sensors

MDPI

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

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 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.

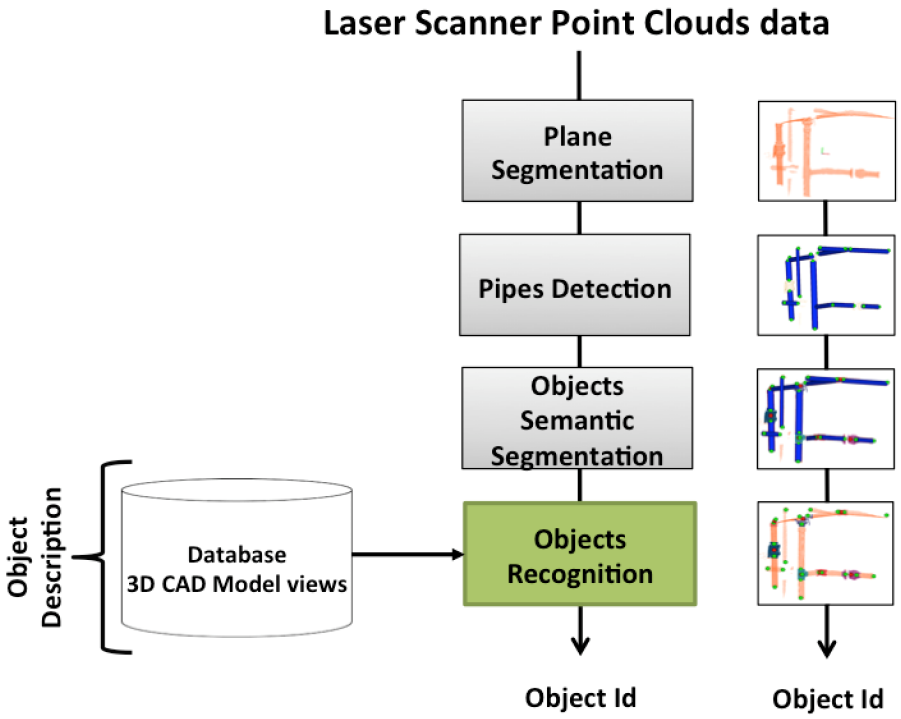


Figure 1. 3D Object Recognition Pipeline.

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	

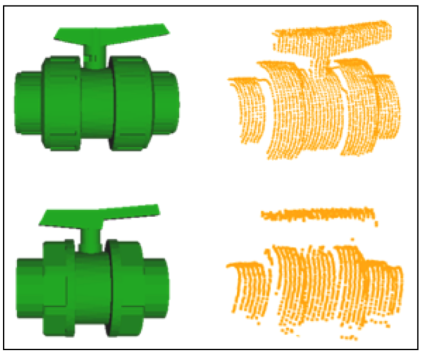


Figure 2. Ball-valve (**top**) and Ball-valve-s (**bottom**) with their respective segmented scan.

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 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 where 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

function `DetectPipes`(***in:*** *scan*, ***out:*** P_I)**:**
  `// Returns the set of pipes` P_I `detected in the scan using RANSAC`
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}`)`
function `PipeSegmentation`(***in:*** *scan*, ***out:*** P_O)**:**
  `// Returns the set of non duplicated pipes present in the scan`
  P_I =`DetectPipes`(*scan*) `// get the set detected pipes`
  forall $P_i \in P_I$ **do**
    `//` M_{P_i} `set of duplicated pipes to be merged`
    $M_{P_i} = \{P_i\} \cup \{P_j \in P_I | \exists P_k \in M_{P_i}, (Colinear(P_j, P_k) \wedge Overlapped(P_j, P_k))\}$
    $P_{M_{P_i}}$ = `MergePipes`(M_{P_i}) `// merge the duplicated pipes`
    $P_O = P_O \cup \{P_{M_{P_i}}\}$ `// add to the set of detected pipes`
    $P_I = P_I \setminus M_{P_i}$ `// substract` M_{P_i} `form` P_I
return P_O

