation of the framework objectives have been an on-going effort throughout the development of the thesis, and as such they are reflected in all the publications of this compendium. Every publication contains a state-of-the-art review, and also experimental evaluation of the proposed algorithms.

1.3 Context

The work presented in this thesis has been supported by the *Personal Investigador Pre-doctoral en Formacion(FPI)* 2015 grant from the *the Spanish Government* and has been developed at the Centre de Investigació en Robòtica Submarina (CIRS) research group of the Universitat de Girona (UdG), which is part of the VICOROB research institute. The group started researching in underwater vision and robotics in 1992 and it is currently composed of pre-doctoral researchers, engineers, technicians, post-doctoral fellows and permanent staff. The group is a leading team in the research and development of Autonomous Underwater Vehicles (AUVs) for accurate seafloor mapping and light intervention. It has participated in several European-funded and National-funded projects (of both basic and applied research) and it has also been involved in technology transfer projects and contracts with companies and institutions worldwide.

1.4 Outline Of The Thesis

The material presented in this thesis is organized as follows:

Chapter 2 presents, through the publication “*3D Object Recognition Based on Point Clouds in Underwater Environment with Global Descriptors: A Survey*”, the problem of

object recognition from colorless 3D point clouds in underwater environments. It presents a performance comparison of state-of-the-art global descriptors, and experiments conducted from both virtual scans and from real data collected with an AUV equipped with a fast laser sensor.

Chapter 3 through the publication “*Underwater Object Recognition Using Point-Features, Bayesian Estimation and Semantic Information*”, presents an approach for application scenarios such as IMR of underwater industrial structures consisting of pipes and connecting objects. It discusses in detail the Bayesian and semantic approaches used in object recognition and highlights some limitations of descriptor-based approaches, especially those related to object variations due to environmental noise. In addition, the chapter describes the use of semantic segmentation object tracking to limit the drift of the vehicle during scanning and data acquisition during the experiment.

Chapter 4 through the publication “*Semantic Mapping for Autonomous Subsea Intervention*”, presents a semantic map approach to tackle the problem described in the previous chapter. The approach is based on the integration of feature based SLAM and 3D object recognition using a data base of *a priori* known objects. A central concept is the bi-directional exchange of information between the recognition module and the SLAM module, in order to improve the the recognition using Bayesian inference and object semantics.

The final part of this thesis consists of two chapters. **Chapter 5** provides a summary of results obtained by the methods described in the previous three chapters. **Chapter 6** summarizes the contributions from this thesis, and is followed by concluding remarks and directions for future research.

2

3D OBJECT RECOGNITION BASED ON POINT CLOUD IN UNDERWATER ENVIRONMENT WITH GLOBAL DESCRIPTORS: A SURVEY

This chapter therefore examines the state of the art of 3D global descriptors, particularly those applied directly to colourless point clouds, and works towards a clearer understanding and summary of the descriptors that will guide this thesis. As a result not only the advantages and benefits, but also the pitfalls to avoid in real experiments, are made clear. The goal of this study is to overview methods for the recognition of 3D objects based on a global descriptor, and using as input no coloured 3D point clouds collected using a laser scanner. This paper contributes to the state-of-the-art as being the first work on the comparison and performance-evaluation of methods for underwater object recognition. It is also the first effort using a comparison of methods for data acquired with a free-floating underwater platform. All proposed work was described in detail and published in the following journal paper:

Title: 3D Object Recognition Based on Point Clouds in Underwater Environment with Global Descriptors: A Survey

Authors: **K. Himri**, P. Ridao, and N. Gracias

Journal: Sensors

Volume: 19, Number: 20, Pages: 4451, Published: 2019

Quality index: JCR2019 Instruments & Instrumentation IF 3.275, Q1 (15/64)



Article

3D Object Recognition Based on Point Clouds in Underwater Environment with Global Descriptors: A Survey

Khadidja Himri * , Pere Ridao and Nuno Gracias

Underwater Robotics Research Center (CIRS), Computer Vision and Robotics Institute (VICOROB), University of Girona, Parc Científic i Tecnològic UdG C/Pic de Peguera 13, 17003 Girona, Spain; pere@eia.udg.edu (P.R.); ngracias@silver.udg.edu (N.G.)

* Correspondence: khadidja.himri@udg.edu (K.H.); Tel.: +34-972-418-905

Received: 16 August 2019; Accepted: 4 October 2019; Published: date



Abstract: This paper addresses the problem of object recognition from colorless 3D point clouds in underwater environments. It presents a performance comparison of state-of-the-art global descriptors, which are readily available as open source code. The studied methods are intended to assist Autonomous Underwater Vehicles (AUVs) in performing autonomous interventions in underwater Inspection, Maintenance and Repair (IMR) applications. A set of test objects were chosen as being representative of IMR applications whose shape is typically known a priori. As such, CAD models were used to create virtual views of the objects under realistic conditions of added noise and varying resolution. Extensive experiments were conducted from both virtual scans and from real data collected with an AUV equipped with a fast laser sensor developed in our research centre. The underwater testing was conducted from a moving platform, which can create deformations in the perceived shape of the objects. These effects are considerably more difficult to correct than in above-water counterparts, and therefore may affect the performance of the descriptor. Among other conclusions, the testing we conducted illustrated the importance of matching the resolution of the database scans and test scans, as this significantly impacted the performance of all descriptors except one. This paper contributes to the state-of-the-art as being the first work on the comparison and performance evaluation of methods for underwater object recognition. It is also the first effort using comparison of methods for data acquired with a free floating underwater platform.

Keywords: 3D object recognition; point clouds; global descriptors; laser scanner; underwater environment; pipeline detection; inspection; maintenance and repair; AUV; autonomous manipulation

1. Introduction

The last few years have seen a multitude of object detection and recognition approaches appear in the literature. This development effort has been driven by the growing need to have autonomous systems that can interact with poorly structured, poorly organized and dynamic real-world situations.

Significant progress has been made in object recognition for mobile robots over the last decade. An application scenario that achieved a promising degree of performance is the use of robots in kitchen environments [1–3]. Robots are able to identify everyday objects such as bowls, plates and cups using color and depth cameras, in order to locate and grasp them in an automated way. More recently, a new artificial intelligence robotics research lab was opened by NVIDIA where the main focus is to teach a robotic arm to navigate an IKEA kitchen [4] and recognize different utensils. Stereo vision systems were used for identifying and grasping objects [5–7], where the robots aimed to accurately localize parts of the object from images and determine the correct grasping points.

The application of recognition in indoor environments using mobile robots has extended to a wide range of other applications. These include domestic assistance to elderly people or those with a certain degree of disability [8–10], agricultural [11,12] and industrial applications [13,14], and in advanced driver-assisted systems [15–18].

Autonomous driving and indoor service robotics are two main application scenarios which are partially responsible for the surge in work on object detection and recognition. Both scenarios imply robots that operate alongside humans, and whose actions can be potentially dangerous to human life. In this sense, there has been a drive towards increasing both the robustness and speed of the recognition process. For land robotics the increase of robustness can be achieved in part by the use of different complementary sensory modalities, such as laser scanners, Light Detection and Ranging (LIDAR), color cameras, and depth sensors based on texture projection. However, in other application scenarios such as in underwater robotics, the use of complementary sensors may be severely restricted or impossible, due to payload limitations and environmental conditions that are adverse to these types of sensors.

The underwater environment is one of the most challenging in terms of sensing in general and in terms of object perception in particular. The rapid attenuation and scattering of light and other electromagnetic waves implies that object detection and recognition when using optical sensing can only be conducted at very short distances from the objects, in the order of just a few meters. Acoustic propagation allows much longer ranges in terms of sensing distance, but the object representations obtained are far too noisy and coarse in resolution to allow precise object identification and localization for autonomous object grasping. Comparatively fewer applications of object recognition were reported underwater than in the above-water counterpart. These include pipeline identification and inspections based on optical images in seabed survey operations [19], cable identification and pipeline tracking based on acoustic images [20], and recognition of different geometric shapes such as cylinders and cubes [21] using acoustic imaging cameras.

In this paper, we are interested in exploring methods suitable for object recognition underwater, with the future aim of grasping and manipulating such objects. The long-term potential application scenarios are wide ranging, and include:

- Inspection, maintenance and repairing of offshore structures, which are frequently carried out by the oil and gas industries [22].
- Safe and secure exploration of inaccessible, polluting and dangerous maritime resources, including the detection of man-made objects [23,24].
- Subsea collision avoidance, by using systems to identify and locate a different obstacles [25], for example in the early assessment of accident sites.
- Detection and identification of marine wildlife, with the aim of studying their physical environment [26].

1.1. Objectives and Contributions

This paper addresses the problem of 3D object recognition in underwater environments. The main goal of this work is to compare the performance of state-of-the-art global methods for the recognition of different man-made objects. It focuses on the use of global descriptors available in the open source library “Point Cloud Library (PCL)” [27].

As elaborated in Section 2, global descriptors have the advantage of better representing the whole object in a compact way, by exploiting relationships between points on different parts of the object. The main drawback of these methods lies in their inability to deal with severe occlusions, and with cluttered data comprising multiple objects. Although local descriptors are more adequate in realistic and clutter scenarios, they have a much higher computation cost. This makes them less suitable for real-time data processing on vehicles with limited computational resources, such as the case of AUVs. Global methods, on the contrary, are more adequate to real-time operation and, for this reason, are the focus of this paper.

The chosen test objects are all related to underwater piping and tubing, and include different types of valves and sections, as detailed in Section 5.1. These objects were selected because they are representative of the building blocks of existing underwater structures where autonomous manipulation is expected to have a high impact in the near future. Given that our primary concern is the recognition of objects whose shape is known a priori, we used Computer Aided Design (CAD) models of the objects in our testing. The CAD models provide a noise-free description of the shape, from which virtual views of the objects can be produced under realistic conditions of added noise and varying resolution.

The results from experiments with real data, collected by an AUV equipped with a fast laser sensor developed in our research centre [28] are used to illustrate how each descriptor works and performs, under realistic subsea conditions. These conditions include, for example, the acquisition of data by a moving platform, which can create deformations in the perceived shape of the objects. Contrarily to aerial laser scanning applications, where the longer imaging range and extra sensing devices (such as GPS) can assist in correcting the effects of the moving platform, in underwater environments these effects are considerably more difficult to correct using the typically available sensors, such as IMUs and DVLs.

This paper contributes to the state-of-the-art as being the first work on the comparison and performance evaluation of methods for underwater object recognition. It is also the first effort using comparison of methods for data acquired with a free floating underwater platform.

Regarding the specific application of pipe-related object recognition, few publications exist in the literature using 3D point clouds as the main (or only) source of information. To the best of our knowledge these are all above-water application scenarios, using high resolution LiDAR. Examples include the work of Huang et al. [13] and Pang et al. [29], where a complex pipeline structure is divided and modelled as a set of interconnecting parts, using a SVM-based approach and a single local feature descriptor (Fast Point Feature Histogram, mentioned in Section 2). Another noteworthy application to pipeline classification is the work of Kumar et al. [30] where an aerial vehicle equipped with a low-cost LiDAR is able to map and identify pipes of different sizes and radii. The pipe identification is based on the analysis of smaller ROIs where information about curvature is gathered. Since the focus of that work is on the real-time mapping, there is no attempt to detect and classify other objects apart from pipes.

1.2. Structure of the Paper

The paper is structured as follows. Section 2 presents an overview of object recognition approaches with selected examples of the most relevant previous work. Section 3 provides a deeper description of the class of object recognition methods (Global methods) that are used in this paper, including a description of each of the methods tested. Section 4 explains the algorithmic pipeline used for the processing and testing. Section 5 details the experimental setup. Section 6 provides comparative results obtained first in simulated conditions, and then in a real experiment, using an AUV equipped with an underwater scanner developed at our lab [28]. Finally Section 9 draws the main conclusions and provides future work directions.

2. Overview of Object Recognition from Point Clouds

This section presents an overview of the most relevant approaches in the literature related to 3D object recognition from point clouds.

Object recognition approaches can be divided into three broad categories: global, local and hybrid. Global methods aim at representing the whole object as a single vector of values. A definite advantage of these methods is that they are suited for real-time data processing, due to their low computation cost. However, they present the disadvantage of being disturbed by cluttered scenes. To overcome this sensibility to the presence of multiple objects, global methods require a preliminary step of object segmentation, in order to isolate individual objects previous to the recognition. Conversely, the local

methods are generally more specific for a local area and computed from salient points, which make them more robust to clutter and occlusion. These methods seek to describe the object as a collection of small salient areas of the object, whose geometric arrangement is also taken into account. However, these methods suffer from larger computation cost due to the large number of points-descriptors per object. The last category is hybrid methods, which aim at incorporating the strengths of both global and local descriptors.

2.1. Local Recognition Pipelines

The use of local descriptors for 3D object recognition was reported in several review papers.

Alexandere et al. [31] assessed the different descriptors implemented in the PCL, considering only the methods that could be applied directly on a point cloud. As such, some methods were excluded, namely the Spin Image Descriptor [32] which is based on a mesh representation, the Global Fast Point Feature Histogram (GFPPFH) [33] which assumes labelling of the points, and the Camera Roll Histogram [34] given that they were mainly interested in evaluating the recognition process without estimating the pose of the objects. The tests were carried out based on an RGB-D object dataset [35]. The authors singled out the Colored Signature of Histogram of Orientation (CSHOT) descriptor [36] given that it offered a good balance between recognition performance and time complexity.

A comprehensive survey paper by Guo et al. [37] reported and reviewed the most important local descriptors applied on mesh surface or point clouds. These authors considered the existing local descriptors published between 2001 and 2015. The local descriptors presented in [37,38] were tested on four relevant benchmark datasets: Bologna [39], the UWA 3D Object Recognition (U3OR) [40], the Queen's [41] and the Ca' Foscari Venezia Dataset [42].

For details of 3D local feature descriptors, we refer the reader to [37,38]. In the following, we provide a brief review of several local and especially global descriptors based on point clouds that relate to our work.

After 2015, the literature continues and includes other local descriptors. For instance, the Equivalent Circumference Surface Angle Descriptor (ECSAD) [43] is a 3D shape feature designed for detecting the 3D shape edges, and is best suited when the objects have clear prominent edges. Another local descriptor based on contour information is the Rotational Contour Signatures (RCS), presented in [44]. The RCS computes several signatures from 2D contour information, obtained from 3D-to-2D projection of the local surface. The key contribution of these authors consisted in building a geometry encoding, where the local surface is rotated toward a predefined local reference frame, thus enabling the gathering of multi-view contour information. The RCS descriptor was compared against five state-of-the-art descriptors, including Spin Image (SI) [32], SNAPSHOTS [45], Fast Point Feature Histograms (FPFH) [46], SHOT [47] and Rotational Projection Statistics (RoPS) [48] and using the two standard databases: the Bologna [39] and the UWA Object Recognition (UWAOR) datasets [40].

Recently, using Mobile Laser Scanning (MLS) point cloud data, Zhenwei et al. [49] classified pole-like objects from unstructured MLS point cloud data. The authors used the random sample consensus (RANSAC) [50] and principal component analysis (PCA) [51] to detect the vertical cylinder model and principle direction of the point set. Along the same line, a good description of the state-of-the-art for mobile laser scanning systems is presented in [52]. The authors cite several methods based on the point cloud data used for gathering information on road and transport, with emphasis on relevant methods for feature extraction, segmentation and object detection.

2.2. Global Recognition Pipelines

Global descriptors describe the characteristics of the entire object and they are often used as a coarse representation suitable for real-time applications. Most of the existing global descriptors evolved from local feature representations. An example of this is the Viewpoint Feature Histogram (VFH) [53], which is an extension of the local descriptor Fast Point Feature Histograms (FPFH) [46], that encodes information on the whole object and viewpoint. Rusu et al. [53] validated the efficiency of the VFH

applied on 60 IKEA kitchenware objects collected using stereo cameras. The method was compared against the state-of-the-art Spin Image (SI) [32] with favorable results for VFH. This descriptor is designed to accomplish both recognition and pose identification.

The Global Radius-based Surface Descriptor (GRSD) [54] is a global version of the local Radius-based Surface Descriptor (RSD) [55] suitable to mobile manipulation applications. To evaluate their approach, numerous experiments were performed in an indoor environment. These experiments rely on geometric and appearance-based data, where everyday objects were used. The recognition approach used both images collected with a stereo camera and 3D depth data from a range scanner. Marton et al. [54] proposed an approach aimed at combining the Speeded-up Robust Features (SURF) [56] 2D descriptor with the GRSD 3D descriptor. The authors defined a hierarchical classification system, where the GRSD is used as a first step to reduce the number of choice of objects to those of similar shape, followed by the use of the SURF descriptor to accurately identify the object in a particular instance. Gunji et al. [57] proposed the Bag-of-Features (BoF)-based object recognition pipeline, which is suited to processing large-scale scene point clouds acquired in indoor environments with a laser rangefinder. This method follows a two-step approach. The first step is a preprocessing of data which includes an unsupervised training of codebook (collection of vector-quantized features) using K-means, where the codebook consists of centroids of clusters of FPFH, a local descriptor. The second step is the recognition of the target model, which is implemented by computing the BoF inside a sliding window. The authors performed trials using real data, where they showed that the proposed approach based on BoF has better performance in terms of precision and recall compared to the 3D Hough voting [58] with SHOT descriptor [47] method. Similar work based on Bag-Of-Feature was proposed in [59].

Jain et al. [60] introduced another global descriptor derived from a local descriptor. The authors presented a manipulation framework for grasping objects based on Global Principal Curvature Shape Descriptor (GPCSD). The GPCSD aimed to categorize object-clusters that are geometrically equivalent into similar primitive shape categories. The GPCSD is based on the local Principal Curvatures (PC) values. The computation of the descriptor is similar to GRSD presented in [54]. However, rather than labelling the voxel using Radius-Based Surface Descriptor (RSD), the authors applied Principal Curvatures (PC). The performance of GPCSD is compared against Global Radius-Based Surface Descriptor (GRSD), using the Washington RGB-D dataset and real-time data from a Microsoft Kinect. The results showed that both descriptors performed well, although GRSD was found to be more robust to distance variations.

2.3. Hybrid Recognition Pipelines

From the results of the studies above, it is natural to expect a better performance by merging global and local information. The following reports present different hybrid recognition pipelines that combine both approaches.

In [61], Aldoma et al. presented a hybrid pipeline allowing the processing of data from different modalities. The method is based on three different descriptors: the SIFT 2D local descriptor, the 3D global descriptor OUR-CVFH descriptor [62], that exploits the color, shape, and object size information, and the SHOT [47] descriptor, a 3D local descriptor. The two local (2D and 3D) and the 3D global descriptors were combined using an optimization-based hypothesis-verification method, which aimed at validating a subset of hypotheses belonging to recognition hypotheses .

Alhamzi et al. [63] used state-of-the-art 3D descriptors to recognize the objects and estimate their pose, using the PCL Library. The authors selected the Viewpoint Feature Histogram (VFH) [53], a global descriptor, to recognize the objects of interest, whereas the Fast Point Feature Histogram (FPFH) [46], a local descriptor, was applied to estimate the position of the object. The performance of VFH was compared to the state-of-the-art descriptors, namely the Ensemble of Shape Functions (ESF) [64] and the Clustered Viewpoint Feature Histogram (CVFH) [34]. Then the authors integrate the result of the VFH descriptor, with five various types of local descriptors: SHOT [47], CSHOT [36],

PFH [65] PFHRGB [27] and FPFH [46]. Alhamzi et al. concluding that the couple VFH and FPFH achieved the best result. The performance of the hybrid method-based VFH and FPFH was validated using the Willow Garage dataset [66].

Sels et al. [67] presented a new fully automated Laser Doppler Vibrometer (LDV) measurement technique. Their measurement technique was remarkable in using data from a 3D Time-of-Flight camera jointly with a CAD file of the test object to automatically obtain measurements at predefined locations. The authors adopted the same pipeline presented in [63], where the global VFH descriptor was used for recognition, and the local FPFH descriptor to estimate the pose of the object.

3. Global Descriptors

As mentioned before, the central idea behind the methods for object recognition from 3D points is that an object can be characterized by a set of combined features, either local or global. This section presents a more detailed summary of the class of methods that are used in this paper: global descriptors. All the approaches that are tested and compared in the results section are here described.

The global features describe and encode the shape or geometry information of the object in a very compact way, allowing a low computational effort. The local features represent the objects by encoding subsets of neighbouring points around each salient point, which implies a much larger dimension of the feature space. However, the local descriptors have the advantage of dealing with high object cluttering and occlusions.

The data used in the underwater experiments of this paper was collected using a laser scanner developed in-house, that generates point cloud data without color information. Acquiring reliable color information underwater is quite a challenging task, due to the absorption and attenuation which are strongly dependant on distance and on the light wavelength. As such, from the robustness point of view, it is important to develop and use methods that do not rely on color. A set of descriptors were therefore selected which do not require color information and are available in the Point-Cloud Library. The only exception is the Global Orthographic Object Descriptor (GOOD) that is not integrated in the current version 1.8 of PCL [27].

In this study, the evaluation of a set of global descriptors is performed taking into account their performance, whether they retain their descriptiveness under flexible transformations based on how the object was scanned, under variations in the density of point clouds, and under different levels of noise. These descriptors are represented by a histogram, whose the size depends on the descriptors themselves.

The recent literature shows a gradually increasing interest in using methods available in the Point Cloud Library (PCL library). The methods evaluated and compared in our study were considered to be the most relevant in the literature, and are shown in Table 1, and include the Viewpoint Feature Histogram (VFH) [53], the Clustered Viewpoint Feature Histogram (CVFH) [34], the Oriented, Unique and Repeatable CVFH (OUR-CVFH) [62], the Global Orthographic Object Descriptor (GOOD) [68], the Ensemble of Shape Functions (ESF) [64], the Global Fast Point Feature Histogram (GFPFH) [33] and the Global Radius-based Surface Descriptors (GRSD) [54]. The list is ordered by chronological descending order of the methods they are based on.

Table 1. Summarized characteristics of the seven descriptors used in this paper. The “based on” column indicates if the descriptor evolved directly from another approach. The “use of normals” indicates whether the method uses surface normals for computing the descriptor, while the last column indicates the length of the descriptor vector.

Descriptor	Main Characteristics		
	Based on	Use of Normals	Descriptor Size
Global Orthographic Object Descriptor (GOOD)-2016—[68]	-	No	75
The Ensemble of shape functions (ESF)-2011—[64]	Shape function [69]	No	640
Global Radius-based Surface Descriptors (GRSD)-2010—[54]	RSD [55]	Yes	21
Viewpoint Feature Histogram (VFH)-2010—[53]	Fast Point Feature Histogram (FPFH) [46]	Yes	308
Global Fast Point Feature Histogram (GFPFH)-2009—[33]	Fast Point Feature Histogram (FPFH) [46]	Yes	16
Clustered Viewpoint Feature Histogram (CVFH)-2011—[34]	VFH [53]	Yes	308
Oriented, Unique and Repeatable C VFH (OUR-CVFH)-2012—[62]	CVFH [34]	Yes	308

In the following subsections, the 3D global descriptors of the study are briefly introduced.

3.1. Global Orthographic Object Descriptor (GOOD)

The Global Orthographic Object Descriptor (GOOD) [68] aims at providing reliable information in real time. To boost the robustness, a unique and repeatable object reference frame was applied. When computing the local reference frame, a sign ambiguity arises, which is solved with a proposed method based on eigenvalues and Principal Component Analysis (PCA). Using this reference frame, three principal orthographic projections are created (XoZ , XoY , and YoZ). Each orthographic projection is partitioned into bins, where the number of points falling into each bin is counted. The authors performed several tests, changing the number of bins, to find an adequate bin size that achieves best performance. These bins were presented as distribution matrices. The descriptor is finally obtained by concatenating these distribution matrices, where the sequence of projection was determined based on the highest entropy and variance of the projections. The size of GOOD histogram equal 75 floats, 25 per each one of the matrix of distribution of the three projections.

The crucial advantage of GOOD, is the fact that it is represented by three orthographical projections, which make it rich in terms of information suited for manipulation tasks. As illustrated in Figure 1, it is essential to know the true dimensions of the object in order to adjust the gripper, and this information can be obtained from the dimension on orthographic projection grid.

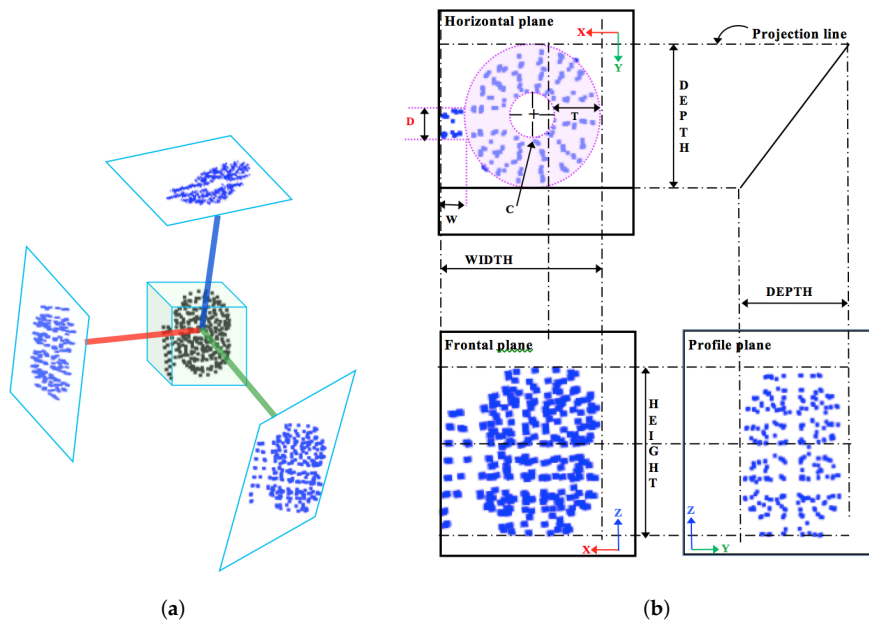


Figure 1. Example of how the three orthogonal projections are built: (a) Local reference frame and projections; (b) Projections in multi-view layout. GOOD can also be used for object manipulation. In the plan view of the object, the symbols C, W, D and T represent how the projection can be further processed where the features for the manipulation task can be extracted, namely inner radius (C), thickness (T), handle length (W) and handle thickness (D).

3.2. The Ensemble of Shape Functions (ESF)

The Ensemble of shape functions (ESF) was introduced by Osada et al. [69]. The authors suggested a way to characterize any 3D polygonal model, using a geometric shape function based on five measurements. Later on, Wohlkinger et al. [64] used the same principle, but reduced the number of measurements from five to three. The ESF descriptor combines a set of ten 64-bin-sized histograms of shape functions, describing geometric properties of the point cloud. The descriptor uses a voxel grid to approximate a real surface. Then, for each point in the cloud, three points are chosen randomly. These points are used to compute the three shape functions, as illustrated in Figure 2:

- The distance D2: This is the distance calculated between two points, then classified into one of three categories based on whether the connection line falls in the surface, off the surface, or is mixed (with one part in and the other off the surface). To characterize the distribution of the voxels along the line, the authors added the ratio of line distance (D2 ratio). This ratio is equal to zero if the line falls off the surface, equal to one if inside, and equal to the numbers of the voxels, along with the connection, if the line is mixed.
- The angle A3: This is the angle computed between two lines, then the line opposite to this angle is classified in one of the three categories (in, out, or mixed).
- The area D3: This is the square root of the area formed by the three points, based on the Heron Formula (1).

$$D3 = \sqrt{\sqrt{s(s-a)(s-b)(s-c)}},$$

$$s = \frac{a+b+c}{2},$$
(1)

where s represents the semi-perimeter and a , b and c the side lengths of the triangle. The area is classified similarly as in $D2$.

The total length of the ESF descriptor is 640 bins which are divided into ten sections: three for the angle component ($A3$), three for the area component ($D3$), three for the distance component ($D2$) and one for the ratio of distance component ($D2$ ratio).

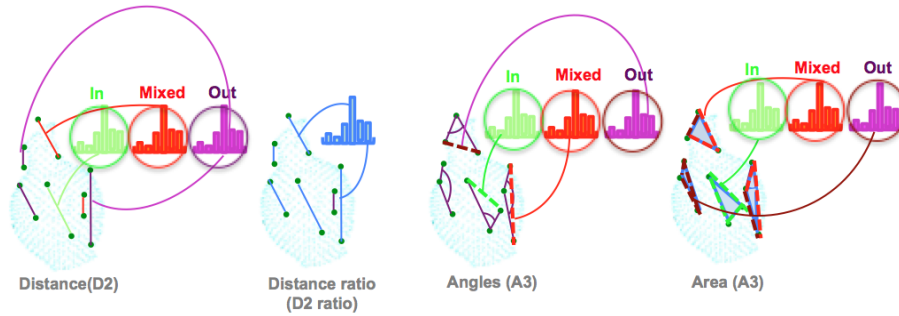


Figure 2. The Ensemble of Shape Function (ESF). Illustration of how shape functions are computed for a point cloud of an amphora. Left: point distance distributions. The green histogram represents points in, the red histogram points out, and the purple represents mixed points; Middle left: Distance Ratio; Middle right: angle distributions; Right: the area covered by triplets of sampled points.

3.3. Global Radius-Based Surface Descriptors (GRSD)

The Global Radius-based Surface Descriptor (GRSD) was introduced by Marton et al. [54]. The descriptor is considered as the global version of the Radius-based Surface Descriptor (RSD) [54] which is a local descriptor.

To better understand the GRSD, we start by describing how the RSD is computed. The RSD descriptor encodes the radial relationship between every pair of points lying in a surface (defined by a radius r). For each query point p and its neighbour points p_i , the distance and angle α formed between the two normals of the pair p and p_i are computed. We could draw an imaginary sphere around the pairs of points, where the point p belongs to each of these spheres. From all the possible cases, only points with the largest and smallest spheres are chosen, and their radii are selected to build a descriptor of the point $radius = [r_{min}, r_{max}]$.

$$d = \sqrt{2r} \sqrt{1 - \cos(\alpha)},$$

$$r \approx \frac{d}{\alpha},$$
(2)

To compute the GRSD descriptor, first, the input point cloud is voxelized. Once the neighborhood is defined based on the current and surrounding voxels, the RSD descriptor is computed as explained above. Based on the estimation of the two principal radii r_{min} and r_{max} , the surfaces are categorized based on intuitive rules defined in [54]. These surfaces are classified into: planes (large r_{min}), cylinders (medium r_{min} , large r_{max}), edges (small r_{min} , r_{max}), rims (small r_{min} , medium to large r_{max}), and spheres (similar r_{min} , and r_{max}). Once all voxels are categorized locally into one of these classes, the GRSD histogram is binned based on the number of transitions between all these local labels. The GRSD descriptor labels these transitions between distinctive surface types for an object.

$$b = \frac{s \cdot (s + 1)}{2}$$
(3)

where s is the number of possible categories, resulting in 21 dimensions for these 6 possible categories.

3.4. Global Descriptors Based on Fast Point Feature Histogram (FPFH)

This section outlines the family of global descriptors based on the computation of the local descriptor Fast Point Feature Histogram (FPFH) [46].

3.4.1. Viewpoint Feature Histogram (VFH)

The VFH was introduced by [53] as a global version of both the Point Feature Histogram (PFH) and the Fast Point Feature Histogram (FPFH) [46]. VFH describes the whole point cloud while PFH/FPFH are based on describing the local geometry around the key-points. The VFH has two components:

1. The Extended Fast Point Feature Histogram (EFPFH). This is an extended version of the FPFH. The difference between EFPFH and its predecessors lies in the way the geometry characteristic of the features is computed. For each point inside the point cloud, instead of comparing each couple of points inside predefined radii, the EFPFH compares each point with the centroid of the point cloud. The histogram is computed using the following steps, where the object is assumed as being a single cluster of points.
 - The centroid of the point cloud (c) is computed together with a normal (n_c).
 - For each point p_i and its normal n_i in the cluster, a reference frame is defined for each pair p_i and c , with origin in c , where the 3 axes of the frame are $u = n_c$, $v = (p_i - c) \times u$, and $w = u \times v$.
 - As illustrated in Figure 3a. From c , p_i , n_c and n_i , a set of features are computed from 3 angles (α , θ , ϕ) and the distance γ , as:

$$EFPFH = \begin{cases} \alpha &= \arccos(v \cdot n), \\ \phi &= \arccos(u \cdot \frac{p_i - c}{\|p_i - c\|}), \\ \theta &= \text{atan2}(w \cdot n, u \cdot n), \\ \gamma &= \|p_i - c\| \end{cases} \quad (4)$$

2. The viewpoint histogram. It is a histogram of the angles between the two vectors as shown in Figure 3b; the vector $v_p - p_i$ formed from the point p_i to the viewpoint v_p , and the normal n_i of the point p_i .

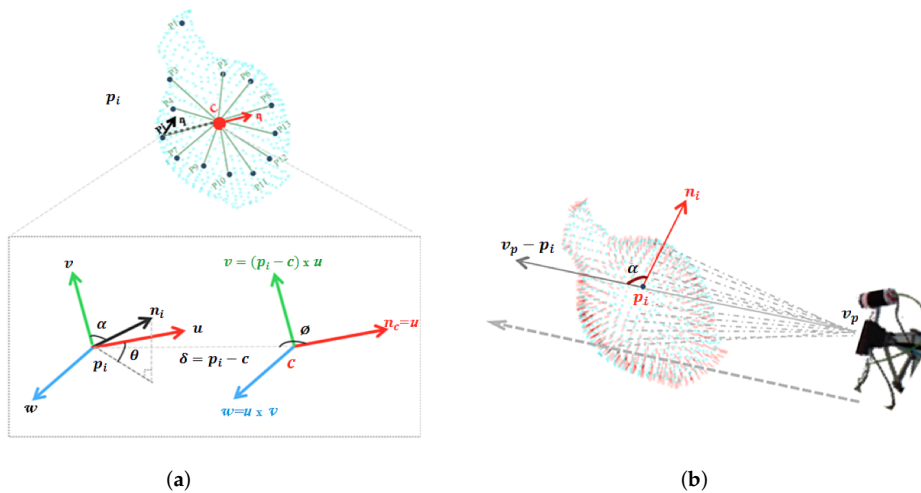


Figure 3. The two components of VFH: (a) Extended Fast Point Feature Histogram (EFPFH), the Computation of the three angular features. (b) the Viewpoint histogram. The histogram contains the stack of the angle α formed by the viewpoint toward the point p_i and its normal.

The complete size of the VFH descriptor is 308 bins composed of The EFPFH; 45 bins for each one $(\alpha, \theta, \phi, \gamma)$, plus 128 bins for the viewpoint component.

3.4.2. Global Fast Point Feature Histogram (GFPFH)

The Global Fast Point Feature Histogram (GFPFH) [33] is an extended version of the Fast Point Feature Histogram (FPFH) local descriptor. It is based on computing the number of angle histograms between angles of normals of each surface point and its neighboring points as explained in the Section 3.4.1. However, instead of comparing each point with the centroid of the point cloud, FPFH compares each pair of points p_i and p_j inside predefined radii, considering only the pairs with their direct neighbors.

The GFPFH descriptor needs a preliminary step, which consists of categorizing the surface into classes. These classes depend on the object and how it can be handled or decomposed for grasping. As an example, a cup is composed of a cylindrical body and handle where it can be grasped. Then, for each point, the FPFH is computed. In [33] the authors used The Conditional Random Field model [70] to label each surface with one of the object-classes.

Using the categorization results, the GFPFH descriptor is computed. The first step consists of representing the input point clouds by an octree, where each leaf contains a set of points. For each leaf, a probability of belonging to a particular class is assigned. This probability is computed as the ratio of the number of points in the labeled leaf according to that class over the total number of points.

In the following step, a line segment is created as illustrated in Figure 4, where the intersected leaf in its path is checked to see if it is occupied. The results are stored in a histogram based on the leaf occupancy: 0 if it is empty and, the leaf probabilities if it is occupied.

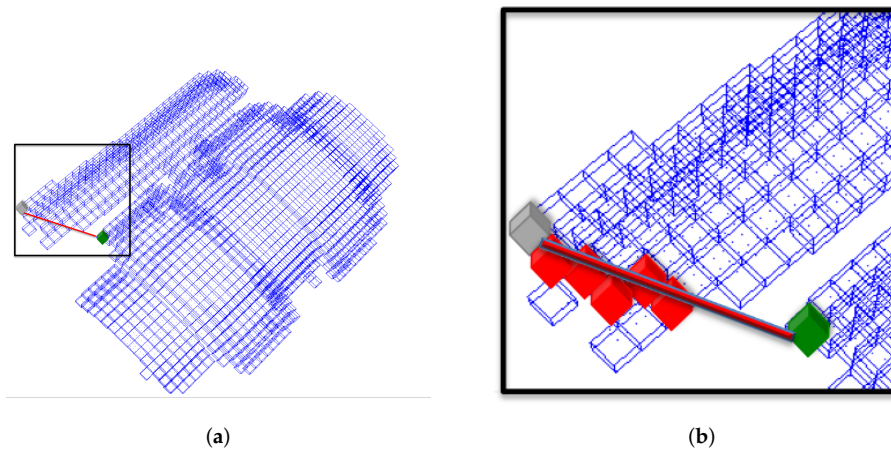


Figure 4. The estimation of a GFPFH for a 3D point cluster. (a) Octree representation of the point clouds. (b) illustrated zoom in of the region marked in (a), for every two pairs of leaves, a ray is cast from the start leaf (green) to the goal one (grey). All intersections with other leaves and free space are recorded and ported into a leaf class pair histogram.

3.4.3. Clustered Viewpoint Feature Histogram (CVFH)

The CVFH global descriptor was proposed by Aldoma et al. in [34] as an extension version of Viewpoint Feature Histogram. It aimed at solving the major limitations in VFH, that were essentially sensitivity to noise and occlusions (where the object has missing parts).

To make CVFH robust against occlusion, the authors proposed discarding the histogram normalization step used in computation of VFH, allowing the CVFH to become scale independent. To be able to distinguish objects with identical size and normals distribution, the authors added a Shape Distribution Component (SDC) in the computation of the histogram. The SDC encodes information about the distribution of the points p_i around the centroid of the region measured by the distances:

$$SDC = \frac{(c - p_i)^2}{\max((c - p_i)^2)} \quad (5)$$

where $i = 1, 2, \dots, N$, c denotes the centroid of the whole surface points and N represents the total number of the whole object surface points.

Rather than computing a single VFH histogram for the entire cluster, the main idea of the CVFH consists in splitting the object into stable regions by using smooth region growing segmentation algorithm [63]. For each region a VFH descriptor is computed.

The main advantages of the descriptor compared to its predecessor comes from the decomposition into a set of descriptors of the set of VFH clusters, which represents a multivariate description of the partial view. As long as any of the stable regions is visible, occlusions can be handled. The size of the CVFH is equal to the size of the VFH, where the number of bins used for this component is again 45 thus making a total size of 308 for CVFH.

3.4.4. Oriented, Unique and Repeatable CVFH (OUR-CVFH)

Despite the good result obtained in 3D recognition using CVFH [34], this descriptor suffers from two major drawbacks. On one hand, there is an absence of an aligned Euclidean space, causing the feature to lack a proper spatial description. On the other hand, it is invariant to rotations around the roll of the camera axis, thus restricting the pose estimation to 5 DoF.

In [62] Aldoma et al. presented the oriented, unique and repeatable CVFH (OUR-CVFH) descriptor, the last extension of FPFH. OUR-CVFH descriptor used semi-global unique and repeatable reference frames (SGURF) on object surfaces. The objective of using SGURF is to overcome the limitations of CVFH by defining multiple repeatable coordinate systems on the surface S .

4. Object Recognition Pipeline

To compare the descriptors, a 3D object recognition pipeline was used, which is described hereafter. Its block diagram appears in Figure 5. The pipeline is fed with an input scan coming either from a laser scanner (real experiments) or a virtual 3D camera (simulation). Then, a three step process is followed.

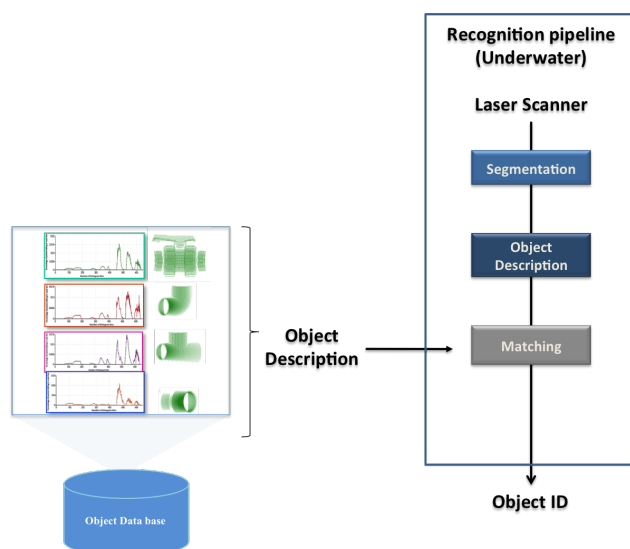


Figure 5. Block diagram of the proposed method.

1. Segmentation: Real scans pass through a segmentation phase, to remove any point not belonging to the object view. For instance, if the object is lying on the bottom of a water tank, removing

the principal plane (the bottom) is enough to correctly segment it. This is actually how it was implemented for the real experiments reported in Section 7. This step is skipped in the simulated results. The proposed 3D recognition pipeline requires a segmentation step, which aims to separate the 3D points belonging to the objects of interest from the rest of the scene. It consists of regrouping the points representing the object into one homogeneous group based on similar characteristics following the approach proposed in [71].

2. Description: This block uses the global descriptors, presented in the previous section, to encode the segmented object (input scan) in a compact way. The global object descriptors are also used to encode the object views stored in the database (object model). In this way the segmented input scan can be matched against the object model views in the database.
3. Matching: This step compares the segmented input scan with all the views of the object models in the database. The matching stage is based on computing Chi-square distance as proposed in [65,72]. The selected view corresponds to the one with the minimum distance.















The output of the recognition module is the object ID of the recognized object, as well as its matching view.

5. Experimental Setup

The main goal of this paper is to compare the robustness and performance of the global descriptors and to select the most adequate one for object recognition in an industrial underwater environment. A series of experiments were conducted in order to study the capabilities of the descriptors to distinguish among objects commonly present in Inspection Maintenance and repair (IMR) applications. The proposed objects database was composed of seven objects (Table 2) which are representative of an industrial scenario. The influence of the following parameters in the object recognition capabilities were studied:

- The use of full vs. partial views.
- The point cloud resolution.
- The presence of noise.

Table 2. Polyvinylchloride (PVC) pressure pipes objects used in the experiments.

PVC Objects	Id Name	Size (mm ³)	PVC Objects Views (12)
	1-Ball-Valve	198 × 160 × 120	
	2- Elbow	122.5 × 122.5 × 77	
	3- R-Tee	122.5 × 168 × 77	
	4- R-Socket	88 × 75 × 75	
	5- Ball-Valve-S	174 × 160 × 118	
	6- Butterfly-Valve	287.5 × 243 × 121	
	7- 3-Way-Ball-Valve	240 × 160 × 172	

Two types of experiments were performed:

1. **Simulated Experiments**, which involved the use of a virtual camera to generate a simulated point cloud of the object, grabbed from a random point of view. The virtual scan was characterized with all the descriptors being used to recognize the object. For each *<object, resolution, noise, full/partial*

view> combination, $n = 100$ Montecarlo runs of the experiment were performed, computing the average object recognition, and the confusion matrix.

2. **Real Experiments** involved the use of a laser scanner mounted on the GIRONA 500 AUV operating in a water tank scenario. Four objects were placed on the bottom of the water tank. The GIRONA500 vehicle was tele-operated to follow an approximated square trajectory, starting from a position where the reducing socket was within the field of view of the laser scanner. The trajectory ended with the robot on the ball valve, after passing over the elbow and reducing tee. The vehicle performed 3 complete loops, allowing it to acquire multiple views of the same object, each time it passed above it. The laser scanner was mounted looking towards the bottom providing full views of the objects.

5.1. Object Database

The experiments were conducted using the objects illustrated in Table 2. Each one was modelled as a complete set of potentially overlapping views stored as point clouds, covering the full object. The views were virtually scanned from the 3D CAD model using a method similar to the one reported in [34].

5.2. Virtual Scan

A simulated point cloud is generated using the tessellated sphere module from the PCL library [27]. The 3D CAD model of the object is placed at the origin of 3D space. Next, a sphere with a radius equal to the intended camera-to-object distance is used. The sphere is converted into a polyhedron depending on the level of tessellation, as illustrated in Figure 6. The virtual camera is then placed at each corner of the polyhedron. The number of views acquired is therefore equal to the number of corners (Equation (6)).

$$g(l) = (4^l \times 20) / 2 + 2 ; \text{ being } l \text{ the level of tessellation} \quad (6)$$

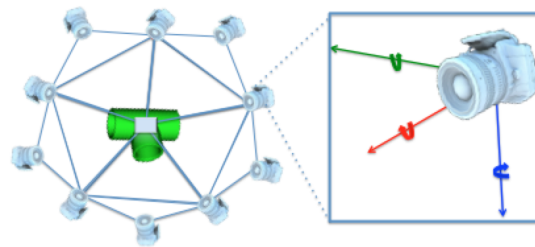


Figure 6. View points used for each object model stored in the database.

The two types of object views were used:

- **Full Object Views** using a level of tessellation fixed to 1, resulting in an icosahedron composed of 20 triangles and in 12 corners. The virtual camera was placed at each corner, at 0.5 m distance looking towards the origin, resulting in 12 full object views (Figure 6). These are the type of views used to represent the object in the database.
- **Partial Object Views** using a random vertex of the icosahedron to place the camera. The camera-to-object distance is randomly selected within the 0.2 to 1 m interval which is representative of the typical range for manipulation operations. The camera is also rotated around the three axes with a random angle of up to $\pm 10^\circ$.

5.3. Resolution

The XY resolution of the virtual 3D camera used to grab the point cloud was set at 150×150 pixels providing a dense point cloud at the working distances. Nevertheless, the point cloud was sub-sampled at different voxel sizes (Table 3) to study the influence of the scan resolution in the object recognition results.

Table 3. Views of the Ball-Valve scan with different resolution and noise levels used in the experiment: the line “VX” indicates the different voxel size resolutions and column “ σ ” indicates the different standard deviations of the added noise. All values are in meters.

$\sigma(m)$ \ VX(m)	0.003	0.005	0.007	0.009	0.011	0.013	0.015	0.017	0.019	0.021	0.023	0.025
0												
0.00625												
0.0125												
0.025												
0.05												
0.1												

6. Results on Simulated Data

The diagram below summarizes the experimental approach followed in this study as shown in the experiments column (Figure 7). The last column indicates a set of criteria that was used for the structuring the performance comparisons and the interpretation of the results. These criteria are the following: Difference of resolution between the scan and the object model in the database, Scan Resolution, Full vs. Partial Object View, Best Descriptor and Object Confusion.

Four different experiments were performed, depending on whether a full or partial view was used, and if the resolution of the scan and the database object models was the same or not:

1. Full View Same Resolution Experiment (FVSR).
2. Full View Different Resolution Experiment (FVDR).
3. Partial View Same Resolution Experiment (PVSR).
4. Partial View Different Resolution Experiment (PVDR).

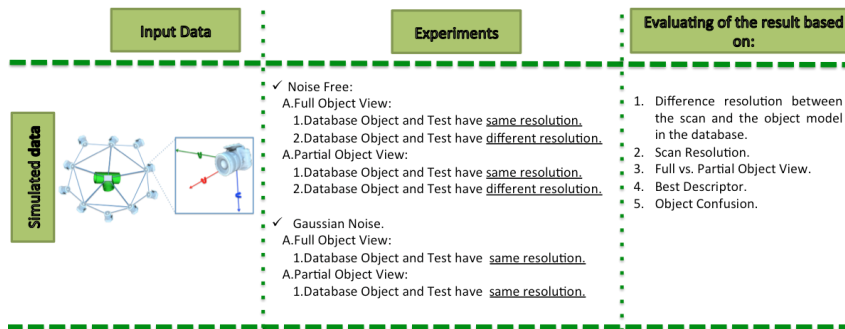


Figure 7. Diagram of the proposed simulated experiment.

Each experiment involved 100 Montecarlo runs, and their results are respectively shown in Figures 8–11. Moreover, the average results among all the objects for all the resolutions are shown in Table 4. Finally, a summary of the results taking into account all the objects and all the studied resolutions is shown in Table 5.

Table 4. Average of recognition per resolution for all descriptors: (top-left) Using full object views and the same resolution between the model and the measurement; (top-right) Using partial object views and the same resolution between the model and the measurement; (bottom-left) Using full object views and different resolution between the model and the measurement; (bottom-right) Using partial object views and different resolution between the model and the measurement.

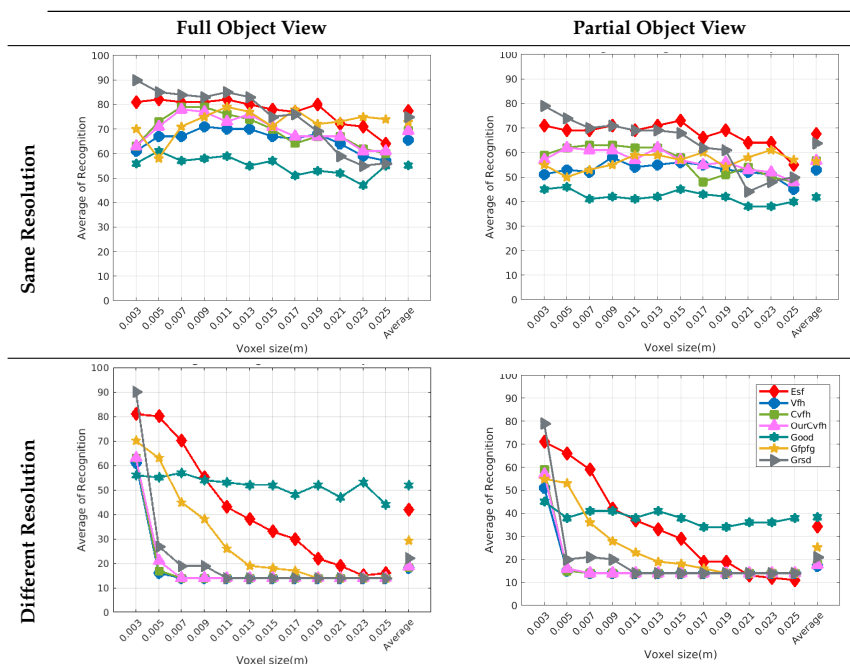


Table 5. Summary of results for all the objects and all the resolutions. The best descriptor is marked in green, while the worst one is marked in red.

Experiment	View	Resolution	Descriptors							Average Over Descriptors
			ESF	VFH	CVFH	OURCVFH	GOOD	GPPFH	GRSD	
FVSR	Full	Same	77.4	65.3	69.5	69.3	55.1	72.8	75.0	69.2
FVDR		Different	41.8	18.1	18.3	18.7	51.9	29.3	22.3	28.6
PVSR	Partial	Same	67.6	52.8	56.7	56.8	41.9	56.5	63.8	56.6
PVDR		Different	34.3	17.2	17.8	17.8	38.3	25.3	21.0	24.5
FVSR/ FVDR	Average Over	Full View	59.6	41.7	43.9	44.0	53.5	51.0	48.6	48.9
PVSR/ PVDR		Partial View	50.9	35.0	37.3	37.3	40.1	40.9	42.4	40.5
FVSR/ PVSR		Same Res	72.5	59.1	63.1	63.0	48.5	64.6	69.4	62.9
FVDR/ PVDR		Diff Res	38.0	17.6	18.1	18.2	45.1	27.3	21.6	26.6
FVSR/ FVDR/ PVSR/ PVDR		Full Average	55.3	38.4	40.6	40.6	46.8	46.0	45.5	44.7

6.1. Difference of Resolution Between the Scan and the Object Model in the Database

Figure 8 shows the average recognition for every descriptor and every resolution when both the database and the measured scan contain full views of the object and have the same resolution (FVSR). On the other hand, Figure 9 shows the same data for a second set of experiments (FVDR), when the resolution used for the object model in the database and the resolution of the measured scan are different. The blue color (indicating a high average recognition rate) the top row Figure 8 is significantly more present than in Figure 9, meaning that better results are achieved using the same resolution instead of different ones. The same can be appreciated in Figures 10 and 11 for the more realistic case when the measured scan shows only a partial view of the object (PVSR and PVDR experiments). This can be clearly observed in Figure 4 which shows the average recognition rate for all the descriptors and all the resolutions for full and partial object views in both cases, with the same and different resolutions. The lower row of the figure clearly shows a significant drop in the average recognition rate for resolutions beyond 0.005 for both cases, partial and full object views. Finally, the last column of Table 5 (*Average over descriptors*) shows the recognition rate averaged for all objects, all resolutions and all the descriptors. When the same resolution is used the recognition rate is 62.9 reducing to only 26.6 when the resolutions differ.

6.2. Scan Resolution

To study how the resolution affects the recognition let us have a look at the upper row of Table 4. In both cases, full view (shown at left side) and partial view (shown at right side), it can be appreciated how the performance decreases with the reduction of the resolution for the two better descriptors: ESF and GRSD. Interestingly, the performance of the GOOD descriptor remains almost constant across the different resolutions, even when different resolutions are used among the database object models and the scan. The behaviour for the rest of descriptors is more arbitrary not showing a clear trend. The lower row of Table 4 clearly shows how the performance decreases as the difference between the database model and the measured scan increases (the object model is at resolution 0.003 and the input scan resolution is varied during the experiment), with the remarkable exception of the GOOD descriptor.

6.3. Full vs. Partial Object View

The importance of measuring a view as wide as possible is shown in Table 4 as well. The left-hand column of the table corresponds to the case when the full view of the object is observed (FVSR and FVDR). The right-hand column corresponds to a partial view (PVSr and PVDR). In both cases, for same and different resolution between the model and the measured scans, a $\approx 10\%$ decrease in the average recognition rate is observed. This decrease in the average recognition is confirmed in Table 5 where the averaged recognition rate for all descriptors is 8.4% better for full view (48.9) than for partial view (40.5%). Unfortunately, the observation of partial views is the more realistic case so its results should be considered more representative of the reality.

6.4. Best Descriptor

Qualitatively, the best performing descriptor can be inferred from the top row of Figures 8–11. The descriptors whose $object \times voxel_size$ grid is predominantly blue are the ones performing better, while those predominantly yellow, orange or red are progressively the worst ones. In the results of the FVSR experiment, Figure 8, it can be clearly appreciated that GRSD and ESF are the best descriptors, while GOOD is the worst one. When partial views are used instead, PVSr experiment, a decrease of performance (colors shifted towards green) can be seen, but with essentially the same results. ESF and GRSD continue being the best descriptors while GOOD is the one with lower performance. In the FVDR and PVDR experiments, when different resolutions between the model and the scan are used the scenario is totally different. In this case, only the GOOD descriptor is able to provide significant results.

The results may also be analyzed quantitatively. Table 5 shows how well each descriptor performed (averaged among objects and resolutions). The results averaged among all the experiments using full views, show clearly that the best performing descriptor is ESF (59.6%) followed by GOOD (53.5%) while CVFH (43.9%) is the worst one. If partial views are used instead, the best descriptor is still ESF (50.9%), followed in this case by GRSD (42.4%), with VFH (35.0%) being the worst one. If we focus only on the dimension related to the same/different resolution, then, using the same resolution ESF (72.5%) is the best one followed by GRSD (69.4%) with GOOD (48.5%) being the one performing worst. When different resolutions are used instead, GOOD (45.1%) becomes the best one followed by ESF (38.0%), and VFH (17.6%) the worst. If we average the results among all the experiments (last row of the table), we conclude that ESF (55.3%) is the one performing better in general followed by GOOD (46.8%), and VFH (38.4%) the worst performing one. In our opinion, the most relevant results corresponds to the PVSr experiment because having full views is not always possible and, at least in our case, having the same resolution is always easy. In this case, ESF (67.6%) and GRSD (63.8%) are the best descriptors and GOOD (41.8%) is the one providing poorest results.

6.5. Object Confusion

Besides looking at the average recognition rate and, in order to understand the descriptor capabilities for object recognition, it is good to examine the confusion matrices. For every object, they show the object-class that is recognized, but also, when it is mis-recognized, which are the classes that generate the confusion. It is worth noting, hence, that the smaller the recognition rate the higher the confusion. Figures 8–11 show, in their bottom row, the confusion matrices for the different experiments. To extract conclusions about confusion, regardless of the descriptor, we averaged the results among all the descriptors in Table 6. Examining them we can extract the following general conclusions:

1. The lower the resolution the higher the confusion. This can be appreciated in Table 4 since the recognition rate decreases with the resolution.
2. The recognition rate is higher than the mis-recognition (The addition of all the confusion percentages) only when the same resolution is used. Using the same resolution leads to less

confusion (60.3% average recognition rate), while using different resolutions leads to significantly higher confusion (27% average recognition rate).

3. The use of full views also leads to less confusion (49.4 average recognition rate) than using partial views (37.9 average recognition rate).

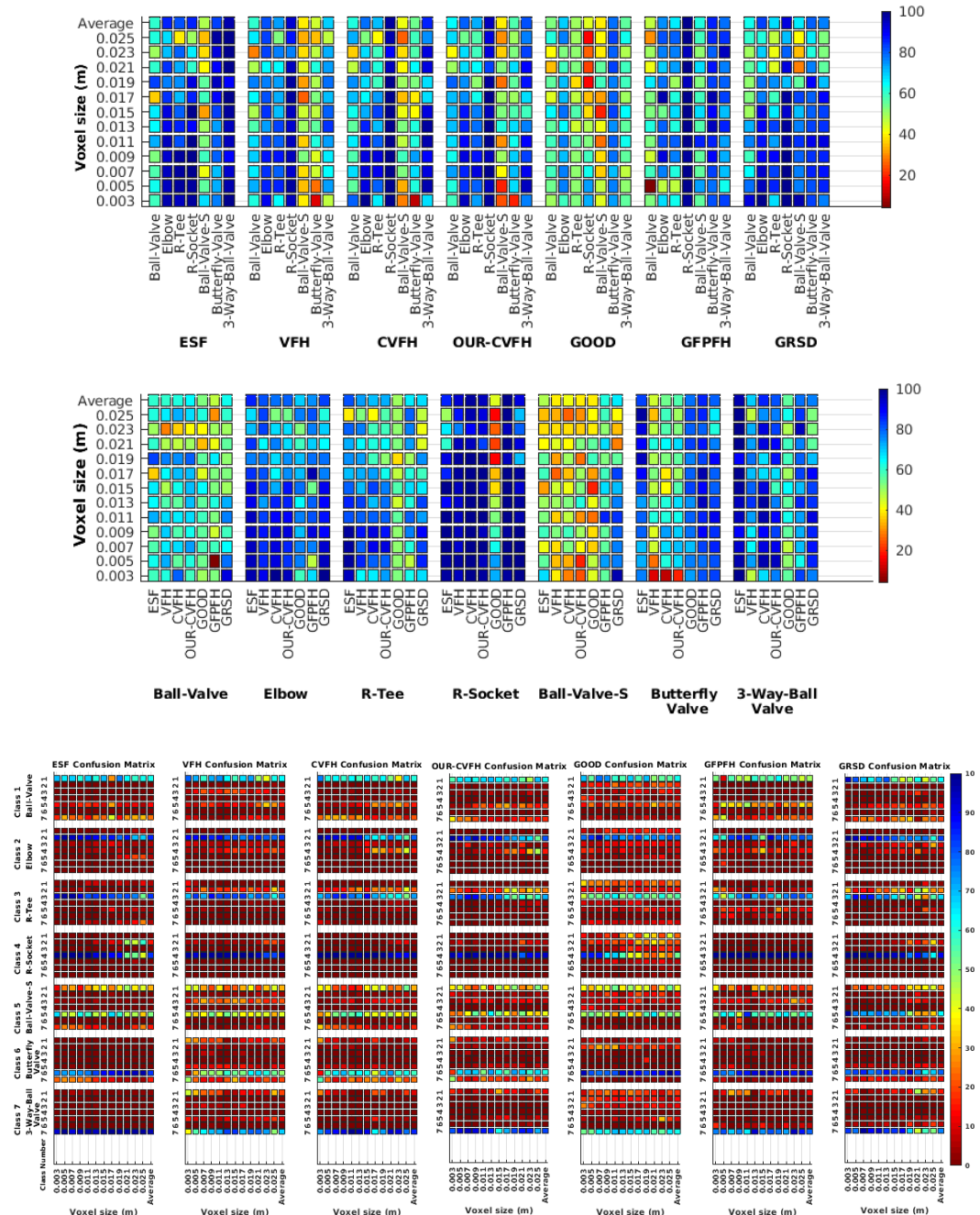


Figure 8. Average of recognition per resolution for all descriptors, using full views and having the same resolution for the model and the measurement: (Top) Grouped by descriptor; (Middle) Grouped by object; (Bottom) Confusion Matrix.

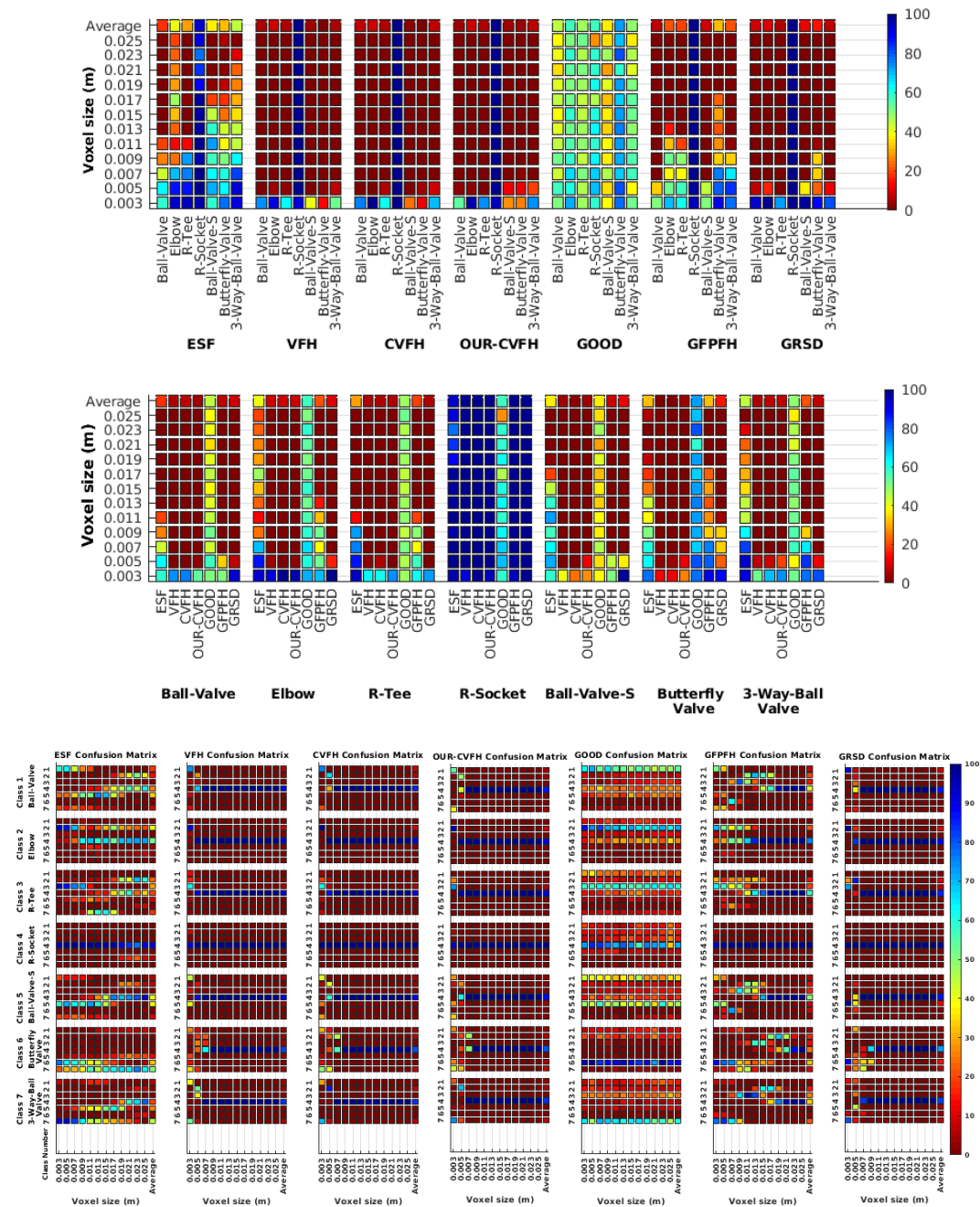


Figure 9. Average of recognition per resolution for all descriptors, using full views and having the different resolution for the model and the measurement: **(Top)** Grouped by descriptor; **(Middle)** Grouped by object; **(Bottom)** Confusion Matrix.

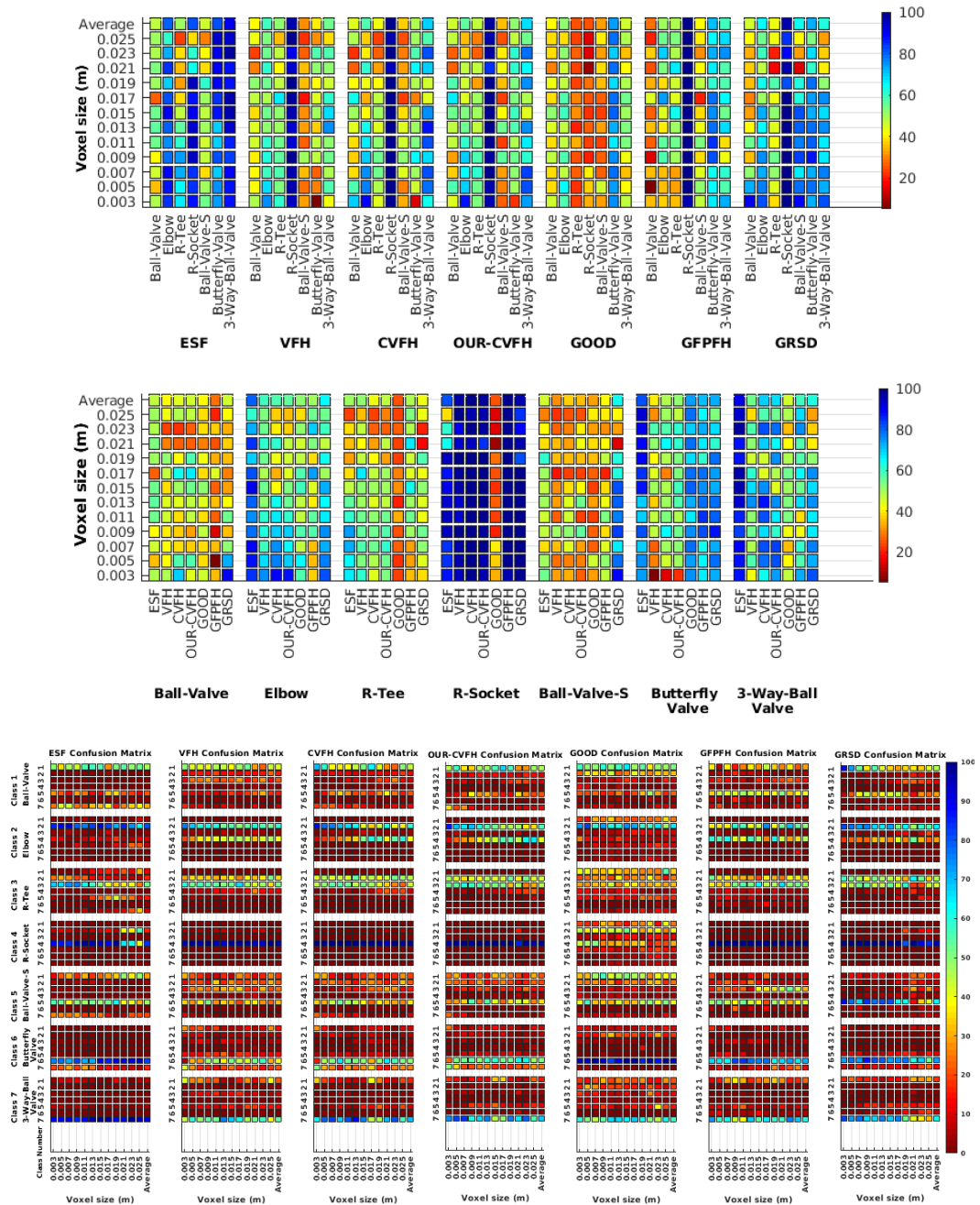


Figure 10. Average of recognition per resolution for all descriptors, using partial views and having the same resolution for the model and the measurement: (Top) Grouped by descriptor; (Middle) Grouped by object; (Bottom) Confusion Matrix.

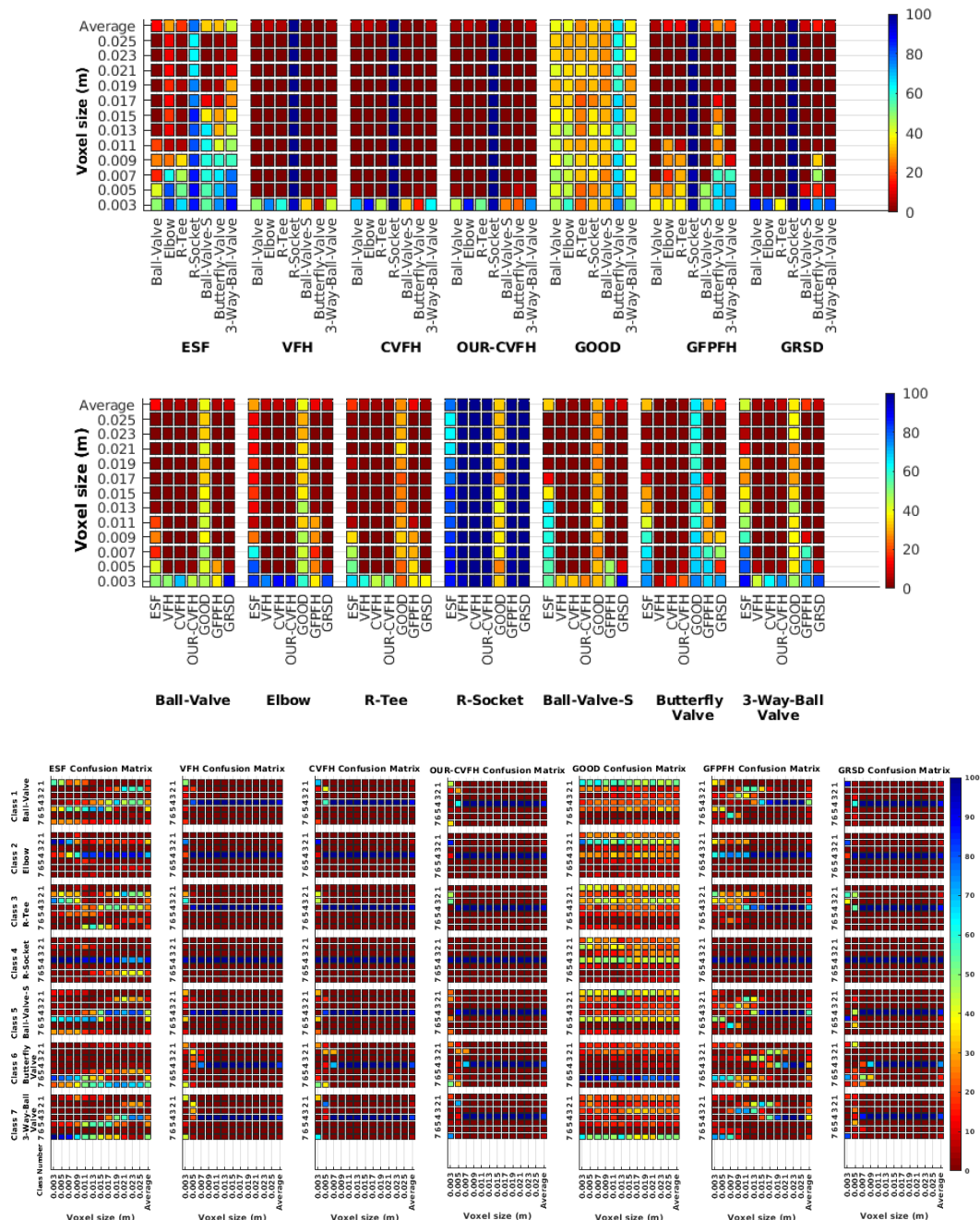


Figure 11. Average of recognition per resolution for all descriptors, using partial views and having different resolution for the model and the measurement: (Top) Grouped by descriptor; (Middle) Grouped by object; (Bottom) Confusion Matrix.

Now, several relevant questions arise:

1. *Which is the descriptor provoking most confusion?* Let us focus on the results when using the same resolution which we consider the most interesting ones. In this case, see Table 5, the descriptor with the lowest recognition rate is GOOD, being hence the one leading to higher confusion. This can be confirmed looking at its confusion matrices in Figures 8 and 10. On the other side

of the spectrum we find ESF and GRSD which have good recognition rates (72.5% and 69.4% respectively), leading to less confusion as can be appreciated in Figures 8 and 10.

- When confusion arises, which are the objects more prone to be confused? As stated above, the two most interesting scenarios are the ones corresponding to the same resolution, FVSR and PVSR. Figure 12 shows how objects are confused in those scenarios. The green arrows correspond to the confusions appearing (those whose percentages is higher than 5%) when using full views. In this case, most of the confusion appears either among the valves (O_1, O_5, O_6, O_7) or among the Elbow, R-Tee and the R-Socket objects. Moreover The R-Tee is also confused with Ball-Valve-S (O_5). When partial views are used instead, the blue arrows add on top of the green ones showing new confusions (The black ones still exist with partial views), making the object identification more challenging. The graph shows clearly how the use of partial views leads to more confusion.

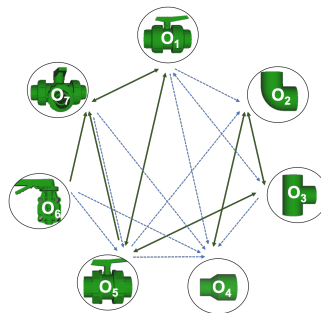


Figure 12. Confusion per Object Graph.

Table 6. Summary of confusion matrices for all the objects and all descriptors averaged along resolutions. Marked in blue are those confusions greater than 5% which were used to build the object confusion graph of Figure 12.

Experiment	View	Resolution	Objects																											
			Ball Valve							Elbow							R-Tee							R-Socket						
			1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7
FVSR	Full	Same	58,8	2,8	4,3	0,7	16,7	0,5	16,3	1,6	78,8	6,8	10,7	1,4	0,6	0,1	4,6	15,6	69,6	1,6	5,3	0,9	2,3	2,8	8,3	1,3	86,3	1,0	0,1	0,2
FVDR		Different	13,5	7,5	3,8	62,3	8,0	1,1	3,8	1,7	22,0	1,5	73,3	0,8	0,6	0,1	3,1	8,6	17,0	64,3	2,3	1,0	3,8	1,5	2,1	1,7	92,7	1,0	0,8	0,2
PVSR	Partial	Same	36,6	9,9	8,7	13,5	19,8	1,1	10,4	3,0	51,5	3,2	39,0	2,0	0,7	0,7	5,6	28,5	38,6	18,6	5,7	0,7	2,2	2,7	7,8	1,2	85,4	1,6	0,7	0,6
PVDR		Different	11,3	7,9	3,3	63,7	8,5	1,5	3,7	2,9	15,5	1,4	77,2	1,4	0,9	0,8	3,9	11,1	10,6	66,8	3,3	1,1	3,2	2,6	4,1	1,9	87,0	1,2	2,5	0,6
FVSR/ FVDR	Average Over	Full View	36,1	5,2	4,0	31,5	12,4	0,8	10,0	1,6	50,4	4,1	42,0	1,1	0,6	0,1	3,9	12,1	43,3	32,9	3,8	1,0	3,1	2,1	5,2	1,5	89,5	1,0	0,4	0,2
PVSR/ PVDR		Partial View	24,0	8,9	6,0	38,6	14,2	1,3	7,1	2,9	33,5	2,3	58,1	1,7	0,8	0,7	4,7	19,8	24,6	42,7	4,5	0,9	2,7	2,7	6,0	1,6	86,2	1,4	1,6	0,6
FVSR/ PVSR		Same Res	47,7	6,3	6,5	7,1	18,3	0,8	13,3	2,3	65,2	5,0	24,8	1,7	0,7	0,4	5,1	22,1	54,1	10,1	5,5	0,8	2,3	2,8	8,1	1,3	85,9	1,3	0,4	0,4
FVDR/ PVDR		Diff Res	12,4	7,7	3,5	63,0	8,3	1,3	3,8	2,3	18,8	1,4	75,2	1,1	0,7	0,5	3,5	9,9	13,8	65,5	2,8	1,1	3,5	2,0	3,1	1,8	89,9	1,1	1,7	0,4
FVSR/ FVDR/ PVSR/ PVDR	Full Average	Same	30,1	7,0	5,0	35,0	13,3	1,1	8,5	2,3	42,0	3,2	50,0	1,4	0,7	0,4	4,3	16,0	34,0	37,8	4,1	0,9	2,9	2,4	5,6	1,5	87,9	1,2	1,0	0,4
		Different	12,4	7,7	3,5	63,0	8,3	1,3	3,8	2,3	18,8	1,4	75,2	1,1	0,7	0,5	3,5	9,9	13,8	65,5	2,8	1,1	3,5	2,0	3,1	1,8	89,9	1,1	1,7	0,4

Experiment	View	Resolution	Objects														Average Recognition For all objects							
			Ball Valve-S				Butterfly Valve					3-Way Valve												
			1	2	3	4	5	6	7	1	2	3	4	5	6	7		1	2	3	4	5	6	7
FVSR	Full	Same	29,4	3,4	9,4	0,7	47,8	0,8	8,6	6,2	3,8	1,3	0,4	4,8	70,1	13,6	12,4	1,7	2,1	0,2	4,3	2,4	76,8	69,8
FVDR		Different	6,9	4,9	3,7	65,4	14,6	1,5	2,9	4,0	6,6	2,8	49,0	2,6	22,5	12,4	4,1	6,3	7,2	55,9	5,2	1,0	20,4	29,0
PVSR	Partial	Same	18,5	14,1	13,0	14,0	33,1	1,4	6,0	7,0	4,5	2,7	11,3	5,2	54,3	14,9	14,5	3,0	4,0	12,2	8,2	1,6	56,5	50,9
PVDR		Different	6,1	5,8	3,4	66,6	12,6	1,9	3,5	5,2	6,1	3,1	51,0	3,0	20,4	11,3	5,1	7,3	6,3	57,4	5,7	0,9	17,2	24,9
FVSR/ FVDR	Average Over	Full View	18,1	4,2	6,5	33,0	31,2	1,2	5,8	5,1	5,2	2,0	24,7	3,7	46,3	13,0	8,3	4,0	4,7	28,0	4,7	1,7	48,6	49,4
PVSR/ PVDR		Partial View	12,3	10,0	8,2	40,3	22,8	1,6	4,8	6,1	5,3	2,9	31,1	4,1	37,4	13,1	9,8	5,2	5,2	34,8	6,9	1,3	36,8	37,9
FVSR/ PVSR		Same Res	23,9	8,8	11,2	7,3	40,5	1,1	7,3	6,6	4,2	2,0	5,9	5,0	62,2	14,2	13,5	2,3	3,0	6,2	6,3	2,0	66,6	60,3
FVDR/ PVDR		Diff Res	6,5	5,4	3,5	66,0	13,6	1,7	3,2	4,6	6,4	2,9	50,0	2,8	21,5	11,8	4,6	6,8	6,8	56,6	5,4	0,9	18,8	27,0
FVSR/ FVDR/ PVSR/ PVDR	Full Average	Same	15,2	7,1	7,4	36,7	27,0	1,4	5,3	5,6	5,3	2,5	27,9	3,9	41,8	13,0	9,1	4,6	4,9	31,4	5,8	1,5	42,7	43,6
		Different	6,5	5,4	3,5	66,0	13,6	1,7	3,2	4,6	6,4	2,9	50,0	2,8	21,5	11,8	4,6	6,8	6,8	56,6	5,4	0,9	18,8	27,0

6.6. Gaussian Noise

In this section Gaussian noise is introduced in the simulation to study how it affects the recognition rate. Only the case of the same resolution is evaluated since from previous results it is clear that it provides the best results. The two experiments, the full and partial views, were considered. In both cases 100 Montecarlo runs were executed over 12 different resolutions for 6 different noise levels (see Table 3). Moreover, the average results among all the objects for all the resolutions are shown in Table 7. Finally, a summary of the results taking into account all the objects and all the studied resolutions is shown in Table 8.

Assuming 0.007 is the resolution for the scanner used in the real experiments reported in the next section, Figures 13 and 14 show the recognition rate, for all the descriptors, and every noise level, at this resolution, respectively for:

1. Noisy Full View Same Resolution Experiment (NFVSR).
2. Noisy Partial View Same Resolution Experiment (NPVSR).

6.6.1. Scan Resolution

Table 7, shows the recognition rate averaged for all objects, detailed for every resolution. It is interesting to note that while most of the descriptors' performance decreases with the resolution as well as with the noise, some of them show a very poor result for the high-resolution and high-noise combination. This is the case for the GRSD and ESF (at higher noise ratios). We attribute this to the fact that ESF is based on the shape function computing distances and angles between random points and GRSD used the radial relationships to describe the geometry of points at each voxel. Accordingly, the impact of these two factors could be amplified when both resolution and noise level are high.

6.6.2. Full vs. Partial Object View

As observed in the previous experiments, better recognition rates are achieved using full views (see Table 8). The improvement with respect to the use of partial view ranges from $\approx 15\%$ at $\sigma \in [0 - 0.0125]$, $\approx 15\%$ at $\sigma \in [0.025 - 0.05]$, and $\approx 3\%$ at $\sigma = 0.1$.

6.6.3. Best Descriptor

The results reported in Table 8 show that ESF is either the best, or the second best, descriptor except for the highest noise level where its performance drops significantly, making it one of the 2 worst performing descriptors. GRSD works well at low noise levels but its performance drops significantly when the noise is medium to high. GOOD is the worst performing one at low noise levels, but performs well at high noise. VFH performs poorly across the whole noise spectrum while CVFH, OUR-CVFH and GPPFH present intermediate performance levels. On the other hand, Table 9 shows the recognition rate, for all the descriptors, averaged by object, for the resolution 0.007. This is the assumed resolution for the scanner used in the real experiments reported in the next section. There we can see that for full view and low noise ($\sigma = 0.00625$), which is the case corresponding to our sensor, GRSD is the best descriptor followed by ESF, GPPFH, OUR-CVFH and CVFH. GOOD and VFH are the worst ones. If we go to the other extreme, high noise ($\sigma = 0.1$), GOOD becomes the best descriptor with results close to CVFH and OUR-CVFH. GRSD is the worst performing one closely followed by ESF, GPPFH and VFH. For partial views and low noise ESF and GRSD are the best ones, followed by OUR-CVFH and CVFH (medium-high performance), VFH and GPPFH (medium-low performance) and GOOD (worst performance). At high noise levels the best one is GOOD followed by GPPFH, CVFH and ESF with less performance while VFH and GRSD are the worst ones.

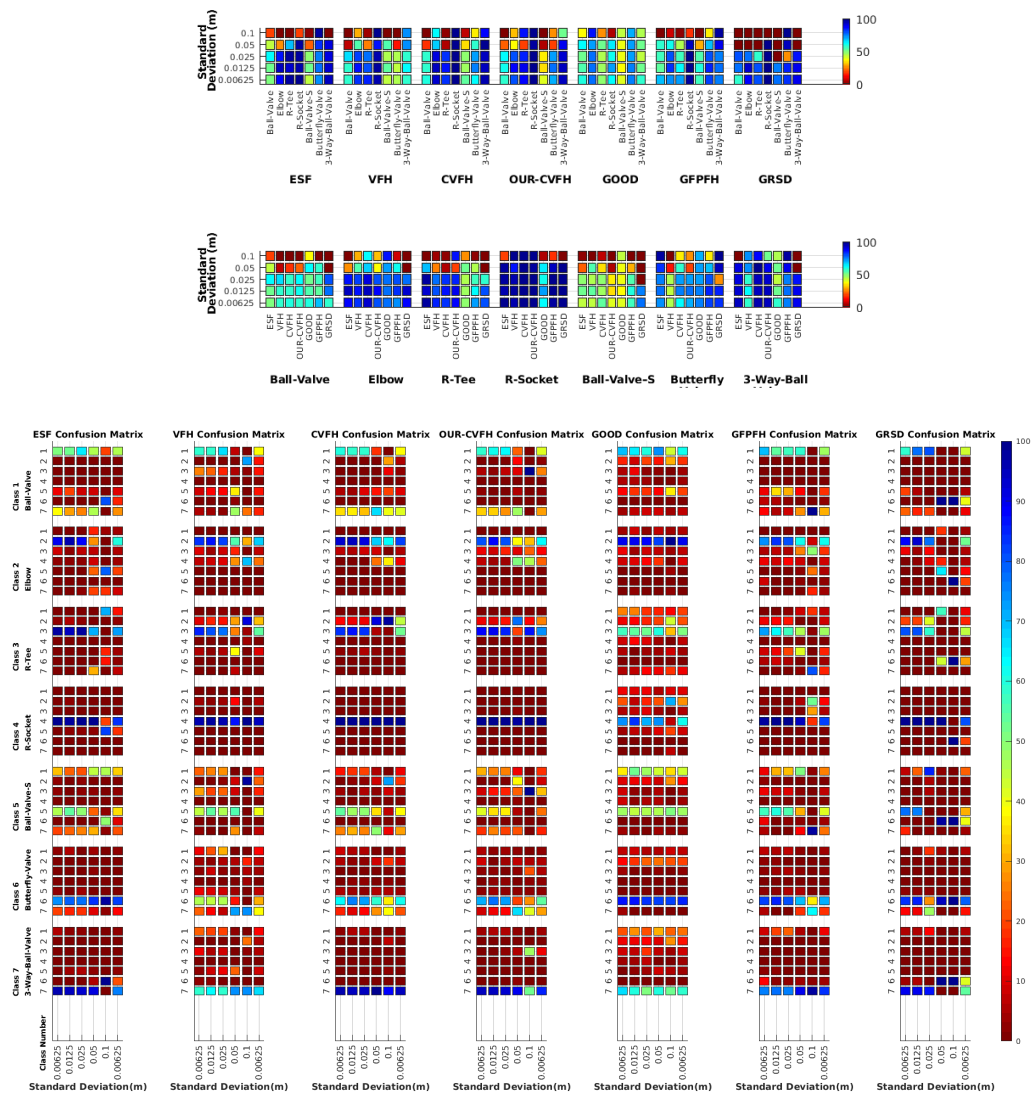


Figure 13. Average of recognition for the resolution 0.007 for all descriptors, using full views and having the same resolution for the model and the measurement: (Top) Grouped by descriptor; (Middle) Grouped by object; (Bottom) Confusion Matrix.

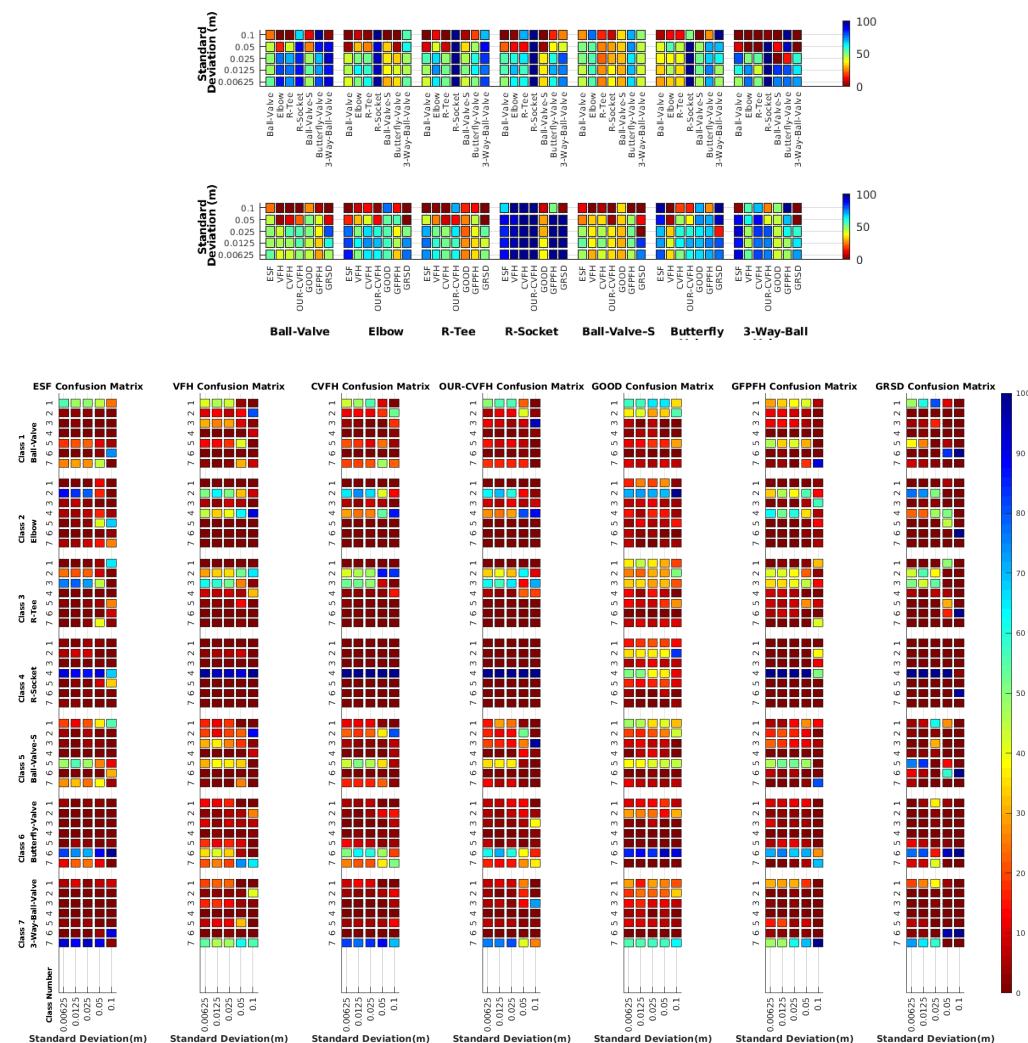
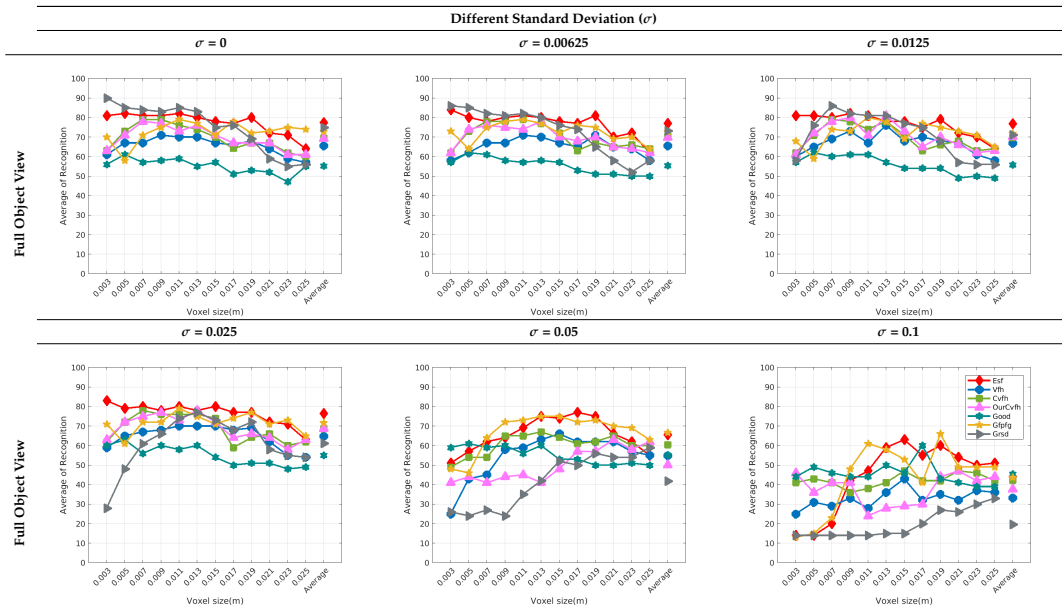


Figure 14. Average of recognition for the resolution 0.007 for all descriptors, using partial views and having different resolution for the model and the measurement: (Top) Grouped by descriptor; (Middle) Grouped by object; (Bottom) Confusion Matrix.