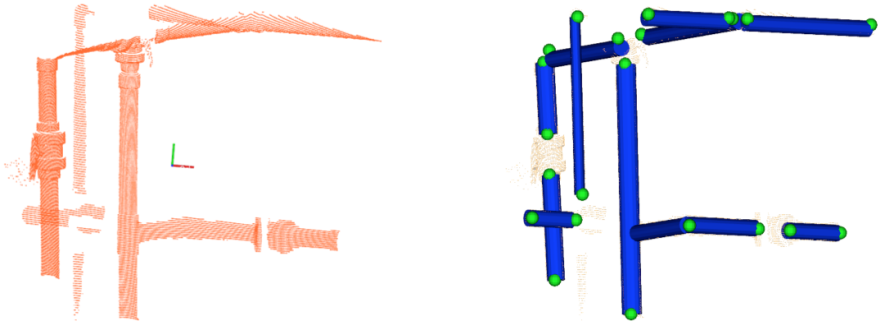


Figure 3. Pipes detection: (**left**) 3D laser scan point cloud; (**right**) pipes with their respective endpoints.

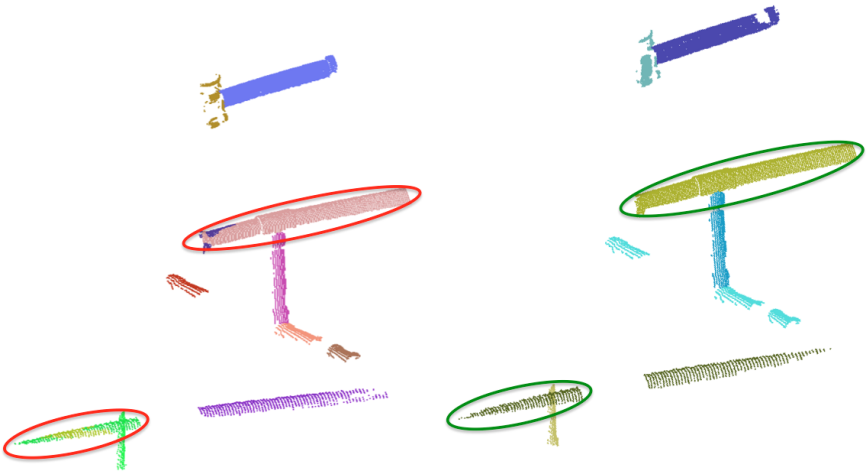


Figure 4. Pipes Merging: (**left**) Pipe detection result previous to merging showing, within circles, multiple pipe detections of the same pipe; (**right**) Result after merging where the multiple detections have been merged into a single one.

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.

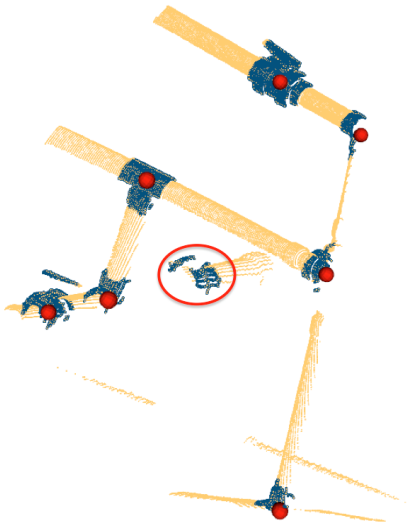


Figure 5. Semantic Segmentation: Red points represent the centroids of segmented objects. The red circle shows a segmented object located at an isolated extremity.

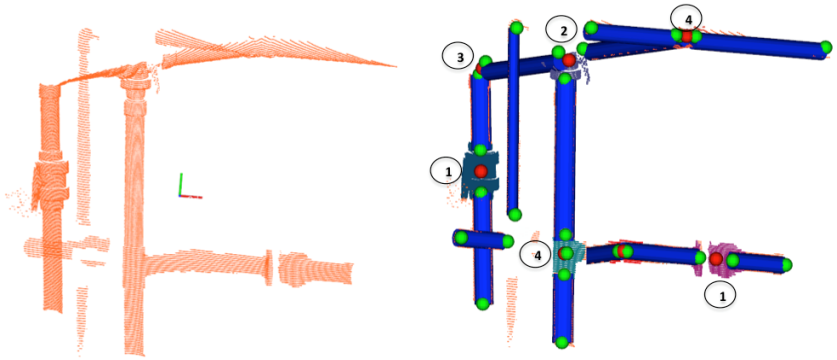


Figure 6. Semantic Segmentation: (**Left**) Input 3D point cloud; (**Right**) Pipes (blue cylinders) with their endpoints (green spheres), and the centroids of the objects to be segmented (red spheres) along with the segmented objects point clouds (colored). The objects 1, 2, 3, 4 represent respectively: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

Algorithm 2: Semantic Segmentation

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
  (c_i, c_j)=`LineToLineSegment(`P_i, P_j`)`
  $d = ||c_i - c_j||$ `//` Compute the line to line distance
  $intersection = (c_i + c_j)/2$ `//` midpoint $\equiv$ intersection $\equiv$ obj pos
  $d1_{P_i} = ||intersection - Extremity(1, P_i)||$ `//` distance to extremities
  $d2_{P_i} = ||intersection - Extremity(2, P_i)||$
  $d1_{P_j} = ||intersection - Extremity(1, P_j)||$
  $d2_{P_j} = ||intersection - Extremity(2, P_j)||$
  if $(d < \tau_d)$ **then** `//` coplanar?
    if $(d1_{P_i} < \tau_d)$ **&&** $(d1_{P_j} < \tau_d)$ **then return** *connected*=***true***;
    if $(d1_{P_i} < \tau_d)$ **&&** $(d2_{P_j} < \tau_d)$ **then return** *connected*=***true***;
    if $(d2_{P_i} < \tau_d)$ **&&** $(d1_{P_j} < \tau_d)$ **then return** *connected*=***true***;
    if $(d2_{P_i} < \tau_d)$ **&&** $(d2_{P_j} < \tau_d)$ **then return** *connected*=***true***;
  else
    return *connected*=***false***
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
function `ComputeExtremityLocations(`***in:*** P, C_i, ***out:*** C_e`)`**:**
  `//` Returns the set of obj locations at the isolated pipe extremities
function `SemanticSegmentation(`***in:*** *scan*, P, ***out:*** O`)`**:**
  `//` Returns the set O of objects locations and their cropped point clouds
  $C_P = \emptyset$ `//` set of pipe pairs intersections
  forall $(P_i, P_j) \in P \times P | i \neq j$ **do**
    if `Connected(`$P_i, P_j, intersection$`)` **then** $C_P = C_P \bigcup \{(intersection, P_i, P_j)\}$;
  C_i =`ComputeIntersectionLocations(`C_P`)` `//` set of obj pos at pipe intersections
  C_e =`ComputeExtremityLocations(`P, C_i`)` `//` set of obj pos at pipe extremes
  $O = \emptyset$
  forall $objpos \in C_i \cup C_e$ **do**
    $objpc$ =`CropObject(`$objpos, scan$`)` `//` crop the obj point cloud
    $O = O \cup \{< objpos, objpc >\}$
  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 $E = \{\boldsymbol{e_1}, \ldots, \boldsymbol{e_m}\}$

and $F = \{\boldsymbol{f_1}, \ldots, \boldsymbol{f_n}\}$ 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:

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

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:

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

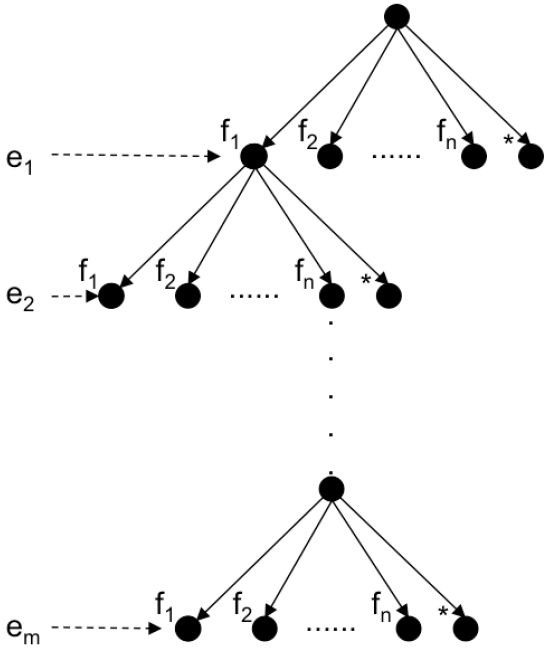


Figure 7. Interpretation tree stating, for each object e_i (level i) its potential associations $f_{1\ldots n}$, representing the (*) node, a spurious measurement.