Table 7. Average of recognition per resolution for all descriptors using the same resolution between the model and the scan. The results are shown for 6 different noise levels and for 2 cases, full and partial object views.



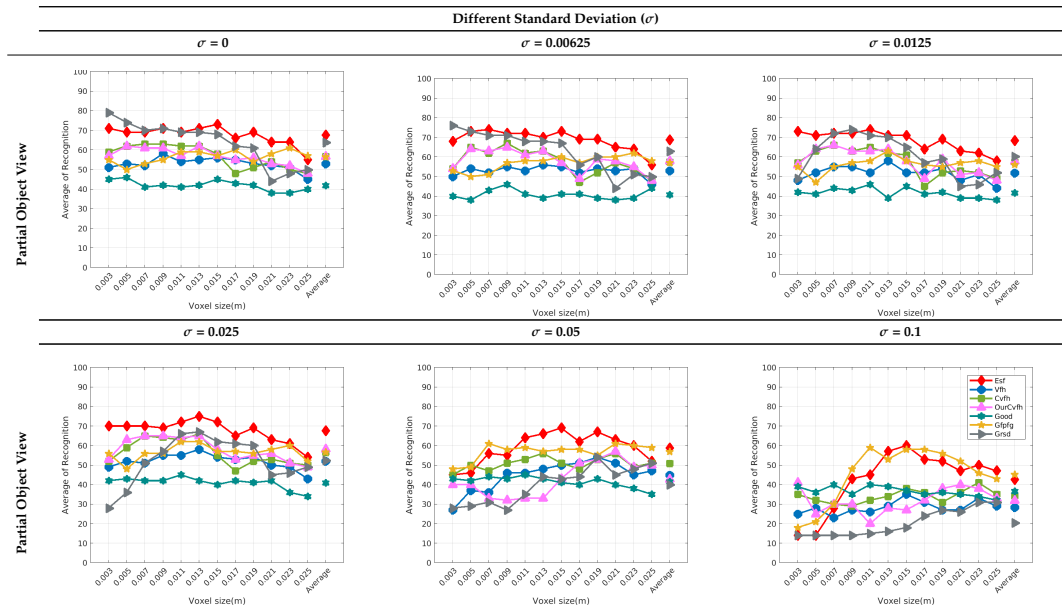
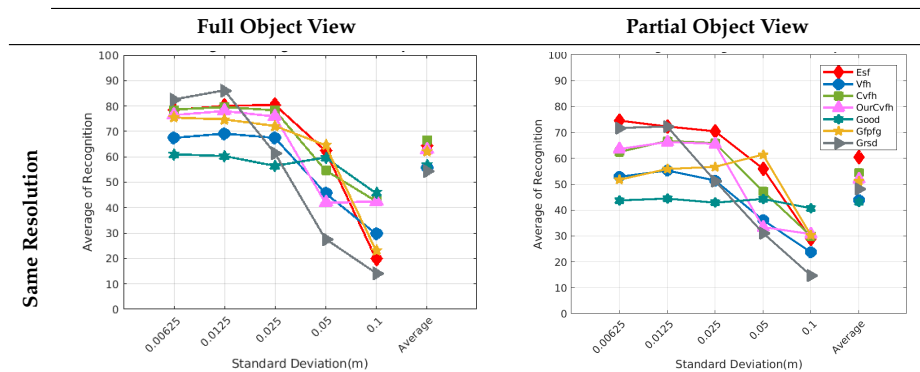


Table 8. Average Recognition for all noise levels and all descriptors, averaged by resolution. The two best performing descriptors are marked in green and light green respectively, and the worst performing ones in dark and light red.

Noise Std	Experiment	View	Resolution	Descriptors						Average Over Descriptors	
				ESF	VFH	CVFH	OURCVFH	GOOD	GPPFH		GRSD
$\sigma = 0$	FVSR	Full	Same	80,7	65,9	77,7	77,0	61,9	71,1	85,6	74,3
	PVSR	Partial	Same	69,7	52,3	63,9	61,4	41,7	53,4	70,4	59,0
$\sigma = 0.00625$	FVSR	Full	Same	78,4	67,4	78,6	76,4	61,0	75,4	82,6	74,3
	PVSR	Partial	Same	74,6	52,9	62,4	63,6	43,7	51,7	71,7	60,1
$\sigma = 0.0125$	FVSR	Full	Same	80,0	69,1	79,6	78,0	60,3	74,7	86,1	75,4
	PVSR	Partial	Same	72,3	55,3	66,7	66,3	44,4	55,9	72,3	61,9
$\sigma = 0.025$	FVSR	Full	Same	80,4	67,4	78,3	75,9	56,4	72,1	61,4	70,3
	PVSR	Partial	Same	70,4	51,4	65,9	65,6	42,9	56,7	51,3	57,7
$\sigma = 0.05$	FVSR	Full	Same	62,4	45,7	54,7	41,9	59,7	64,6	27,7	51,0
	PVSR	Partial	Same	56,0	36,1	36,1	33,4	44,3	61,4	31,1	42,7
$\sigma = 0.1$	FVSR	Full	Same	20,0	29,7	42,6	42,4	45,7	23,3	14,3	31,1
	PVSR	Partial	Same	28,9	23,7	30,0	30,7	40,7	30,3	14,7	28,4
Average	FVSR/PVSR	Average	Same	64,5	51,4	61,4	59,4	50,2	57,6	55,8	57,2

Table 9. Average of recognition per different standard deviation for the resolution 0.007 and for all descriptors: (top-left) Using full object views and the same resolution between the model and the measurement; (top-right) Using partial object views and the same resolution between the model and the measurement.



6.6.4. Object Confusion

Table 10 shows the confusion tables for all object and all noise combinations averaged for all the descriptors. Notice that the cells marked in blue are those above 5%. The 'average recognition for all objects' column shows that on average, for all the objects, the recognition works well for noise levels equal to or below 0.025. Beyond that, the recognition rate falls below 50%. It can be observed, that for $\sigma \leq 0.025$ all the objects are recognized with a recognition rate over 50% for both cases, full and partial views, except for the Ball-Valve and the Ball-Valve-S objects. For them, the recognition rate is sometimes below 50%, especially for partial views. It can also be appreciated, that the number of cells with a percentage of confusion over 5% (cells in blue) increases when using partial views than when using full views. This effect is observed in almost every object indicating that the use of partial views leads to more confusion. For $\sigma \geq 0.05$ the recognition rate decrease significantly below 50% with the exception of the R-Socket and the Butterfly-Valve objects. Figure 15 shows the confusion graph for the lowest and highest noise cases. For the first case, it can be seen how new confusion links appear when partial views are used instead of full ones. This is consistent with the results of the previous section. In the case of high noise, most of the confusion actually appears even using full views, and only two

more confusions appear when using partial views. As expected, this shows how increasing the noise also increases the percentage of confusions.

Table 10. Confusion table for all objects and noises.

Experiment	View	Resolution	Objects																												
			Ball Valve							Elbow							R-Tee							R-Socket							
			1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7	
$\sigma = 0$	FVSR	Full	Same	59.9	2.9	23.6	0.6	13.7	0.4	17.4	0.7	84.7	7.0	7.0	0.1	0.4	0.0	3.9	15.3	71.9	1.7	3.9	2.0	1.4	0.9	2.3	0.3	95.9	0.7	0.0	0.0
	PVSR	Partial	Same	45.7	9.0	7.9	0.4	21.1	0.3	15.6	3.6	66.0	5.3	22.6	1.1	0.6	0.9	3.9	31.0	54.7	5.1	3.9	0.6	0.9	2.1	6.1	0.9	88.6	1.3	0.3	0.7
	FVSR/PVSR	Average		52.8	5.9	15.7	0.5	17.4	0.4	16.5	2.1	75.4	6.1	14.8	0.6	0.5	0.4	3.9	23.1	63.3	3.4	3.9	1.3	1.1	1.5	4.2	0.6	92.2	1.0	0.1	0.4
$\sigma = 0.00625$	FVSR	Full	Same	57.9	3.0	6.1	0.7	11.9	1.0	19.4	0.4	81.1	8.6	8.3	0.1	1.3	0.1	3.4	14.3	75.7	2.0	3.0	0.9	0.7	1.4	3.7	0.7	93.7	0.4	0.0	0.0
	PVSR	Partial	Same	45.6	9.1	8.7	0.3	21.4	0.3	14.6	1.9	64.4	4.9	26.1	0.4	0.9	1.4	4.1	30.3	55.4	5.1	3.4	0.9	0.7	2.1	5.0	0.6	89.3	2.0	0.4	0.6
	FVSR/PVSR	Average		51.7	6.1	7.4	0.5	16.6	0.6	17.0	1.1	72.8	6.7	17.2	0.3	1.1	0.8	3.8	22.3	65.6	3.6	3.2	0.9	0.7	1.8	4.4	0.6	91.5	1.2	0.2	0.3
$\sigma = 0.0125$	FVSR	Full	Same	58.9	2.7	6.0	0.7	14.6	0.7	16.4	1.0	85.9	4.4	7.9	0.1	0.4	0.3	3.4	16.7	72.9	2.3	2.6	1.3	0.9	1.3	1.7	0.9	95.0	1.1	0.0	0.0
	PVSR	Partial	Same	48.9	10.4	7.6	1.0	17.4	0.7	14.0	2.9	69.1	4.6	20.7	1.6	0.4	0.7	5.3	32.1	54.7	3.0	2.3	1.3	1.3	2.1	6.3	1.0	88.3	1.7	0.3	0.3
	FVSR/PVSR	Average		53.9	6.6	6.8	0.9	16.0	0.7	15.2	1.9	77.5	4.5	14.3	0.9	0.4	0.5	4.4	24.4	63.8	2.6	2.4	1.3	1.1	1.7	4.0	0.9	91.6	1.4	0.1	0.1
$\sigma = 0.025$	FVSR	Full	Same	62.0	4.3	3.6	0.6	12.9	0.3	16.4	0.3	79.1	8.7	11.1	0.1	0.1	0.4	3.0	18.7	69.0	1.7	5.1	0.7	1.7	1.4	2.7	1.3	93.6	1.0	0.0	0.0
	PVSR	Partial	Same	54.3	7.4	10.0	0.9	15.1	0.0	12.3	2.3	59.7	6.7	28.3	1.0	0.6	1.4	4.4	31.6	53.6	5.1	3.1	1.0	1.1	3.0	6.1	1.3	87.0	2.0	0.4	0.1
	FVSR/PVSR	Average		58.1	5.9	6.8	0.7	14.0	0.1	14.4	1.3	69.4	7.7	19.7	0.6	0.4	0.9	3.7	25.1	61.3	3.4	4.1	0.9	1.4	2.2	4.4	1.3	90.3	1.5	0.2	0.1
$\sigma = 0.05$	FVSR	Full	Same	30.7	7.9	2.4	0.4	13.1	15.0	30.4	5.9	42.4	9.9	26.0	12.7	0.4	2.7	11.4	28.7	30.7	3.6	12.3	6.3	7.0	0.7	5.3	0.4	91.9	1.7	0.0	0.0
	PVSR	Partial	Same	25.4	12.4	5.2	0.3	21.1	12.8	22.9	5.3	30.9	4.0	43.7	13.0	0.6	2.6	11.3	32.9	25.9	5.3	14.5	2.6	7.5	2.4	6.7	1.0	85.9	3.3	0.7	0.0
	FVSR/PVSR	Average		28.0	10.1	3.8	0.4	17.1	13.9	26.7	5.6	36.6	6.9	34.9	12.9	0.5	2.6	11.4	30.8	28.3	4.4	13.4	4.4	7.3	1.6	6.0	0.7	88.9	2.5	0.4	0.0
$\sigma = 0.1$	FVSR	Full	Same	8.3	18.7	15.9	0.3	7.3	25.7	23.9	1.1	27.6	8.9	27.9	15.0	14.4	5.1	14.1	34.0	15.3	3.3	3.3	16.3	13.7	1.3	16.4	4.1	49.0	14.0	14.3	0.9
	PVSR	Partial	Same	8.1	25.1	17.7	1.4	4.4	24.7	18.4	0.7	20.7	11.7	33.7	13.3	14.9	5.0	15.9	29.1	17.1	6.9	7.9	15.6	7.6	1.4	14.7	2.3	61.3	5.9	13.9	0.6
	FVSR/PVSR	Average		8.2	21.9	16.8	0.9	5.9	25.2	21.1	0.9	24.1	10.3	30.8	14.1	14.6	5.1	15.0	31.6	16.2	5.1	5.6	15.9	10.6	1.4	15.6	3.2	55.1	9.9	14.1	0.7

Experiment	View	Resolution	Objects																												Recognition							
			Ball Valve-S							Butterfly Valve							3-Way Valve							Average Confusion														
			1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7								
$\sigma = 0$	FVSR	Full	Same	23.6	5.0	10.4	1.0	48.3	0.6	11.1	6.9	7.7	3.0	0.0	5.7	65.9	10.9	12.3	2.4	2.9	0.7	3.1	0.7	77.9	8.0	5.9	7.9	1.8	4.5	0.7	6.8							
	PVSR	Partial	Same	20.6	14.1	10.9	1.1	42.1	1.4	9.7	6.9	5.7	4.3	0.6	6.7	61.4	14.4	15.6	3.9	4.1	1.0	5.0	0.6	69.9	8.8	11.6	5.5	5.1	6.5	0.6	7.0							
	FVSR/PVSR	Average		22.1	9.6	10.6	1.1	45.2	1.0	10.4	6.9	6.7	3.6	0.3	6.2	63.6	12.6	13.9	3.1	3.5	0.9	4.1	0.6	73.9	8.4	8.8	6.7	3.5	5.5	0.7	6.9							
$\sigma = 0.00625$	FVSR	Full	Same	19.1	4.6	10.4	1.0	50.7	1.7	12.4	6.3	4.1	1.9	0.1	4.3	69.3	14.0	9.9	2.1	3.9	0.4	2.1	3.0	78.6	6.8	5.3	5.3	2.1	3.6	1.3	7.8							
	PVSR	Partial	Same	18.0	11.3	11.1	0.9	44.7	2.7	11.3	6.6	5.6	3.9	0.4	7.4	64.9	11.3	15.9	2.9	4.3	0.4	6.3	1.1	69.1	8.1	10.7	5.6	5.5	6.8	1.0	6.6							
	FVSR/PVSR	Average		18.6	7.9	10.8	0.9	47.7	2.2	11.9	6.4	4.9	2.9	0.3	5.9	67.1	12.6	12.9	2.5	4.1	0.4	4.2	2.1	73.9	7.4	8.0	5.4	3.8	5.2	1.2	7.2							
$\sigma = 0.0125$	FVSR	Full	Same	27.7	3.6	6.6	0.4	52.3	0.4	9.0	7.4	4.0	2.4	0.4	2.9	73.4	9.4	12.4	1.7	2.6	0.6	3.3	1.6	77.9	8.9	5.1	3.8	2.0	4.1	0.7	6.0							
	PVSR	Partial	Same	15.3	9.4	11.9	1.4	49.1	1.3	11.6	5.9	3.7	1.1	1.1	6.4	68.9	12.9	16.9	3.4	4.6	0.7	7.7	0.9	65.9	8.0	10.9	5.1	4.7	6.2	0.8	6.8							
	FVSR/PVSR	Average		21.5	6.5	9.2	0.9	50.7	0.9	10.3	6.6	3.9	1.8	0.8	4.6	71.1	11.1	14.6	2.6	3.6	0.6	5.5	1.2	71.9	8.5	8.0	4.5	3.4	5.1	0.8	6.4							
$\sigma = 0.025$	FVSR	Full	Same	35.7	5.3	8.9	0.9	39.7	0.3	9.3	11.1	5.6	1.9	0.4	3.7	62.4	14.9	13.1	2.0	3.9	0.9	3.7	0.6	75.9	10.8	6.4	4.7	2.6	4.4	0.3	7.1							
	PVSR	Partial	Same	25.3	12.1	14.0	1.9	35.9	1.1	9.7	12.7	5.3	3.3	0.4	6.0	54.9	17.4	18.9	3.4	4.0	0.0	4.3	0.3	69.1	11.1	11.0	6.5	6.1	5.3	0.6	7.0							
	FVSR/PVSR	Average		30.5	8.7	11.4	1.4	37.8	0.7	9.5	11.9	5.4	2.6	0.4	4.9	58.6	16.1	16.0	2.7	3.9	0.4	4.0	0.4	72.5	10.9	8.7	5.6	4.3	4.8	0.5	7.1							
$\sigma = 0.05$	FVSR	Full	Same	20.3	11.0	7.0	2.6	26.9	14.1	18.1	4.1	6.7	2.0	0.4	2.6	61.7	22.4	8.6	3.6	3.0	0.0	5.1	15.7	64.0	8.5	10.5	4.1	5.5	7.9	8.6	13.5							
	PVSR	Partial	Same	18.4	17.2	11.6	1.9	27.1	8.6	15.2	5.5	4.5	2.6	0.5	4.4	52.0	30.5	9.2	4.4	3.8	0.6	11.4	14.9	55.7	8.7	13.0	4.7	8.7	11.3	6.7	13.1							
	FVSR/PVSR	Average		19.3	14.1	9.3	2.3	27.0	11.4	16.7	4.8	5.6	2.3	0.5	3.5	56.9	26.5	8.9	4.0	3.4	0.3	8.3	15.3	59.9	8.6	11.8	4.4	7.1	9.6	7.6	13.3							
$\sigma = 0.1$	FVSR	Full	Same	11.1	27.9	15.3	0.3	7.7	21.3	16.4	1.3	8.1	6.6	1.3	1.0	50.9	30.9	3.1	8.0	11.1	0.1	1.4	28.6	47.6	5.4	18.9	10.3	5.5	7.0	20.1	15.1							
	PVSR	Partial	Same	13.3	29.3	16.4	1.9	7.6	18.6	13.0	0.9	10.0	5.0	1.3	1.3	49.1	32.4	3.6	11.4	8.6	0.1	2.4	27.1	46.7	6.0	20.0	10.3	7.5	5.9	19.1	12.8							
	FVSR/PVSR	Average		12.2	28.6	15.9	1.1	7.6	19.9	14.7	1.1	9.1	5.8	1.3	1.1	50.0	31.6	3.4	9.7	9.9	0.1	1.9	27.9	47.1	5.7	19.4	10.3	6.5	6.4	19.6	14.0							
			Average	8.2	10.8	6.2	4.8	6.1	5.1	9.1																												

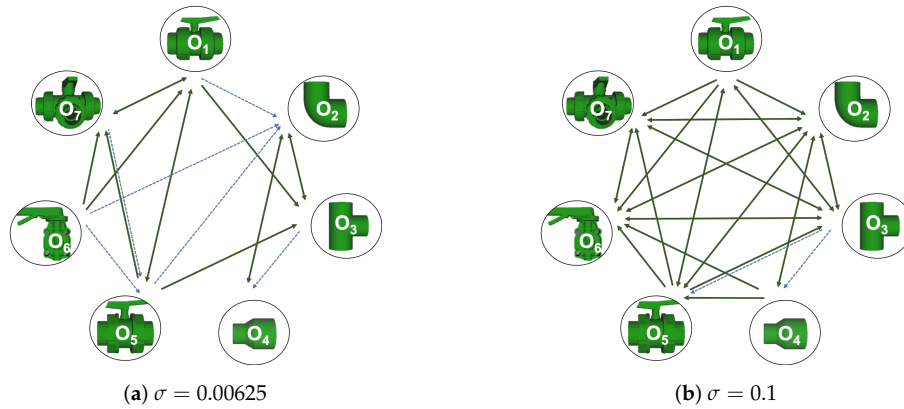


Figure 15. Object Confusion Graph.

7. Results on Underwater Testing

In this section we present experimental results and compare them with the previous simulations. The setting and the analytic process of the experiment are summarized in Figure 16. We took advantage of an already collected dataset which was previously used for semantic Simultaneous Localization And Mapping (SLAM) [73]. The data was collected using an in-house-developed laser scanner [28] mounted on GIRONA500 Autonomous Underwater Vehicle (AUV) [74], which was performing a trajectory in a small water tank. The experiment involved 25, 29, 48 and 48 observations of full view scans corresponding to the Ball-Valve, Elbow, R-Tee, and R-Socket objects respectively. Although the data-set only used 4 out of the 7 objects used in this survey, we think the results are representative.

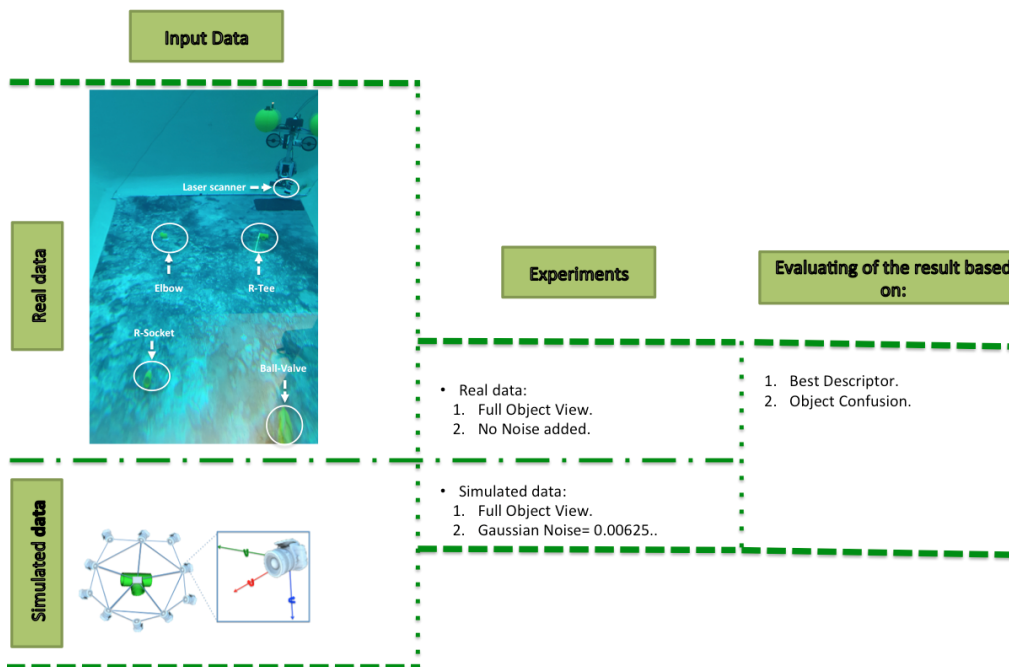


Figure 16. Diagram of the real versus simulated experiment. The upper left figure illustrates the experimental setup, with the Girona 500 AUV deploying an underwater laser scanner inside a water tank, where the four objects were laid on the bottom.

7.1. Real vs. Simulated Results

To compare the results obtained in the real experiment to the simulation results, a supplementary simulation was performed. The experiment involved 100 Montecarlo runs, as in the previous simulations, considering solely the four objects that were involved in the real experiment namely: Ball-Valve, Elbow, R-Tee, and R-Socket. The simulation parameters $\langle objects = 4, resolution = 0.007, noise = 0.00625, view = Full \rangle$ represent the case closest to the real data. The simulated scans were generated assuming a distance to the object $d = 1.11 \pm 0.56m$, and yaw&roll angles were varied between -0.4 and 7.4 degrees, while the pitch ranged between -2.2 and 3.9 degrees. The simulated values were chosen randomly within those ranges, corresponding to the ones observed in the real experiment.

Table 11, shows the corresponding percentages of how many times each different object class was recognized for both real and simulated runs, respectively, so they can be compared. Notice that the yellow cells represent the objects with their respective class number.

As expected, the recognition works better with the simulated scans than with the real ones. This is understandable, taking into account that real scans can be potentially affected by errors due to: (1) non-perfect scanner calibrations and (2) motion induced distortion. The latter may be significant, since the scanner works by steering a laser beam (which takes time to scan the scene) and assumes the sensor is static during the process (which is never the case since the robot is floating). To illustrate the problem we can look at Table 12. The top row shows a successful recognition example corresponding to the ESF descriptor. At the left, is shown a laser scan of a Ball-Valve (in black) and the corresponding object view (in red) matched in the data base. At the right, both histograms, the one corresponding to the scan and the one corresponding to the matched view show good agreement. On the other hand, the bottom row shows an example of mis-recognition of an R-Tee object using the GPFH descriptor. At the left, is shown the laser scan (in black), the matched object view in the data base (in red), and the most similar view in the database manually selected by us (in blue). The corresponding histograms are shown at the right side of the bottom row. Although we perceive the black scan to be closer to the blue view than to the red one, the difference is evaluated quantitatively by the corresponding histograms, and it is clear that the black histogram is closer to the red one than to the blue. It is worth noting the distortion present in the black scan, which is probably the origin of the mis-recognition.

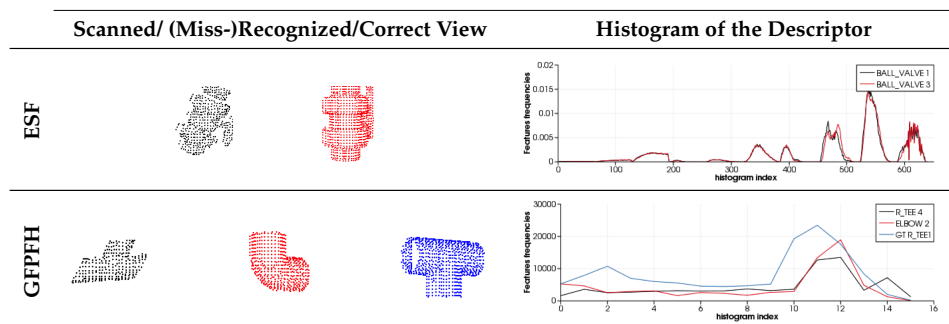
7.1.1. Best Descriptor

Table 11 shows the average recognition rate for the real and the synthetic experiments, highlighting in green the two better performing descriptors and in red the two worst ones. In the real experiments, the best performing descriptor is CVFH (95.4%) followed by OUR-CVFH (91.5%). This result is in agreement with the simulated one (both at 100%). The worst one is GPFH (54.1%) followed by GOOD (58.0%) in the real experiment, with GOOD being (75.0%) and VFH (88.0%) in the simulated ones. We think that this disagreement is due to the fact that our results for GPFH differ significantly between reality (54.1%) and simulation (96.8%), probably affected by problems like the one commented above which is illustrated in Table 12.

Table 11. Confusion Matrix for the real and synthetic data, for all descriptors, represented in a table.

Descriptors	Experiment	Objects																Average of recognition	
		Ball Valve				Elbow				R-Tee				R-Socket				Real	Synthetic
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4		
ESF	Real	100	0	0	0	0	76	24	0	8	0	92	0	21	2	0	77	86,2	
	Synthetic	100	0	0	0	1	99	0	0	0	0	100	0	0	0	0	100		99,8
VFH	Real	100	0	0	0	31	41	21	7	4	6	90	0	19	2	0	60	72,8	
	Synthetic	100	0	0	0	0	100	0	0	0	0	100	0	0	48	0	52		88,0
CVFH	Real	100	0	0	0	0	93	3	3	0	4	95	0	0	6	0	94	95,4	
	Synthetic	100	0	0	0	0	100	0	0	0	0	100	0	0	0	0	100		100
OURCVFH	Real	100	0	0	0	0	83	0	17	0	0	96	4	0	13	0	88	91,5	
	Synthetic	100	0	0	0	0	100	0	0	0	0	100	0	0	0	0	100		100
GOOD	Real	100	0	0	0	34	66	0	0	100	0	0	0	33	0	0	67	58,0	
	Synthetic	100	0	0	0	0	100	0	0	100	0	0	0	0	0	0	100		75,0
GFPFH	Real	60	0	0	40	0	48	0	52	4	73	23	0	0	15	0	85	54,1	
	Synthetic	100	0	0	0	0	88	12	0	1	0	99	0	0	0	0	100		96,8
GRSD	Real	88	0	12	0	0	76	0	24	0	33	67	0	0	2	0	98	82,1	
	Synthetic	100	0	0	0	0	98	2	0	0	0	100	0	0	0	0	100		99,5
Average Real		92,6	0,0	1,7	5,7	9,3	69,0	6,9	14,8	16,7	16,6	65,9	0,6	10,4	5,6	0,0	81,2		

Table 12. Example of the performance of the two descriptors ESF and GFPFH: (First column) scanned view with the corresponding model view and the correct view in case of mis-recognition; (Second column) visualisation of the corresponding histograms for each descriptor.



7.1.2. Object Confusion

Figure 17 and Table 11 show the confusion matrices for the real and the synthetic experiments. First it is worth noting that, as expected, real experiments lead to more confusion than synthetic ones. Second, focusing on the real results, it can be observed that in general (averaging over all descriptors), the Ball-Valve is the most easily recognizable object, while the R-Tee and the elbow were the objects leading to more confusion (average recognition of 65.9% and 69% respectively). Nevertheless, focusing on CVFH and OUR-CVFH (the best performing descriptors), we can see that the first one does an excellent job having only one confusion beyond the 5% boundary (R-Socket confused with Elbow), while the second one adds a second confusion beyond the 5% limit (Elbow confused with socket). Finally, it called our attention to the experimental and simulated results corresponding to the GOOD descriptor and the R-Tee object. In Table 11, it can be appreciated how it fails to recognize the object (0%) and confuses it (100%) systematically with the Ball-Valve. We attribute this result to the fact that GOOD works based on the orthographic projection on XoZ, XoY, and YoZ. Checking the object-database (Figure 2) it can be observed that there is no R-Tee view corresponding to the top view observed by the laser in the real experiment, while there are two views of the Ball-Valve in the database which projected

onto XoZ, XoY, and YoZ look similar to the top view of the R-Tee Object. This problem illustrates how important it is to have representative views in the database of those objects which will be observed. Please note that this problem did not arise in the [FVSR](#) and [PVSR](#) experiments, since in those cases, the scans were not forced to be taken from the top as happens in the water-tank experiment where a downward-looking laser scanner was used.

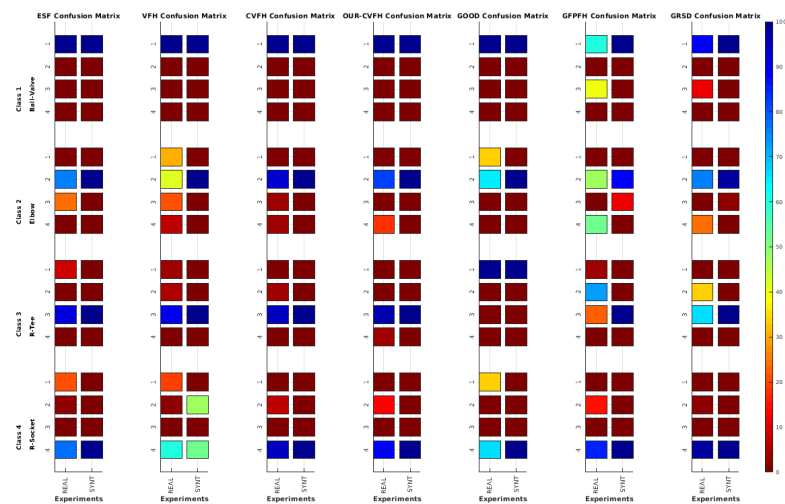


Figure 17. Confusion Matrix for the real and synthetic data, for all descriptors, where the first column corresponds to real data and the second to synthetic data.

This could explain the divergence of the simulated and real results reported for some of the entries of Table 11.

8. Interpretation of Results

This section provides further interpretation of the results of all the experiments performed in this study.

8.1. Simulated Data

The results detailed in Section 6, obtained using simulated data, are summarized in Figure 18.

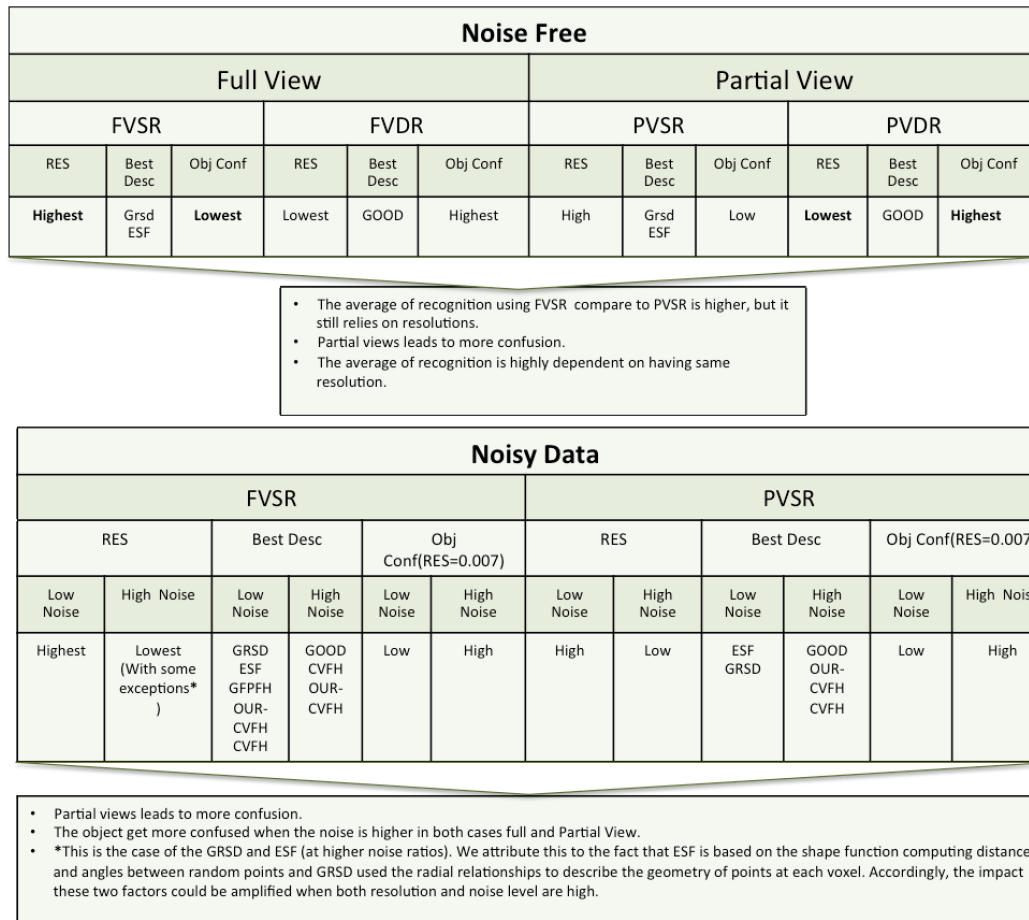


Figure 18. Summary of the result section for the simulated data. Res, Best Desc and Obj Conf represent respectively: Resolution, Best Descriptor and Object Confusion

The following points contain the main findings from the experiments, under testing conditions involving the use of: full or partial views, same or different resolution of the scan and the database object models, and full or partial views under added noise.

1. Full View Same Resolution and Partial View Same Resolution Experiment
The best performance in both cases was achieved using ESF and GRSD. The average of recognition is slightly better using the full view than the partial one, where the trend of the recognition accuracy with respect to the descriptors was monotonic in both cases.
2. Full View Different Resolution and Partial View different Resolution Experiment
Only the GOOD descriptor provided significantly valid results. The change of resolution did not affect the performance of the descriptor in either of the scenarios (FVSR and PVSR). This invariance of the performance regarding the resolution was reported in the original work of the authors in [68].
3. Same resolution versus different Resolution Experiment
From the cases of FVDR and PVDR, changing the resolution in the database and test led to poor recognition. Conversely, the cases of FVSR and PVSR, show that having the same resolution in both the database and test leads to higher recognition rates.
4. Full Views versus Partial Views
Predictably, using full object views instead of just a partial view, leads to better results.

This behaviour is somehow expectable, given the global nature of the methods tested, and the fact that the descriptors in the database were always computed from full views. Additionally, from the confusion matrices, the objects that are prone to confusion when using partial views are a superset of the ones for the case of using full views.

5. Noisy Full View versus Noisy Partial View

In these experiments only the case where the resolution of the database and test are similar were taking into account. As general assessment, the results of NFVSR versus NPVSR follow the same trend as discussed in the noise-free experiment, where the performance of the descriptor decreases with lower resolution and higher noise ratio, except for GRSD and ESF where the performance of the descriptors decreased for high resolution and noise. We attribute this difference to the fact that ESF is based on the shape function computing distances and angles between random points, while GRSD used the radial relationships to describe the geometry of points at each voxel. Accordingly, the impact of these two factors could be amplified when both resolution and noise level are high.

As a specific assessment, the confusion matrix for the $resolution = 0.007$, which is the resolution of the laser scan used in the real experiment, was computed at a different noise level. The results showed that the object got more confusion at a high noise level compared to a low level.

8.2. Underwater Data

The figure below (Figure 19) summarizes the results based on real underwater versus simulated data .

Underwater Data			Simulated Data		
Average of Recognition	Best Desc	Obj Conf	Average of Recognition	Best Desc	Obj Conf
Good	CVFH OUR-CVFH	Low	Highest	CVFH OUR-CVFH	Lowest

- The average recognition using simulated data is higher compare to using real data.
- Real data got a higher confusion compared to the simulated.

Figure 19. Summary of the results section for real versus simulated data.

From the results presented in Section 7, it is worth noting that CVFH and OUR-CVFH were the two best performing descriptors in both real and simulated data. Also that the recognition based on the descriptors GFPGH and GOOD gave slightly different values when using the real and the simulated data. These differences can be explained by several factors:

- Real data inevitably suffers from noise generated from the changes of the position of the laser during the acquisition of the point cloud, causing a distortion of the object shape and leading to a different descriptor representation. These motion distortions were present in real but not in simulated data.
- Most of the object descriptors used in this study are based on use of a surface normals. Noise causes a modification in the surface which causes a change in the normal for each point.

9. Conclusions

This paper presented a survey and comparison of global descriptors for 3D object recognition purposes when a 3D model of the object is available a priori. Because our focus of interest is centered in underwater IMR applications, we selected seven representative objects commonly present in submerged pipe infrastructures. Using their CAD models, we set up a database containing 12 views

of each object. Next, seven global descriptors available in the Point Cloud Library were selected and compared exhaustively in simulations and through water tank experiments. Different criteria were evaluated: (1) the use of partial vs. global views, (2) the use of same vs. different resolution between the object model and the input scan, (3) the effect of resolution and (4) the effect of noise.

Our results demonstrate that, as intuition suggests, using global views provides better results than using partial views. Less intuitive is the conclusion that using the same resolution in the views of the database and the input scan leads to significantly better results. The combination of both cases is therefore the best scenario: full view/same resolution. When the resolution of the scan is analyzed, in general, for most descriptors the higher the resolution the better the recognition rates. Hence decreasing the resolution leads to a decrease in the performance, with the exception of the GOOD descriptor whose performance remains constant over the studied resolutions.

Another parameter studied was the noise. In this case, the results follow intuition and the higher the noise the worse the recognition rate and the higher the object confusion. The exception is again the GOOD descriptor which is the best one for high levels of noise.

It is not straightforward to single out the best performing descriptor, since this depends on the particular combination of the different parameters studied. Therefore, the numerous graphs provided for each one of them may help other researchers to make their own decisions based on the particular constraints of their own application.

10. Future Work

A central goal of our work is the use of a real-time laser scanner mounted on an intervention AUV to detect, identify and locate objects in the robot's surroundings, and to use this information to allow the robot to decide which manipulation actions may be performed on each type of object. Therefore our next step is going to focus on implementing a method to recognize objects within a point cloud which may contain several of them. This will require a method to segment the different objects so that they can be recognized later on. Once recognized they will then be located and introduced into a SLAM algorithm to set-up a semantic map of the robot environment. As an example, Figure 20 illustrates a test structure containing multiple object instances that is currently being used for this purpose.



Figure 20. Structure of PVC objects.

Author Contributions: Conceptualization, K.H. and P.R.; Investigation, K.H.; Methodology, K.H., N.G. and P.R.; Software, K.H. ; Supervision, P.R. and N.G.; Writing, K.H, P.R. and N.G.

Funding: This work was supported by the Spanish Government through a FPI Ph.D. grant to K. Himri, as well as by the Spanish Project DPI2017-86372-C3-2-R (TWINBOT-GIRONA1000) and the H2020-INFRAIA-2017-1-twostage-731103 (EUMR).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rusu, R.B.; Marton, Z.C.; Blodow, N.; Dolha, M.; Beetz, M. Towards 3D point cloud based object maps for household environments. *Robot. Auton. Syst.* **2008**, *56*, 927–941.
2. Rusu, R.B.; Gerkey, B.; Beetz, M. Robots in the kitchen: Exploiting ubiquitous sensing and actuation. *Robot. Auton. Syst.* **2008**, *56*, 844–856.
3. Blodow, N.; Goron, L.C.; Marton, Z.C.; Pangercic, D.; Rühr, T.; Tenorth, M.; Beetz, M. Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 4263–4270.
4. Houser, K. A Robot Is Learning to Cook and Clean in an Ikea Kitchen. 2019. Available online: <https://futurism.com/the-byte/robots-cook-clean-ikea-kitchen> (accessed on 09 May 2019)
5. Saxena, A.; Wong, L.; Quigley, M.; Ng, A.Y. A vision-based system for grasping novel objects in cluttered environments. In *Robotics Research*; Springer: Berlin, Heidelberg, 2010; pp. 337–348.
6. Zhu, M.; Derpanis, K.G.; Yang, Y.; Brahmabhatt, S.; Zhang, M.; Phillips, C.; Lecce, M.; Daniilidis, K. Single image 3D object detection and pose estimation for grasping. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 3936–3943.
7. Farahmand, F.; Pourazad, M.T.; Moussavi, Z. An intelligent assistive robotic manipulator. In Proceedings of the IEEE Engineering in Medicine and Biology 27th Annual Conference, Shanghai, China, 17–18 January 2006; pp. 5028–5031.
8. Boronat Roselló, E. ROSAPL: Towards a Heterogeneous Multi-Robot System and Human Interaction Framework. Master's Thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, 2014.
9. Vargas, J.A.C.; Garcia, A.G.; Oprea, S.; Escolano, S.O.; Rodriguez, J.G. Detecting and manipulating objects with a social robot: An ambient assisted living approach. In *ROBOT 2017: Third Iberian Robotics Conference*; Springer: Cham, Switzerland, 2017; pp. 613–624.
10. Vargas, J.A.C.; Garcia, A.G.; Oprea, S.; Escolano, S.O.; Rodriguez, J.G. Object recognition pipeline: Grasping in domestic environments. In *Rapid Automation: Concepts, Methodologies, Tools, and Applications*; IGI Global: Hershey, PA, USA, 2019; pp. 456–468.
11. Bontsema, J.; Hemming, J.; Pekkeriet, E. CROPS: High tech agricultural robots. In Proceedings of the International Conference of Agricultural Engineering AgEng 2014, Zurich, Switzerland, 6–10 July 2014.
12. Tao, Y.; Zhou, J. Automatic apple recognition based on the fusion of color and 3D feature for robotic fruit picking. *Comput. Electr. Agric.* **2017**, *142*, 388–396.
13. Huang, J.; You, S. Detecting objects in scene point cloud: A combinational approach. In Proceedings of the International Conference on 3D Vision, 3DV '13 2013, Seattle, WA, USA, 29 June–1 July 2013; pp. 175–182, doi:10.1109/3DV.2013.31.
14. Holz, D.; Behnke, S. Fast edge-based detection and localization of transport boxes and pallets in rgb-d images for mobile robot bin picking. In Proceedings of the ISR 2016: 47st International Symposium on Robotics. VDE, Munich, Germany, 21–22 June 2016; pp. 1–8.
15. Geronimo, D.; Lopez, A.M.; Sappa, A.D.; Graf, T. Survey of pedestrian detection for advanced driver assistance systems. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *32*, 1239–1258.
16. Fu, M.Y.; Huang, Y.S. A survey of traffic sign recognition. In Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition, Qingdao, China, 11–14 July 2010; pp. 119–124.
17. Fan, Y.; Zhang, W. Traffic sign detection and classification for Advanced Driver Assistant Systems. In Proceedings of the 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Zhangjiajie, China, 15–17 August 2015; pp. 1335–1339.
18. Markiewicz, P.; Długosz, M.; Skruch, P. Review of tracking and object detection systems for advanced driver assistance and autonomous driving applications with focus on vulnerable road users sensing. In *Polish Control Conference*; Springer: Cham, Switzerland, 2017; pp. 224–237.
19. Foresti, G.L.; Gentili, S. A hierarchical classification system for object recognition in underwater environments. *IEEE J. Ocean. Eng.* **2002**, *27*, 66–78.