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 $\boldsymbol{\eta_k}$ (yellow) moving, during a small time interval Δt, a displacement $\boldsymbol{\Delta\eta}$ to achieve a new pose $\boldsymbol{\eta_{k+1}}$ (green). Let $\hat{\boldsymbol{\eta}}_k$, $\boldsymbol{\Delta\hat{\eta}}$ and $\hat{\boldsymbol{\eta}}_{k+1}$ be the estimates of the corresponding vectors. If $\boldsymbol{\Delta\hat{\eta}}$ is incorrect due to a failure in the navigation sensors, the estimated robot location at time $k+1$ ($\hat{\boldsymbol{\eta}}_{k+1}$) is also erroneous (frame $\{E_{k+1}\}$ in orange). Now, let us consider 3 equidistant objects: $\boldsymbol{o_1}$, $\boldsymbol{o_2}$ and $\boldsymbol{o_3}$, observed from $\{S_k\}$ as: $\boldsymbol{e_1}$, $\boldsymbol{e_2}$ and $\boldsymbol{e_3}$ as well as from $\{S_{k+1}\}$ as: $\boldsymbol{f_1}$, $\boldsymbol{f_2}$ and $\boldsymbol{f_3}$. Since the 3 inter-distances are equal, 6 possible pairings exist ($\{\boldsymbol{e_1f_1, e_2f_2, e_3f_3}\}$, $\{\boldsymbol{e_1f_2, e_2f_3, e_3f_1}\}$, $\{\boldsymbol{e_1f_3, e_2f_1, e_3f_2}\}$, $\{\boldsymbol{e_1f_3, e_2f_2, e_3f_1}\}$, $\{\boldsymbol{e_1f_1, e_2f_3, e_3f_2}\}$, $\{\boldsymbol{e_1f_2, e_2f_1, e_3f_3}\}$), 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 H_L = IJCBB([],[],1,e_{1...m},f_{1...n})
// H_L[i] = j ⇒ (e_i, f_j) is a pairing; H_L[i] = 0 ⇒ e_i is not paired
// e_{1...m} : object locations in the first scan S_E indexed by i
// f_{1...n} : object locations in the second scan S_F indexed by j
1  procedure IJCBB(in: H, Best, i; out: H, Best):
2      if i > m then                                              /* leaf node?    */
3          if Pairings(H) > Pairings(Best) then
4              Best ← H;
5      else
6          for j=1 to n do
7              if JointCompatible(H, i, j) then
8                  IJCBB([H j], Best, i + 1)                      /* (e_i, f_j) accepted */
9          if Pairings(H) + m − i > Pairings(Best) then
10             IJCBB([H 0], Best, i + 1)                          /* star node, e_i, not paired */