20. Bagnitsky, A.; Inzartsev, A.; Pavin, A.; Melman, S.; Morozov, M. Side scan sonar using for underwater cables & pipelines tracking by means of AUV. In Proceedings of the IEEE Symposium on Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies, Tokyo, Japan, 5–8 April 2011; pp. 1–10.
21. Yu, S.C.; Kim, T.W.; Asada, A.; Weatherwax, S.; Collins, B.; Yuh, J. Development of high-resolution acoustic camera based real-time object recognition system by using autonomous underwater vehicles. In Proceedings of the OCEANS 2006, Boston, MA, USA, 18–21 September 2006.
22. Hegde, J.; Utne, I.B.; Schjøberg, I. Applicability of current remotely operated vehicle standards and guidelines to autonomous subsea IMR operations. In Proceedings of the ASME 34th International Conference on Ocean, Offshore and Arctic Engineering, 31 May–5 June 2015 June, St. John's, NL, Canada; American Society of Mechanical Engineers: New York, NY, USA, 2015; doi:10.1115/OMAE2015-4162.
23. Barat, C.; Rendas, M.J. A robust visual attention system for detecting manufactured objects in underwater video. In Proceedings of the OCEANS 2006, Boston, MA, USA, 18–21 September 2006.
24. Gordan, M.; Dancea, O.; Stoian, I.; Georgakis, A.; Tsatos, O. A new SVM-based architecture for object recognition in color underwater images with classification refinement by shape descriptors. In Proceedings of the IEEE International Conference on Automation, Quality and Testing, Robotics, Cluj-Napoca, Romania, 25–28 May 2006; Volume 2, pp. 327–332.
25. Sivčev, S.; Rossi, M.; Coleman, J.; Omerdić, E.; Dooly, G.; Toal, D. Collision detection for underwater ROV manipulator systems. *Sensors* **2018**, *18*, 1117.
26. Gobi, A.F. Towards generalized benthic species recognition and quantification using computer vision. In Proceedings of the Fourth Pacific-Rim Symposium on Image and Video Technology, Singapore, Singapore, 14–17 November 2010; pp. 94–100.
27. Rusu, R.B.; Cousins, S. 3D is here: Point cloud library (PCL). In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 1–4.
28. Palomer, A.; Ridaou, P.; Ribas, D.; Forest, J. Underwater 3D laser scanners: The deformation of the plane. In *Lecture Notes in Control and Information Sciences*; Fossen, T.L., Pettersen, K.Y., Nijmeijer, H., Eds.; Springer: Cham, Switzerland, 2017; Volume 474, pp. 73–88, doi:10.1007/978-3-319-55372-6_4.
29. Pang, G.; Qiu, R.; Huang, J.; You, S.; Neumann, U. Automatic 3D industrial point cloud modeling and recognition. In Proceedings of the 14th IAPR international conference on machine vision applications (MVA), Tokyo, Japan, 18–22 May 2015; pp. 22–25.
30. Kumar, G.; Patil, A.; Patil, R.; Park, S.; Chai, Y. A LiDAR and IMU integrated indoor navigation system for UAVs and its application in real-time pipeline classification. *Sensors* **2017**, *17*, 1268.
31. Alexandre, L.A. 3D descriptors for object and category recognition: a comparative evaluation. In Proceedings of the Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal, 7–12 October 2012; Volume 1, p. 7.
32. Johnson, A.E.; Hebert, M. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 433–449.
33. Rusu, R.B.; Holzbach, A.; Beetz, M.; Bradski, G. Detecting and segmenting objects for mobile manipulation. In Proceedings of the IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, Kyoto, Japan, 27 September–4 October 2009; pp. 47–54.
34. Aldoma, A.; Vincze, M.; Blodow, N.; Gossow, D.; Gedikli, S.; Rusu, R.B.; Bradski, G. CAD-model recognition and 6DOF pose estimation using 3D cues. In Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 6–13 November 2011; pp. 585–592.
35. Lai, K.; Bo, L.; Ren, X.; Fox, D. A large-scale hierarchical multi-view rgb-d object dataset. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 1817–1824.
36. Tombari, F.; Salti, S.; Di Stefano, L. A combined texture-shape descriptor for enhanced 3D feature matching. In Proceedings of the 18th IEEE International Conference on Image Processing, Brussels, Belgium, 11–14 September 2011; pp. 809–812.
37. Guo, Y.; Bennamoun, M.; Sohel, F.; Lu, M.; Wan, J. 3D object recognition in cluttered scenes with local surface features: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2270–2287.
38. Guo, Y.; Bennamoun, M.; Sohel, F.; Lu, M.; Wan, J.; Kwok, N.M. A comprehensive performance evaluation of 3D local feature descriptors. *Int. J. Comput. Vis.* **2016**, *116*, 66–89.

39. Tombari, F.; Salti, S.; Di Stefano, L. Unique shape context for 3D data description. In Proceedings of the ACM Workshop on 3D Object Retrieval, Firenze, Italy, 25 October 2010; ACM: New York, NY, USA, 2010; pp. 57–62.
40. Mian, A.S.; Bennamoun, M.; Owens, R. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 1584–1601.
41. Taati, B.; Greenspan, M. Local shape descriptor selection for object recognition in range data. *Comput. Vis. Image Underst.* **2011**, *115*, 681–694.
42. Rodolà, E.; Albarelli, A.; Bergamasco, F.; Torsello, A. A scale independent selection process for 3d object recognition in cluttered scenes. *Int. J. Comput. Vis.* **2013**, *102*, 129–145.
43. Jørgensen, T.B.; Buch, A.G.; Kraft, D. Geometric edge description and classification in point cloud data with application to 3D object recognition. In Proceedings of the 10th International Conference on Computer Vision Theory and Applications International Conference on Computer Vision Theory and Applications. Institute for Systems and Technologies of Information, Control and Communication, Berlin, Germany, 11–14 March 2015; pp. 333–340.
44. Yang, J.; Zhang, Q.; Xian, K.; Xiao, Y.; Cao, Z. Rotational contour signatures for robust local surface description. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3598–3602.
45. Malassiotis, S.; Strintzis, M.G. Snapshots: A novel local surface descriptor and matching algorithm for robust 3D surface alignment. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1285–1290.
46. Rusu, R.B.; Blodow, N.; Beetz, M. Fast point feature histograms (FPFH) for 3D registration. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'09, Kobe, Japan, 12–17 May 2009; pp. 3212–3217.
47. Salti, S.; Tombari, F.; Di Stefano, L. SHOT: Unique signatures of histograms for surface and texture description. *Comput. Vis. Image Underst.* **2014**, *125*, 251–264.
48. Guo, Y.; Sohel, F.; Bennamoun, M.; Lu, M.; Wan, J. Rotational projection statistics for 3D local surface description and object recognition. *Int. J. Comput. Vis.* **2013**, *105*, 63–86.
49. Shi, Z.; Kang, Z.; Lin, Y.; Liu, Y.; Chen, W. Automatic recognition of pole-like objects from mobile laser scanning point clouds. *Remote Sens.* **2018**, *10*, 1891, doi:10.3390/rs10121891.
50. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395.
51. Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52.
52. Chen, F.; Selvaggio, M.; Caldwell, D.G. Dexterous grasping by manipulability selection for mobile manipulator with visual guidance. *IEEE Trans. Ind. Inform.* **2019**, *15*, 1202–1210.
53. Rusu, R.B.; Bradski, G.; Thibaux, R.; Hsu, J. Fast 3D recognition and pose using the viewpoint feature histogram. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 18–22 October 2010; pp. 2155–2162.
54. Marton, Z.C.; Pangercic, D.; Rusu, R.B.; Holzbach, A.; Beetz, M. Hierarchical object geometric categorization and appearance classification for mobile manipulation. In Proceedings of the 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids), Nashville, TN, USA, 6–8 December 2010; pp. 365–370.
55. Marton, Z.C.; Pangercic, D.; Blodow, N.; Kleinhellefort, J.; Beetz, M. General 3D modelling of novel objects from a single view. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 3700–3705.
56. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359.
57. Gunji, N.; Niigaki, H.; Tsutsuguchi, K.; Kurozumi, T.; Kinebuchi, T. 3D object recognition from large-scale point clouds with global descriptor and sliding window. In Proceedings of the 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 721–726.
58. Tombari, F.; Di Stefano, L. Object recognition in 3D scenes with occlusions and clutter by hough voting. In Proceedings of the Fourth Pacific-Rim Symposium on Image and Video Technology, Singapore, Singapore, 14–17 November 2010; pp. 349–355.
59. Garstka, J. Learning Strategies to Select Point Cloud Descriptors for Large-Scale 3-D Object Classification. Ph.D. Thesis, University of Hagen, Hagen, Germany, 2016.

60. Jain, S.; Argall, B. Estimation of Surface Geometries in Point Clouds for the Manipulation of Novel Household Objects. In Proceedings of the RSS 2017 Workshop on Spatial-Semantic Representations in Robotics, Cambridge, MA, USA, 16 July 2017.
61. Aldoma, A.; Tombari, F.; Prankl, J.; Richtsfeld, A.; Di Stefano, L.; Vincze, M. Multimodal cue integration through hypotheses verification for rgb-d object recognition and 6dof pose estimation. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 2104–2111.
62. Aldoma, A.; Tombari, F.; Rusu, R.B.; Vincze, M. OUR-CVfH-oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6DOF pose estimation. In *Joint DAGM (German Association for Pattern Recognition) and OAGM Symposium*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 113–122.
63. Alhamzi, K.; Elmogy, M.; Barakat, S. 3d object recognition based on local and global features using point cloud library. *Int. J. Adv. Comput. Technol.* **2015**, *7*, 43.
64. Wohlkinger, W.; Vincze, M. Ensemble of shape functions for 3d object classification. In Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), Karon Beach, Phuket, Thailand, 7–11 December 2011; pp. 2987–2992.
65. Rusu, R.B.; Marton, Z.C.; Blodow, N.; Beetz, M. Persistent point feature histograms for 3D point clouds. In Proceedings of the 10th International Conference on Intel Autonomous System (IAS-10), Baden-Baden, Germany, 2008; pp. 119–128.
66. Aldoma, A.; Fäulhammer, T.; Vincze, M. Automation of “ground truth” annotation for multi-view RGB-D object instance recognition datasets. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 5016–5023.
67. Sels, S.; Ribbens, B.; Bogaerts, B.; Peeters, J.; Vanlanduit, S. 3D model assisted fully automated scanning laser Doppler vibrometer measurements. *Opt. Lasers Eng.* **2017**, *99*, 23–30.
68. Kasaei, S.H.; Lopes, L.S.; Tomé, A.M.; Oliveira, M. An orthographic descriptor for 3D object learning and recognition. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 4158–4163.
69. Osada, R.; Funkhouser, T.; Chazelle, B.; Dobkin, D. Matching 3D models with shape distributions. In Proceedings of the SMI 2001 International Conference On Shape Modeling and Applications, Genova, Italy, 7–11 May 2001; pp. 154–166.
70. McCallum, A. Efficiently inducing features of conditional random fields. In Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence, Acapulco, Mexico, 7–10 August 2003; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2003; pp. 403–410.
71. Rusu, R.B.; Blodow, N.; Marton, Z.C.; Beetz, M. Close-range scene segmentation and reconstruction of 3D point cloud maps for mobile manipulation in domestic environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009, St. Louis, MO, USA, 10–15 October 2009; pp. 1–6.
72. Hetzel, G.; Leibe, B.; Levi, P.; Schiele, B. 3D object recognition from range images using local feature histograms. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, Kauai, HI, USA, 8–14 December 2001; Volume 2, p. II.
73. Himri, K.; Ridao, P.; Gracias, N.; Palomer, A.; Palomeras, N.; Pi, R. Semantic SLAM for an AUV using object recognition from point clouds. *IFAC-PapersOnLine* **2018**, *51*, 360–365.
74. Ribas, D.; Palomeras, N.; Ridao, P.; Carreras, M.; Mallios, A. Girona 500 AUV: From survey to intervention. *IEEE/ASME Trans. Mechatron.* **2012**, *17*, 46–53.



3

UNDERWATER OBJECT RECOGNITION USING POINT-FEATURES, BAYESIAN ESTIMATION AND SEMANTIC INFORMATION

IN this chapter, a 3D object recognition method for uncolored point clouds using point features is proposed for inspection, maintenance, and repair (IMR) of underwater industrial structures consisting of pipes and connecting objects. The objectives of this work were twofold: First, methods for pre-processing point cloud data were developed to facilitate the recognition task. These methods include plane and pipe detection, semantic segmentation, and object tracking based on the Joint Compatibility Branch and Bound(JCBB) algorithm. The JCBB-based tracking is intended to correct the effects of inconsistencies in the robot navigation, which prevented the tracking of the objects along with the scans. The second objective is to compare three established methods, namely descriptor-based, Bayesian-based and semantic-based recognition. All the proposed works have been described in detail and published in the following journal.:

Title: Underwater Object Recognition Using Point-Features, Bayesian Estimation and Semantic Information

Authors: **K. Himri**, P. Ridao, and N. Gracias

Journal: Sensors

Volume: 21, Number: 5, Published: 2021

DOI: 10.3390/s21051807

Quality index: JCR2019 Instruments & Instrumentation IF 3.275, Q1 (15/64)



Article

Underwater Object Recognition Using Point-Features, Bayesian Estimation and Semantic Information

Khadidja Himri *, Pere Ridao * and Nuno Gracias *

Underwater Robotics Research Center (CIRS), Computer Vision and Robotics Institute (VICOROB),
University of Girona, Parc Científic i Tecnològic UdG C/Pic de Peguera 13, 17003 Girona, Spain

* Correspondence: khadidja.himri@udg.edu (K.H.); pere@eia.udg.edu (P.R.); ngracias@silver.udg.edu (N.G.)

Abstract: This paper proposes a 3D object recognition method for non-coloured point clouds using point features. The method is intended for application scenarios such as Inspection, Maintenance and Repair (IMR) of industrial sub-sea structures composed of pipes and connecting objects (such as valves, elbows and R-Tee connectors). The recognition algorithm uses a database of partial views of the objects, stored as point clouds, which is available *a priori*. The recognition pipeline has 5 stages: (1) Plane segmentation, (2) Pipe detection, (3) Semantic Object-segmentation and detection, (4) Feature based Object Recognition and (5) Bayesian estimation. To apply the Bayesian estimation, an object tracking method based on a new Interdistance Joint Compatibility Branch and Bound (IJCBB) algorithm is proposed. The paper studies the recognition performance depending on: (1) the point feature descriptor used, (2) the use (or not) of Bayesian estimation and (3) the inclusion of semantic information about the objects connections. The methods are tested using an experimental dataset containing laser scans and Autonomous Underwater Vehicle (AUV) navigation data. The best results are obtained using the Clustered Viewpoint Feature Histogram (CVFH) descriptor, achieving recognition rates of 51.2%, 68.6% and 90%, respectively, clearly showing the advantages of using the Bayesian estimation (18% increase) and the inclusion of semantic information (21% further increase).

Keywords: 3D object recognition; point clouds; global descriptors; semantic segmentation; semantic information; Bayesian probabilities; laser scanner; underwater environment; pipeline detection; inspection; maintenance and repair; AUV; autonomous manipulation; multi-object tracking; JCBB



Citation: Himri, K.; Ridao, P.; Gracias, N. Underwater Object Recognition Using Point-Features, Bayesian Estimation and Semantic Information. *Sensors* **2021**, *21*, 1807. <https://doi.org/10.3390/s21051807>

Academic Editor: Nikolaos Doulamis

Received: 4 February 2021
Accepted: 23 February 2021
Published: 5 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the recent developments in the robotics industry there has been an increasing use of vehicle-mounted sensors. These sensors seek to provide useful information to the user, such as a clear perception of the environment, or provide more specific details such as obstacles to be avoided or objects to interact with. The outputs of these different sensors lead to different representations of the environment, depending on the sensor used and the task to be accomplished.

Previous work on methods for collecting and interpreting spatial data for mobile robotics could be broadly divided into three main categories. The first focuses prominently on data providing a 2D representation of the environment, such as images from cameras. The second relies on 3D point cloud data from sensors like laser scanners or acoustic ranging. The third uses hybrid data, either combining data from two different sensors or using a composite sensor such as the Microsoft Kinect that provides both images and point clouds. Over the last decade 3D point clouds have been widely used in computer vision and mobile robotics applications, opening the door to important but challenging tasks such as 3D object recognition [1–6] and semantic segmentation [7–9], which are core steps for scene understanding.

Understanding scenes and being able to navigate while detecting objects of interest is a fundamental task for self-driving vehicles and autonomous robots. To navigate an

environment, the robot needs to build a representation of the content of the scene that encapsulates the location of objects of interest within the environment.

In this line of research, the combined use of 3D object recognition and semantics has contributed to the development of better approaches to scene understanding. In the last decade various methods based on point clouds have been proposed, aiming to solve semantic segmentation. Semantic segmentation [10–12] can be broadly defined as the task of grouping parts of the input data, which can be 2D or 3D images or even 3D point clouds, which belong to the same object class, thus classifying each pixel or 3D point in the input according to a category.

Most of the recent methods deploy deep learning techniques while considering object models as black boxes. This trend is highlighted in the survey published by Guo et al. [13] on recent work on deep learning methods for point clouds, including semantic segmentation. Their survey reviews the most relevant applications for point cloud understanding, within the topics of 3D shape classification, 3D object detection and tracking and 3D point cloud segmentation. A review of state-of-the-art Deep Learning methods is presented using various publicly available datasets.

Semantic segmentation was inspired by the success of Deep Learning methods in producing an accurate result [10,13,14], but these techniques require an extremely large amount of data to train the network. Such large datasets may be difficult to obtain, or not provide adequate information, such as the case of man-made structures captured by sensors that only provide colourless point clouds.

3D object recognition based on point clouds has been studied across various disciplines, with an emphasis on deep neural network based approaches and feature point based methods. Relevant research in this area has been summarized and organized in various survey, using global and local methods [3,15]. Global recognition methods describe the entire object as a single vector of values, whereas local recognition methods are more focused on local regions and are only based on salient points.

Accurate and efficient algorithms for segmentation and recognition are required for the emerging Inspection, Maintenance and Repair (IMR) applications, especially given the recent advances in laser scanning technology. An example of critical application scenarios, that are attracting increasing research interest, are construction sites such as refineries which have extensive networks of industrial pipelines, that need frequent inspection and intervention.

Research in segmentation and recognition for pipeline sites has been conducted by Huang et al. [16] and Pang et al. [17], where a complex pipeline structure is partitioned and modeled as a set of interconnected parts using a Support Vector Machine (SVM)-based approach and a single local feature descriptor. Another notable application to pipeline classification is the work of Kumar et al. [18], in which an aerial vehicle equipped with a low-cost Light Detection and Ranging (LIDAR) is able to map and identify pipes of different lengths and radii. Ramon et al. [19] proposed a visual algorithm based on a semantic Convolutional Neural Networks (CNN) to detect pipes. The authors presented an approach based on a drone capable of autonomously landing on pipes, for inspection and maintenance in industrial environments. More recently, Kim et al. [20] presented an automatic pipe-elbow detection system in which pipes and elbows were recognized directly from laser-scanned points. The methods they used are based on curvature information and CNN-based primitive classification.

Regarding marine applications, the use of vision sensors underwater is becoming widespread. However, these sensors impose strong requirements related to water turbidity and the presence of light, to capture high quality images. Since the underwater images are subjected to rapid attenuation and scattering of light, object detection and recognition can only be performed at very short distances from objects, of the order of a few meters. Acoustic propagation allows much longer ranges in terms of sensing distance, but the object representations obtained are much too noisy and coarse in resolution to allow accurate object identification and localization for autonomous object grasping.

A comparatively small number of object recognition applications have been reported underwater. These include pipeline identification and inspections based on optical images in seabed survey operations [21], cable identification and pipeline tracking based on acoustic images [22] and recognition of different geometric shapes such as cylinders and cubes [23] using acoustic imaging cameras.

Similarly to in-air applications, Deep Learning methods have been quickly adapted to handle object recognition in underwater environments. In [24], Yang et al. applied both YOLOv3 [25] and Fast Region-based Convolutional Network (Faster R-CNN) [26] methods based on deep learning to localise and classify the images from their dataset *Underwater Robot Picking Contest (URPC)* into three categories—sea cucumber, sea urchin and scallop. The two algorithms were used in comparative experiments, to select the best algorithm and model for target detection and recognition, as part of their underwater detection robot. In [27], a detailed review of Deep Learning-based object recognition is given, whether it is underwater or surface target recognition. Surface object recognition is mostly based on images, while underwater objects are recognized based on videos, target radiated noises [28] and acoustic noises [29]. While Deep Learning outperforms traditional machine-learning methods when large amounts of training data are available, it also imposes additional effort in the annotation of those large amounts of data. Work on detection and mapping of pipelines and related objects has focused, almost exclusively, on above-water scenarios. An exception is the work of Martin et al. [30], which presents an approach based on a deep neural network PointNet [31]. These authors are able to detect pipes and valves from 3D point clouds with RGB color information, obtained with a stereo camera, using their own dataset to train and test the network.

1.1. Objectives and Contributions

The present paper develops a semantic Bayesian model for the recognition of 3D underwater pipeline structures. The proposed approach builds upon our previous work in [3], and extends it in several directions. The present work was motivated by the challenges stemming from real data collected under realistic underwater conditions with an AUV equipped with a fast laser scanner developed at our research center [32]. An example of the challenging conditions is the fact that data is collected by a free-floating, platform whose movements create deformations of the perceived shape of the objects which are difficult to be corrected with the typically available sensors, such as Inertial Measurement Unit (IMU) and Doppler Velocity Log (DVL). Three main contributions of the present paper can be summarized as follows.

- The 3D complexity of pipeline structures makes segmentation a difficult issue to deal with. Our test structure, which is described in further detail in Section 5, includes four different types of objects: two different valves (*Butterfly-Valve* and *Ball-Valves*), an *Elbow* and a *r-R-Tee*. These objects are connected by cylindrical pipes. In this paper a semantic segmentation method is proposed, based on geometric constraints together with rules for decomposing connected pipe structures. The aim of this method is to separate and distinguish, at the point cloud level, the points that belong to objects and those that belong to connecting pipes.
- Most global 3D descriptor methods assume that the point clouds are de-noised, complete, and consistent. This is not always the case, specially for the conditions that we are targeting in this paper, where the objects may be partially occluded due to the cluttered nature of the pipelines, and the point clouds may be inconsistent due to unmodeled deformations caused scanner motions during acquisition. These conditions commonly lead to false detection and overall failure of the global descriptor methods. Additionally, the similarity between objects can also lead to confusion when only a small or non-informative part of the object is observed. To overcome these limitations, a Bayesian semantic model is proposed. Taking advantage of the results obtained in our previous work [3], a confusion matrix was created for different global descriptors and objects. In this study, only the two best performing descriptors were considered:

- CVFH and Oriented, Unique and Repeatable (OUR-CVFH).
- To feed the Bayesian estimation model, observations of the same object across multiple scans are required. However, the underwater data suffer from the lack of DVL tracking during the descent of the AUV and sometimes during the mission when, for example, the sensor beams touch the side slopes of the test tank facility. The loss of DVL tracking leads to a rapid degradation of the estimates of the absolute pose of the pipeline structure with respect to the vehicle, which in turn hinders the ability to correctly perform the tracking of the objects. To overcome this problem, a multi-object tracking method inspired in the Joint Compatibility Branch and Bound (JCBB) algorithm [33] was proposed.

1.2. Structure of the Paper

The remainder of the paper is organized as follows. Section 2 describes the processing pipeline that is proposed in this paper. It includes a description of the object database, the algorithms used for pipe detection and semantic object detection, and the object recognition based on global descriptors. Section 3 describes the Bayesian Recognition component of our approach. It details the object tracking and Bayesian estimation processes. In Section 4, the algorithm developed for the recognition based on semantic information is detailed. Section 5 presents a description of the experimental hardware, the testing conditions and the analysis of the experimental results. This analysis is separated in terms of average and class-by-class performance, followed by a discussion of results. Finally Sections 6 and 7 present the overall conclusions of this work and lines for further research, respectively.

2. 3D Object Recognition Pipeline

Our recognition strategy focuses on object recognition of connected objects, which includes polyvinylchloride (PVC) pipes and attached elements, such as simple pipe connectors and valves suitable for manipulation and intervention. The proposed recognition pipeline is shown in Figure 1. The method uses, as input, a 3D point cloud acquired by a laser scanner mounted on an AUV. The scene contains objects for which 3D models are available *a priori* in a database. These objects are interconnected through pipes. The goal of the algorithm is to identify these objects by returning the class of the object with its associated Bayesian probability.

As shown in Figure 1, the recognition pipeline is divided into different modules described in the following subsections.

2.1. Object Data Base

The data base contains 3D models of the *a priori* known objects. Each one is modelled as a set of overlapping partial views stored as point clouds and covering the full object. The details on how the data base was built are presented in [3]. The only difference regarding the database used in the present paper is that, given the similarities of the partial views of *Ball-Valve* and the *Ball-Valve-S* (as can be seen in Figure 2) it was decided to merge these two classes into a single class labelled *Ball-Valve*.

The most relevant characteristics of the objects in the database are illustrated in Table 1 including their views.

2.2. Plane Segmentation

Our recognition system was tested in a robotics testing pool, as described in Section 5. The pool walls appear in the scans as large co-planar sets of points. These surfaces need to be removed in order to avoid unnecessary interference with the semantic segmentation that will be applied to the industrial pipe structure. In order to achieve this, a plane segmentation procedure was implemented using the Random Sample Consensus (RANSAC) [34] algorithm already available in Point Cloud Library (PCL) [35]. Due to the fact that the AUV is free-floating and moving during the scan acquisitions, the sets of points corresponding to the pool walls are not precisely co-planar. In fact, they follow a slightly curved but almost

flat surface, which is not straightforward to describe parametrically. However good results for the plane extraction can be achieved by properly adjusting the acceptance threshold in the plane-fitting algorithm.

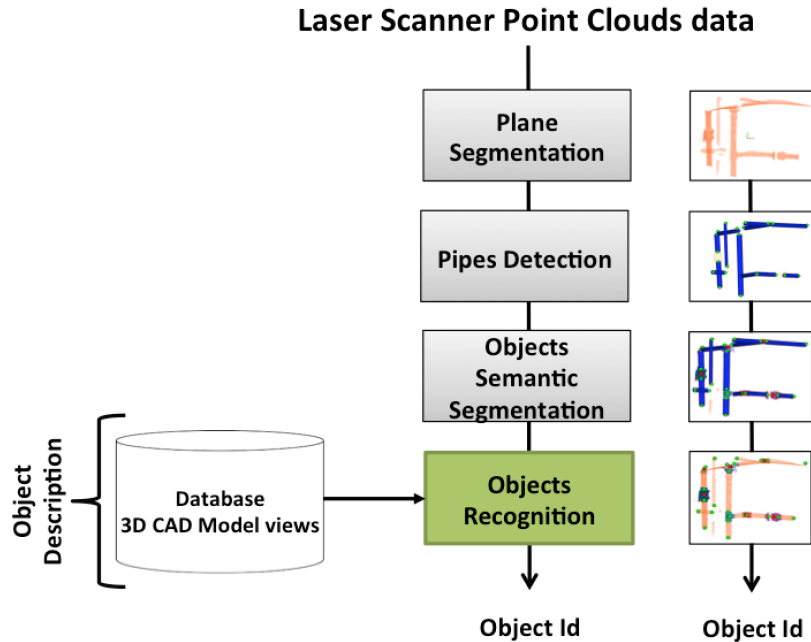














Figure 1. 3D Object Recognition Pipeline.

Table 1. Polyvinylchloride (PVC) pressure pipes objects used in the experiments.

PVC Objects	Id Name	Size (mm ³)	PVC Objects Views (12)
	1-Ball-Valve	198 × 160 × 120	
	2- Elbow	122.5 × 122.5 × 77	
	3- R-Tee	122.5 × 168 × 77	
	4- R-Socket	88 × 75 × 75	
	5- Butterfly-Valve	287.5 × 243 × 121	
	6- 3-Way-Ball-Valve	240 × 160 × 172	

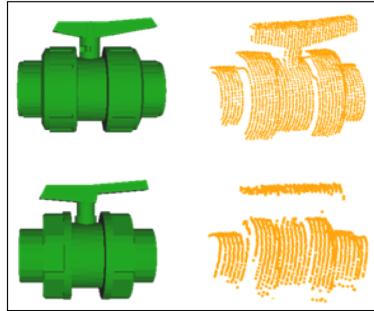


Figure 2. Ball-valve (top) and Ball-valve-s (bottom) with their respective segmented scan.

2.3. Pipe Detection

The next step is to detect the pipes that are visible within the current scan. A variety of methods exist to estimate the parameters of primitive geometric shapes such as planes, spheres, cylinders, cones, within 3-D point clouds [36–40]. In our case, a method based on RANSAC-PCL [35] has been applied to detect the pipes in the scene which are modelled as cylinders of similar radii. The RANSAC-PCL method uses a seven parameter description of the cylinders, where the first three represent a point on the axis, the second three represent the direction of the axis, and the last one represents the radius of the cylinder. Since the diameters of the pipes are known and equal to 0.064 m, we look for potential candidate cylinders whose radii are within a tolerance of this value.

Once the set of points belonging to a cylinder has been identified, the location of the extremities and the length can be computed by projecting the points on the cylinder axis and calculating the maximum and minimum of the segment defined by the projection. Figure 3 shows, for a given scan, all detected pipes with their respective endpoints. Unfortunately, in some cases, the same pipe may generate two different cylindrical point clouds. As shown in the encircled area of the left Figure 4, two pipes were detected, one appearing in red (the long one) and the other in blue (small section of a pipe). This happens due to small deformations of the scan caused by the motion induced distortion present in the underwater laser scanner [41]. Therefore, it is necessary to identify and fuse the point clouds that correspond to the same pipe segment (Algorithm 1) in order to provide a set of non duplicated pipes as input to the next module. The right side of Figure 4, shows the result after the merging.

Algorithm 1: Detection of Pipes and Extremities

```

1 function DetectPipes(in: scan, out:  $P_I$ ):
  | // Returns the set of pipes  $P_I$  detected in the scan using RANSAC
2 function MergePipes(in:  $M_{P_i}$ , out:  $P_{M_{P_i}}$ ):
  | // Returns a single pipe ( $P_{M_{P_i}}$ ) result of merging the input set of
  | pipes ( $M_{P_i}$ )
3 function PipeSegmentation(in: scan, out:  $P_O$ ):
  | // Returns the set of non duplicated pipes present in the scan
4  $P_I = \text{DetectPipes}(\text{scan})$  // get the set detected pipes
5 forall  $P_i \in P_I$  do
  | //  $M_{P_i}$  set of duplicated pipes to be merged
6  $M_{P_i} = \{P_i\} \cup \{P_j \in P_I | \exists P_k \in M_{P_i}, (\text{Colinear}(P_j, P_k) \wedge \text{Overlapped}(P_j, P_k))\}$ 
7  $P_{M_{P_i}} = \text{MergePipes}(M_{P_i})$  // merge the duplicated pipes
8  $P_O = P_O \cup \{P_{M_{P_i}}\}$  // add to the set of detected pipes
9  $P_I = P_I \setminus M_{P_i}$  // subtract  $M_{P_i}$  form  $P_I$ 
10 return  $P_O$ 

```

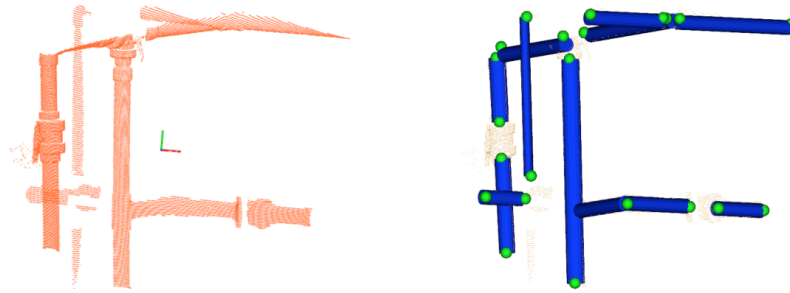


Figure 3. Pipes detection: (left) 3D laser scan point cloud; (right) pipes with their respective endpoints.

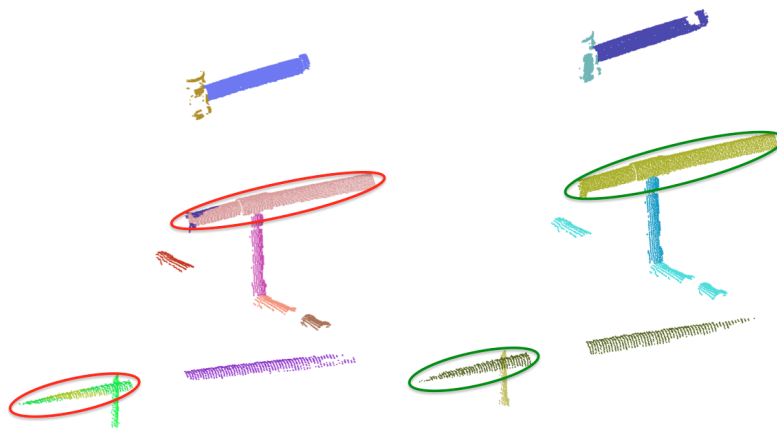


Figure 4. Pipes Merging: (left) Pipe detection result previous to merging showing, within circles, multiple pipe detections of the same pipe; (right) Result after merging where the multiple detections have been merged into a single one.

2.4. Semantic Object-Segmentation

This block of the procedure handles segmenting the object point clouds, from an input scan, containing pipes and objects. Instance segmentation is the process of clustering of input data (e.g., image or point cloud) into multiple contiguous parts without regard to understanding the context of its environment. One of the drawbacks of instance segmentation is that it relies on object detection methods to find the individual instances, which results in segmenting only the detected instances, so its performance in terms of over- or under-segmentation, depends on the result of the object detection method used.

By contrast, semantic segmentation partitions the scenes into semantically meaningful parts, based on the understanding of what these parts represent, classifying each part into one of the pre-determined classes: pipes and objects. Therefore, semantic segmentation can be used to segment point clouds corresponding to challenging scenes where objects are connected to pipes. Since the pipes have been already detected, and because they are connected through objects, it is possible to exploit the connectivity and pipe intersections to guide the segmentation process. The *SemanticSegmentation*(\cdot) (Algorithm 2) is organized in 4 steps:

1. Compute pipe intersections: This is done by the *Connected* function (Line 1) which, for each pair of pipes, checks if they are connected through an object and returns the pipe intersection point. To be connected, the axes of both pipes should be co-planar and two of their extremities should be close enough. By close enough, we consider that their distance should be smaller than the object size. Ideally, co-planarity means that the axes, when taken as infinite lines, will intersect. In reality, the axis lines estimated

for the two pipes may not intersect, but will have a small distance between them. Therefore co-planarity is assessed by checking the inter-line distance.

2. Compute candidate object locations at the intersections: Each pair of pipes defines an 'intersection' point. Therefore, if we have 3 pipes connected to an object (e.g., the *R-Tee*), we have 3 pairs of 2 pipes having, therefore, 3 intersection points. The function *ComputeIntersectionLocations* in line 16 clusters the intersection points corresponding to the same object and computes their centroids, to obtain a single location for each object.
3. Compute candidate object locations at isolated pipe extremities: Because of the iterative nature of the scanning process it may happen that a pipe appears in a scan together with an object at its extremity, while the other pipes connected to the object have not yet been detected. The function *ComputeExtremityLocations* in line 17 computes the object locations in these cases. The outcome of this step is shown in Figure 5.
4. Crop the objects from the input scan: Once the object locations are known ($C_i \cup C_e$), and knowing the dimensions of the objects, the points contained in a predefined bounding box are cropped (line 25) and returned for object recognition.

Figure 6 shows an example of semantic segmentation where the candidate object locations can be appreciated together with the segmented point clouds.

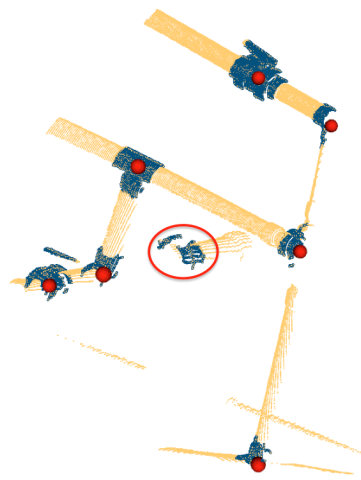


Figure 5. Semantic Segmentation: Red points represent the centroids of segmented objects. The red circle shows a segmented object located at an isolated extremity.

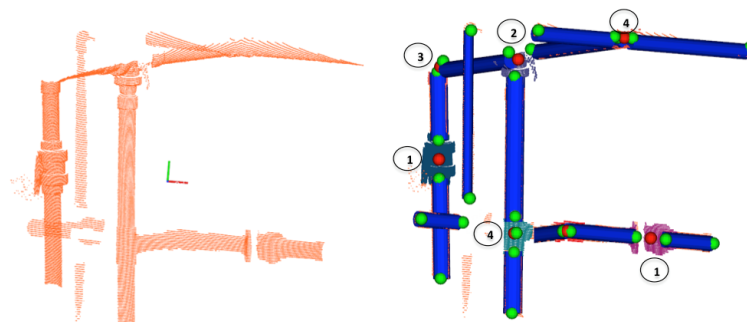


Figure 6. Semantic Segmentation: (Left) Input 3D point cloud; (Right) Pipes (blue cylinders) with their endpoints (green spheres), and the centroids of the objects to be segmented (red spheres) along with the segmented objects point clouds (colored). The objects 1, 2, 3, 4 represent respectively: a *Ball-Valve*, a *3-Way-Valve*, an *Elbow* and a *R-Tee*.

Algorithm 2: Semantic Segmentation

```

1 function Connected(in:  $P_i, P_j$ , out: connected, intersection):
    // Computes the intersection point between  $P_i, P_j$  and returns if they are
    // connected
    // Compute the points on the pipe axis lines defining the shortest
    // distance segment
2    $(c_i, c_j) = \text{LineToLineSegment}(P_i, P_j)$ 
3    $d = \|c_i - c_j\|$  // Compute the line to line distance
4    $\text{intersection} = (c_i + c_j)/2$  // midpoint  $\equiv$  intersection  $\equiv$  obj pos
5    $d1_{P_i} = \|\text{intersection} - \text{Extremity}(1, P_i)\|$  // distance to extremities
6    $d2_{P_i} = \|\text{intersection} - \text{Extremity}(2, P_i)\|$ 
7    $d1_{P_j} = \|\text{intersection} - \text{Extremity}(1, P_j)\|$ 
8    $d2_{P_j} = \|\text{intersection} - \text{Extremity}(2, P_j)\|$ 
9   if ( $d < \tau_d$ ) then // coplanar?
10    if ( $d1_{P_i} < \tau_d$ ) &&& ( $d1_{P_j} < \tau_d$ ) then return connected=true;
11    if ( $d1_{P_i} < \tau_d$ ) &&& ( $d2_{P_j} < \tau_d$ ) then return connected=true;
12    if ( $d2_{P_i} < \tau_d$ ) &&& ( $d1_{P_j} < \tau_d$ ) then return connected=true;
13    if ( $d2_{P_i} < \tau_d$ ) &&& ( $d2_{P_j} < \tau_d$ ) then return connected=true;
14  else
15    return connected=false
16 function ComputeIntersectionLocations(in:  $C_p$ , out:  $C_i$ ):
    // Given the set of pipe pairs intersections ( $C_p$ ), returns the set of
    // obj locations ( $C_i$ ) at the pipe intersections
17 function ComputeExtremityLocations(in:  $P, C_i$ , out:  $C_e$ ):
    // Returns the set of obj locations at the isolated pipe extremities
18 function SemanticSegmentation(in: scan,  $P$ , out:  $O$ ):
    // Returns the set  $O$  of objects locations and their cropped point
    // clouds
19    $C_p = \emptyset$  // set of pipe pairs intersections
20   forall  $(P_i, P_j) \in P \times P | i \neq j$  do
21     if Connected( $P_i, P_j, \text{intersection}$ ) then  $C_p = C_p \cup \{(\text{intersection}, P_i, P_j)\}$ ;
22    $C_i = \text{ComputeIntersectionLocations}(C_p)$  // set of obj pos at pipe
    // intersections
23    $C_e = \text{ComputeExtremityLocations}(P, C_i)$  // set of obj pos at pipe extremes
24    $O = \emptyset$ 
25   forall objpos  $\in C_i \cup C_e$  do
26     objpc = CropObject(objpos, scan) // crop the obj point cloud
27      $O = O \cup \{< \text{objpos}, \text{objpc} >\}$ 
28   return  $O$ ; // return the set of obj locations and point clouds

```

2.5. 3D Object Recognition Based on Global Descriptors

Object recognition is based on the use of the global descriptors that we studied and compared in [3]. The Clustered Viewpoint Feature Histogram (CVFH) [42] and the Oriented, Unique and Repeatable CVFH (OUR-CVFH) [43] were the two descriptors that achieved the best overall performance, so we have selected only these two descriptors. A summary of their characteristics is presented in Table 2.

The descriptors are used to encode, in a compact way, the objects segmented in the previous step. They also encode the object views stored in the database (see Table 1). In this way, the segmented objects can be matched against the model views, comparing the segmented input scan, with all the views of the object models in the database. Using the

chi-square distance, as proposed in [44,45], the database view corresponding to the smallest distance is selected.

Table 2. Summarized characteristics of the two descriptors used in this paper, respectively CVFH and OUR-CVFH. The “based on” column indicates if the descriptor evolved directly from another approach. The “use of normals” indicates whether the method uses surface normals for computing the descriptor, while the last column indicates the length of the descriptor vector.

Descriptor	Main Characteristics		
	Based on	Use of Normals	Descriptor Size
Clustered Viewpoint Feature Histogram (CVFH)-2011—[42]	Viewpoint Feature Histogram(VFH) [46]	Yes	308
Oriented, Unique and Repeatable CVFH (OUR-CVFH)-2012—[43]	CVFH [42]	Yes	308

3. Bayesian Recognition

One of the problems of performing single view object recognition as proposed above (in Section 2.5) is that several objects may have similar views. Partial views of the *R-Tee* may be easily confused with the *Elbow* for instance. In [3] we studied the confusion matrices for the different objects. The confusion matrices state, for n observations of a given object, how many of them were recognised as *object-class-1*, how many as *object-class-2* and so on. Therefore, they can be easily converted into probabilities which can be used to implement a Bayesian estimation method for object recognition to attain more robust results. This is achieved by combining several observations to compute the probability that an object belongs to each object-class, selecting, then, the one with highest probability as the solution. To do this, first it is necessary to be able to track the objects across the scans (as described in Section 3.1) so that their Bayesian probabilities can be iteratively computed (Section 3.2).

3.1. Object Tracking

To track objects across the scans we have to solve the data association problem. The simplest way to do it is to use the Individual Compatibility Nearest Neighbour (ICNN). This can be done if a reasonable dead reckoning navigation is available. In presence of significant uncertainty ICNN is not enough, and more powerful strategies such as the JCBB [33] are required. JCBB explores the interpretation tree (Figure 7) searching for the hypothesis with largest number of jointly consistent pairings between measurements (e_i) and features (f_j). The validation of the hypothesis is based on two conditions: (1) the candidate set of pairings must be individually and jointly compatible and (2) only those hypotheses that may increase the current number of pairings are explored (bound condition). The first condition is achieved by comparing the Mahalanobis distance of the set of candidate pairings with a threshold, defined at a given confidence level, of the related Chi-square distribution. The second condition is met by estimating the maximum number of pairings we can achieve if we keep exploring the current branch. Since each depth level of the tree represents a potential pairing, the number of levels below the node of the current hypothesis is an estimate of the maximum number of pairings we may add by exploring the current branch. Then it is only worth continuing exploring if the number of pairings of the current hypothesis plus the maximum number of achievable pairings is higher than the one associated with the current best hypothesis.

Unfortunately, when an AUV navigates close to vertical 3D structures, like a water tank, some DVL beams may suffer from multi-path effects leading to incorrect localization (position jumps). This type of error cannot be solved using the standard JCBB. For this reason, a navigation-less variation of the JCBB algorithm based on the intra-scan inter-object distances is proposed in Algorithm 3, which will be referred to as IJCBB. In this case, the algorithm pairs objects that are present in two scans so that all their inter-distances in both scans remain unaltered. Let us consider two sets of object locations $E = \{e_1, \dots, e_m\}$

and $F = \{f_1, \dots, f_n\}$ segmented from two given scans (S_E and S_F), whose objects we want to associate. A matching hypothesis is defined as a set of non-duplicated potential pairings from both scans:

$$\mathcal{H} = \{p_{ij} = (e_i, f_j) \in E \times F / \forall p_{kl} \in \mathcal{H} \implies i \neq k \text{ and } j \neq l\}. \quad (1)$$

An hypothesis is considered to be jointly compatible if and only if, the distance between any two objects in scan S_i and the corresponding distance of their matching objects in scan S_j also matches:

$$\mathcal{H} \text{ Jointly compatible} \iff (\forall p_{ij}, p_{kl} \in \mathcal{H} \implies \|e_i - e_k\| = \|f_j - f_l\|). \quad (2)$$

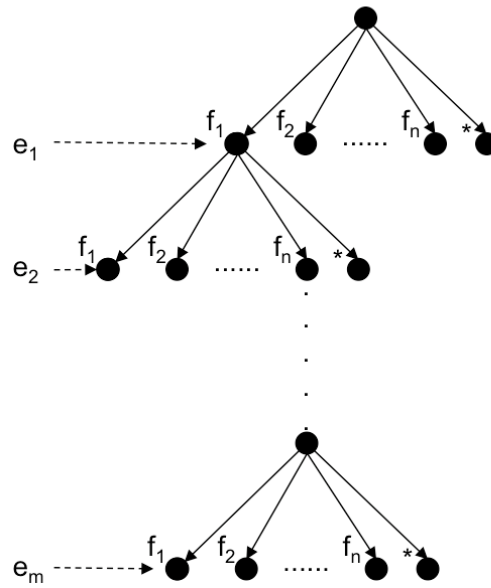


Figure 7. Interpretation tree stating, for each object e_i (level i) its potential associations $f_{1\dots n}$, representing the (*) node, a spurious measurement.

Then, as stated above, the goal of IJCBB (Algorithm 3) is to find the largest hypothesis \mathcal{H}_L for which the condition in Equation (2) holds. Once \mathcal{H}_L has been computed, the roto-translation transformation between both scans can be computed using Single Value Decomposition (SVD) [47]. The minimum number of matching pairs required to solve for the roto-translation is 3, which defines 3 inter-distances. Figure 8 shows an example of the ambiguities that may arise using 3 pairs only.

Let us consider a robot located at a pose η_k (yellow) moving, during a small time interval Δt , a displacement $\Delta\eta$ to achieve a new pose η_{k+1} (green). Let $\hat{\eta}_k$, $\Delta\hat{\eta}$ and $\hat{\eta}_{k+1}$ be the estimates of the corresponding vectors. If $\Delta\hat{\eta}$ is incorrect due to a failure in the navigation sensors, the estimated robot location at time $k + 1$ ($\hat{\eta}_{k+1}$) is also erroneous (frame $\{E_{k+1}\}$ in orange). Now, let us consider 3 equidistant objects: o_1 , o_2 and o_3 , observed from $\{S_k\}$ as: e_1 , e_2 and e_3 as well as from $\{S_{k+1}\}$ as: f_1 , f_2 and f_3 . Since the 3 inter-distances are equal, 6 possible pairings exist ($\{e_1f_1, e_2f_2, e_3f_3\}$, $\{e_1f_2, e_2f_3, e_3f_1\}$, $\{e_1f_3, e_2f_1, e_3f_2\}$, $\{e_1f_3, e_2f_2, e_3f_1\}$, $\{e_1f_1, e_2f_3, e_3f_2\}$, $\{e_1f_2, e_2f_1, e_3f_3\}$), the first 3 (the ones involving a rotation in the plane only) are shown in Figure 8. The other three are not considered since they involve a motion (in pitch) which the robot cannot manage. The actual solution corresponds to frame $\{S_{1,k+1}\}$ (in green) while the others ($\{S_{2,k+1}\}$ and $\{S_{3,k+1}\}$ both in grey) are not correct. Given the fact that we are tracking the robot pose, Δt is very small so the smallest motion (lower $\Delta\psi_i$) can be considered the correct one. In case only two inter-distances are equal, then four pairings exist and only two are relevant. Again, the

smallest motion heuristic can be applied. When all the inter-distances are different a single pairing exists.

Algorithm 3: IJCBB

```

// The algorithm is called as  $\mathcal{H}_L = IJCBB([], [], 1, e_{1\dots m}, f_{1\dots n})$ 
//  $\mathcal{H}_L[i] = j \Rightarrow (e_i, f_j)$  is a pairing;  $\mathcal{H}_L[i] = 0 \Rightarrow e_i$  is not paired
//  $e_{1\dots m}$ : object locations in the first scan  $S_E$  indexed by  $i$ 
//  $f_{1\dots n}$ : object locations in the second scan  $S_F$  indexed by  $j$ 
1 procedure IJCBB(in:  $\mathcal{H}$ , Best, i; out:  $\mathcal{H}$ , Best):
2   if  $i > m$  then                                     /* leaf node? */
3     if Pairings( $\mathcal{H}$ ) > Pairings(Best) then
4        $Best \leftarrow \mathcal{H}$ ;
5   else
6     for  $j=1$  to  $n$  do
7       if JointCompatible( $\mathcal{H}, i, j$ ) then
8         IJCBB( $[\mathcal{H} j]$ , Best,  $i + 1$ )                 /*  $(e_i, f_j)$  accepted */
9       if Pairings( $\mathcal{H}$ ) +  $m - i >$  Pairings(Best) then
10        IJCBB( $[\mathcal{H} 0]$ , Best,  $i + 1$ )                 /* star node,  $e_i$ , not paired */
11 function JointCompatible(in:  $\mathcal{H}, i, j$ , out: compatible):
// Returns true if
//  $\tau_\psi$ : maximum rotation  $\Delta\psi$  that can be experimented in  $\Delta t$  seconds
12 switch  $i$  do                                       // number of pairings in the hypothesis
13   case 1 do
14     return true
15   case 2 do
16     return InterdistanceCompatible( $\mathcal{H}, i, j$ )
17   otherwise do
18     return (InterdistanceCompatible( $\mathcal{H}, i, j$ ) && GetRotation() <  $\tau_\psi$ )

```

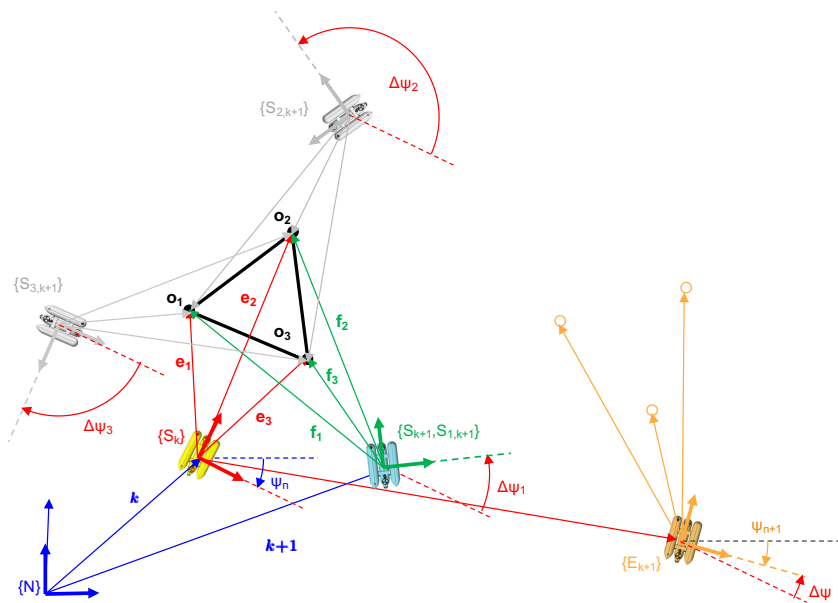


Figure 8. Roto translation estimation.

It may also happen that Equation (2) holds for an incorrect data-association hypothesis. This means that we can have two different sets of objects, having the same inter-distances. This may happen when scanning repetitive structures, for instance. Again, because Δt is small, the small motion heuristic also works providing the correct roto-translation. For these reasons, the *JointCompatible*(\cdot) function in Algorithm 3 checks the rotation angle implied by the hypothesis \mathcal{H} , which should be small enough to be considered jointly compatible.

Figure 9 shows the tracking of two consecutive scans using IJCBB. The red objects were detected from S_E and the blue ones from S_F , corresponding to the previous and the current scan. In this case five objects were paired, while other three were discarded.

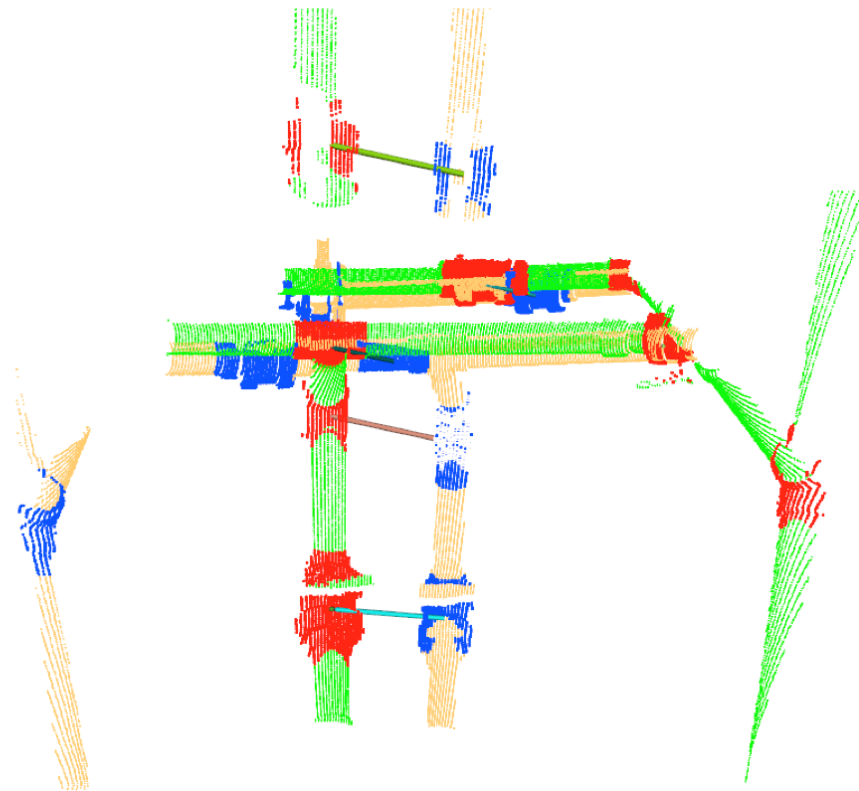


Figure 9. Tracking objects over two consecutive scans, represented in green/red and yellow/blue. The significant displacement between the two scans is the results of navigation inaccuracies from noisy Doppler Velocity Log (DVL) readings in the test pool. The solid lines indicate the objects associated by the tracking.

3.2. Bayesian Estimation

The objects can often be confused with others. This happens because we are dealing with partial views of the objects appearing in the scans, which may match several views of other objects in the database. To overcome this problem, we propose to use Bayesian estimation. The object confusion matrix, already computed in [3], can be used as an estimate of the conditional probabilities needed for this purpose. Let Z be the object class recognized with the global descriptor, X its actual class and let their sub-indexes represent each one of the potential classes (*Ball-Valve:1, Elbow:2, R-Tee:3, R-Socket:4, Butterfly-Valve:5, 3-Way-Valve:6*), then $P(Z_C|X_i)$ provides the probability of recognising an object as belonging to class Z_C when its actual class is X_i . If $C = i$ then it is a True Positive (TP), otherwise ($C \neq i$) it is a False Positive (FP). Tracking the objects across the scans allows computing its class probabilities in an iterative way, selecting the one with highest probability as the recognized one.

The proposed Bayesian recognition method is shown in Algorithm 4. The observation probabilities $P(Z_j|X_i)$ contained in the $P_{Z|X}$ matrix are computed from the synthetic confusion matrix (Table 3). Then, given an Object O and the class Z_C resulting from the descriptor-based recognition, the next procedure is followed. If the object is observed for the first time (line 8) its prior probability is initialized considering each potential class as equi-probable (line 11). Lines 12–16 use the Bayes Theorem to compute the probability of the object belonging to each potential class j , given the observed class Z_C and its prior probability $O.P[j]$. Finally, the most likely class is returned as the one recognised by the method.

Algorithm 4: Bayesian-based Recognition

```

// Ball-Valve:1, Elbow:2, R-Tee:3, R-Socket:4, Butterfly-Valve:5,
// 3-Way-Valve:6
1 function CompatibleClasses(O):
2   return [1,2,3,4,5,6]
3 function BayesianRecognition(in: O, ZC; out: O):
4   return Recognition(O, ZC, O)
5 function Recognition(in: O, ZC; out: O):
  // ZC ∈ {1, ..., 6} Detected Class
  // O = {seen: boolean, P = [P(X1), ..., P(X6)], np, id}
  // Observation probabilities extracted from the Confusion Matrix
  6 
$$P_{Z|X} = \begin{pmatrix} P(Z_1|X_1) & P(Z_1|X_2) & \dots & P(Z_1|X_6) \\ P(Z_2|X_1) & P(Z_2|X_2) & \dots & P(Z_2|X_6) \\ \vdots & \vdots & \ddots & \vdots \\ P(Z_6|X_1) & P(Z_6|X_2) & \dots & P(Z_6|X_6) \end{pmatrix}$$

  7 SC=CompatibleClasses(O) // Set of compatible classes
  8 if ¬(O.seen) then
  9   O.seen = true // First Observation of O
 10   forall j ∈ SC do
 11     O.P[j] = 1/#SC // All classes are equiprobable
 12   forall j ∈ SC do
 13     P[j] = PZ|X[ZC, j] * Oi.P[j] // Non normalized Bayesian prob:
 14     P(ZC|Xj) * P(Xj)
 15   η = 1/∑j∈SC P[j] // Compute the Normalizer
 16   forall j ∈ SC do
 17     O.P[j] = η * P[j] // Normalized Bayesian probabilities
 18   O.id = argmaxj∈SC O.P[j] // Select the Most likely class
  return O.id

```

Table 3. Confusion Matrices expressed as a numerical %.

Descriptors	Experiment	Objects																													
		Ball Valve						Elbow						R-Tee						Butterfly-Valve						3-Way-Ball-Valve					
		1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
CVFH	SYN	63	10	7	1	2	19	2	75	7	14	1	1	4	27	65	2	1	1	17	5	1	1	54	21	9	3	1	1	1	84
	DESC	72.5	9.5	1	2.5	3	11.5	10	86.67	3.33	0	0	0	23.5	8	41.5	0	2.5	24.5	50.67	0	1.33	0	25.33	22.67	58.82	0	0	0	11.76	29.41
	BAYS	100	0	0	0	0	0	10	90	0	0	0	0	5.5	0	57	0	7	30.5	4	0	0	0	96	0	100	0	0	0	0	0
	SEM	100	0	0	0	0	0	0	96.67	0	0	0	3.33	1	0	57	0	1.5	40.5	4	0	0	0	96	0	0	0	0	0	0	100
OUR-CVFH	SYN	62	8	11	1	2	16	2	68	11	17	1	1	2	22	71	3	1	1	13	7	4	1	63	13	10	3	4	1	1	81
	DESC	49	28	1	4	1	16	10	86.67	0	0	0	3.33	3	30	40	0	0	26	28	4	0	0	58.67	9.33	64.71	0	0	0	17.65	17.65
	BAYS	60	40	0	0	0	0	6.67	93.33	0	0	0	0	1	10.5	46.5	0	0	42	0	0	0	0	98.67	1.33	35.29	0	0	0	64.71	0
	SEM	84	15	1	0	0	0	0	96.67	0	0	0	3.33	1	0	57	0	0	42	0	0	0	0	98.67	1.33	0	0	11.76	0	0	88.24

4. Semantic-Based Recognition

The recognition rate can be further improved using semantic information about the number of pipes connected to the object and their geometry. This information can be used to constrain the set of potential compatible classes for a given object. As an example, if we know that an object is connected to 3 pipes, then only two candidate classes are possible—the *R-Tee* and the *3-Way-Valve*. Then, we can compute the Bayesian probabilities for these candidate classes only, assigning zero probability to the rest. Because we track the pipes to segment the objects, we can use this already available information to estimate the connectivity of the objects, and use this semantic information to improve the recognition results. The method has the potential to disambiguate confusing objects having different connectivity. For instance, certain views of the *Ball-Valve* can be easily confused with the *3-Way-Valve* (See Figure 10). This ambiguity can be easily resolved by taking into account the connectivity. Algorithm 5 shows this modification with respect to the Bayesian method algorithm discussed above. The function *CompatibleClasses(O)*, originally returning the 6 classes, now returns only the set of classes compatible with the object connectivity geometry. It is worth noting that, given the iterative nature of the scanning process, a certain object may appear connected to a single pipe at first, and connected to two or three pipes later on. Therefore, 4 different geometric configuration may arise (Table 4):

1. Three pipes: 2 collinear and one orthogonal. This group contains the *R-Tee* and the *3-Way-Valve*.
2. Two orthogonal pipes: This group contains the *Elbow* but also the members of the previous group, since it is possible that the third pipe has not been observed yet.
3. Two collinear pipes: All objects are included in this group, except the *Elbow* (because it is orthogonal) and the *R-Sockets* (because only one side can be connect to a pipe of the given radius). The remaining objects admit a collinear connection to 2 pipes.
4. Single or no connection: All objects are considered as potential candidates.

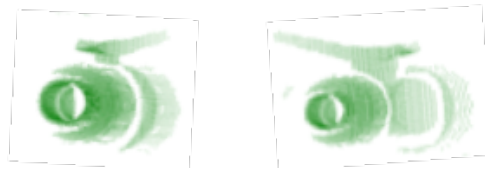
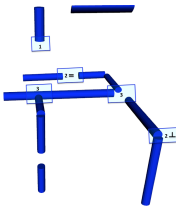






Figure 10. Confusing Views of the Ball-Valve and 3-Way-Valve objects.

Algorithm 5: Semantic-based Recognition

```
// Ball-Valve:1, Elbow:2, R-Tee:3, R-Socket:4, Butterfly-Valve:5,
   3-Way-Valve:6
1 function ConnectedPipes(O):
2   return Number of pipes connected to the object O
3 function Collinear(O):
4   return true if the object connected pipes are collinear, false: otherwise
5 function CompatibleClasses(O):
6   if ConnectedPipes(O) = 3 then return [3,6];
7   if ConnectedPipes(O) = 2 && ¬Collinear(O) then return [2,3,6];
8   if ConnectedPipes(O) = 2 && Collinear(O) then return [1,3,5,6];
9   return [1,...,6]
10 function SemanticRecognition(in: O, ZC; out: O):
11  return Recognition(O, ZC, O)
```

Table 4. Semantic connection of Objects.

Type of Connection	Pipes Disposition			Potential Objects Candidate
	n_p	=	\perp	
	3	2	1	
	2	0	2	
	2	2	0	
	1 0	1 0	1 0	

5. Experimental Results

5.1. Test Platform and Laser Scanner

Testing was conducted using the Girona 500 AUV, a lightweight intervention- and survey-capable vehicle rated for 500m depth with dimensions of 1m in height and width, and 1.5m in length. The lower hull houses the heavier elements such as the batteries and removable payload, whereas the upper hulls contain flotation material and lighter components. This arrangement enables the vehicle to be very stable in roll and pitch due to the distance between the centers of mass and flotation. The pressure sensor, the Attitude and Heading Reference System (AHRS), the Global Positioning System (GPS), the acoustic modem and the DVL provide measurements to estimate the pose of the vehicle. The current configuration of thrusters provides the AUV with 4 degrees of freedom (DoF) which can be controlled in force, velocity and position. Finally, the vehicle software architecture is integrated in Robot Operating System (ROS) [48] simplifying the systems integration.

The laser scanner was designed and developed in-house [49]. It contains a laser line projector, a moving mirror driven by a galvanometer, a camera and two flat viewports, one for the camera and one for the laser. The galvanometer is electrically synchronized with the camera, such that the image acquisition is only performed when the galvanometer is stopped, thus producing an image with only one single laser line. The sensor generates a 3D point cloud by triangulating all the laser points corresponding to the different mirror positions during a full scan. For the experiments in this paper, the scanner was configured to acquire scans at a rate of 0.5 Hz generating ≈ 200 k points/s and 400 lines/scan. At a nominal distance of 3 m, the distance between scan lines is ≈ 4.5 mm.

5.2. Experimental Setup

The experiments consisted in exploring an underwater industrial structure made of pipes and valves, having approximate dimensions of 1.4 m width, 1.4 m depth and 1.2 m height (see Figure 11). During the experiment, the Girona 500 AUV was tele-operated to move around the structure. To reduce the distortions within each scan produced by the vehicle motion, the AUV was put in station-keeping mode during the acquisition of each scan. The structure was mapped at a distance ranging from 2 to 3.5 m.

During the experiment, 100 scans were processed containing a total of 523 object observations of 13 different objects from 6 different classes.

To evaluate the performance, ground truth was created by manually labelling objects appearing in the scans.

The following three object recognition methods, described in this paper, have been evaluated:

1. The Object Recognition Pipeline described in Section 2.
2. The Bayesian estimation extension presented in Section 3.
3. The Semantic Bayesian estimation extension presented in Section 4.

The three methods were tested using the two descriptors—CVFH and OUR-CVFH. These descriptors were selected because they provided the best experimental results in our previous survey paper [3].

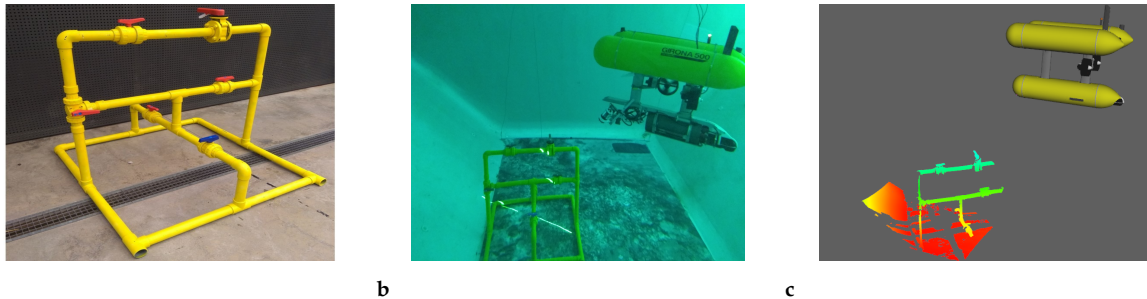


Figure 11. Image of the Girona 500 AUV inspecting the structure. The mapped structure before deployment (a), underwater view of the water tank (b) and online 3D visualizer with a scan of the structure (c).

The IJCBB method (Algorithm 3) was used to address and solve the issue of the navigation jumps, thus allowing tracking of objects across the scans. Consistent tracking of objects is required for the Bayesian estimation to work properly. The effect of the IJCBB method can be seen in Figure 12. The left side shows the accumulation of object instances using only the dead reckoning from the vehicle navigation data. The large navigation errors and the close proximity of some objects leads to some of the tracked objects being incorrectly assigned over time. The right side illustrates the improvement in the localization of these objects by using the tracking based on the IJCBB.

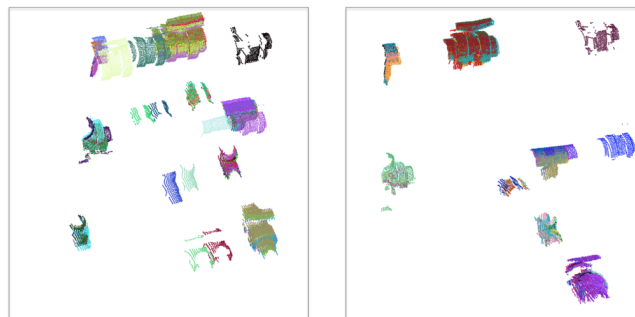


Figure 12. Mapped object point clouds: (Left) Located at their dead reckoning position; (Right) Located at the position estimated by the tracking using the IJCBB algorithm on the right.

Figure 13 and Table 3 show the graphical and numerical representation of the confusion matrices computed for the following cases:

1. The Synthetic Confusion Matrix.
2. The Confusion Matrix based on global descriptors only.
3. The Confusion Matrix incorporating Bayesian estimation.
4. The Confusion Matrix incorporating Bayesian estimation and semantic information.

The first confusion matrix was computed based on the results of our previous paper [3]. It was obtained by averaging the confusion matrices corresponding to the partial and the global view experiments for the noise matching and resolution in the order of magnitude of the one of our scanner ($\sigma = 0.00625$, $resolution = 0.007$ [m]) and for the case where the same resolution is used for the scan and the object 3D model in the data base. The other 3 were computed from the results of the experiment.

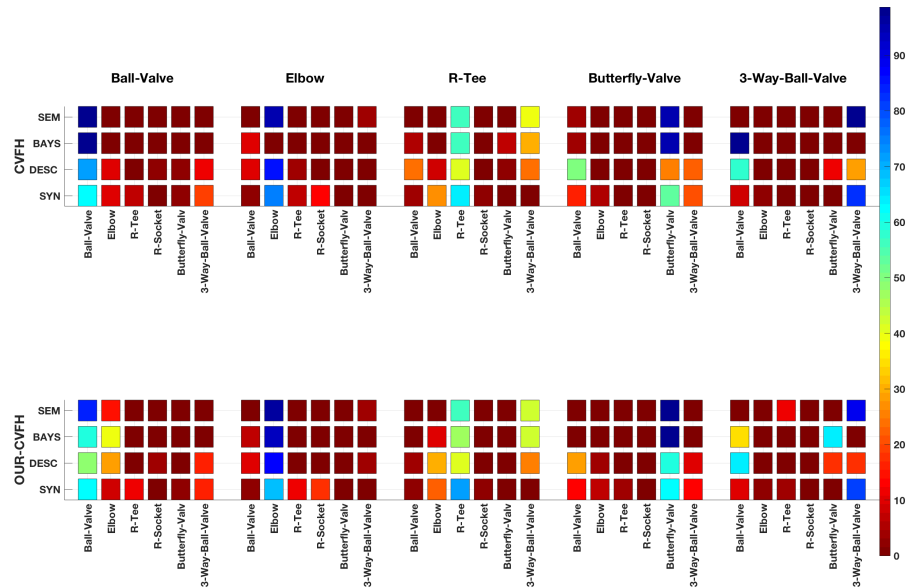


Figure 13. Graphical representation of the Confusion Matrices.

5.3. Average Performance

The average object recognition rate (percentage of correctly recognized objects) for both descriptors, CVFH and OUR-CVFH, is summarized in the last column of Table 5. It can be appreciated that, as hypothesised, in both cases the Bayesian estimation improves the recognition rate achieved with the descriptor alone. Moreover, the use of semantic information further improves the results. When using the OUR-CVFH descriptor improvements (with respect to the semantic method) of 9% and 25% respectively are observed, achieving a final average recognition rate of 85%. Nevertheless, the best results are achieved using the CVFH descriptor, where the Bayesian method improves recognition by 18% and the semantic variant provides a further improvement of 21%, reaching an average recognition rate of 90%.

Table 5. Average of recognition per Object and methods for all descriptors, represented in a table.

Descriptors	Experiment	Average
CVFH	Descriptor	51.2
	Bayesian	68.6
	Semantic	90
OUR-CVFH	Descriptor	50.8
	Bayesian	59.8
	Semantic	85

5.4. Class-by-Class Performance

Now let us focus on the class-by-class performance. To provide a better insight, the evaluation is based on the performance metrics (recall, precision and accuracy) for each descriptor-method-class combination reported in Table 6, and illustrated graphically in Figure 14.

Table 6. Assessment of the recognition performance through Accuracy, Recall and Precision. Qualitative labels used in the text: bad (0–0.2); poor (0.2–0.4); medium; good; excellent.

Descriptors	Experiment	Objects																	
		Ball Valve			Elbow			R-Tee			Butterfly-Valve			3-Way-Ball-Valve			Average		
		Accuracy	Recall	Precision	Accuracy	Recall	Precision	Accuracy	Recall	Precision	Accuracy	Recall	Precision	Accuracy	Recall	Precision	Accuracy	Recall	Precision
CVFH	DESC	0.65	0.73	0.60	0.88	0.87	0.43	0.70	0.42	0.95	0.80	0.25	0.59	0.73	0.29	0.05	0.75	0.51	0.52
	BAYS	0.92	1.00	0.85	0.99	0.90	1.00	0.83	0.57	1.00	0.96	0.96	0.84	0.84	0.00	0.00	0.91	0.69	0.74
	SEM	0.99	1.00	0.98	1.00	0.97	1.00	0.83	0.57	1.00	0.99	0.96	0.96	0.84	1.00	0.17	0.93	0.90	0.82
OUR-CVFH	DESC	0.64	0.49	0.70	0.67	0.87	0.18	0.68	0.41	0.98	0.88	0.59	0.90	0.70	0.18	0.03	0.71	0.50	0.56
	BAYS	0.78	0.60	0.92	0.75	0.93	0.22	0.75	0.47	1.00	0.96	0.99	0.87	0.76	0.00	0.00	0.80	0.60	0.60
	SEM	0.92	0.84	0.99	0.93	0.97	0.49	0.82	0.57	0.97	1.00	0.99	1.00	0.82	0.88	0.15	0.90	0.85	0.72

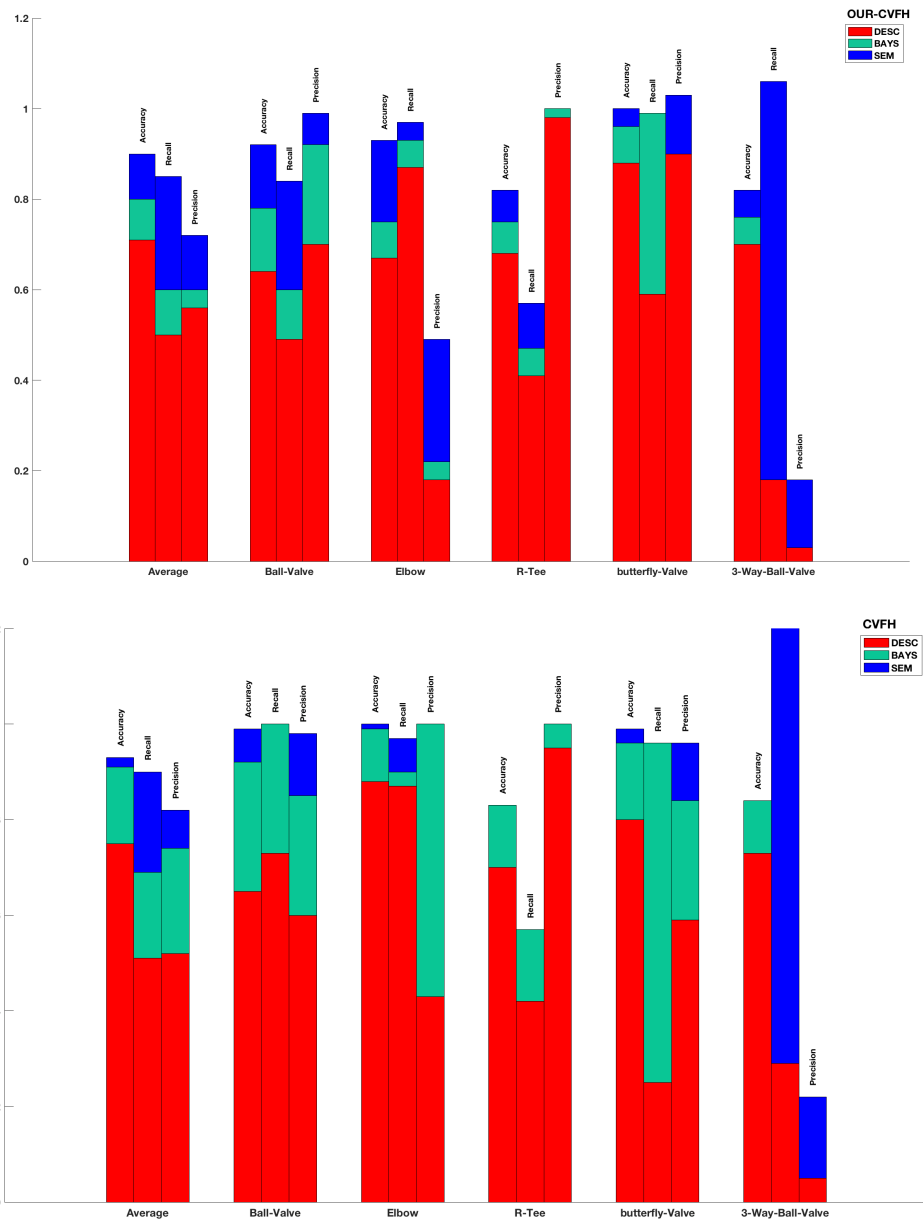


Figure 14. Evaluation of the recognition performance using Accuracy, Recall and Precision for descriptor-based, Bayesian-based and semantic-based method for both: **(Top)** OUR-CVFH; **(Bottom)** CVFH.

5.4.1. Descriptor Based Recognition Pipeline

When using only the descriptor based recognition, the performance varies across the object classes. For CVFH, the recall is excellent for the *Elbow*, good for the *Ball-Valve*, medium for the *R-Tee* and poor for the *Butterfly-Valve* and the *3-Way-Valve*. On the other hand, the precision is excellent for the *R-Tee*, good for the *Ball-Valve* and the *Butterfly-Valve*, medium for the *Elbow* and poor for the *3-Way-Valve*. Similar results are obtained for the OUR-CVFH descriptor which achieves an excellent recall for the *Elbow*, medium for the *Ball-Valve*, the *R-Tee* and the *Butterfly-Valve* and again bad for the *3-Way-Valve*. In this case

the precision is excellent for the *R-Tee* and the *Butterfly-Valve*, good for the *Ball-Valve* and poor for the *Elbow* and the *3-Way-Valve*.

5.4.2. Bayesian Estimation

When applying Bayesian estimation with the CVFH descriptor, both performance metrics improve significantly becoming excellent for the *Ball-Valve*, the *Elbow* and the *Butterfly-Valve*. For the *R-Tee* the recall is medium with an excellent precision, but for the *3-Way-Valve* both metrics are actually worse. The precision remains excellent for the *R-Tee* and improves to excellent for the *Ball-Valve*, the *Elbow* and *Butterfly-Valve*, but remains poor for the *3-Way-Valve*. For the OUR-CVFH descriptor, the performance improves slightly less. The recall remains excellent for the *Elbow* and improves to excellent for the *3-Way-Valve*. It remains good for the *R-Tee* and improves to good for the *Ball-Valve*, but still poor for the *3-Way-Valve*. On the other hand, the excellent precision of the *R-Tee* and the *Butterfly-Valve* is maintained while it evolves from good to excellent for the *Butterfly-Valve*, and from bad to poor for the *Elbow*, but remains poor for the *3-Way-Valve*. However, in general all the metrics improve.

5.4.3. Bayesian Estimation and Semantic Information

When semantic information is included in the Bayesian estimation, the performance further improves. For CVFH, the recall and precision qualitative performance remains the same (mostly excellent) but their numerical values increase slightly. Moreover, the poor performance in the Bayesian estimation of the *3-Way-Valve*, improves to excellent. The OUR-CVFH descriptor improves significantly in this case. The recall, remains excellent for the *Elbow* and the *Butterfly-Valve* and improves to excellent for the *Ball-Valve* and the *3-Way-Valve* while maintaining the medium performance (but increasing by 10%) for the *R-Tee*. Its precision remains excellent for the *Ball-Valve* and *Butterfly-Valve* (in both cases increasing numerically), evolving from poor to medium for the *Elbow*, although still poor (but increasing the value) for the *3-Way-Valve*. Again, all the numerical values of the statistics improve.

The overall best results are obtained using the CVFH and the semantic-method with excellent recall and precision for every object class except the *R-Tee* which has medium recall and the *3-Way-Valve* which has poor precision.

5.4.4. Discussion

Analysing the results reported in Table 4, we realise that for the *Butterfly-Valve* and the *3-Way-Valve* object classes the recall achieved with the descriptor method is significantly below the average recall. Moreover, for the *3-Way-Valve*, the Bayesian method is not improving the results but causing troubles. To understand what happens let us examine the synthetic confusion matrix (Figure 13) for both descriptors. It can be appreciated that the *Butterfly-Valve* is commonly confused with the *Ball-Valve* and the *3-Way-Valve*. For both descriptors the *Butterfly-Valve* (TP) observation probability is significantly higher (>50%) than the probabilities of the *Ball-Valve* and the *3-Way-Valve* (False Negatives (FNs)). However, when the confusion matrix is computed from the experimental data similar recognition percentages are found for OUR-CVHF while they are reversed for the CVFH, with higher probabilities for the FNs (*Ball-Valve* and *3-Way-Valve*) than for the TP (*Butterfly-Valve*). Using the partial views observed with the scanner in the experiment, CVFH is not working as well as it did with the synthetic ones simulated in [3]. Instead, the experimental and synthetic behaviours of OUR-CVFH are closer.

The problem is more severe with the *3-Way-Valve* whose experimental and synthetic recognition percentages are also reversed, and in addition suffering a poor accuracy, indicating that most of the observations are actually FNs. If we take a close look at the partial views obtained after the segmentation (see Figure 15) we can see that unfortunately most of them correspond to challenging scans (in red). Recognizing the object from those views is difficult, if not unfeasible, even for the human perception. This suggests that a

method should be designed to decide which view is representative and therefore worth attempting to recognize and which one should just be ignored.

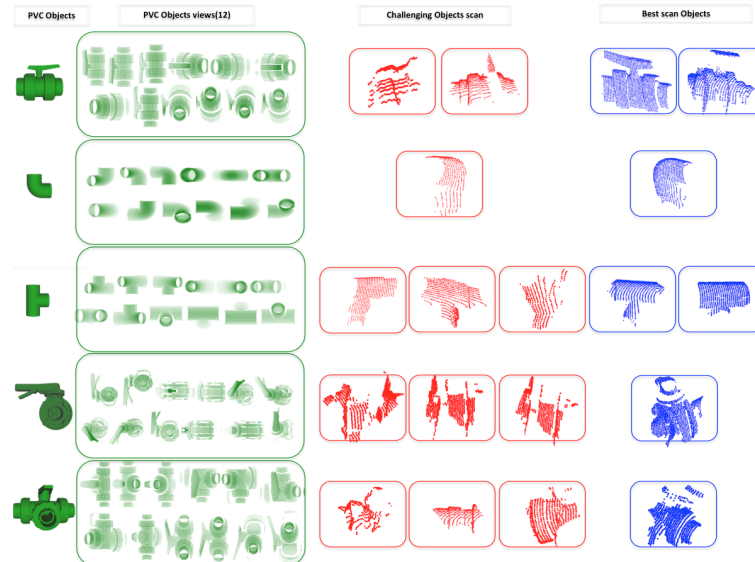


Figure 15. PVC objects used in the experiment (first column) with their respective database views (second column). The last two columns provide manually selected examples of segmented objects from the experiments, with the most difficult in red and the easiest in blue.

For the *Butterfly-Valve*, the Bayesian method does an excellent job, bringing the recall and the precision to 0.96 and 0.84 for CVFH and to 0.99 and 0.87 for OUR-CVFH. To understand why, let us focus on the CVFH descriptor. The TP probability ($P(Z_5|X_5) = 0.54$) is discriminant in comparison to the FP probabilities ($P(Z_5|X_1) = 0.02$, $P(Z_5|X_2) = 0.01$, $P(Z_5|X_3) = 0.01$, $P(Z_5|X_4) = 0.1$, $P(Z_5|X_6) = 0.01$). This means that a single TP observation assigns more weight to the probability of the TP-class than several FP observations do with their counterparts. Its accuracy (0.59) also helps, since it means that there are more TPs than FPs, driving therefore, the Bayesian estimation towards the correct class. The same happens for OUR-CVFH where we start from a much better point with a recall of 0.59 and a good precision of 0.9. Unfortunately this is not the case for the *3-Way-Valve* whose performance even decreases for both descriptors when using the Bayesian method. For the CVFH case, even though the TP probability ($P(Z_6|X_6) = 0.84$) is high, there are two significant FP probabilities in play ($P(Z_6|X_1) = 0.19$, $P(Z_6|X_5) = 0.21$). Adding this to the very high number of FPs (where precision is only 5%) explains the fact that the Bayesian method is not helping but actually making it worse. It is worth remembering that the origin of the problem is the fact that the *3-Way-Valve* partial views obtained after the segmentation are poor representatives of the object class.

When semantics are taken into account during the Bayesian estimation process, the results improve further. Now, Bayesian estimation only affects those classes which are compatible in terms of pipe connectivity. Because classes having a significant confusion, like the *3-Way-Valve* and the *Ball-Valve* for instance, have different connectivity (3 and 2 respectively), so they can be easily distinguished by the number of connected pipes. This further improves the results of all the object classes, recovering, in particular the recall of the *3-Way-Valve*. However, the precision is still poor because there are a significant number of FPs which are compatible in terms of connectivity. This is the case of the *R-Tee* class, which is often confused with the *3-Way-Valve*. Because both classes are equivalent in terms of connectivity, the semantic-based method is not able to help. Again, it is worth noting that the origin of the problem is the poor *3-Way-Valve* views observed in the experiment.

6. Conclusions

Detecting and recognizing multiply connected objects in underwater environments is a complex task that must be performed under the constraints of the sensor, the acquisition platform and the nature of the shapes of the objects we wish to detect. In this paper, we have presented a method to recognize 3D objects as part of a pipeline for acquiring and processing non-colored point clouds using point features. The presented method is intended to be used for Inspection, Maintenance and Repair (IMR) of industrial underwater structures. As a representative example for testing, the developed methods were applied to a test structure consisting of pipes and connected PVC objects. These objects pose considerable challenges for an object recognition system, due to view-dependant similarities in their appearance. As such, the testing conditions capture the main difficulties of a real scenario for underwater Inspection, Maintenance and Repair (IMR).

An initial goal of this paper was to develop methods for the pre-processing of point cloud data that would potentiate and facilitate the recognition task. These methods include plane and pipe detection, semantic segmentation, and object tracking based on the IJCBB algorithm. Semantic segmentation aimed at better obtaining a set of points that belong to the objects, in order to reduce the negative impact of the presence of parts of the pipes, during recognition. The semantic segmentation involved determining the pipe intersections, to then allow for computing candidate object locations and therefore perform a better crop of the input scan so that it tightly encapsulates the object to be recognized. The IJCBB-based tracking aimed at correcting the effects of inconsistencies in the robot navigation, which appeared in the form of sudden jumps in the estimated pose of the AUV that preclude the tracking of the objects along scans.

The second goal, which conveys the most important contributions of this study, is the comparison of three established methods, namely descriptor-based, Bayesian-based and semantic-based recognition.

The descriptor-based method, which was used in our previous work [3] to detect individual objects attained good performance, especially when the scans contained a complete, occlusion-free view of the objects. Considerably better results were obtained by tracking objects along scans and using a Bayesian framework to keep recognition probabilities assigned to each object, achieving, for the CVFH descriptor an 18% increase in the average recognition rate.

It should be noted that there is a significant increase in the recognition rate when the object to be detected satisfies the conditions that a relevant part of the object shape is present, and that distinctive features of the objects are visible. Clear examples where these conditions were not met were the *Butterfly-valve* and the *3-way-ball-valve*. These two objects were affected by poorly segmented views, which resulted in the loss of the distinctive features needed for discrimination among objects. In this case, the distinctive features are the handle for the *Butterfly-valve* and the part of the opening of the *3-way-ball-valve*.

These problems have been addressed by semantics-based recognition, which considers a set of rules based on pipe intersections that allow computation and updating of the Bayesian estimation approach, considering only objects that verify these rules. For the CVFH descriptor, the inclusion of semantic rules increases the average recognition rate by 21% with respect to the Bayesian method.

7. Future Work

Although there are clear advantages to using semantic information with the Bayesian method for recognition, the dependence of the recognition system on the segmented views makes it vulnerable in some cases. Motivated by the improved results achieved by using semantic information within the Bayesian approach, near-term future work will concentrate on the integration of the approach within a Simultaneous Localization And Mapping (SLAM) framework. Among other advantages, such a framework will further facilitate the association of observations of objects, releasing the constraint of needing sufficient temporal overlap between scans, which is implicitly required in the tracking

process. Moreover, SLAM will provide a consistent long term drift-less navigation, allowing to explore the structure from different viewpoints. This will enrich the set of views used during the Bayesian recognition providing more robust results.

From the experiments with the database views generated from the CAD models, we concluded that significant perceptual differences were observed between the rendered views in the database and the real views captured by the laser scanner. Such differences impact the recognition performance negatively. This problem will be addressed, in the near future, by collecting database views with the laser scanner used in the tank during the experiment.

As longer term future work, the approach will be used as a building block towards a complete system for autonomous intervention by I-AUVs working in industrial underwater scenarios.

Author Contributions: Conceptualization, K.H., P.R. and N.G.; Investigation, K.H.; Methodology, K.H., P.R. and N.G.; Software, K.H.; Supervision, P.R. and N.G.; Writing, K.H., P.R. and N.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Spanish Government through a FPI Ph.D. grant to K. Himri, as well as by the Spanish Project DPI2017-86372-C3-2-R (TWINBOT-GIRONA1000) and the H2020-INFRAIA-2017-1-twostage-731103 (EUMR).

Institutional Review Board Statement: Not relevant as no human or animal subjects were used in this study.

Informed Consent Statement: No human subjects were used in this study.