11 function JointCompatible(in: H, i, j, out: compatible):
       // Returns true if
       // τ_ψ : maximum rotation Δψ that can be experimented in Δt 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(H,i,j)
17         otherwise do
18             return (InterdistanceCompatible(H,i,j) && GetRotation() < τ_ψ)
```

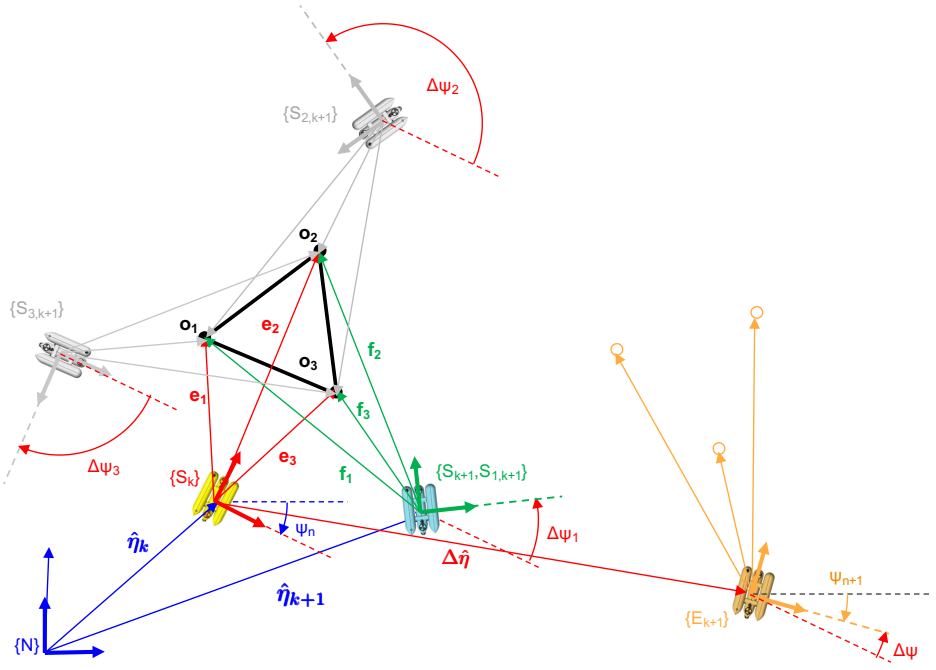


Figure 8. Roto translation estimation.

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.

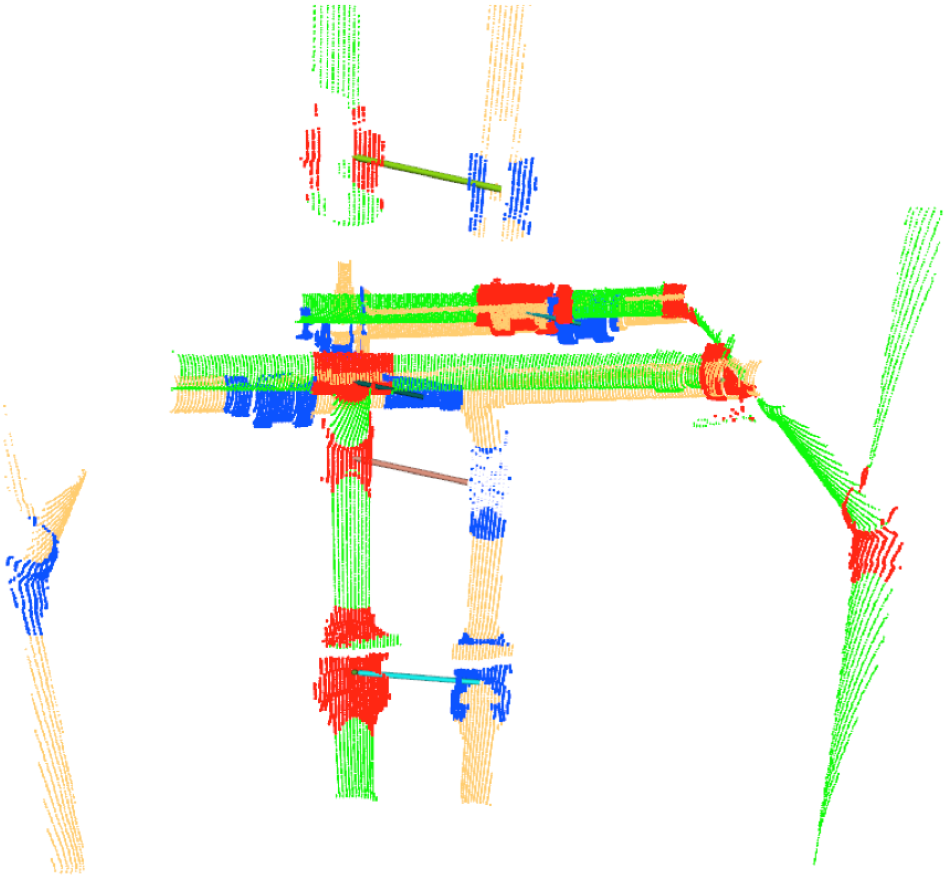


Figure 9. Tracking objects over two consecutive scans, represented in green/red and yellow/blue. The significant displacement between the two scans is the results of navigation inaccuracies from noisy Doppler Velocity Log (DVL) readings in the test pool. The solid lines indicate the objects associated 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, Z_C; out: O):
4      return Recognition(O, Z_C, O)
5  function Recognition(in: O, Z_C; out: O):
       // Z_C ∈ {1,...,6} Detected Class
       // O = ⟨seen : boolean, P = [P(X_1),···,P(X_6)], n_p, id⟩}
       // Observation probabilities extracted from the Confusion Matrix
6      P_{Z|X} = ( P(Z_1|X_1)  P(Z_1|X_2)  ...  P(Z_1|X_6)
                   P(Z_2|X_1)  P(Z_2|X_2)  ...  P(Z_2|X_6)
                       ⋮           ⋮        ⋱       ⋮
                   P(Z_6|X_1)  P(Z_6|X_2)  ...  P(Z_6|X_6) )
7      S_C=CompatibleClasses(O)                    // Set of compatible classes
8      if ¬(O.seen) then
9          O.seen = true                           // First Observation of O
10         forall j ∈ S_C do
11             O.P[j] = 1/#S_C                     // All classes are equiprobable
12     forall j ∈ S_C do
13         P[j] = P_{Z|X}[Z_C, j] * O_i.P[j]       // Non normalized Bayesian prob:
                  P(Z_C|X_j) * P(X_j)
14     η = 1/Σ_{j∈S_C} P[j]                        // Compute the Normalizer
15     forall j ∈ S_C do
16         O.P[j] = η * P[j]                       // Normalized Bayesian probabilities
17     O.id = argmax_{j∈S_C} O.P[j]                // Select the Most likely class
18     return O.id
```