Data Availability Statement: Data sharing is not applicable to this article as no new data were created in this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhu, Q.; Chen, L.; Li, Q.; Li, M.; Nüchter, A.; Wang, J. 3d lidar point cloud based intersection recognition for autonomous driving. In Proceedings of the 2012 IEEE Intelligent Vehicles Symposium, Madrid, Spain, 3–7 June 2012; pp. 456–461.
2. Chen, C.S.; Chen, P.C.; Hsu, C.M. Three-dimensional object recognition and registration for robotic grasping systems using a modified viewpoint feature histogram. *Sensors* **2016**, *16*, 1969. [\[CrossRef\]](#)
3. Himri, K.; Ridao, P.; Gracias, N. 3D Object Recognition Based on Point Clouds in Underwater Environment with Global Descriptors: A Survey. *Sensors* **2019**, *19*, 4451. [\[CrossRef\]](#)
4. Li, D.; Wang, H.; Liu, N.; Wang, X.; Xu, J. 3D Object Recognition and Pose Estimation From Point Cloud Using Stably Observed Point Pair Feature. *IEEE Access* **2020**, *8*, 44335–44345. [\[CrossRef\]](#)
5. Lee, S.; Lee, D.; Choi, P.; Park, D. Accuracy–Power Controllable LiDAR Sensor System with 3D Object Recognition for Autonomous Vehicle. *Sensors* **2020**, *20*, 5706. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Gomez-Donoso, F.; Escalona, F.; Cazorla, M. Par3DNet: Using 3DCNNs for Object Recognition on Tridimensional Partial Views. *Appl. Sci.* **2020**, *10*, 3409. [\[CrossRef\]](#)
7. Landrieu, L.; Simonovsky, M. Large-scale point cloud semantic segmentation with superpoint graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4558–4567.
8. Lowphansirikul, C.; Kim, K.S.; Vinayaraj, P.; Tuarob, S. 3D Semantic Segmentation of Large-Scale Point-Clouds in Urban Areas Using Deep Learning. In Proceedings of the 2019 11th International Conference on Knowledge and Smart Technology (KST), Phuket, Thailand, 23–26 January 2019; pp. 238–243.
9. Xie, Y.; Tian, J.; Zhu, X.X. A review of point cloud semantic segmentation. *arXiv* **2019**, arXiv:1908.08854.
10. Ma, J.W.; Czerniawski, T.; Leite, F. Semantic segmentation of point clouds of building interiors with deep learning: Augmenting training datasets with synthetic BIM-based point clouds. *Autom. Constr.* **2020**, *113*, 103144. [\[CrossRef\]](#)
11. Gupta, S.; Girshick, R.; Arbeláez, P.; Malik, J. Learning rich features from RGB-D images for object detection and segmentation. In Proceedings of the European conference on computer vision, Zurich, Switzerland, 6–12 September 2014; pp. 345–360.
12. Arbeláez, P.; Maire, M.; Fowlkes, C.; Malik, J. Contour Detection and Hierarchical Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 898–916. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep learning for 3d point clouds: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**. [\[CrossRef\]](#)

14. Fernandes, D.; Silva, A.; Névoa, R.; Simões, C.; Gonzalez, D.; Guevara, M.; Novais, P.; Monteiro, J.; Melo-Pinto, P. Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy. *Inf. Fusion* **2021**, *68*, 161–191. [[CrossRef](#)]
15. Guo, Y.; Bennamoun, M.; Sohel, F.; Lu, M.; Wan, J. 3D object recognition in cluttered scenes with local surface features: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2270–2287. [[CrossRef](#)] [[PubMed](#)]
16. Huang, J.; You, S. Detecting Objects in Scene Point Cloud: A Combinational Approach. In Proceedings of the 2013 International Conference on 3D Vision, Seattle, WA, USA, 29 June–1 July 2013; 3DV '13, pp. 175–182. [[CrossRef](#)]
17. Pang, G.; Qiu, R.; Huang, J.; You, S.; Neumann, U. Automatic 3d industrial point cloud modeling and recognition. In Proceedings of the 2015 14th IAPR International Conference on Machine Vision Applications (MVA), Tokyo, Japan, 18–22 May 2015; pp. 22–25.
18. Kumar, G.; Patil, A.; Patil, R.; Park, S.; Chai, Y. A LiDAR and IMU integrated indoor navigation system for UAVs and its application in real-time pipeline classification. *Sensors* **2017**, *17*, 1268. [[CrossRef](#)]
19. Ramon-Soria, P.; Gomez-Tamm, A.; Garcia-Rubiales, F.; Arrue, B.; Ollero, A. Autonomous landing on pipes using soft gripper for inspection and maintenance in outdoor environments. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 4–8 November 2019; pp. 5832–5839.
20. Kim, Y.; Nguyen, C.H.P.; Choi, Y. Automatic pipe and elbow recognition from three-dimensional point cloud model of industrial plant piping system using convolutional neural network-based primitive classification. *Autom. Constr.* **2020**, *116*, 103236. [[CrossRef](#)]
21. Foresti, G.L.; Gentili, S. A hierarchical classification system for object recognition in underwater environments. *IEEE J. Ocean. Eng.* **2002**, *27*, 66–78. [[CrossRef](#)]
22. Bagnitsky, A.; Inzartsev, A.; Pavin, A.; Melman, S.; Morozov, M. Side scan sonar using for underwater cables & pipelines tracking by means of AUV. In Proceedings of the 2011 IEEE Symposium on Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies, Tokyo, Japan, 5–8 April 2011; pp. 1–10.
23. Yu, S.C.; Kim, T.W.; Asada, A.; Weatherwax, S.; Collins, B.; Yuh, J. Development of High-Resolution Acoustic Camera based Real-Time Object Recognition System by using Autonomous Underwater Vehicles. In Proceedings of the OCEANS 2006, Boston, MA, USA, 18–21 September 2006; pp. 1–6.
24. Yang, H.; Liu, P.; Hu, Y.; Fu, J. Research on underwater object recognition based on YOLOv3. *Microsyst. Technol.* **2020**, 1–8. [[CrossRef](#)]
25. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
26. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
27. Wang, N.; Wang, Y.; Er, M.J. Review on deep learning techniques for marine object recognition: Architectures and algorithms. *Control. Eng. Pract.* **2020**, 104458. [[CrossRef](#)]
28. Chen, Y.; Xu, X. The research of underwater target recognition method based on deep learning. In Proceedings of the 2017 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Xiamen, China, 22–25 October 2017; pp. 1–5.
29. Cao, X.; Zhang, X.; Yu, Y.; Niu, L. Deep learning-based recognition of underwater target. In Proceedings of the 2016 IEEE International Conference on Digital Signal Processing (DSP), Beijing, China, 16–18 October 2016; pp. 89–93.
30. Martin-Abadal, M.; Piñar-Molina, M.; Martorell-Torres, A.; Oliver-Codina, G.; Gonzalez-Cid, Y. Underwater Pipe and Valve 3D Recognition Using Deep Learning Segmentation. *J. Mar. Sci. Eng.* **2020**, *9*, 5. [[CrossRef](#)]
31. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv* **2017**, arXiv:1706.02413.
32. Palomer, A.; Ridao, P.; Ribas, D.; Forest, J. Underwater 3D laser scanners: The deformation of the plane. In *Lecture Notes in Control and Information Sciences*; Fossen, T.I., Pettersen, K.Y., Nijmeijer, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2017; Volume 474, pp. 73–88. [[CrossRef](#)]
33. Neira, J.; Tardós, J.D. Data association in stochastic mapping using the joint compatibility test. *IEEE Trans. Robot. Autom.* **2001**, *17*, 890–897. [[CrossRef](#)]
34. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [[CrossRef](#)]
35. Rusu, R.B.; Cousins, S. 3d is here: Point cloud library (pcl). In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 1–4.
36. Rabbani, T.; Van Den Heuvel, F. Efficient hough transform for automatic detection of cylinders in point clouds. *ISPRS Wg Iii/3, Iii/4* **2005**, *3*, 60–65.
37. Liu, Y.J.; Zhang, J.B.; Hou, J.C.; Ren, J.C.; Tang, W.Q. Cylinder detection in large-scale point cloud of pipeline plant. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 1700–1707. [[CrossRef](#)]
38. Tran, T.T.; Cao, V.T.; Laurendeau, D. Extraction of cylinders and estimation of their parameters from point clouds. *Comput. Graph.* **2015**, *46*, 345–357. [[CrossRef](#)]
39. Xu, Y.; Tuttas, S.; Hoegner, L.; Stilla, U. Geometric primitive extraction from point clouds of construction sites using vgs. *IEEE Geosci. Remote. Sens. Lett.* **2017**, *14*, 424–428. [[CrossRef](#)]

40. Jin, Y.H.; Lee, W.H. Fast cylinder shape matching using random sample consensus in large scale point cloud. *Appl. Sci.* **2019**, *9*, 974. [[CrossRef](#)]
41. Palomer, A.; Ridaio, P.; Ribas, D. Inspection of an underwater structure using point-cloud SLAM with an AUV and a laser scanner. *J. Field Robot.* **2019**, *36*, 1333–1344. [[CrossRef](#)]
42. Aldoma, A.; Vincze, M.; Blodow, N.; Gossow, D.; Gedikli, S.; Rusu, R.B.; Bradski, G. CAD-model recognition and 6DOF pose estimation using 3D cues. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 6–13 November 2011; pp. 585–592.
43. Aldoma, A.; Tombari, F.; Rusu, R.B.; Vincze, M. OUR-CVFH-oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6DOF pose estimation. In *Joint DAGM (German Association for Pattern Recognition) and OAGM Symposium*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 113–122.
44. Rusu, R.B.; Marton, Z.C.; Blodow, N.; Beetz, M. Persistent point feature histograms for 3D point clouds. In Proceedings of the 10th International Conference Intel Autonomous Systems (IAS-10), Baden-Baden, Germany, 23–25 July 2008; pp. 119–128.
45. Hetzel, G.; Leibe, B.; Levi, P.; Schiele, B. 3D object recognition from range images using local feature histograms. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, Kauai, HI, USA, 8–14 December 2001; Volume 2.
46. Rusu, R.B.; Bradski, G.; Thibaux, R.; Hsu, J. Fast 3d recognition and pose using the viewpoint feature histogram. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 18–22 October 2010; pp. 2155–2162.
47. Arun, K.S.; Huang, T.S.; Blostein, S.D. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *9*, 698–700. [[CrossRef](#)]
48. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Berger, E.; Wheeler, R.; Mg, A. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009; Volume 3, p. 5.
49. Palomer, A.; Ridaio, P.; Forest, J.; Ribas, D. Underwater Laser Scanner: Ray-Based Model and Calibration. *IEEE/ASME Trans. Mechatronics* **2019**, *24*, 1986–1997. [[CrossRef](#)]

4

SEMANTIC MAPPING FOR AUTONOMOUS SUBSEA INTERVENTION

This chapter presents a semantic map approach to the problem described in the previous chapters. The approach is based on the integration of feature based SLAM and 3D object recognition using a data base of *a priori* known objects. The object recognition module performs an initial recognition of pipes and objects within a scan, and passes this information to the SLAM. Depending on whether these object have been observed in the past, the SLAM module either adds them to the map or otherwise uses them to correct the map and the robot navigation. The use of the SLAM module has two important advantages: (1) it provides a consistent map and a drift-free navigation, and (2) it provides a global identifier for every observed object instance and its pipe connectivity. This information is then used by the object recognition module, to improve the the recognition using Bayesian inference over the set of object classes compatible in terms of pipe connectivity.

All the proposed work is described in detail on the following submitted journal article:

Title: Semantic Mapping for Autonomous Subsea Intervention
Authors: G. Villacrosa, K. Himri , P. Ridao, and N. Gracias
Journal: Submitted to MDPI Sensors
Volume: , Published: 2021
Quality index: JCR2019 Instruments & Instrumentation IF 3.275, Q1 (15/64)

Article

Semantic Mapping for Autonomous Subsea Intervention

Guillem Vallicrosa *, Khadidja Himri *, Pere Ridao * and Nuno Gracias *

Underwater Robotics Research Center (CIRS), Computer Vision and Robotics Institute (VICOROB),
Universitat de Girona, Parc Científic i Tecnològic de la UdG. C/Pic de Peguera 13, 17003 Girona, Spain

* Correspondence: gvallicrosa@eia.udg.edu (G.V.); khadidja.himri@udg.edu (K.H.); pere@eia.udg.edu (P.R.);
ngracias@silver.udg.edu (N.G.)

Abstract: This paper presents a method to build a semantic map to assist an underwater vehicle-manipulator system in performing intervention tasks autonomously in a submerged man-made pipe structure. The method is based on the integration of feature-based simultaneous localization and mapping (SLAM) and 3D object recognition using a database of a priori known objects. The robot uses Doppler velocity log (DVL), pressure, and attitude and heading reference system (AHRS) sensors for navigation and is equipped with a laser scanner providing non-coloured 3D point clouds of the inspected structure in real time. The object recognition module recognises the pipes and objects within the scan and passes them to the SLAM, which adds them to the map if not yet observed. Otherwise, it uses them to correct the map and the robot navigation if they were already mapped. The SLAM provides a consistent map and a drift-less navigation. Moreover, it provides a global identifier for every observed object instance and its pipe connectivity. This information is fed back to the object recognition module, where it is used to estimate the object classes using Bayesian techniques over the set of those object classes which are compatible in terms of pipe connectivity. This allows fusing of all the already available object observations to improve recognition. The outcome of the process is a semantic map made of pipes connected through valves, elbows and tees conforming to the real structure. Knowing the class and the position of objects will enable high-level manipulation commands in the near future.

Keywords: 3D object recognition; point clouds; global descriptors; semantic segmentation; semantic information; Bayesian probabilities; laser scanner; underwater environment; pipeline detection; inspection, maintenance and repair; AUV



Citation: Vallicrosa, G.; Himri, K.; Ridao, P.; Gracias, N. Semantic Mapping for Autonomous Subsea Intervention. *Sensors* **2021**, *21*, 6740. <https://doi.org/10.3390/s21206740>

Academic Editor: Vassilis S. Kogiannis, John Lygouras

Received: 27 August 2021
Accepted: 29 September 2021
Published: 11 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

State-of-the-art autonomous underwater vehicles (AUVs) are commonly used for seafloor mapping in predominantly flat environments using multiple sensors, including side-scan sonar (SSS), multibeam echosounder (MBES), forward-looking sonar (FLS) and cameras, among others. The use of unmanned underwater vehicles (UUVs) for inspection, maintenance and repair (IMR) applications is nowadays limited to the use of remotely operated vehicles (ROVs) in inspection and/or intervention tasks. Nevertheless, during the last decade, the research community has made a significant effort defining a new class of UUV, the intervention autonomous underwater vehicle (I-AUV). This class of vehicles is expected to replace intervention ROVs in IMR tasks in the future [1]. Though several autonomous manipulation tasks have already been demonstrated, often only in water tank conditions, most are just proof of concept demonstrations oriented to very particular targets. Tasks such as valve turning [2,3], connector plug/unplug [4] and object search and recovery [5] are clear examples. Nevertheless, in all these tasks, custom algorithms have usually been used to detect and track a particular manipulation goal. Often the targets have been labeled with markers to simplify the problem, or the robot was limited to performing a particular manipulation action over a particular target object. In contrast, a truly autonomous I-AUV should be able to obtain and use semantic knowledge of its

surroundings. As such, the vehicle should be capable of identifying which objects are around it, which class they belong to, and which tasks can be performed on them. For instance, if a safety valve has to be manipulated in case of an alarm, the I-AUV needs to know which valve, where it is and how it can be opened or closed. This leads to the semantic map concept—a map containing the objects position and their specific class. Semantic mapping is a key technique to endow the I-AUV with autonomous reasoning capabilities.

1.1. Objectives

This paper tackles the semantic map building problem for an I-AUV equipped with a real-time high-resolution laser scanner and working on IMR operations. It extends our prior work [6], where a point feature-based 3D object recognition method was proposed. The method used Bayesian estimation as a probabilistic framework to integrate multiple detections into a single, and more robust, object class identification. To do so, it was necessary to track objects along the sequentially grabbed scans. For this purpose, a Interdistance Joint Compatibility Branch and Bound (IJCBB) object tracking method was proposed, which was able to track the objects in the presence of navigation glitches due to sporadic failures of the Doppler velocity log (DVL) measurements. Moreover, the method exploited semantic information related to object pipe connectivity (number of pipes connected to the object) to constrain the potential set of compatible object classes used during the Bayesian estimation. Nevertheless, the IJCBB must establish at least three pairings between two scans to be able to register them. Otherwise, the tracking fails and the object detections in this scan cannot contribute to the Bayesian estimation. On the other hand, the iterative nature of the tracking algorithm reduces the drift, but is not able to cancel it. Therefore, the natural next step is to employ simultaneous localization and mapping (SLAM) techniques using the pipes and objects as features to build a drift-less consistent map of the structure. Using conventional data association algorithms, between the objects in a scan and the objects in the SLAM, it is possible to track the objects and apply the Bayesian estimation. The outcome of the process is a semantic map of pipes and objects, which provides the I-AUV with an accurate navigation as well as with the semantic knowledge of the manipulable objects around it.

1.2. Contributions

The main contributions of the present paper are the following:

- A feature-based extended Kalman filter (EKF) SLAM method is proposed which uses line and point features to represent the pipes and the objects, respectively. The method solves two problems: (1) it provides a drift-less navigation; and (2) it assigns a globally consistent identifier to every object in every scan, enabling Bayesian estimation. When conveniently combined with the object recognition results, it becomes a semantic map endowing the I-AUV with the semantic knowledge required to perform high-level commands, such as *Open Valve X*, for example.
- It provides a method for plane segmentation which partitions the point cloud according to the average maximum curvature and classifies the partitions either as planes or as a curved region. The method allows separation of the flat surfaces corresponding to the walls of the water tank, where the experiment was performed, from the pipe structure itself.
- It provides an extension to the semantic object segmentation method already proposed in [6], ensuring the correct segmentation of the valve handle, which proved problematic in the previous paper.

1.3. Structure of the Paper

The remainder of the paper is organized as follows. Section 2 describes the state of the art on underwater SLAM, object recognition and semantic mapping. Section 3 describes the object recognition pipeline from the segmentation of the scans to the Bayesian recognition. Section 4 describes the feature-based SLAM for object and pipe feature tracking. Section 5

describes the experimental setup and the results obtained. Sections 6 and 7 provide conclusions and future work on the results obtained.

2. State of the Art

2.1. Underwater SLAM

Many outdoor field robots rely on absolute measurements to bound the dead reckoning (DR) navigation drift, such as the Global Positioning System (GPS). However, in underwater robotics, those sensors are unavailable due to electromagnetic attenuation; underwater robots instead have to rely on acoustic localization methods such as long baseline (LBL) [7], short baseline (SBL) [8], ultra-short baseline (USBL) [9] or GPS intelligent buoys (GIB) [10]. Those methods require deployment of the beacons and/or a support vessel to provide the GPS positioning to be composed with the measured acoustic position. Unfortunately, those methods restrict the vehicle to a predefined zone (LBL) or decrease their precision with increasing depth of the vehicle (SBL, USBL and GIB).

A solution to overcome these issues and have a completely independent AUV is to correlate the vehicle sensor measurements with a map of the environment to reliably locate its position with Terrain-Based Navigation (TBN) techniques [11]. However, precise maps are not widely available, and so many researchers rely on SLAM methods, where the robot incrementally builds a model of the environment and simultaneously uses it to estimate its position within it.

Underwater SLAM can be categorised according to the type of sensors used to perceive the environment. On the one hand, vision-based sensors perceive the environment at high rates and high precision, but they are very sensitive to water visibility, which greatly limits their range. On the other hand, acoustic-based sensors provide low-rate and low-precision measurements regardless of visibility. Regarding acoustic SLAM, we can further classify SLAM into feature-based and featureless methods. Feature-based methods are generally used in man-made environments, where features are easier to extract [12–14], while featureless methods are primarily used in natural environments [15–21].

In contrast, underwater vision-based SLAM relies heavily on visual features extracted from the texture of the environment [22–27]. If the environment is texture-less, an alternative is to use laser-camera systems, where the laser produces the necessary texture to extract point clouds from the environment. Initial developments of this approach relied on a fixed laser scanner that, combined with the vehicle motion, produces the point clouds [28,29], but suffers from navigation drift.

A new laser scanner based on a moving mirror provides scans at a maximum rate of 6 Hz, fast enough to allow the vehicle drift during a single scan to be neglected [30]. This laser scanner has already been tested on motion planning applications in an unknown environment [31] and in a pose-based SLAM for mapping [32]. In the present work, we focus on the application of this laser scanner to semantically extract features that serve as input for the SLAM algorithm and ease the recognition of the object features on pre-trained models of the different objects.

2.2. Object Recognition

Object recognition is a domain of 3D scene exploration and understanding associated with applications such as autonomous driving and housekeeping robots. 3D object recognition has emerged thanks to pre-existing 2D methods translated into 3D and the advanced availability of different types of 3D sensors.

In the field of object recognition based on point clouds, several surveys have been carried out in which methods and ideas based on global and local descriptors have been presented [33–35]. Global recognition methods interpret the entire object as a unique vector of values, while local recognition methods focus more on a local region and are computed from salient points. Recently, deep learning has gained increasing attention. The following two publications are representative examples. In [36], Guo et al. summarized deep learning methods applied to 3D point clouds. The authors aimed to select the most

relevant applications for point cloud understanding, considering 3D shape classification, 3D object detection and tracking, and 3D point cloud segmentation. They evaluated the quality of the performance of state-of-the-art methods based on deep learning and compared the methods with different publicly available datasets. In Tian et al. [37], the authors proposed a dynamic graph convolutional broad network (DGCB-Net) for feature extraction and object recognition from point clouds, and their method was tested on several public datasets and one dataset which they collected.

However, fewer papers have focused on underwater application scenarios, with the exception of the paper by Martin et al. [38], in which a processing pipeline is presented, based on the use of a deep PointNet neural network. The proposed method was able to detect pipes and valves from 3D RGB point clouds in underwater environments using a generated dataset to train and test the network. Recent work by Pereira et al. [39] is also based on a deep learning approach, where a convolutional neural network was used for recognizing a docking structure from point clouds. Their methods were evaluated with simulated and real datasets.

Although deep learning approaches have been reported to have attained accurate results, such methods are very demanding in terms of the amount of training data to ensure proper learning generalization. In the case of man-made structures observed by sensors that provide only colourless point clouds, the collection of the required training data is a difficult and time-consuming task.

The work described in Martin et al. [38] used a similar man-made structure as the one in our work, comprising valves interconnected by pipes. Furthermore, the experiments in both papers were conducted in an underwater environment. Their work is directly related to the problem we are trying to solve, i.e., the recognition of man-made objects underwater, because it formed part of the same research project TWINBOT [40] in which both groups participated. In the following paragraphs, we provide a comparison of the two works, which highlights the trade-offs between the two approaches.

- In the present work, we have used a feature-based SLAM approach to object recognition using a 3D point cloud with no RGB information, obtained with a laser scanner. The process can be summarised as follows:
 - The segmentation of the ground was performed using the methods explained in Section 3.2. The segmentation of pipes was performed separately from the recognition of the objects;
 - Five object classes were defined in the experiments, which were segmented based on the pipe connections;
 - The knowledge database was generated from the object's CAD model using a process described in our previous article [33]. The test data was collected in the test pool of our laboratory, and included 1268 point clouds for individual objects, extracted from 245 laser scans;
 - The main recognition performance results are found in Section 5.4.
- In the work of Martin et al. [38], a deep learning approach was applied for the detection of pipes and valves. The network used, as input, 3D point clouds with RGB information obtained from stereo cameras, and the following steps were performed:
 - Ground truth data were manually created from the point cloud, and divided into three classes: pipes, valves and background;
 - Two datasets were used. The first dataset was acquired in a test tank and contained 262 point clouds. This dataset was divided into two subsets, the first containing 236 point clouds which were used to train the network and the remainder used as test samples. The second dataset was collected in the sea and included 22 point clouds that were used only as a test set;
 - 13 experiments were conducted varying the hyper-parameters in the training phase: batch size, learning rate, block-stride and number of points;

- To assess the performance of the neural network and estimate how the model is expected to perform, a 10-fold cross-validation was performed. Overall, 9 subsets of 213 point clouds were used for training, and 1 subset of 23 point clouds was used for testing. The final classification result was obtained by averaging the performance of these ten different results;
- From the results presented, it can be seen that the background class was predominant, followed by the pipe and valve classes in both pool and sea experiments.

2.3. Semantic Mapping

Semantic mapping started indoors with scene recognition [41–44] and then moved outdoors. It has been applied on various input data, such as cameras [45,46], depth cameras [47,48] or laser scanners (usually LIDARs) [49–51]. Implementations vary from supervised to unsupervised methods, where semantic classes are a priori unknown.

Adding semantic information to underwater maps contributes to a better spatial awareness of nearby terrains and objects, enabling higher-level tasks to be performed. This is especially important for IMR tasks where robots have to be aware of the different components and how to interact with them.

In the underwater environment, it has been mainly used for semantic image segmentation [52,53], which can also be applied to exploration [54]. To the best of the authors' knowledge, semantic mapping has not yet been applied to point clouds obtained underwater with a laser scanner for IMR tasks, and thus, this paper goes beyond the state of the art.

3. Object Recognition Pipeline

As can be seen in Figure 1, the object recognition pipeline is divided into several modules. First, the floor and lateral walls/slopes of the water tank where the experiment takes place are segmented and subtracted from the scanned point-cloud. Then, pipes are detected and the resulting point cloud is used as input for the semantic object segmentation. Having extracted the planes and the pipes from the scan, objects are segmented.

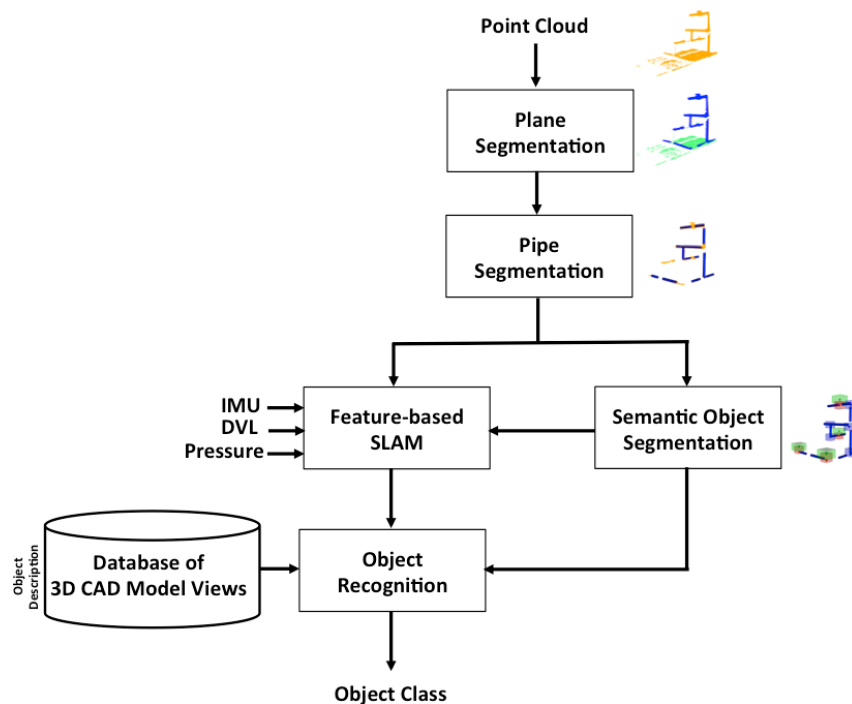














Figure 1. 3D object recognition pipeline.

A feature-based SLAM is continuously running, integrating DVL, pressure and attitude and heading reference system (AHRS) measurements. The input pipes and objects are used as features of the SLAM, which simultaneously estimates the robot pose, and the position of the already-observed pipes and objects. Therefore, solving the association of the objects segmented from the scan with those already mapped, it is possible to associate a global identifier with them. Finally, the object recognition module uses the point feature descriptors of the partial views of the segmented objects, matching them against those stored in the object database, identifying the object class. Since the global identifier of the observed object instance is known thanks to the SLAM output, it is possible to use several past object class estimations to compute its global object class, achieving more robust results. Hereafter, the different modules are described in more detail.

3.1. Object Data Base

A database of point clouds was created (Table 1), containing overlapping partial views of isolated objects. These views were created from 3D CAD models and captured using a virtual camera. This database was useful for the design of simulated experiments and for their statistical analysis, as presented in our previous work [6]. Details on the creation of the database can be found in the same publication.

Table 1. Polyvinylchloride (PVC) pressure pipe objects used in the experiments (reprinted with permission from ref. [6]. 2021 Sensors)

PVC Objects	Id Name	Size (mm ³)	PVC Objects Views (12)
	1-Ball-Valve	198 × 160 × 120	
	2- Elbow	122.5 × 122.5 × 77	
	3- R-Tee	122.5 × 168 × 77	
	4- R-Socket	88 × 75 × 75	
	5- Butterfly-Valve	287.5 × 243 × 121	
	6- 3-Way-Ball-Valve	240 × 160 × 172	

3.2. Plane Segmentation

In our previous work [6], planes were detected using random sample consensus (RANSAC). Unfortunately, in several scans, the principal plane detected did not correspond to the floor or the walls of the water tank. Sometimes, points belonging to different pipes and even objects, and others belonging to the slopes, became co-planar, forming the most significant plane in the scene. However, removing it would wrongly eliminate a significant number of points in the pipes and objects, making the recognition more challenging. To avoid this problem, an alternative method is proposed in this paper.

The problem of plane segmentation can be seen as an unsupervised classification problem, where the goal is to group the points into regions defined according to their curvature, which is an attribute describing the local geometry around a point. In Point Cloud Library (PCL), the curvature of a point is computed performing an eigen-decomposition of the points in the neighbourhood. The eigenvector corresponding to the smallest eigenvalue provides the direction of the normal, and the other two provide the tangent plane. The curvature κ is defined as the ratio between the smallest eigenvalue and the addition of the three eigenvalues:

$$\kappa = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \text{ where } \lambda_0 < \lambda_1 < \lambda_2. \quad (1)$$

To remove the planar surfaces, first we segmented the point cloud into several regions using the region-growing method [55]. The algorithm begins by selecting as a seed point the one with least curvature. Then, the region is computed by growing the seed to those adjacent points in the neighbourhood whose angles between normals (the normal of the seed and the local normal at the point) are within a pre-defined threshold. Next, the points within the region with a curvature below a threshold are considered as new seeds, and the algorithm is iterated until no more seeds are available. At this point, the first region has been segmented and the algorithm is applied again to the rest of the point cloud. The result is a set of regions having a smooth evolution of the angle among their normals. The regions are separated either for having a sudden change in their normals (smoothness), or because they are spatially separated, as shown in Figure 2. The threshold angle between the normal vectors was set to 30 degrees. If the points are on the same plane, then the normals of the fitting planes of these two points are approximately parallel.

Second, the resulting regions are classified into two categories based on an empirical threshold on their mean curvature (Fig. 2). We evaluated the curvature of each region in a neighbourhood of 50 points and chose an empirical threshold of 0.025 (Fig. 3). Each region from the growing regions result is classified as: (a) points on flat areas such as the bottom and the slopes on both sides of the water tank, or (b) points on the rest of the cloud, such as objects and pipes of the structure.

Subsequently, the flat regions are deleted, and the remainder are merged into a single region containing the non-flat areas to be further processed. The proposed plane segmentation method is shown in Algorithm 1.

Algorithm 1: Plane Segmentation

```

1 function RegionsGrowingSegmentation(in: scan, out: RI):
  | // Returns the set of region Ri detected in the scan using Growing
  |   Regions Algorithm
2 return {RI}
3 function MergeRegions(in: RI, out: SRI, PRI):
4   if (Rcurvature > τd) then // Non planar?
  |   | // Returns a pipes and objects (structure) regions (SRI) result
  |   |   of merging the input set of non-plane regions
5   |   else
  |   |   | // Returns a plane regions (PRI) result of merging the input
  |   |   |   set of plane regions
6 return {< SRI, PRI >}
7 procedure PlaneSegmentation(in: scan; out: SRI, PRI):
8   | RI=RegionsGrowingSegmentation(scan) // Set of regions Ri
9   | forall Ri ∈ RI do
10  |   | {< SRI, PRI >}=MergeRegions(Ri)

```

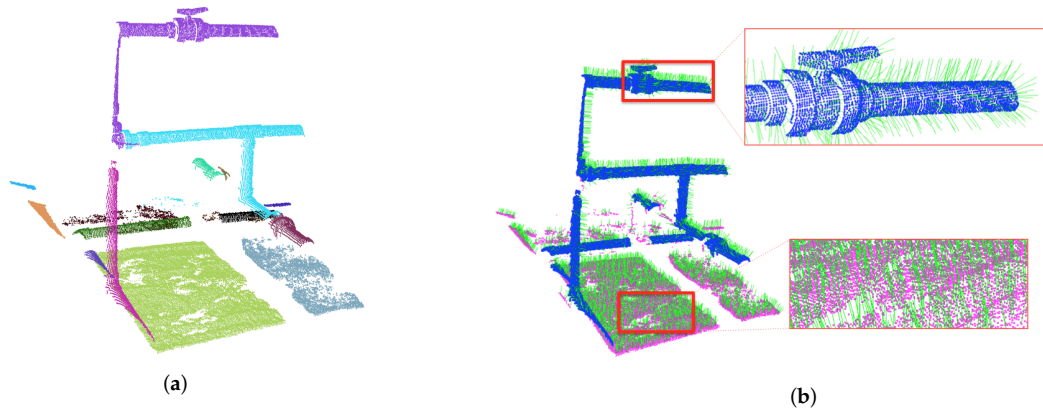


Figure 2. Plane segmentation. (a) Outcome regions of the region-growing method. (b) Segmentation of the point cloud into two regions with normals in green: I) non-flat areas in blue, and II) flat areas in pink.

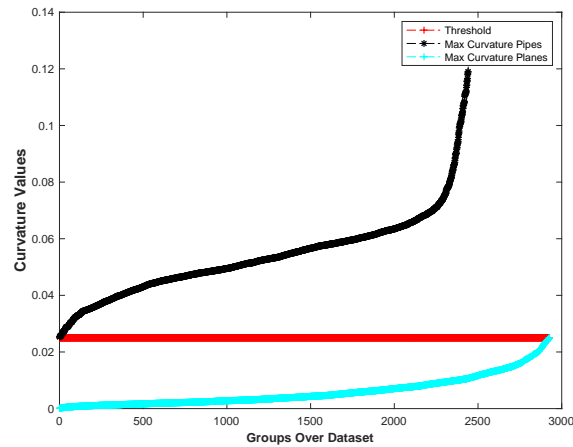


Figure 3. Mean curvature threshold separating the pipes from the flat areas. The horizontal axis represents, for all the dataset scans, the regions obtained using the region-growing method. The vertical axis provides, for each region, its mean curvature.

3.3. Pipe Detection

For detecting pipes in the current scan, a method based on the RANSAC implementation in PCL was used. This method models the pipes as cylinders with seven parameters, consisting of the 3D position of a point on the axis, axis direction, and cylinder radius. The scan is divided into two categories, namely the pipe cloud category and non-pipe cloud category. Since the radius of the pipes is known and objects have a maximum size, only the segmented cylinders with length more than 0.30 m and maximum radius of 0.064 m are considered as pipes. To calculate the endpoints of the pipes, the selected set of points is projected onto the pipe axis, and the points at the extreme ends are considered as the limits of the pipe.

Scan deformations caused by motion-induced distortions during the acquisition of the laser scan [32] can occasionally lead to two different detections being generated for the same pipe. The solution for such cases as well as details on the implementation of the pipe detection are provided in [6]. An example of pipe detection with their respective endpoints is given in Figure 4.

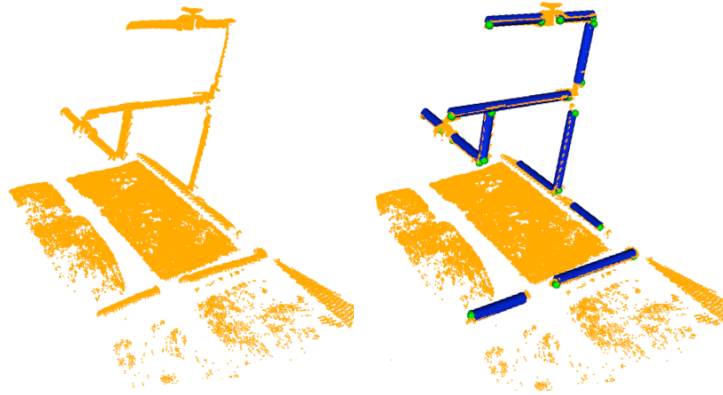


Figure 4. Pipe detection: (left) 3D laser scan point cloud; (right) Pipes in blue with their respective endpoints in green.

3.4. Semantic Object Segmentation

The proposed semantic 3D object segmentation is inspired and motivated by the fact that objects are found at the extremities of pipes. Knowledge about these objects includes detailed information about the connectivity of the objects and structural knowledge, such as the fact that valves with two parallel connections are characterised by handles, which is an important feature for objects like butterfly valves to distinguish them from their homologous valves. In addition, functional knowledge is needed for these features, allowing the robot to infer whether the valves can be turned on or off based on the position of a handle. To this end, the semantic segmentation problem can be formulated as follows: Given an object with one or two parallel connections (Figure 5), it is possible to find a potential handle, as shown in the right part of the figure, where objects with one or two parallel connections are segmented using a ‘mushroom’ shape (green cube on the top of the red one). The base is defined for the body of the object and the parallel pipe shape for the potential handle, while if the object has more connections or two perpendicular connections, only the base is segmented, as shown with blue cubes.

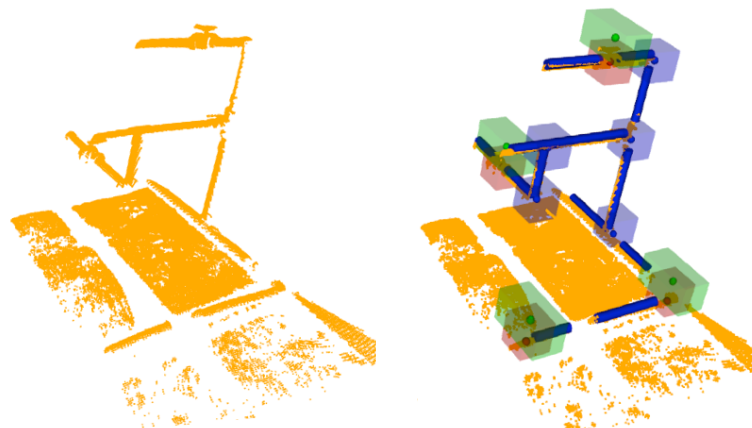


Figure 5. Semantic object segmentation: (left) 3D laser scan point cloud; (right) Example of segmentation and how objects with different connectivity are treated differently.

3.5. 3D Object Recognition Based on Global Descriptors

Object recognition is an essential part of building a semantic map of the environment. In [33] we studied and compared several descriptors using synthetic and real data. The best results involving experimental data were achieved using the Clustered Viewpoint Feature Histogram (CVFH) [56] descriptor, which is therefore used in this paper.

3.6. Bayesian Recognition

A disadvantage of object recognition with a single-view approach is that multiple objects may have similar views. A study based on the confusion matrices for the various objects was carried out in our previous work [33]. Given a set of observations of a particular object, we can use confusion matrices to determine how many observations were recognized as *object-class-n*, where *n* indicates the class name of the object. Given this information, we can estimate a probability for each class as well as the confusion between classes, which is used to implement a Bayesian estimation method to improve object recognition results.

For this purpose, several observations were combined to calculate the probability that an object belongs to each object class. The selected object was assigned to the class with the highest probability. This method required continuous observation of the same objects across the scans, so a tracking method was required to iteratively compute the Bayesian probabilities. In our previous work [6], this tracking was performed using a navigation-less variant of the Joint Compatibility Branch and Bound (JCBB) algorithm, based on the distances between objects within a scan, and referred to as IJCBB. In the present work, we use the SLAM solution described in Section 4, which achieves significantly higher performance.

3.7. Bayesian Estimation

In order to solve the common problem of ambiguous observations caused by having only partial views of the objects in the scans, a Bayesian estimator is applied. In [33] we have already computed the object confusion matrix; this matrix is used as an estimate of the required conditional probabilities. The object class recognised with the global descriptor is denoted as Z_C . X is the actual class of this object, and *Ball-Valve*, *Elbow*, *R-Tee*, *R-Socket*, *Butterfly-Valve*, *3-Way-Valve* are potential class candidates, sub-indexed with numbers 1 to 6 respectively. $P(Z_C|X_i)$ indicates the probability that the object is recognised as class Z_C when its actual class is X_i . If $C = i$ then it is a true positive (TP), otherwise ($C \neq i$) it is a false positive (FP).

3.8. Semantic-Based Recognition

By knowing the number of pipes connected to the object and their geometry, the recognition rate can be further improved. This method was presented in [6] and is briefly summarized here for completion.

The information about the number of pipe connections and their geometry is used to reduce the set of possible classes for a given object by considering only those classes that are compatible with that configuration. For example, if we know that an object is connected to 3 pipes, then only 2 candidate classes are possible: the *R-Tee* and the *3-Way-Valve*. Thus, the Bayesian probabilities are computed only for the compatible candidate classes and considered zero for the rest.

Four different geometric configurations may arise:

Configuration 1 Three pipes: two collinear and one orthogonal. This group contains the *R-Tee* and the *3-Way-Valve*;

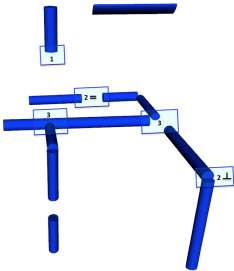




Configuration 2 Two orthogonal pipes: This group contains the *Elbow* but also the members of the previous group, since it is possible that the third pipe has not yet been observed;

Configuration 3 Two collinear pipes: All objects are included in this group, except the *Elbow* and the R-Sockets. The remaining objects admit a collinear connection to two pipes;

Configuration 4 Single or no connection: All objects are considered as potential candidates.

It can be seen from Table 2 that these configurations have a hierarchy in the sense that the first is the most restrictive, the second is less restrictive and encompasses the objects of the first group, and so on. One exception is group 3, for 2 collinear pipes where the *Elbow* of group 2 is not present. It is worth noting that the laser scanning process often provides only partial views of the objects due to occlusions and the limited field of view. As such, a certain object may appear as connected to a single pipe in the first observation, then connected to three pipes on the second observation and then just to a single pipe in the third observation. Since objects are mapped in the SLAM, we can use the knowledge of the previously observed configurations to better compute the probabilities. As an example, if an object is observed in configuration 1 and then configuration 2, then the probabilities for the second observation will be computed as for configuration 1 (which is the most restrictive).

Table 2. Semantic connection of objects. The number of pipes connected to an object is indicated by n_p (reprinted with permission from ref. [6]. 2021 Sensors).

Type of Connection	Pipe Disposition			Potential Object Candidates
	n_p	=	\perp	
	3	2	1	
	2	0	2	
	2	2	0	
	1 0	1 0	1 0	

4. Simultaneous Localization and Mapping for Object and Pipe Tracking

Once the pipes and objects are segmented from the scans, they are sent as input to a SLAM algorithm that integrates AUV navigation with those features in order to improve navigation and track the features, keeping a single global ID for each of them. The output of the SLAM to the semantic Bayesian recognition are the global IDs for each object detected in the scan. This ensures that different observations of the same object are used together to better estimate the object class.

4.1. Line Feature Representation

The pipes are represented using an ortho-normal line representation [57] consisting of three angles of rotation ($\alpha \beta \gamma$) and the shortest distance from the frame origin to the line ρ (2) (Figure 6).

$$L = [\alpha \quad \beta \quad \gamma \quad \rho] \quad (2)$$

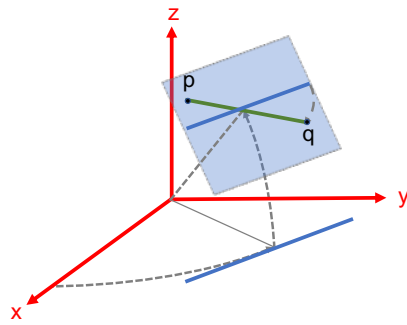


Figure 6. Line feature parametrization.

Given the segment endpoints (p and q) provided by the pipe detection algorithm (see Section 3.3), the ortho-normal representation is computed using Plücker coordinates [58].

$$\mathbf{n} = \mathbf{p} \times \mathbf{q} \quad (3)$$

$$\mathbf{v} = \mathbf{q} - \mathbf{p} \quad (4)$$

$$\mathbf{n}_u = \mathbf{n} / \|\mathbf{n}\| \quad (5)$$

$$\mathbf{v}_u = \mathbf{v} / \|\mathbf{v}\| \quad (6)$$

$$\mathbf{r}_u = \mathbf{v}_u \times \mathbf{n}_u \quad (7)$$

$$\mathbf{R} = [\mathbf{r}_u \ \mathbf{v}_u \ \mathbf{n}_u] = \text{Rot}(\gamma, z) \text{Rot}(\beta, y) \text{Rot}(\alpha, x) \quad (8)$$

$$\rho = \|\mathbf{n}\| / \|\mathbf{v}\| \quad (9)$$

where \mathbf{v}_u represents the line direction and \mathbf{n}_u is perpendicular to the plane formed by the two endpoints and the frame origin (Figure 7). The three angles of rotation can be extracted from the rotation matrix \mathbf{R} as:

$$\alpha = \text{atan2}(v_{uz}, n_{uz}) \quad (10)$$

$$\beta = \text{asin}(r_{uz}) \quad (11)$$

$$\gamma = \text{atan2}(r_{uy}, r_{ux}) \quad (12)$$

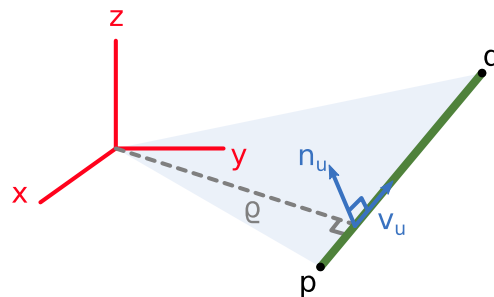


Figure 7. Ortho-normal representation of a pipe segment.

The line is computed from the pipe endpoints which are known in the vehicle sensor frame $\{S\}$, which is the frame of reference of the point cloud. Therefore it is initially referenced to $\{S\}$ and has to be transformed to the world frame $\{W\}$:

$${}^W\rho \cdot {}^W\mathbf{n}_u = {}^W\mathbf{R}_S \cdot {}^S\rho \cdot {}^S\mathbf{n}_u + {}^W\mathbf{t}_S \times ({}^W\mathbf{R}_S \cdot {}^S\mathbf{v}_u) \quad (13)$$

$${}^W\mathbf{v}_u = {}^W\mathbf{R}_S \cdot {}^S\mathbf{v}_u \quad (14)$$

where ${}^W\mathbf{R}_S$ and ${}^W\mathbf{t}_S$ are, respectively, the rotation and the translation that transform from the sensor frame $\{S\}$ to the world frame $\{W\}$.

Similarly, the opposite transformation is computed as:

$${}^S\rho \cdot {}^S\mathbf{n}_u = {}^W\mathbf{R}_S^T \cdot {}^W\rho \cdot {}^W\mathbf{n}_u - {}^W\mathbf{R}_S^T \cdot {}^W\mathbf{t}_S \times {}^W\mathbf{v}_u \quad (15)$$

$${}^S\mathbf{v}_u = {}^W\mathbf{R}_S^T \cdot {}^W\mathbf{v}_u \quad (16)$$

From this, we can calculate the frame change Jacobians with respect to the line representation in the frame and to the sensor position in the world.

4.2. State Vector

The state is represented with the Gaussian random vector $\mathbf{x}(k)$:

$$\mathbf{x}(k) = [\mathbf{x}_v(k) \quad \mathbf{f}_1(k) \quad \mathbf{f}_2(k) \quad \dots \quad \mathbf{f}_n(k)]^T \quad (17)$$

defined by 2 parameters, the mean:

$$\hat{\mathbf{x}}(k) = [\hat{\mathbf{x}}_v(k) \quad \hat{\mathbf{f}}_1(k) \quad \hat{\mathbf{f}}_2(k) \quad \dots \quad \hat{\mathbf{f}}_n(k)]^T \quad (18)$$

and the covariance matrix $\mathbf{P}(k)$, which provides the covariance of the vehicle and the feature lines, as well as their cross-correlations:

$$\mathbf{P}(k) = E([\mathbf{x}(k) - \hat{\mathbf{x}}(k)][\mathbf{x}(k) - \hat{\mathbf{x}}(k)]^T) = \begin{bmatrix} \mathbf{P}_v(k) & \mathbf{P}_{vf_1}(k) & \dots & \mathbf{P}_{vf_n}(k) \\ \mathbf{P}_{f_1v}(k) & \mathbf{P}_{f_1}(k) & \dots & \mathbf{P}_{f_1f_n}(k) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{f_nv}(k) & \mathbf{P}_{f_nf_1}(k) & \dots & \mathbf{P}_{f_n}(k) \end{bmatrix} \quad (19)$$

The vehicle state $\mathbf{x}_v = [x \ y \ z \ \phi \ \theta \ \psi \ u \ v \ w]^T$ has nine dimensions, including vehicle position $[x \ y \ z]^T$ and the vehicle orientation $[\phi \ \theta \ \psi]$, both represented in the world reference frame $\{W\}$. This frame is located at the water surface, being aligned with the north (i.e., north-east-down (NED) reference frame). The linear velocities $[u \ v \ w]^T$, instead, are referenced to the vehicle's frame $\{B\}$. This is also the minimum dimension of the state vector at the beginning of the execution. The state vector is initialized with the vehicle at rest on the surface when the first depth, AHRS and DVL measurements are received.

The line and object features $[\mathbf{f}_1(k) \ \mathbf{f}_2(k) \ \dots \ \mathbf{f}_n(k)]$ are static and defined in the world reference frame $\{W\}$. Line features are represented with ortho-normal coordinates (see Section 4.1) and objects are represented by their coordinates xyz . The number of line features in the state vector is represented by n_l and the number of object features is represented by n_o , with the total number of features being $n = n_l + n_o$.

4.3. Prediction

A six degrees of freedom (DoF) constant-velocity kinematics model is used to predict the vehicle state evolution from time $k - 1$ to time k . The attitude rate of change (Euler angle derivatives), available from the AHRS, is used as the system input ($\mathbf{u}(k) = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$). The uncertainty is modeled as a white Gaussian noise in linear acceleration (w_l) and attitude velocity (w_a). This model can be formulated as:

$$\mathbf{x}_v(k|k-1) = f(\mathbf{x}_v(k-1), \mathbf{u}(k), \mathbf{w}(k)) \quad (20)$$

$$\mathbf{x}_v(k|k-1) = \begin{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \mathbf{Rot}(\phi, \theta, \psi) \left(\begin{bmatrix} u \\ v \\ w \end{bmatrix} \Delta t + \mathbf{w}_l \frac{\Delta t^2}{2} \right) \\ \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} + (\mathbf{u} + \mathbf{w}_a) \Delta t \\ \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \mathbf{w}_l \Delta t \end{bmatrix} \quad (21)$$

where Δt is the time between $k-1$ and k , and $\mathbf{w} = [\mathbf{w}_l \ \mathbf{w}_a] \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ is a white Gaussian noise representing the uncertainty of the linear acceleration $\mathbf{w}_l = [w_{\dot{u}} \ w_{\dot{v}} \ w_{\dot{w}}]$ and the attitude velocity $\mathbf{w}_a = [w_{\dot{\phi}} \ w_{\dot{\theta}} \ w_{\dot{\psi}}]$. In contrast, the features are static and are kept constant throughout the prediction. Hence, the whole state can be predicted using:

$$\mathbf{x}(k|k-1) = [f(\mathbf{x}_v(k-1), \mathbf{u}(k), \mathbf{w}(k)) \quad \mathbf{f}_1(k-1) \quad \mathbf{f}_2(k-1) \quad \dots \quad \mathbf{f}_n(k-1)]^T \quad (22)$$

4.4. Navigation Sensor Updates

The different navigation sensors present on the vehicle (pressure sensor, DVL and AHRS) provide direct observations of the state vector. Therefore, a linear observation model can be used. The general model in this case is:

$$\mathbf{z}(k) = \mathbf{H}(k) \cdot \mathbf{x}(k|k-1) + \mathbf{m}(k) \quad (23)$$

where \mathbf{z} is the measurement vector, and $\mathbf{m} \equiv \mathcal{N}(\mathbf{0}, \mathbf{R})$ is a white Gaussian noise vector with $\mathbf{0}$ mean and covariance \mathbf{R} . The size of the observation matrix \mathbf{H} , as well as the size of \mathbf{R} , changes between the different types of observations.

A pressure sensor produces a 1 DoF position measurement which is a direct observation of the vehicle's depth (i.e., z position). Therefore, the resulting observation matrix is:

$$\mathbf{H}_{\text{DEPTH}}(k) = [0 \quad 0 \quad 1 \quad \mathbf{0}_{1 \times 6} \quad \mathbf{0}_{1 \times (4n_l + 3n_o)}] \quad (24)$$

and $\mathbf{R}_{\text{DEPTH}}$ is the covariance of the pressure sensor:

$$\mathbf{R}_{\text{DEPTH}} = \sigma_{\text{DEPTH}}^2 \quad (25)$$

An AHRS produces 3 DoF angular measurements, which are direct observations of the vehicle attitude (Euler angles). The resulting observation matrix is:

$$\mathbf{H}_{\text{AHRS}}(k) = [\mathbf{0}_{3 \times 3} \quad \mathbf{I}_{3 \times 3} \quad \mathbf{0}_{3 \times 3} \quad \mathbf{0}_{3 \times (4n_l + 3n_o)}] \quad (26)$$

and the covariance matrix \mathbf{R}_{AHRS} is a 3×3 square matrix with the uncertainties of each angle observation:

$$\mathbf{R}_{\text{AHRS}}(k) = \begin{bmatrix} \sigma_{\phi}^2 & 0 & 0 \\ 0 & \sigma_{\theta}^2 & 0 \\ 0 & 0 & \sigma_{\psi}^2 \end{bmatrix} \quad (27)$$

A DVL produces 3 DoF velocity measurements, which are direct observations of the vehicle velocity in its own frame:

$$\mathbf{H}_{\text{DVL}}(k) = [\mathbf{0}_{3 \times 3} \quad \mathbf{0}_{3 \times 3} \quad \mathbf{I}_{3 \times 3} \quad \mathbf{0}_{3 \times (4n_l + 3n_o)}] \quad (28)$$

and the covariance matrix R_{DVL} is a 3×3 square matrix with the uncertainties of each velocity estimation.

$$R_{DVL}(k) = \begin{bmatrix} \sigma_u^2 & 0 & 0 \\ 0 & \sigma_v^2 & 0 \\ 0 & 0 & \sigma_w^2 \end{bmatrix} \quad (29)$$

4.5. Line Feature Observation

From the pipe detector (see Section 3.3), line features are received as pairs of endpoints in the sensor frame $\{S\}$. A first merging filter is used to join collinear segments onto bigger segments. This is done by checking the point-to-line distance of the endpoints against the line defined by the other segment and vice-versa. If all the distances are below a threshold, the segments are joined and the longest possible segment from the two pairs of endpoints is retained (Figure 8).

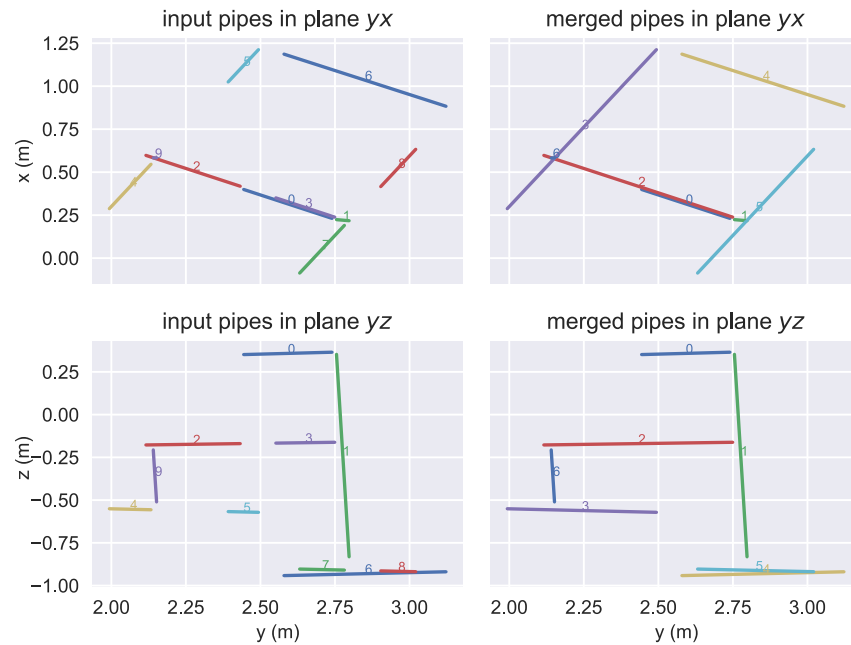


Figure 8. (left) Original pipes received from the pipe detector. (right) Merged pipes before SLAM update.

As observations of a highly angular structure, the angular threshold between lines is not very sensitive, and in this case, a maximum value of 0.175 rad is used. However, the distance threshold is more sensitive due to the existence of parallel lines. A maximum value around the half distance between the closest lines in the real structure, 0.3 m, is used.

The merged segments are converted to the line feature representation in the sensor frame using Equations (3)–(12). The first step in the feature update process is feature association. Already mapped features in the state vector are transformed to the sensor frame together with their uncertainty. A JCBB algorithm is used to ensure consistency in the associations, as opposed to standard individual compatibility [59]. Once this association is solved, we have two kinds of observations: re-observed features that were already in the state vector, or new features that are candidates to be added to the state vector.

For better representation of the line features when observing the results, the endpoints provided by the pipe detector are saved and re-projected to their associated line at the end of every feature observation.

4.5.1. Line Feature Re-Observation

Given a feature observation $z(k)$, associated with an already mapped feature f_j , the non-linear observation equation is defined as:

$$z(k) = h_{f_j}(x(k), v_j(k)) = h_j(x_v(k), f_j(k), v_j(k)) \quad (30)$$

$$v_j(k) \equiv \mathcal{N}(\mathbf{0}, \mathbf{R}_{f_j}(k)) \quad (31)$$

where the h_j function uses the the robot pose x_v and the feature parameters $f_j = [{}^W\alpha \ {}^W\beta \ {}^W\gamma \ {}^W\rho]$ are represented in the world frame to transform the line parameters to be referenced to the sensor frame. To do so, first, (3)–(7) are used to compute the vectors ${}^W r_u$, ${}^W v_u$, ${}^W n_u$ and ${}^W \rho$. Next, Equations (15)–(16) are used to compute their counterparts in the sensor frame and, finally, (10)–(12) compute the angles of the new line parametrization in the sensor frame.

The linearised observation matrix is given by:

$$\mathbf{H}_{f_j}(k) = \left. \frac{\partial h_{f_j}(x(k), v(k))}{\partial x(k)} \right|_{x(k)=\hat{x}(k)} \quad (32)$$

$$\mathbf{H}_{f_j}(k) = \begin{bmatrix} \mathbf{J}_{1_j}(k) & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{J}_{2_j}(k) & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}_{4 \times (9+4n_l+3n_o)} \quad (33)$$

where \mathbf{J}_{1_j} is a 4×9 Jacobian matrix that represents the partial derivative of transforming f_j from the world frame $\{W\}$ to the sensor frame $\{S\}$ with respect to the vehicle state, and \mathbf{J}_{2_j} is a 4×4 Jacobian matrix that represents the partial derivative of transforming f_j from the world frame $\{W\}$ to the sensor frame $\{S\}$ with respect to the features in the world frame ${}^W f_j$:

$$\mathbf{J}_{1_j}(k) = \left. \frac{\partial h_j(x_v(k), f_j(k))}{\partial x_v(k)} \right|_{x_v(k)=\hat{x}_v(k), f_j(k)=\hat{f}_j(k)} \quad (34)$$

$$\mathbf{J}_{2_j}(k) = \left. \frac{\partial h_j(x_v(k), f_j(k))}{\partial f_j(k)} \right|_{x_v(k)=\hat{x}_v(k), f_j(k)=\hat{f}_j(k)} \quad (35)$$

Next, observation matrices are stacked to form a single observation matrix:

$$\mathbf{H}(k) = \begin{bmatrix} \mathbf{H}_{f_1}(k) \\ \mathbf{H}_{f_2}(k) \\ \vdots \\ \mathbf{H}_{f_s}(k) \end{bmatrix}_{4s \times (9+4n_l+3n_o)} \quad (36)$$

with s being the number of observed features. Similarly, the covariance matrices \mathbf{R}_i are used to form a block diagonal matrix of uncertainty:

$$\mathbf{R}(k) = \begin{bmatrix} \mathbf{R}_{f_1}(k) & \mathbf{0}_{4 \times 4} & \cdots & \cdots \\ \mathbf{0}_{4 \times 4} & \mathbf{R}_{f_2}(k) & \cdots & \cdots \\ \vdots & \vdots & \ddots & \mathbf{0}_{4 \times 4} \\ \vdots & \vdots & \mathbf{0}_{4 \times 4} & \mathbf{R}_{f_s}(k) \end{bmatrix}_{4s \times 4s} \quad (37)$$

Then, a standard EKF update is applied using these matrices.

4.5.2. New Line Feature Observation

After updating the filter with all the feature observations which have been associated to map features, the remaining non-associated features are considered as candidates to be incorporated to the state vector. Since the structure is known to have only vertical or horizontal pipes, the candidate features are tested against this condition in order to discard outliers.

To add a feature \mathbf{f}_i observed in the sensor frame $\{S\}$ to the state vector, it is compounded with the current vehicle position to obtain the feature in the world frame $\{W\}$. We denote this operation with the \odot operator to distinguish it from the vehicle-point compounding using the \oplus operator, traditionally defined in the SLAM literature as:

$$\mathbf{f}_i(k) = \mathbf{x}_v(k) \odot \mathbf{f}_i(k) \quad (38)$$

Let the stochastic map at time step k be defined by the stochastic vector $\mathbf{x}(k) \sim \mathcal{N}(\hat{\mathbf{x}}(k), \mathbf{P}(k))$. Then, the augmented state vector, including the new feature, is given by:

$$\mathbf{x}_+(k) \equiv \mathcal{N}(\hat{\mathbf{x}}_+(k), \mathbf{P}_+(k)) \quad (39)$$

where:

$$\hat{\mathbf{x}}_+(k) = [\hat{\mathbf{x}}(k) \quad \hat{\mathbf{x}}_v(k) \odot \hat{\mathbf{f}}_i(k)]^T \quad (40)$$

and:

$$\mathbf{P}_+(k) = \begin{bmatrix} \mathbf{P}(k) & [\mathbf{P}_v^T(k) \mathbf{P}_{f_1}^T(k) \dots \mathbf{P}_{f_m}^T(k)]^T \mathbf{J}_{1\odot}(k)^T \\ [\mathbf{P}_v(k) \mathbf{P}_{v f_1}(k) \dots \mathbf{P}_{v f_m}(k)] \mathbf{J}_{1\odot}(k) & \mathbf{J}_{1\odot}(k) \mathbf{P}_v(k) \mathbf{J}_{1\odot}^T(k) + \mathbf{J}_{2\odot}(k) \mathbf{R}_{f_i}(k) \mathbf{J}_{2\odot}^T(k) \end{bmatrix} \quad (41)$$

where $\mathbf{J}_{1\odot}$ is a 4×9 Jacobian matrix that represents the partial derivative of transforming \mathbf{f}_i from the sensor frame $\{S\}$ to the world frame $\{W\}$ with respect to the vehicle state, and $\mathbf{J}_{2\odot}$ is a 4×4 Jacobian matrix that represents the partial derivative of transforming \mathbf{f}_i from the sensor frame $\{S\}$ to the world frame $\{W\}$ with respect to the feature in the sensor frame:

$$\mathbf{J}_{1\odot}(k) = \left. \frac{\partial \mathbf{x}_v(k) \odot \mathbf{f}_i(k)}{\partial \mathbf{x}_v(k)} \right|_{\mathbf{x}_v = \hat{\mathbf{x}}_v(k), \mathbf{f}_i(k) = \hat{\mathbf{f}}_i(k)} \quad (42)$$

$$\mathbf{J}_{2\odot}(k) = \left. \frac{\partial \mathbf{x}_v(k) \odot \mathbf{f}_i(k)}{\partial \mathbf{f}_i(k)} \right|_{\mathbf{x}_v = \hat{\mathbf{x}}_v(k), \mathbf{f}_i(k) = \hat{\mathbf{f}}_i(k)} \quad (43)$$

Once a feature is added to the state vector, its endpoints are also saved for future re-observations.

4.6. Object Feature Observation

From the object semantic segmentation, object features are received as xyz positions in the sensor frame $\{S\}$. The first step before the update is the feature association. Already-mapped features in the state vector are transformed to the sensor frame together with their uncertainty. As for the line features, a JCBB algorithm is used to ensure consistency in the associations. Once this association is solved, we have two kinds of observations: re-observed features that were already in the state vector or new features that are candidates to be added to the state vector.

4.6.1. Object Feature Re-observation

As in the previous case, each feature observation $\mathbf{z}(k)$ associated with an already mapped feature \mathbf{f}_j has an observation Equation (30). In this case, since we use point features instead of lines, a different \mathbf{h}_j function is used:

$$\mathbf{h}_j(\mathbf{x}_v(k), \mathbf{f}_j(k)) = \ominus \mathbf{x}_v(k) \oplus \mathbf{f}_j(k). \quad (44)$$

where \oplus and \ominus are the conventional compounding and inverse compounding operations commonly used in the SLAM literature.

Given the point feature observation (32), computing the observation matrix \mathbf{H}_{f_j} (33) involves computing the Jacobians \mathbf{J}_{1_j} (34) and \mathbf{J}_{2_j} (35) of the point feature observation function \mathbf{h}_j given in (44). In this case, the matrix size is $3 \times (9 + 4n_l + 3n_o)$ since the points are tri-dimensional. In a similar way as was used in Section 4.5.1, the stacked observation matrix $\mathbf{H}(k)$ can be computed as shown in (36), though, in this case, its dimension is

$3s \times (9 + 4n_l + 3n_o)$. Finally, the covariance matrix of the observation can be built as a block diagonal matrix as shown in (37) being, in this case, a $3s \times 3s$ matrix.

Then, a standard EKF update is applied using these matrices.

4.6.2. New Object Feature Observation

As for the line features, after updating all the object position observations which have been associated to point map features, the remaining non-associated features are considered as candidates to be incorporated to the state vector. The process followed to map the newly discovered objects is equivalent to the one conducted with the pipe lines. The main difference is how the world reference feature position is computed:

$$\mathbf{f}_i(k) = \mathbf{x}_v(k) \oplus \mathbf{f}_i(k) \quad (45)$$

which, in this case, uses the conventional vector compounding operation. Therefore, the vector augmentation equations are equivalent to (40) and (41), substituting \odot by \oplus , $J_{1\odot}$ by $J_{1\oplus}$ and $J_{2\odot}$ by $J_{2\oplus}$. Please note that in this case, the h_j function used to compute the Jacobians is now the one reported in (44).

5. Experimental Results

5.1. Experimental Setup

The underwater test scene consisted of an industrial structure comprising pipes and valves, with an approximate size of 1.4 m width, 1.4 m depth and 1.2 m height (Figure 9). For the testing, this structure was positioned at the bottom of a 5 m deep water tank, while the Girona500 AUV [60] moved in a trajectory around it while always facing the underwater structure. The laser scanner measurements were obtained at a distance ranging from 2 to 3.5 m from the underwater structure at a rate of 0.5 Hz. Maintaining a constant distance to the observed structure ensures better results as observed in [61]. The dataset was acquired and stored in a Robot Operating System (ROS) bagfile to be processed offline, consisting of the AUV navigation data (DVL at 5 Hz, pressure at 8 Hz and AHRS at 20 Hz), and the point clouds of 245 laser scans gathered with our laser scanner.

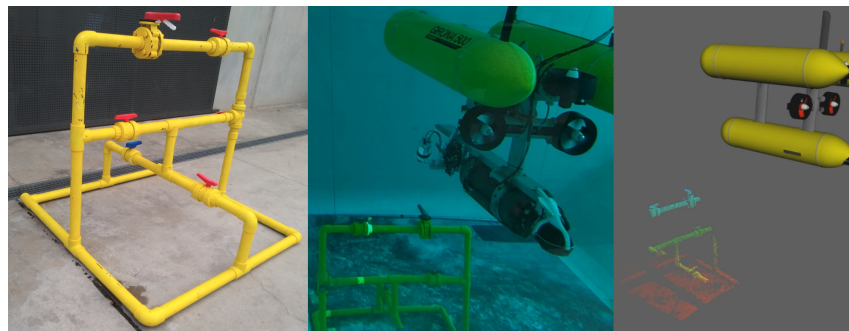


Figure 9. Experimental setup in the water tank with the Girona500 AUV. (left) Industrial structure before deployment. (center) Underwater view of the water tank during the experiments. (right) The 3D visualizer with a scan of the structure.

The 245 scans were processed, containing a total of 1268 object observations of 20 unique objects from 6 different classes, and 1778 pipe observations of 12 unique pipes. More details on the experimental setup can be found in [6].

A video showcasing the segmentation and SLAM results can be found in <https://www.youtube.com/watch?v=flFoUrDN-rc>

5.2. SLAM Results

The proposed SLAM algorithm with line and object features was compared first with the same algorithm without the feature updates, consisting of a DR navigation. Since no features are used in DR, the resulting map contains all the observations received from the semantic segmentation module (Figure 10a). Nevertheless, the SLAM solution provides a consistent map with all the pipes and objects from the structure (Figure 10b). Note that the lower corner is never observed in this dataset, and thus, the corner object, as well as the full length of the bottom pipes, are not included in the final map.

To assert the convergence on the state estimation for pipes and objects, one can look at the volume of the uncertainty bounding ellipsoid, which can be computed as $\prod_i \lambda_i$, where λ_i are the eigen-values of the uncertainty matrix corresponding to the feature. For better numerical stability, by avoiding multiplications of small numbers that can lead to numerical errors, volumes can be calculated in the logarithmic space as $\sum_i \log(\lambda_i)$. Figures 11 and 12 show how the uncertainty-bounding ellipsoids for each feature decrease through time with each re-observation of the feature and maintain constant values when the features are not re-observed.

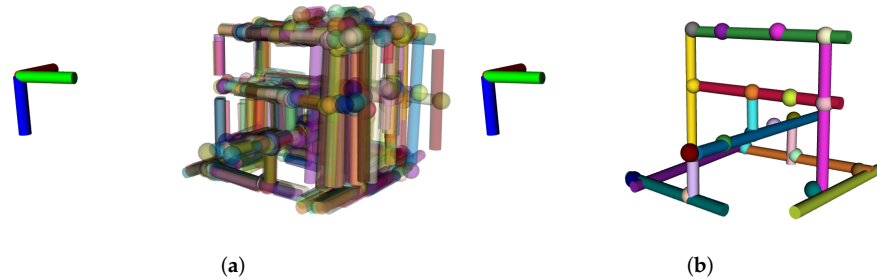


Figure 10. Comparison of maps obtained from DR and SLAM with the NED reference frame. (a) DR with 1778 pipes and 1268 objects. (b) SLAM with 12 pipes and 20 objects.

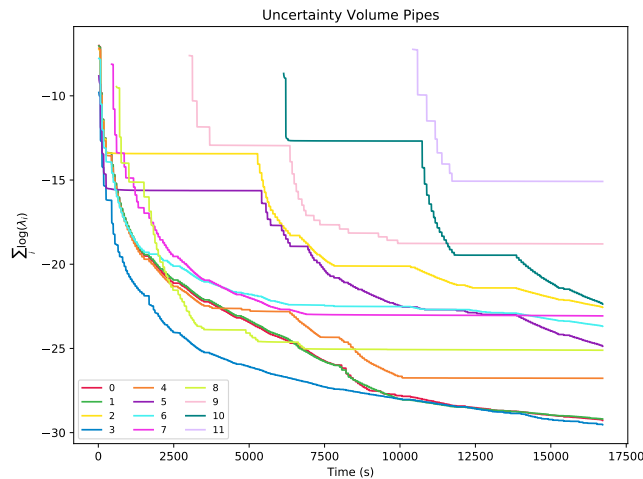


Figure 11. Uncertainty volumes with regards to experiment time for the 12 pipe features.

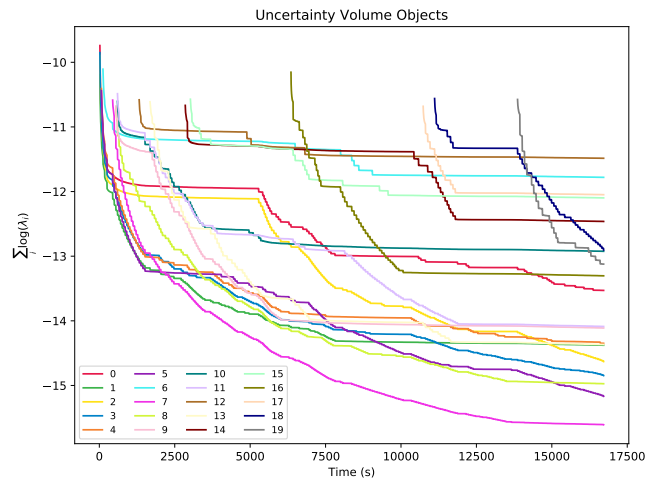


Figure 12. Uncertainty volumes with regards to experiment time for the 20 object features.

Looking at the vehicle state vector, we can observe that vehicle positions in the xy plane reach a significantly smaller uncertainty than the DR solution (Figure 13).

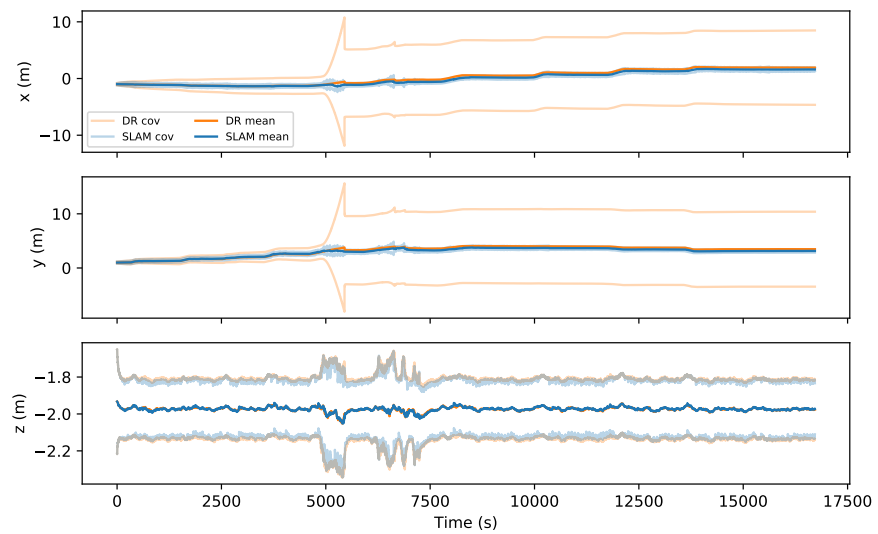


Figure 13. Comparison between DR and SLAM mean values and $\pm 2\sigma$ covariances for robot position.

It is worth noting that the DR covariance shows several peaks of uncertainty due to the DVL not being able to provide velocity measurements to bound the error. This can be clearly seen in the vehicle state velocity (Figure 14), where the DVL failures are more clearly seen by the growing uncertainty. DVL failures are common in water tank experiments due to the beams impacting the vertical walls. However, the SLAM solution greatly reduces the uncertainty during those events, providing a more accurate estimation.

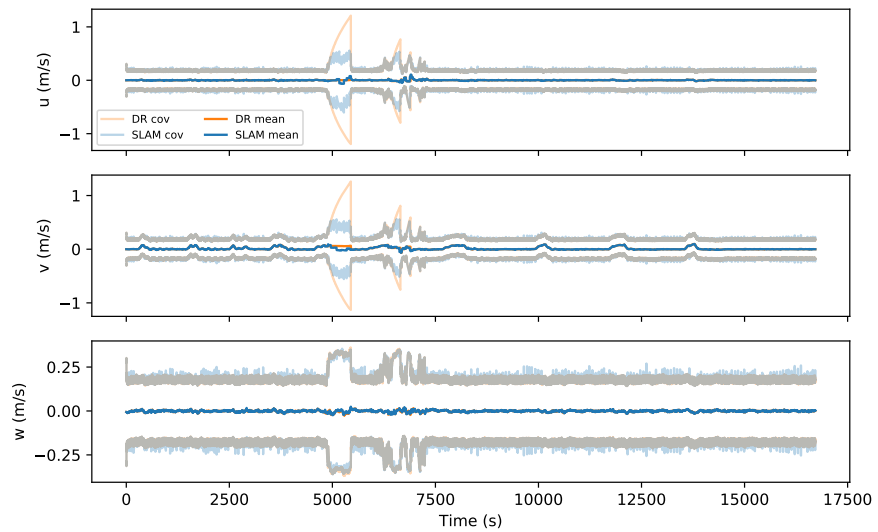


Figure 14. Comparison between DR and SLAM mean values and $\pm 2\sigma$ covariances for robot velocity.

5.3. Object Segmentation Results

Significantly better results have been achieved using the segmentation method described in Section 3.4 compared to our previous solution. The new method correctly segments the handle of the valve, which is a salient feature of this object. This can be appreciated in Figure 15, which is a good example of a *Butterfly* object segmentation. This improvement leads to a better recognition rate with the CVFH descriptor.



Figure 15. Segmented view of the butterfly valve along with the handle.

5.4. Object Recognition Results

Figure 16 and Table 3 show the confusion matrices of the object recognition method. The row labelled *SYN* shows the confusion matrix, which was computed using synthetic data and the CVFH descriptor only. This confusion matrix is the same as the one presented in [33] and is included here for comparison. The rows labeled *DESC*, *BAYS* and *SEM* show the experimental results of applying the method described in this paper to the dataset reported above. These rows show the confusion matrices when using the CVFH alone, together with the Bayesian estimate, and the result of incorporating the semantic information about the pipe connectivity. The figure shows that, in general, for each row

(*SYN*, *DESC*, *BAYS* and *SEM*), the column related to the ground truth class is always the one with the highest recognition rate. It also shows that, in general, the recognition rate grows when incorporating Bayesian estimation and semantic information. Please note that we separate the results (*DESC*, *BAYS* and *SEM*) to provide an insight into how the method works. Nevertheless, the row *SEM*, which corresponds to the output of the complete recognition pipeline, incorporates both Bayesian estimation and semantic information. Therefore, focusing on this row, it can be clearly seen that a good recognition rate is obtained for all objects, with *R-tee* being the most challenging one, since it is often confused with the *3-way-valve*.

On the other hand, Table 4 shows the assessment of the results based on the accuracy, precision, recall and F1 score [62]. Three object classes, namely *Ball-Valve*, *Elbow* and *Butterfly-Valve*, have a balanced trend between recall and precision, resulting in a high F1 score that improves progressively from the descriptor-based to the Bayesian, and then to the semantics-based method.

The *3-Way-Valve* has a high recall, meaning that the system works well recognising it when actually scanning (TP) and that there is a low number of False Negatives (FNs). However, it has a low precision, meaning that the number of FPs is high. Unfortunately, this leads to a poorer F1 score. The high number of FPs (*R-Tees* wrongly detected as *3-way-valves*) may be explained by the fact that most of the *R-Tees* are located at the bottom, on the floor. This means that these objects are far from the laser scanner, and therefore, their point cloud is noisier and of lower resolution (i.e., the point density is considerably lower). The *R-Tees* are particularly sensitive to noise. As can be seen in the database, the object views have smooth continuous curvatures compared to the scanned ones, which produce noisy surfaces. These noisy surfaces distort the results of the descriptor, given that the descriptor is based on the computation of surface normals from the point cloud.

In contrast, the *R-tee* class achieves high precision (low number of FPs) but low recall (high number of FNs) due, again, to the high number of *R-Tees* detected as *3-way-valves*.

Table 3. Confusion matrices expressed as a numerical %. The objects 1, 2, 3, 4, 5 represent, respectively: a *Ball-Valve*, an *Elbow*, a *R-Tee*, a *3-Way-Valve*, and a *Butterfly-Valve*. We highlighted the best recognition for each object, which coincides with the correct class and the usage of semantic information.

Descriptors	Experiment	Objects																								
		Ball Valve					Elbow					R-Tee					3-Way-Ball-Valve					Butterfly-Valve				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
CVFH	SYN	63	10	7	19	2	2	75	7	1	1	4	27	65	1	1	9	3	1	84	1	17	5	1	21	54
	DESC	60.46	18.11	1.28	9.44	10.71	2.74	80.82	4.11	9.59	2.74	1.68	38.13	19.18	28.78	12.23	39.64	5.41	0.9	49.55	4.5	21.76	12.35	7.06	25.29	33.53
	BAYS	82.65	0	0	0	17.35	1.37	90.41	1.37	6.85	0	6.47	10.55	51.32	19.42	12.23	1.8	0	0	95.5	2.7	0	0	0	0	100
	SEM	82.65	0	0	0	17.35	0	73.97	17.81	8.22	0	0	3.12	64.75	32.13	0	0	0	0	100	0	0	0	0	0	100

Table 4. Assessment of the recognition performance through accuracy, recall, precision and F1 score. We highlighted the best results in blue color, which are all consistent with the use of semantic information, except for *3-Way-Ball-Valve*, where the best F1 score was achieved using the Bayesian estimation method.

Descriptors	Experiment	Objects																							
		Average				Ball Valve				Elbow				R-Tee				3-Way-Ball-Valve				Butterfly-Valve			
		Accuracy	Recall	Precision	F1-Score	Accuracy	Recall	Precision	F1-Score	Accuracy	Recall	Precision	F1-Score	Accuracy	Recall	Precision	F1-Score	Accuracy	Recall	Precision	F1-Score	Accuracy	Recall	Precision	F1-Score
CVFH	DESC	0.42	0.49	0.46	0.38	0.4	0.60	0.72	0.66	0.42	0.81	0.19	0.30	0.42	0.19	0.79	0.31	0.42	0.50	0.21	0.29	0.42	0.34	0.36	0.35
	BAYS	0.76	0.84	0.73	0.74	0.76	0.83	0.92	0.87	0.76	0.90	0.60	0.72	0.76	0.51	1	0.68	0.76	0.95	0.55	0.70	0.76	1	0.58	0.74
	SEM	0.80	0.84	0.78	0.78	0.8	0.83	1	0.91	0.80	0.74	0.81	0.77	0.80	0.65	0.95	0.77	0.80	1	0.44	0.61	0.80	1	0.71	0.83

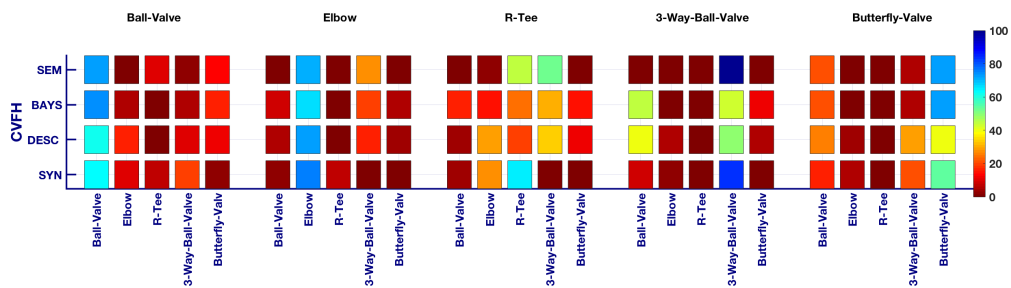


Figure 16. Confusion matrices.

6. Conclusions

This paper has presented a semantic mapping method using non-coloured point clouds and navigation sensor data. The method includes semantic segmentation (of planes, pipes and objects) paired with a feature-based SLAM filter and a semantic-based recognition based on multiple views of each tracked object. The methods were tested against real data gathered with an AUV in a water tank with a man-made pipe structure.

Semantic segmentation attained better performance in selecting the sets of points belonging to each object than in our previous work. This reduced the negative impact of the presence of points belonging to pipes that made recognition more difficult. The "mushroom" shape bounding box used over the pipe intersections allowed the computation of object candidates with the potential presence of handles, thus enabling a better crop of the input scan that tightly encapsulates the object with handle to be recognized.

Feature-based SLAM provided an accurate object tracking that allowed the integration of multiple views of the same object acquired at different times in order to better estimate their class. Moreover, it produced a consistent map of the structure while also providing navigation corrections that compensated for the effects of inconsistencies in navigation due to errors in DVL measurements. The integration of the recognition and the SLAM module, where information is passed back and forth, was instrumental to the higher performance of the approach and to the ability to create a semantic map of all recognized objects.

7. Future Work

Future research plans will continue in the direction of combining the representations of SLAM and object recognition to provide a more accurate and detailed 3D semantic map while providing recognition with more complete views. The object recognition approach we used, based on SLAM, mainly consists of two modules: a Bayesian semantic information-based method for recognition and the SLAM system. Since SLAM provides long-term consistent navigation, one future improvement will be to use this navigation to fuse several scans, which will provide more comprehensive views of the objects. Having more complete views has the potential to improve both the accuracy of object recognition and the reliability of pose estimates, especially in challenging scenarios with significant changes in viewpoints.

A longer term strategy to improve the observation quality is to perform view planning in order to reduce the ambiguity caused by poorly observed objects. Such view planning should be multi-objective in the sense of taking into account multiple objects simultaneously and should be guided towards the next best views that solve the ambiguity between the most probable classes for each object. Continuing this work, future efforts will be directed toward the goal of grasping and manipulating such objects.

Author Contributions: Conceptualization, G.V., K.H., P.R. and N.G.; investigation, G.V., K.H., P.R. and N.G.; methodology, G.V., K.H., P.R. and N.G.; software, G.V., K.H.; supervision, P.R. and N.G.; writing, G.V., K.H., P.R. and N.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Spanish Government through a FPI Ph.D. grant to K. Himri, as well as by the Spanish Project DPI2017-86372-C3-2-R (TWINBOT-GIRONA1000) and the European project H2020-INFRAIA-2017-1-twostage-731103 (EUMarineRobots).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Ridao, P.; Carreras, M.; Ribas, D.; Sanz, P.J.; Oliver, G. Intervention AUVs: The next challenge. *Annu. Rev. Control.* **2015**, *40*, 227–241. doi:<https://doi.org/10.1016/j.arcontrol.2015.09.015>.
- Cieslak, P.; Ridao, P.; Giergiel, M. Autonomous underwater panel operation by GIRONA500 UVMS: A practical approach to autonomous underwater manipulation. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 26–30 May 2015; pp. 529–536. doi:10.1109/ICRA.2015.7139230.
- Carreras, M.; Carrera, A.; Palomeras, N.; Ribas, D.; Hurtós, N.; Salvi, Q.; Ridao, P. Intervention Payload for Valve Turning with an AUV. In *Computer Aided Systems Theory—EUROCAST 2015*; Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 877–884.
- Youakim, D.; Ridao, P.; Palomeras, N.; Spadafora, F.; Ribas, D.; Muzzupappa, M. MoveIt!: Autonomous Underwater Free-Floating Manipulation. *IEEE Robot. Autom. Mag.* **2017**, *24*, 41–51. doi:10.1109/MRA.2016.2636369.
- Sanz, P.J.; Ridao, P.; Oliver, G.; Casalino, G.; Petillot, Y.; Silvestre, C.; Melchiorri, C.; Turetta, A. TRIDENT An European project targeted to increase the autonomy levels for underwater intervention missions. In Proceedings of the 2013 OCEANS-San Diego, San Diego, CA, USA, 23–27 September 2013; pp. 1–10.
- Himri, K.; Ridao, P.; Gracias, N. Underwater Object Recognition Using Point-Features, Bayesian Estimation and Semantic Information. *Sensors* **2021**, *21*, 1807. doi:10.3390/s21051807.
- Kinsey, J.C.; Whitcomb, L.L. Preliminary field experience with the DVLNAV integrated navigation system for oceanographic submersibles. *Control. Eng. Pract.* **2004**, *12*, 1541–1549. doi:10.1016/j.conengprac.2003.12.010.
- Thomas, H.G. GIB Buoys: An Interface Between Space and Depths of the Oceans. In Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles (Cat. No.98CH36290), Cambridge, MA, USA, 21 August 1998 ; pp. 181–184. doi:10.1109/AUV.1998.744453.
- Mandt, M.; Gade, K.; Jalving, B. Integrateing DGPS-USBL position measurements with inertial navigation in the HUGIN 3000 AUV. In Proceedings of the 8th Saint Petersburg International Conference on Integrated Navigation Systems, St. Petersburg, Russia, 28–30 May 2001.
- Alcocer, A.; Oliveira, P.; Pascoal, A. Study and implementation of an EKF GIB-based underwater positioning system. *Control. Eng. Pract.* **2007**, *15*, 689–701. doi:<https://doi.org/10.1016/j.conengprac.2006.04.001>.
- Melo, J.; Matos, A. Survey on advances on terrain based navigation for autonomous underwater vehicles. *Ocean. Eng.* **2017**, *139*, 250–264. doi:<https://doi.org/10.1016/j.oceaneng.2017.04.047>.
- Ribas, D.; Ridao, P.; Domingo, J.D.; Neira, J. Underwater SLAM in Man-Made Structured Environments. *J. Field Robot.* **2008**, *25*, 898–921. doi:10.1002/rob.
- He, B.; Liang, Y.; Feng, X.; Nian, R.; Yan, T.; Li, M.; Zhang, S. AUV SLAM and experiments using a mechanical scanning forward-looking sonar. *Sensors* **2012**, *12*, 9386–9410.
- Fallon, M.F.; Folkesson, J.; McClelland, H.; Leonard, J.J. Relocating underwater features autonomously using sonar-based SLAM. *IEEE J. Ocean. Eng.* **2013**, *38*, 500–513.
- Burguera, A.; González, Y.; Oliver, G. The UspIC: Performing Scan Matching Localization Using an Imaging Sonar. *Sensors* **2012**, *12*, 7855–7885. doi:10.3390/s120607855.
- Mallios, A.; Ridao, P.; Ribas, D.; Carreras, M.; Camilli, R. Toward autonomous exploration in confined underwater environments. *J. Field Robot.* **2016**, *33*, 994–1012.
- Vallicrosa, G.; Ridao, P. H-SLAM: Rao-Blackwellized Particle Filter SLAM Using Hilbert Maps. *Sensors* **2018**, *18*, 1386. doi:10.3390/s18051386.
- Fairfield, N.; Kantor, G.; Wettergreen, D. Towards particle filter SLAM with three dimensional evidence grids in a flooded subterranean environment. In Proceedings 2006 IEEE International Conference on Robotics and Automation (ICRA 2006), Orlando, FL, USA, 15–19 May 2006; pp. 3575–3580. doi:10.1109/ROBOT.2006.1642248.
- Roman, C.; Singh, H. A Self-Consistent Bathymetric Mapping Algorithm. *J. Field Robot.* **2007**, *24*, 23–50. doi:10.1002/rob.
- Barkby, S.; Williams, S.B.; Pizarro, O.; Jakuba, M. A Featureless Approach to Efficient Bathymetric SLAM Using Distributed Particle Mapping. *J. Field Robot.* **2011**, *28*, 19–39. doi:10.1002/rob.
- Palomer, A.; Ridao, P.; Ribas, D. Multibeam 3D Underwater SLAM with Probabilistic Registration. *Sensors* **2016**, *16*, 560. doi:10.3390/s16040560.
- Eustice, R.; Pizarro, O.; Singh, H. Visually Augmented Navigation in an Unstructured Environment Using a Delayed State History. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04), New Orleans, LA, USA, 26 April–1 May 2004; pp. 25–32. doi:10.1109/ROBOT.2004.1307124.
- Williams, S.; Mahon, I. Simultaneous Localisation and Mapping on the Great Barrier Reef. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04), New Orleans, LA, USA, 26 April–1 May 2004; pp. 1771–1776.
- Eustice, R.; Singh, H.; Leonard, J.; Walter, M.; Ballard, R. Visually Navigating the RMS Titanic with SLAM Information Filters. In *Proceedings of the Robotics Science and Systems*; MIT Press: Cambridge, MA, USA, 2005.
- Johnson-Roberson, M.; Pizarro, O.; Williams, S.B.; Mahon, I. Generation and Visualization of Large-Scale Three-Dimensional Reconstructions from Underwater Robotic Surveys. *J. Field Robot.* **2010**, *27*, 21–51. doi:10.1002/rob.