Table 3. Confusion Matrices expressed as a numerical %.

		Objects																											
Descriptors	**Experiment**	**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 candidates 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.

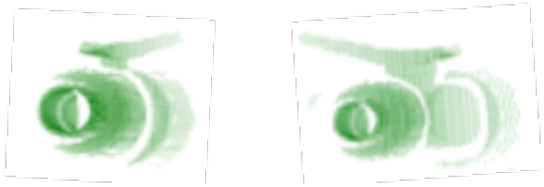
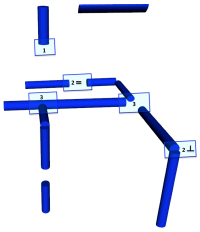


Figure 10. Confusing Views of the Ball-Valve and 3-Way-Valve objects.

Algorithm 5: Semantic-based Recognition

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

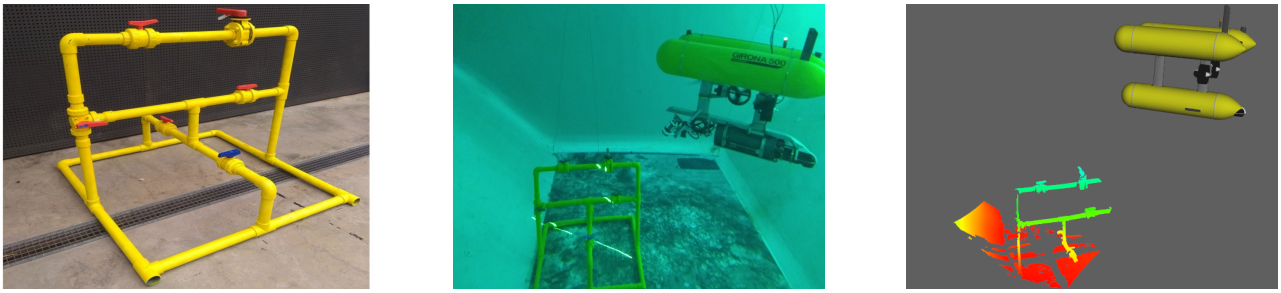
During the experiment, 100 scans were processed containing a total of 523 object observations of 13 different objects from 6 different classes.

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].



a **b** **c**

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.

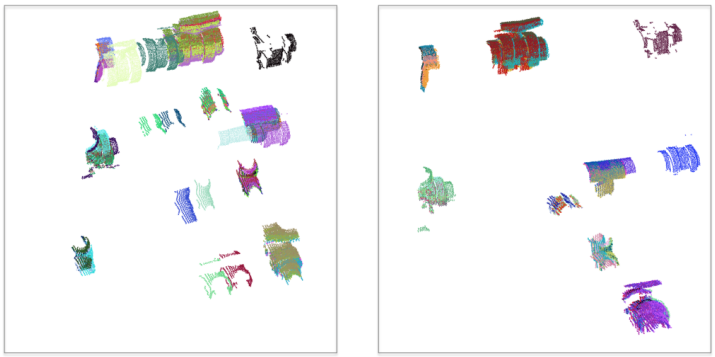


Figure 12. Mapped object point clouds: (**Left**) Located at their dead reckoning position; (**Right**) Located at the position estimated by the tracking using the *IJCBB* algorithm on the right.

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.

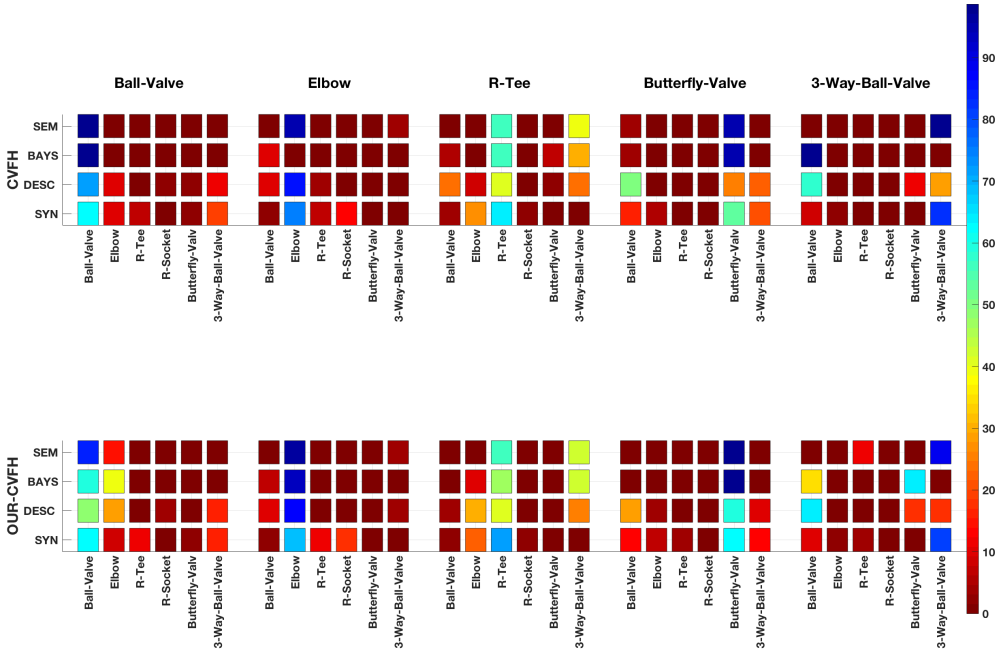


Figure 13. Graphical representation of the Confusion Matrices.

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.

		Objects																	
Descriptors	**Experiment**	**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