26. Gracias, N.; Ridaio, P.; Garcia, R.; Escartin, J.; Cibecchini, F.; Campos, R.; Carreras, M.; Ribas, D.; Magi, L.; Palomer, A.; et al. Mapping the Moon: Using a lightweight AUV to survey the site of the 17th Century ship 'La Lune'. In Proceedings of the MTS/IEEE OCEANS Conference, Bergen, Norway, 10–14 June 2013.
27. Campos, R.; Gracias, N.; Palomer, A.; Ridaio, P. Global Alignment of a Multiple-Robot Photomosaic using Opto-Acoustic Constraints. *IFAC-PapersOnLine* **2015**, *48*, 20–25. doi:10.1016/j.ifacol.2015.06.004.
28. Inglis, G.; Smart, C.; Vaughn, I.; Roman, C. A pipeline for structured light bathymetric mapping. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 4425–4432.
29. Massot-Campos, M.; Oliver, G.; Bodenmann, A.; Thornton, B. Submap bathymetric SLAM using structured light in underwater environments. In Proceedings of the 2016 IEEE/OES Autonomous Underwater Vehicles (AUV), Tokyo, Japan, 6–9 November 2016; pp. 181–188. doi:10.1109/AUV.2016.7778669.
30. Palomer, A.; Ridaio, P.; Forest, J.; Ribas, D. Underwater Laser Scanner: Ray-Based Model and Calibration. *IEEE/ASME Trans. Mechatronics* **2019**, *24*, 1986–1997. doi:10.1109/TMECH.2019.2929652.
31. Palomer, A.; Ridaio, P.; Youakim, D.; Ribas, D.; Forest, J.; Petillot, Y. 3D Laser Scanner for Underwater Manipulation. *Sensors* **2018**, *18*, 1086. doi:10.3390/s18041086.
32. Palomer, A.; Ridaio, P.; Ribas, D. Inspection of an underwater structure using point-cloud SLAM with an AUV and a laser scanner. *Journal of Field Robotics* **2019**, *36*, 1333–1344.
33. Himri, K.; Ridaio, P.; Gracias, N. 3D Object Recognition Based on Point Clouds in Underwater Environment with Global Descriptors: A Survey. *Sensors* **2019**, *19*, 4451.
34. Guo, Y.; Bennamoun, M.; Sohel, F.; Lu, M.; Wan, J. 3D object recognition in cluttered scenes with local surface features: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2270–2287.
35. Alexandre, L.A. 3D descriptors for object and category recognition: a comparative evaluation. In Proceedings of the Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal, 7–12 October 2012; Volume 1, p. 7.
36. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep learning for 3D point clouds: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, doi:10.1109/TPAMI.2020.3005434.
37. Tian, Y.; Chen, L.; Song, W.; Sung, Y.; Woo, S. DGCB-Net: Dynamic Graph Convolutional Broad Network for 3D Object Recognition in Point Cloud. *Remote. Sens.* **2021**, *13*, 66.
38. Martin-Abadal, M.; Piñar-Molina, M.; Martorell-Torres, A.; Oliver-Codina, G.; Gonzalez-Cid, Y. Underwater Pipe and Valve 3D Recognition Using Deep Learning Segmentation. *J. Mar. Sci. Eng.* **2020**, *9*, 5.
39. Pereira, M.I.; Claro, R.M.; Leite, P.N.; Pinto, A.M. Advancing Autonomous Surface Vehicles: A 3D Perception System for the Recognition and Assessment of Docking-Based Structures. *IEEE Access* **2021**, *9*, 53030–53045.
40. Pi, R.; Cieslak, P.; Ridaio, P.; Sanz, P.J. TWINBOT: Autonomous Underwater Cooperative Transportation. *IEEE Access* **2021**, *9*, 37668–37684. doi:10.1109/ACCESS.2021.3063669.
41. Nüchter, A.; Hertzberg, J. Towards semantic maps for mobile robots. *Robot. Auton. Syst.* **2008**, *56*, 915–926.
42. Balaska, V.; Bampis, L.; Boudourides, M.; Gasteratos, A. Unsupervised semantic clustering and localization for mobile robotics tasks. *Robot. Auton. Syst.* **2020**, *131*, 103567.
43. Kostavelis, I.; Gasteratos, A. Learning spatially semantic representations for cognitive robot navigation. *Robot. Auton. Syst.* **2013**, *61*, 1460–1475.
44. Kim, D.I.; Sukhatme, G.S. Semantic labeling of 3d point clouds with object affordance for robot manipulation. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 5578–5584.
45. Civera, J.; Gálvez-López, D.; Riazuelo, L.; Tardós, J.D.; Montiel, J.M.M. Towards semantic SLAM using a monocular camera. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 1277–1284.
46. Tang, Z.; Wang, G.; Xiao, H.; Zheng, A.; Hwang, J.N. Single-camera and inter-camera vehicle tracking and 3D speed estimation based on fusion of visual and semantic features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 108–115.
47. Shao, T.; Xu, W.; Zhou, K.; Wang, J.; Li, D.; Guo, B. An interactive approach to semantic modeling of indoor scenes with an rgb camera. *ACM Trans. Graph. (TOG)* **2012**, *31*, 1–11.
48. Song, S.; Yu, F.; Zeng, A.; Chang, A.X.; Savva, M.; Funkhouser, T. Semantic scene completion from a single depth image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1746–1754.
49. Dewan, A.; Oliveira, G.L.; Burgard, W. Deep semantic classification for 3d lidar data. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 3544–3549.
50. Chen, S.W.; Nardari, G.V.; Lee, E.S.; Qu, C.; Liu, X.; Romero, R.A.F.; Kumar, V. Sloam: Semantic lidar odometry and mapping for forest inventory. *IEEE Robot. Autom. Lett.* **2020**, *5*, 612–619.

51. Milioto, A.; Vizzo, I.; Behley, J.; Stachniss, C. Rangenet++: Fast and accurate lidar semantic segmentation. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 4213–4220.
52. Liu, F.; Fang, M. Semantic segmentation of underwater images based on improved Deeplab. *J. Mar. Sci. Eng.* **2020**, *8*, 188.
53. Miguelanez, E.; Patron, P.; Brown, K.E.; Petillot, Y.R.; Lane, D.M. Semantic knowledge-based framework to improve the situation awareness of autonomous underwater vehicles. *IEEE Trans. Knowl. Data Eng.* **2010**, *23*, 759–773.
54. Girdhar, Y.; Dudek, G. Exploring underwater environments with curiosity. In Proceedings of the 2014 Canadian Conference on Computer and Robot Vision, Montreal, QC, Canada, 6–9 May 2014, pp. 104–110.
55. Rabbani, T.; Heuvel, F.; Vosselman, G. Segmentation of point clouds using smoothness constraint. *Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2006**, *36*, 248–253.
56. Aldoma, A.; Vincze, M.; Blodow, N.; Gossow, D.; Gedikli, S.; Rusu, R.B.; Bradski, G. CAD-model recognition and 6DOF pose estimation using 3D cues. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 6–13 November 2011; pp. 585–592.
57. Yang, Y.; Huang, G. Aided inertial navigation: Unified feature representations and observability analysis. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 3528–3534.
58. Ruifang, D.; Frémont, V.; Lacroix, S.; Fantoni, I.; Changan, L. Line-based monocular graph SLAM. In Proceedings of the 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Daegu, Korea, 16–18 November 2017; pp. 494–500.
59. Neira, J.; Tardós, J.D. Data association in stochastic mapping using the joint compatibility test. *IEEE Trans. Robot. Autom.* **2001**, *17*, 890–897.
60. Ribas, D.; Palomeras, N.; Ridao, P.; Carreras, M.; Mallios, A. Girona 500 AUV: From Survey to Intervention. *IEEE/ASME Trans. Mechatronics* **2012**, *17*, 46–53. doi:10.1109/TMECH.2011.2174065.
61. Himri, K.; Ridao, P.; Gracias, N.; Palomer, A.; Palomeras, N.; Pi, R. Semantic SLAM for an AUV using object recognition from point clouds. *IFAC-PapersOnLine* **2018**, *51*, 360–365.
62. Johnson, J.M.; Khoshgoftaar, T.M. Survey on deep learning with class imbalance. *J. Big Data* **2019**, *6*, 1–54.

5

RESULTS AND DISCUSSION

This chapter summarizes the main results achieved in all work carried out during the work of developing the thesis. Beyond reporting the results, an effort is made to make clear what was learned and how each of the individual results relates to the contributions of the follow-up investigations.

5.1 Evaluation of global point feature descriptors for 3D object recognition in non-coloured point clouds

During the first part of the thesis we carried out a survey comparing the performance of global descriptors for 3D object recognition assuming that 3D models of the objects of interest are available *a priori*. Seven representative objects, commonly present in submerged pipe infrastructures, were selected for the study. The objects included different types of valves, elbows, tees, etc. Using their CAD models, a database containing 12 synthetic views of each object was created. Each view was taken from the vertex of a virtual icosahedron with the laser scanner pointing to the center where the object was placed. The database represents the knowledge the robot has about the objects it needs to recognize. Next, seven global descriptors (ESF, VFH, CVFH, OUR-CVFH, GOOD, GPFH and GRSD) already available in the Point Cloud Library [16] were selected and compared exhaustively, first using synthetic data and then from data collected in water tank experiments. In the comparison, different properties of the object-view point clouds were evaluated: 1) the use of partial vs. global views, 2) the use of same vs. different resolution between the object model and the input scan, 3) the effect of resolution and 4) the effect of noise.

5.1.1 Results using Synthetic Data

Figure 5.1 summarizes the comparative results using synthetic data. In order to study how each of the above listed properties (Full-View/Partial-View, Same-Resolution/Different-Resolution) affect the results independently of the descriptor used, we averaged the recognition rates achieved with all the descriptors. The results showed that:

1. Using global views provides better results than using partial views. The average recognition rate achieved with all the descriptors of 48.9% achieved when using full views, decreased to 40.5% when partial views were used instead.
2. Using the same resolution in the views of the database and the input scan leads to significantly better results. The average recognition rate of 62.9% when the same resolution is used, decreased to only 26.6% when the resolutions differ.

The combination of Full-View/Same-Resolution achieves a recognition rate of 69.2%, being the best scenario, while Partial View /Different Resolution is the poorest, achieving only a 24.5% recognition rate.

When the resolution of the scan is analyzed, in general, for most of descriptors the higher the resolution the better the recognition rates. Hence decreasing the resolution decreases the performance, with the exception of the GOOD descriptor whose performance remains constant over the studied resolutions.

Another parameter studied was the noise (fig. 5.2). In this case, the results follow intuition: the higher the noise the worse the recognition rate and the higher the object confusion. The exception is again the GOOD descriptor which is the best one for high levels of noise.

5.1.2 Results Using Real Data

Results on real data were studied taking advantage of an already existing dataset collected in [4]. Since this experiment involved only 4 out of the 7 objects, a supplementary

Experiment	View	Resolution	Descriptors							Average Over Descriptors
			ESF	VFH	CVFH	OURCVFH	GOOD	GFPFH	GRSD	
FVSR	Full	Same	77.4	65.3	69.5	69.3	55.1	72.8	75.0	69.2
FVDR		Different	41.8	18.1	18.3	18.7	51.9	29.3	22.3	28.6
PVSR	Partial	Same	67.6	52.8	56.7	56.8	41.9	56.5	63.8	56.6
PVDR		Different	34.3	17.2	17.8	17.8	38.3	25.3	21.0	24.5
FVSR/FVDR	Average Over	Full View	59.6	41.7	43.9	44.0	53.5	51.0	48.6	48.9
PVSR/PVDR		Partial View	50.9	35.0	37.3	37.3	40.1	40.9	42.4	40.5
FVSR/PVSR		Same Res	72.5	59.1	63.1	63.0	48.5	64.6	69.4	62.9
FVDR/PVDR		Diff Res	38.0	17.6	18.1	18.2	45.1	27.3	21.6	26.6
FVSR/FVDR/ PVSR/PVDR	Full Average		55.3	38.4	40.6	40.6	46.8	46.0	45.5	44.7

Figure 5.1: Summary of results for all the objects and all the resolutions using synthetic data. The best descriptor is marked in green, while the worst is marked in red. (Extracted from [1])

Noise Std	Experiment	View	Resolution	Descriptors							Average Over Descriptors
				ESF	VFH	CVFH	OURCVFH	GOOD	GFPFH	GRSD	
$\sigma = 0$	FVSR	Full	Same	80.7	65.9	77.7	77.0	61.9	71.1	85.6	74.3
	PVSR	Partial		69.7	52.3	63.9	61.4	41.7	53.4	70.4	59.0
$\sigma = 0.00625$	FVSR	Full	Same	78.4	67.4	78.6	76.4	61.0	75.4	82.6	74.3
	PVSR	Partial		74.6	52.9	62.4	63.6	43.7	51.7	71.7	60.1
$\sigma = 0.0125$	FVSR	Full	Same	80.0	69.1	79.6	78.0	60.3	74.7	86.1	75.4
	PVSR	Partial		72.3	55.3	66.7	66.3	44.4	55.9	72.3	61.9
$\sigma = 0.025$	FVSR	Full	Same	80.4	67.4	78.3	75.9	56.4	72.1	61.4	70.3
	PVSR	Partial		70.4	51.4	65.9	65.6	42.9	56.7	51.3	57.7
$\sigma = 0.05$	FVSR	Full	Same	62.4	45.7	54.7	41.9	59.7	64.6	27.7	51.0
	PVSR	Partial		56.0	36.1	36.1	33.4	44.3	61.4	31.1	42.7
$\sigma = 0.1$	FVSR	Full	Same	20.0	29.7	42.6	42.4	45.7	23.3	14.3	31.1
	PVSR	Partial		28.9	23.7	30.0	30.7	40.7	30.3	14.7	28.4
Average	FVSR/PVSR	Average	Same	64.5	51.4	61.4	59.4	50.2	57.6	55.8	57.2

Figure 5.2: Summary of results for all the objects using the same resolution in the model and the input scan and for different noise levels. The two best descriptors are marked in green, while the two worst are marked in red. (Extracted from [1])

simulation was performed involving 100 Monte Carlo runs, as in the previous simulations, considering solely the four objects that were involved in the real experiment (Ball-Valve, Elbow, R-Tee, and R-Socket). The following simulation parameters: $\langle \text{objects} = 4, \text{resolution} = 0.007, \text{noise} = 0.00625, \text{view} = \text{Full} \rangle$ represent the case closest to the real data. The simulated scans were generated assuming a distance to the object $d = 1.11 \pm 0.56$ m, while varying the yaw and roll angles between -0.4 and 7.4 degrees, while the pitch ranged between -2.2 and 3.9 degrees. The simulated values were chosen randomly within those ranges, being similar to the ones observed in the real experiment. Figure 5.3, shows the respective percentages of how many times each different object class was recognized for both real and simulated runs, so that they can be compared.

		Objects																Average of Recognition	
Descriptors	Experiment	Ball Valve				Elbow				R-Tee				R-Socket				Real	Synth-etic
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4		
ESF	Real	100	0	0	0	0	76	24	0	8	0	92	0	21	2	0	77	86,2	
	Synthetic	100	0	0	0	1	99	0	0	0	0	100	0	0	0	0	100		99,8
VFH	Real	100	0	0	0	31	41	21	7	4	6	90	0	19	2	0	60	72,8	
	Synthetic	100	0	0	0	0	100	0	0	0	0	100	0	0	48	0	52		88,0
CVFH	Real	100	0	0	0	0	93	3	3	0	4	95	0	6	0	94	95,4		
	Synthetic	100	0	0	0	0	100	0	0	0	0	100	0	0	0	0	100		100
OURCVFH	Real	100	0	0	0	0	83	0	17	0	0	96	4	0	13	0	88	91,5	
	Synthetic	100	0	0	0	0	100	0	0	0	0	100	0	0	0	0	100		100
GOOD	Real	100	0	0	0	34	66	0	0	100	0	0	0	33	0	0	67	58,0	
	Synthetic	100	0	0	0	0	100	0	0	100	0	0	0	0	0	0	100		75,0
GFPFH	Real	60	0	0	40	0	48	0	52	4	73	23	0	0	15	0	85	54,1	
	Synthetic	100	0	0	0	0	88	12	0	1	0	99	0	0	0	0	100		96,8
GRSD	Real	88	0	12	0	0	76	0	24	0	33	67	0	0	2	0	98	82,1	
	Synthetic	100	0	0	0	0	98	2	0	0	0	100	0	0	0	0	100		99,5

Figure 5.3: Confusion Matrix for real and synthetic data, for all descriptors. (Extracted from [1])

5.1.3 Discussion

The results of our survey suggested that the best performance for a 3D object recognition system working on 3D point clouds gathered with our custom-developed laser scanner [17] could be achieved when using:

1. Full Views: To use a full view, we need to point the scanner towards the object. This would allow scanning of only one object at a time. Normally, several objects are present in a single scan and, therefore, the system will have to operate in the presence of partial views.
2. The same resolution in the model views and the input scan: In this case a simple decision is to fix the resolution of the views in the data base to the resolution provided by the scanner at the nominal range. If for manipulation purposes we expect to scan objects at 1-1.5 m. distance, the resolution provided by the scanner at this range determines the resolution of the views in the database.
3. The CVFH or OUR-CVFH descriptors: Results on real data suggested that the most promising descriptor was CVFH followed by OURCVFH. Therefore, these two were selected as the candidate descriptors to be used.

Following these guidelines, our next effort focused on implementing a method to recognize several objects within a point cloud. Attempting to recognize several objects in a scan required a method to segment the different objects so that they could be recognized later on.

5.2 3D object recognition method using CVFH, Bayesian estimation and semantic information

The next part of the work focused on developing a method to recognize 3D objects, using the CVFH and OURCVFH point feature descriptors, from uncoloured 3D point clouds scanned. The method is intended to be used for IMR of industrial underwater structures. As a representative example for testing, the developed methods were applied to a test structure consisting of pipes and connected PVC objects, located in the CIRS water tank.

The method was designed and implemented as a processing pipeline involving the following phases:

1. Plane Segmentation: Devoted to removing the floor and the walls of the water tank present in the scans.
2. Pipes Detection: Aiming to detect cylinders representing the pipes.
3. Semantic Object Segmentation: Devoted to segmenting the point clouds corresponding to the 3D objects, which are located either at the pipe intersections or at its extremities.
4. Object Recognition: Used to recognise the class to which each object, segmented at the previous step, belongs.

Unfortunately, the results using only the descriptor for the object recognition phase were poor with respect to the ones identified by our previous work. Averaging over all the objects, a recognition rate of only 51.2% (table. 5.1) was achieved, corresponding to an accuracy of 0.75, a recall of 0.51 and a precision of 0.52. It is worth noting that, in the experimental dataset used in [1], the objects appeared 'unconnected' and isolated. Instead, now, they are inter-connected through pipes and therefore the segmentation is much more challenging, leading to poorer views. Moreover, another problem of performing single-view object recognition is that several objects may have similar views. Partial views of the *R-Tee* may be easily confused with the *Elbow* for instance. Therefore, we proposed to fuse several observations using probabilistic rules, to decide the object class at a certain time. In this direction, a Bayesian estimation method was implemented which computes the conditional probabilities from the confusion matrices estimated in our survey paper [1]. Following this, several observations were used to compute the probability that an object belongs to each object-class, selecting then, the one with highest probability as the solution. To do this, first it is necessary to be able to track the objects across the scans so that their Bayesian probabilities can be iteratively computed.

To reach this objective the IJCBB-based tracking algorithm was proposed to correct the effects of inconsistencies in the robot navigation, which appeared in the form of sudden jumps in the estimated pose of the AUV. These jumps preclude the proper tracking of the objects along scans.

With IJCBB in place, the Bayesian estimation was computed, achieving an improved average accuracy, recall and precision of 0.91, 0.69 and 0.74 respectively. Though the general results averaged over all objects were good, analyzing object-by-object some room for improvement could still be appreciated. In particular, the segmentation method penalized the *Butterfly-valve*, since it often cropped its handle which is a very salient feature.

Since the pipe connectivity of the objects is known in our case, it was natural to constrain the candidate objects to those compatible in terms of pipe connectivity. As an example, if an object is known to have 3 connected pipes, then it cannot be an *Elbow*. It should be either a *R-Tee* or a *3-Way-Valve*. Using the semantic information about pipe connectivity, the object recognition results improved further, reaching a recall of 0.90, an accuracy of 0.93 and a precision of 0.82.

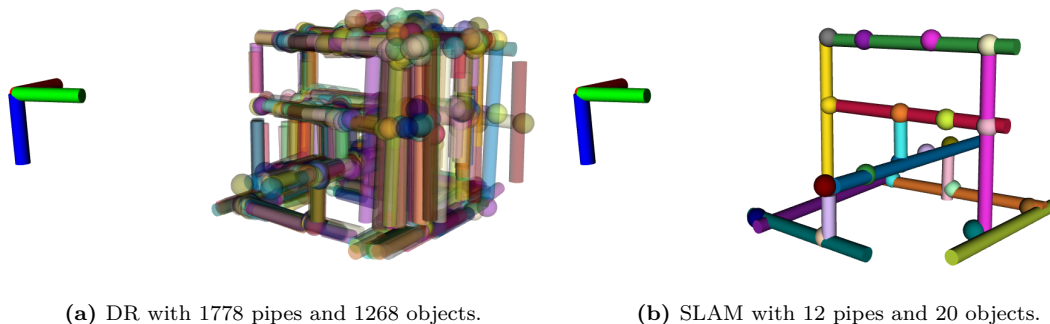
5.2.1 Discussion

The IJCBB is able to solve the object tracking, in the presence of navigation glitches, when at least 3 object pairings between the 2 consecutive scans exist. From a total of 245 scans in the dataset, the use of IJCBB allowed us to process from scan 1 to scan 100. At this point the tracking failed, so it was not possible to associate the object in the posterior scans in order to apply the Bayesian estimation. The results reported in [2] correspond to the first 100 scans. On the other hand, although the tracking allows the building of a map that is significantly better than with the glsDR, it still suffers from some drift. Moreover, our ultimate goal is to implement a drift-less semantic map. Therefore the natural step forward was to formulate the mapping as a feature-based SLAM problem to address both the following issues: 1) To be able to track the objects along the full data set (245 scans), and 2) to integrate the map and the 3D object recognition into a single system.

5.3 Semantic mapping using SLAM and a 3D object recognition system

The last part of the thesis tackles the problem of building a semantic map that is effective for autonomous intervention tasks. In this part, the IJCBB tracking algorithm used to establish the association across the scans is substituted by a SLAM approach. Thus, a combined feature based Extended Kalman Filter (EKF) SLAM method is proposed, using line and point features to represent the pipes and the objects respectively. This method solves 2 problems: 1) it provides a drift-less navigation, 2) it solves the object-association problem for all the scans in the dataset and assigns a globally consistent identifier to every object in every scan, thus enabling Bayesian estimation.

Fig. 5.4 shows the DR and SLAM maps, highlighting the capability of the SLAM to provide a consistent map without duplicated elements.



(a) DR with 1778 pipes and 1268 objects.

(b) SLAM with 12 pipes and 20 objects.

Figure 5.4: Comparison of maps obtained from DR and SLAM with the NED reference frame.

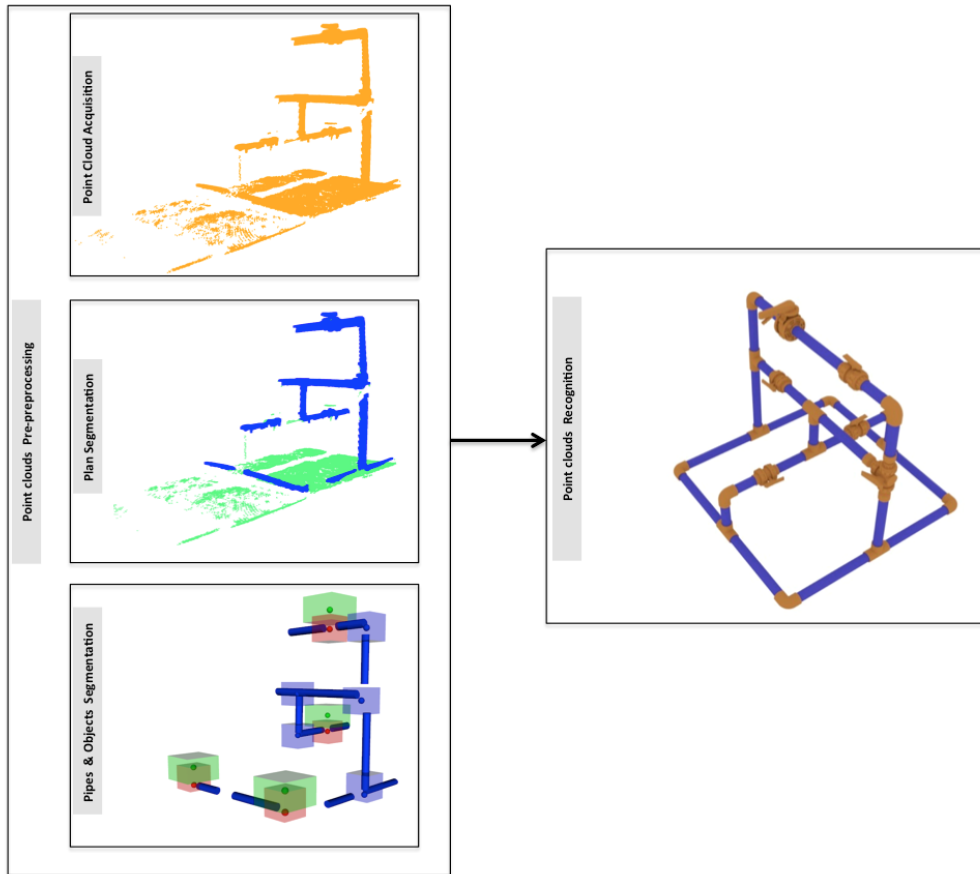


Figure 5.5: 3D Object Recognition Pipeline

Regarding the performance of the object-recognition method, observing the average results reported in table 5.1, it can be appreciated that, as expected, the Bayesian approach improves the results of using the descriptor alone, while the semantic method provides, in this case, the same performance as the Bayesian one. Moreover, when these results are compared with our previous work [2], an increase in performance is observed for the Bayesian method, which can be attributed to the better object segmentation using the 'mushroom' shape, as can be seen in Figure 5.5. The slightly poorer results in the semantic method are due to the fact that the second part of the dataset, from scan 100 to scan 245, is more challenging than the first 100 scans. This becomes clear after noting that the new results, when constrained to the first 100 scans, improve upon the previous reaching the same value for the semantic method.

Conveniently combined with the object-recognition results, the SLAM implements a semantic map endowing the I-AUV with the semantic knowledge required to perform high level commands like *Open Valve X*.

5.3.1 Summary

Table 5.1 summarizes the results achieved during the thesis. First of all, it is worth noting that the recognition rate of an object (also referenced as averaged recognition), as it was referenced in [1], was defined as the number of times an object of a class C is scanned in

Table 5.1: Summary of Results.

Experiment		Survey			ICJBB Tracking 100 scans			SLAM 100 scans			SLAM 245 scans		
Data Type		Synthetic			Real								
Number of Object Detections		-	-	-	523						1163		
Number of Object Classes		7	4		5								
Descriptor	Experiment	Recall	Recall	Recall	Accuracy	Recall	Precision	Accuracy	Recall	Precision	Accuracy	Recall	Precision
CVFH	DESC	0.78	100	95.4	0.75	0.51	0.52	0.48	0.60	0.51	0.42	0.49	0.46
	BAY	-	-	-	0.91	0.69	0.74	0.83	0.85	0.76	0.76	0.84	0.73
	SEM	-	-	-	0.93	0.90	0.82	0.85	0.90	0.81	0.8	0.84	0.78

which the method recognizes it correctly as belonging class C , as a proportion of the total number of times it was scanned. This is actually the recall (number of correct class C recognitions among all the objects of class C scanned). In our survey paper [1], an average object recall of 0.78 was achieved using 7 objects and synthetic data. Since a dataset with 4 of these objects was already available, to make a consistent comparison between synthetic and real data, a new simulation was carried out in similar conditions to those corresponding to the experiment. An average (for all objects) recall of 1 was found in the simulation. It is worth noting that the objects were lying on the floor and that a very good global zenithal view of them was available. The performance slightly decreased to an average object recall of 0.95 when using real data gathered with the AUV in the water tank. When the dataset involving the full structure, where the objects appear connected to the pipes, is processed using the IJCBB tracking algorithm, the average object recall significantly decreased to 0.51. It is worth noting that now the objects do not appear isolated but connected to pipes, so they had to be segmented. Moreover, now many different views of the objects are available, not just simple global zenith views. This explains the decreased performance. Nevertheless, it was possible to improve the performance introducing the Bayesian estimation, reaching an average object recall of 0.69% first, increased to 0.90 later by using the semantic method.

It should also be noted that only the first 100 scans of this dataset were processed, since tracking the objects was not possible beyond scan 100. This was solved later by introducing a feature-based SLAM method able to track the objects along all the dataset scans. In this case, when the whole dataset is processed, an averaged object recall of 0.49, 0.84 and 0.84 was achieved for the Descriptor, Bayesian and Semantic methods respectively. The results are in the same order of magnitude as in the previous case when using the descriptor alone, improved when using the Bayesian approach and provided a slightly lower performance when using the semantic method. To accurately compare the results reported with and without the SLAM it is necessary to restrict the analysis to the first 100 scans only, since the experiment using the IJCBB failed beyond that point. This result is also provided in table 5.1 where it can be appreciated that slightly better results were achieved using SLAM. It should be noted that apart from using SLAM for data association, several improvements in plane and object segmentation were also implemented. This illustrated that the slight decrease in recall of the semantic method is actually due to the fact that the second part of the dataset is more challenging than the first part.

5.4 Computation Time analysis

The computational time needed to process one scan includes the time required for : 1) detection and segmentation of planes; 2) detection of pipes and 3) object segmentation and recognition. For each scan, fig. 5.6 shows the first time in red, and the second one in green. All times reflect the processing on a standard personal computer with a 2.20 GHz i7 processor and 16 GB of memory running Ubuntu Linux 16.04 LTS. As shown in the time histogram of fig. 5.6, the plane segmentation and detection takes an average of 6.08 seconds with an standard deviation of 1.04. The pipe detection takes an average of 8.54 seconds with a standard deviation of 2.46. Moreover the average scan processing time is 15.73 seconds with standard deviation of 2.80. Fig. 5.6 also shows the complete scan processing time in black colour. The time required for object segmentation and recognition is the total processing time (black) minus the sum of the plane (red) and pipe (green) detection times. This time is also plotted in fig. 5.7 for the Bayesian and the Semantic methods. The figure shows, for each scan, the time spent for object segmentation and recognition averaged for each object. When the Bayesian method is used, the average object time is 0.0042 seconds with an standard deviation of 0.0036. For the Semantic method the mean time is 0.0033 seconds and the standard deviation is 0.0028.

During the experiment reported in section 5.1 of Chapter 4, the time required to acquire a scan at the programmed resolution is approximately 2.2 seconds. According to the times reported in fig.5.6, in average, we need 15.73 seconds to process one scan. This reveals that, with the current implementation, it is not possible to build the map in real time. In average, we can only process 1 every 7 scans. This opens future lines of research to achieve real time semantic mapping:

- The first solution is to use a naïve scan-stop & process approach, where the robot is stopped while processing the scan. However this is not a practical solution.
- A more elegant solution would be to avoid segmenting and detecting planes (the costly operation) for every scan. Instead these operations would only be done for a subset of scans. The idea would be to detect pipes in scan k and scan $k + n$, assuming enough overlap between the scans, then associate them and update the filter. A smoothing of the in-between robot trajectory would be performed using a separated filter. Then, the already mapped pipes could be used for the semantic object segmentation of the intermediate scans ($k + 1$ to $k + n - 1$), whose pose had already been estimated with the smoothing filter, leading the point clouds of the object views to be used for Bayesian/Semantic object recognition. This would significantly reduce the processing time.
- Finally, it should be studied the potential processing time reduction that could be achieved by using a GPU based implementation of the PCL. For some algorithms like ICP, a factor of 10, in the processing time, has been reported by the community.

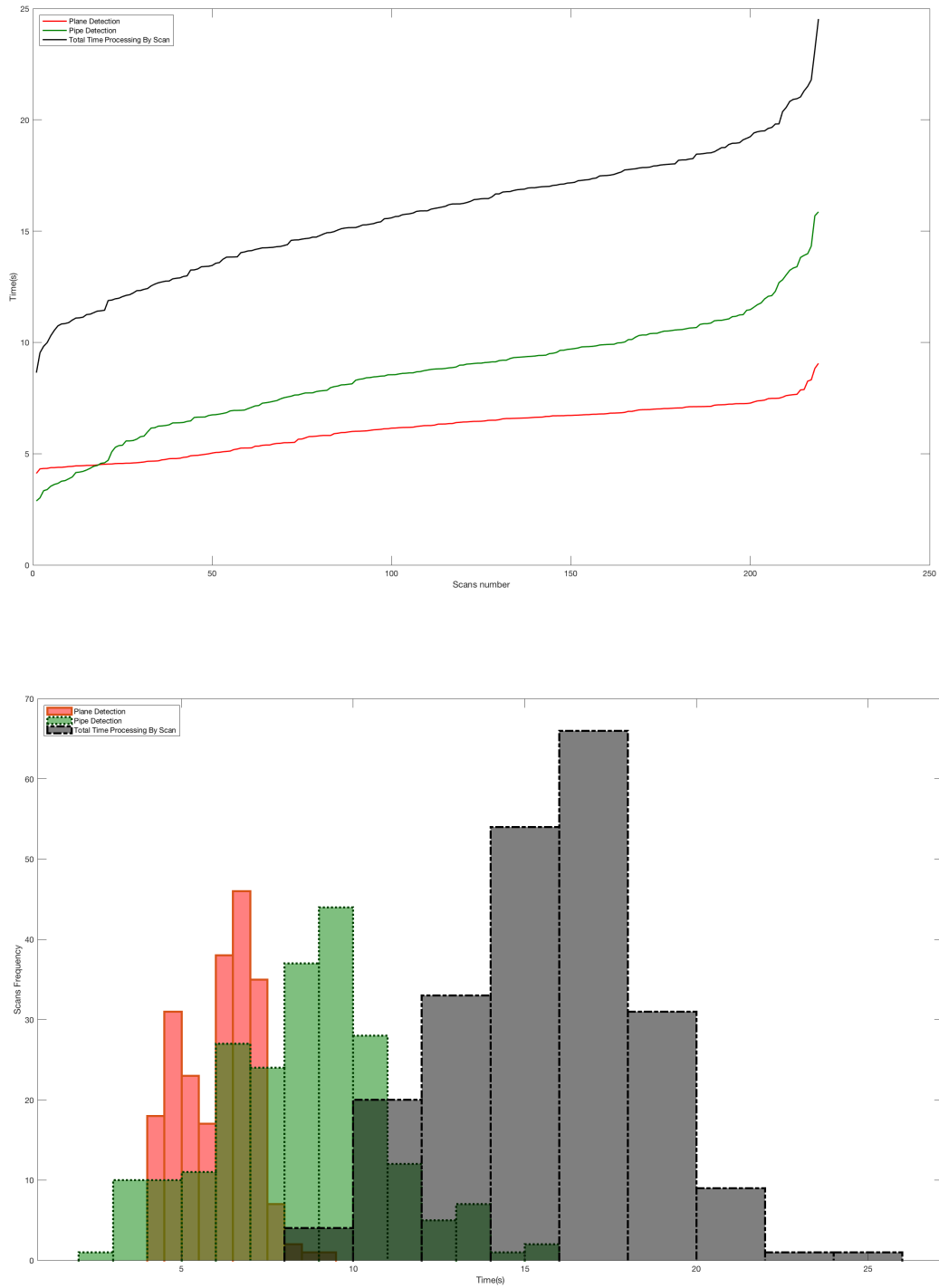


Figure 5.6: Processing time required for Plane segmentation and Pipe detection.

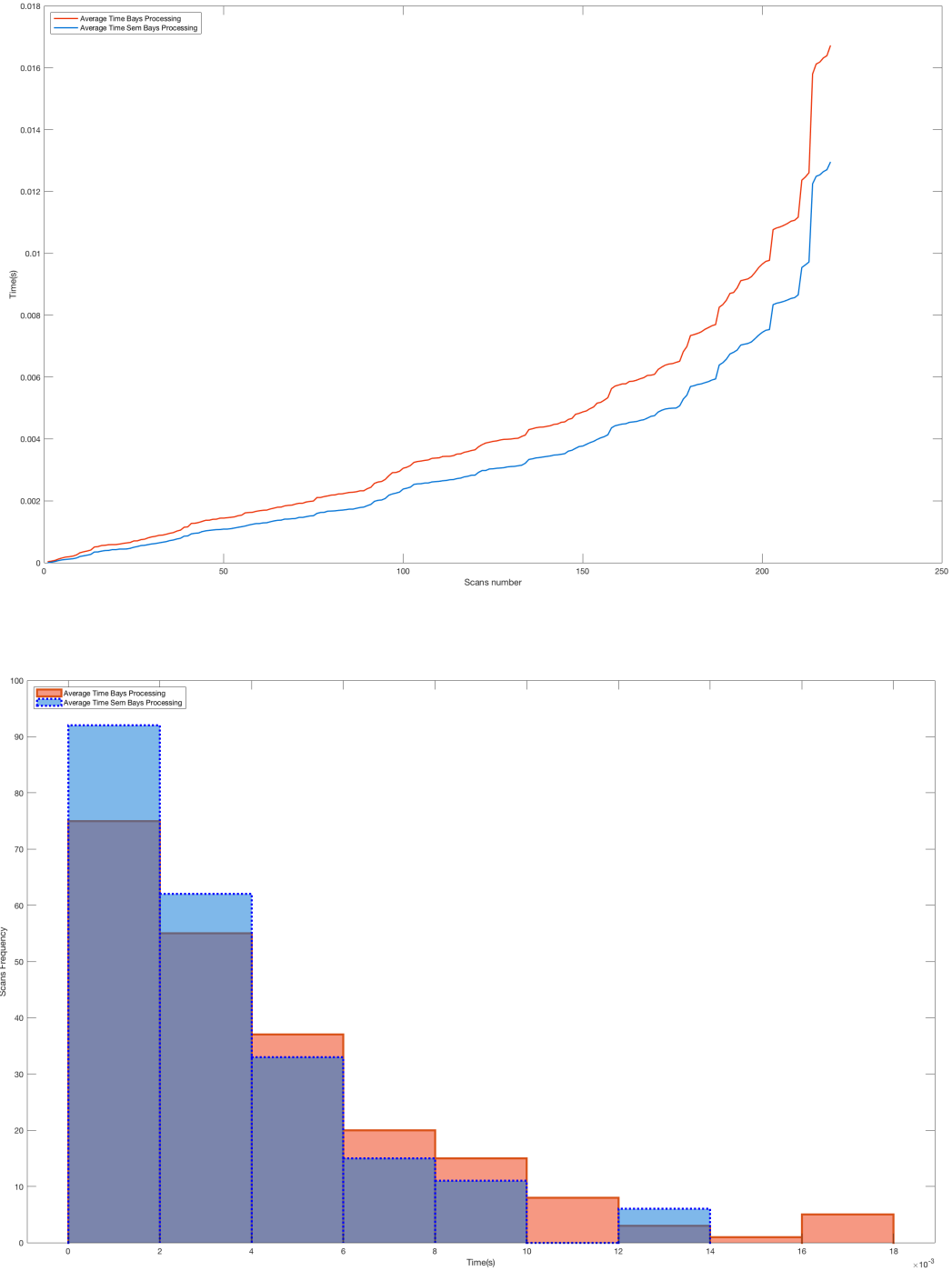


Figure 5.7: Required processing time for Bayesian and semantic based recognition method.

6

CONCLUSIONS

IN this chapter we summarize the work done in this Thesis and explain the main conclusions we have drawn. Later, we present some lines for future research.

6.1 Conclusions

This thesis has proposed a method for semantic mapping by successfully integrating a feature-based SLAM algorithm with a 3D object recognition system, using as input non-coloured 3D point clouds. The system is intended to assist autonomous mobile manipulation underwater by exploiting semantic information. All the objectives detailed in chapter 1.2 have been successfully completed and lead to the following contributions:

1. The proposal of a 3D object recognition system using point-features and a data-base containing the 3D models of the objects to be recognized.

- (a) First, we carried out an exhaustive comparative study of the global point-feature descriptors available in the PCL using both synthetic and real data. This helped to decide the most appropriate descriptor, but also pointed out the relevance of using full views, the same resolution between the data-base views and the scanned ones, and how the resolution and the noise affect recognition performance.
- (b) Next we proposed an object recognition pipeline based on the matching of the current object view, described with the best performing descriptor (with real data), against the object views stored in the database corresponding to a complete set of views of all the *a priori* known objects.

2. The use of Bayesian inference, semantic segmentation and semantic object information to improve the recognition results.

- (a) We proposed the use of an object tracking algorithm (IJCBB) to be able to associate and track the object detections along successive scans. This algorithm relates the detections of a single object along the temporal scans allowing, therefore, fusing of several object observations into a single object class recognition.
- (b) We proposed to use semantic information about object-pipe connectivity to separate and distinguish, at the point-cloud level, the points that belong to objects and those that belong to connecting pipes.
- (c) To advance beyond the performance achievable by using only a single view of the scanned object, we proposed the use of Bayesian estimation to compute the likelihood that an object observation corresponds to each one of the potential classes, selecting the most likely one as the recognized class.
- (d) We refined the method to use semantic information about the object connectivity (number of connected pipes) to decide the set of potential object classes. Thus, given an object observation, the Bayesian estimation only takes into account those classes which are compatible with the actual object connectivity.

3. The implementation of a semantic map integrating the object recognition system with a feature-based SLAM.

- (a) We substituted the use of the IJCBB object tracking, by a feature-based SLAM able to track the objects as well as to provide their drift-less location in a global frame of reference.
- (b) We improved the object segmentation method to avoid unwanted crops of salient features such as the valve handle.

- (c) We improved the plane-detection module using a region-growing segmentation algorithm.

6.2 Future Work

Although the central objectives established for the thesis have been accomplished, there are several relevant directions for future work, which are presented in this section.

There are still many challenges and difficulties to overcome in 3D object recognition systems. First, this is due to the fact that the collected point-clouds from complex scenes contain noise, occlusions, clutter and non-uniform resolution, which significantly affect the accuracy of 3D object recognition. Naturally the more occlusions or ambiguities are present in the scene, the lower the recognition rate will be. In addition, the dependence of the recognition system on the segmented views makes it vulnerable in some cases.

In this dissertation, the focus was on 3D object recognition and the view-selection problem was partially ignored. Future work will address these challenges by:

- Considering better selection of salient views in the database and tested views from the scanned objects, which raises the questions of what are the criteria for defining a good view.
- Fusing several views into a single and wider one, taking advantage of the object pose estimated by the SLAM, and carrying out a further fine-level co-registration. Such wider views would be more informative and thus more discriminant, having the potential to support better recognition.
- A longer term strategy to improve the observation quality is to perform view planning in order to reduce the ambiguity caused by poorly observed objects. Such view planning should be multi-objective in the sense of taking into account multiple objects simultaneously, and should be guided towards the next best views that solve the ambiguity between the most probable classes for each object.

In the longer-term, we plan to use the semantic map as a building block for a complete autonomous intervention system using I-AUVs in IMR scenarios involving industrial underwater structures. Making the robot aware of the objects in its environment and their location, will enable semantic manipulation. Then, it will be possible to issue commands like "TURN VALVE V_i " or "UNPLUG CONNECTOR C_j " instead of controlling a ROV and a robotic arm in a cumbersome configuration space.

BIBLIOGRAPHY

- [1] **K. Himri**, P. Ridaou, and N. Gracias. “3D Object Recognition Based on Point Clouds in Underwater Environment with Global Descriptors: A Survey”. In: *Sensors* 19.20 (2019), p. 4451 (cit. on pp. vii, 12, 15, 117–119, 121, 122).
- [2] **K. Himri**, P. Ridaou, and N. Gracias. “Underwater Object Recognition Using Point-Features, Bayesian Estimation and Semantic Information”. In: *Sensors* 21.5 (2021). ISSN: 1424-8220. DOI: [10.3390/s21051807](https://doi.org/10.3390/s21051807) (cit. on pp. vii, 13, 57, 120, 121).
- [3] G. Villacrosa, **K. Himri**, P. Ridaou, and N. Gracias. “Semantic Mapping for Autonomous Subsea Intervention”. In: *Submitted to MDPI Sensors* (2021) (cit. on pp. vii, 13, 85).
- [4] **K. Himri**, P. Ridaou, N. Gracias, A. Palomer, N. Palomeras, and R. Pi. “Semantic SLAM for an AUV using object recognition from point clouds”. In: *Proceedings of 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2018*. Vol. 51. 29. 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2018. Croatia, Sept. 2018, pp. 360–365. DOI: <https://doi.org/10.1016/j.ifacol.2018.09.497> (cit. on pp. vii, 116).
- [5] **K. Himri**, R. Pi, P. Ridaou, N. Gracias, A. Palomer, and N. Palomeras. “Object Recognition and Pose Estimation using Laser scans for Advanced Underwater Manipulation”. In: *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*. IEEE. 2018, pp. 1–6 (cit. on p. vii).
- [6] E. Che, J. Jung, and M. J. Olsen. “Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review”. In: *Sensors* 19.4 (2019), p. 810 (cit. on p. 8).
- [7] M. Soilán, A. Sanchez-Rodriguez, P. del Rio-Barral, C. Perez-Collazo, P. Arias, and B. Riveiro. “Review of laser scanning technologies and their applications for road and railway infrastructure monitoring”. In: *Infrastructures* 4.4 (2019), p. 58 (cit. on p. 9).
- [8] P. Kim, J. Chen, J. Kim, and Y. K. Cho. “SLAM-driven intelligent autonomous mobile robot navigation for construction applications”. In: *Workshop of the European Group for Intelligent Computing in Engineering*. Springer. 2018, pp. 254–269 (cit. on p. 9).
- [9] Y. Pan, Y. Dong, D. Wang, A. Chen, and Z. Ye. “Three-dimensional reconstruction of structural surface model of heritage bridges using UAV-based photogrammetric point clouds”. In: *Remote Sensing* 11.10 (2019), p. 1204 (cit. on p. 9).

- [10] A. Al-Rawabdeh, F. He, A. Moussa, N. El-Sheimy, and A. Habib. “Using an unmanned aerial vehicle-based digital imaging system to derive a 3D point cloud for landslide scarp recognition”. In: *Remote sensing* 8.2 (2016), p. 95 (cit. on p. 10).
- [11] P. Kim, J. Chen, and Y. K. Cho. “Robotic sensing and object recognition from thermal-mapped point clouds”. In: *International Journal of Intelligent Robotics and Applications* 1.3 (2017), pp. 243–254 (cit. on p. 10).
- [12] P. Kima, J. Chenb, and Y. K. Choa. “Building element recognition with thermal-mapped point clouds”. In: *34th International Symposium on Automation and Robotics in Construction (ISARC 2017)*. 2017 (cit. on p. 10).
- [13] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. “Towards 3D point cloud based object maps for household environments”. In: *Robotics and Autonomous Systems* 56.11 (2008), pp. 927–941 (cit. on pp. 10, 11).
- [14] R. B. Rusu, B. Gerkey, and M. Beetz. “Robots in the kitchen: Exploiting ubiquitous sensing and actuation”. In: *Robotics and Autonomous Systems* 56.10 (2008), pp. 844–856 (cit. on pp. 10, 11).
- [15] N. Blodow, L. C. Goron, Z.-C. Marton, D. Pangercic, T. Rühr, M. Tenorth, and M. Beetz. “Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments”. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE. 2011, pp. 4263–4270 (cit. on pp. 10, 11).
- [16] R. B. Rusu and S. Cousins. “3D is here: Point Cloud Library (PCL)”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China, May 2011 (cit. on p. 116).
- [17] A. Palomer, P. Ridao, D. Youakim, D. Ribas, J. Forest, and Y. Petillot. “3D Laser Scanner for Underwater Manipulation”. In: *Sensors* 18.4 (2018). ISSN: 1424-8220. DOI: 10.3390/s18041086 (cit. on p. 118).