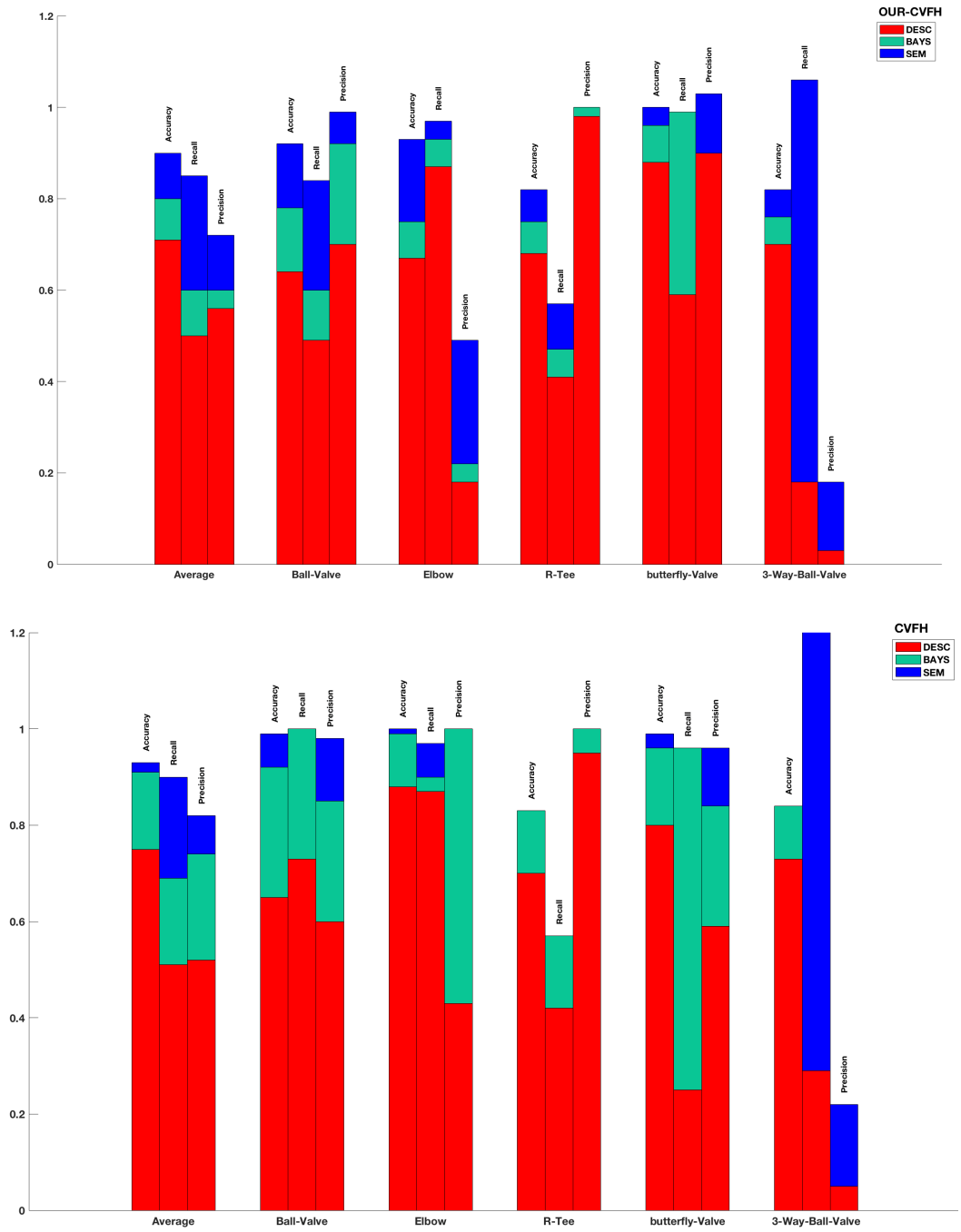


Figure 14. Evaluation of the recognition performance using Accuracy, Recall and Precision for descriptor-based, Bayesian-based and semantic-based method for both: (**Top**) OUR-CVFH; (**Bottom**) CVFH.

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.

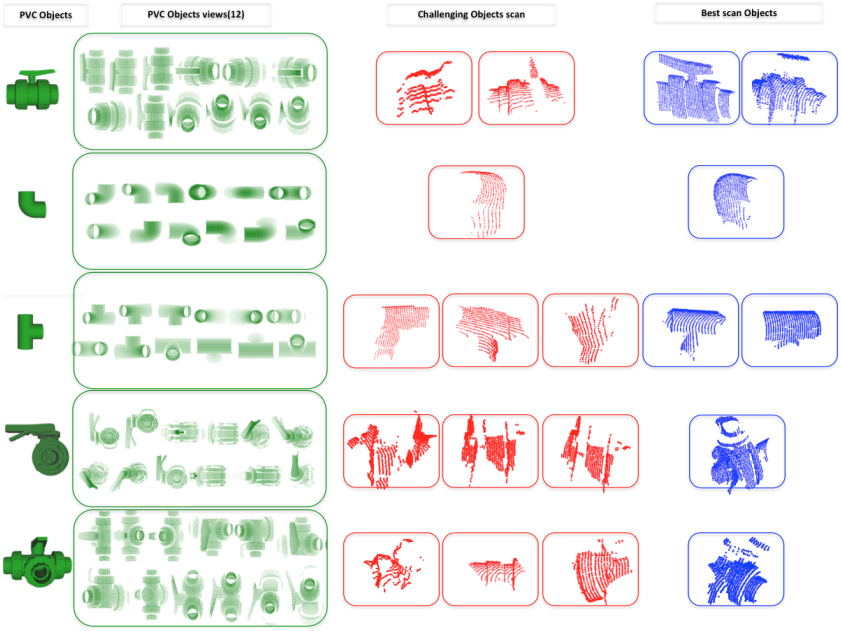


Figure 15. PVC objects used in the experiment (first column) with their respective database views (second column). The last two columns provide manually selected examples of segmented objects from the experiments, with the most difficult in red and the easiest in blue.

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.; Ridao, 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 2011; 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**, 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.; Ridao, P.; Forest, J.; Ribas, D. Underwater Laser Scanner: Ray-Based Model and Calibration. *IEEE/ASME Trans. Mechatronics* **2019**, *24*, 1986–1997. [CrossRef]