sensors

MDPI

Article

Real-Time Pipe and Valve Characterisation and Mapping for Autonomous Underwater Intervention Tasks

Miguel Martin-Abadal, Gabriel Oliver-Codina and Yolanda Gonzalez-Cid *

Department of Mathematics and Computer Science, University of the Balearic Islands, 07122 Palma, Spain
* Correspondence: yolanda.gonzalez@uib.es

Abstract: Nowadays, more frequently, it is necessary to perform underwater operations such as surveying an area or inspecting and intervening on industrial infrastructures such as offshore oil and gas rigs or pipeline networks. Recently, the use of Autonomous Underwater Vehicles (AUV) has grown as a way to automate these tasks, reducing risks and execution time. One of the used sensing modalities is vision, providing RGB high-quality information in the mid to low range, making it appropriate for manipulation or detail inspection tasks. This work presents the use of a deep neural network to perform pixel-wise 3D segmentation of pipes and valves on underwater point clouds generated using a stereo pair of cameras. In addition, two novel algorithms are built to extract information from the detected instances, providing pipe vectors, gripping points, the position of structural elements such as elbows or connections, and valve type and orientation. The information extracted on spatially referenced point clouds can be unified to form an information map of an inspected area. Results show outstanding performance on the network segmentation task, achieving a mean *F1-score* value of 88.0% at a pixel-wise level and of 95.3% at an instance level. The information extraction algorithm also showcased excellent metrics when extracting information from pipe instances and their structural elements and good enough metrics when extracting data from valves. Finally, the neural network and information algorithms are implemented on an AUV and executed in real-time, validating that the output information stream frame rate of 0.72 fps is high enough to perform manipulation tasks and to ensure full seabed coverage during inspection tasks. The used dataset, along with a trained model and the information algorithms, are provided to the scientific community.

Keywords: autonomous intervention; underwater perception; deep learning; point cloud segmentation; pipeline characterisation; pipeline mapping; real-time

Citation: Martin-Abadal, M.; Oliver-Codina, G.; Gonzalez-Cid, Y. Real-Time Pipe and Valve Characterisation and Mapping for Autonomous Underwater Intervention Tasks. *Sensors* **2022**, *22*, 8141. https://doi.org/10.3390/s22218141

Academic Editors: Nuno A. Cruz and Pedro Jorge

Received: 27 September 2022
Accepted: 18 October 2022
Published: 24 October 2022

1. Introduction

The need for conducting underwater intervention tasks has grown significantly in recent decades. Often it is necessary to perform underwater operations in different fields such as archaeology, biology, rescue and recovery or industry that include not only inspection but also interaction with the environment. One of the most relevant cases is the manipulation tasks performed on offshore oil and gas rigs or pipeline networks [1–5].

In the past, the aforementioned tasks were mostly carried out, manually, by scuba divers. Nonetheless, conducting these missions in a hard-to-reach scenario such as open waters tends to be slow, dangerous and resource-consuming. Recently, Remotely Operated Vehicles (ROVs) equipped with diverse sensing systems and manipulators have been used to access deeper and more complex underwater scenarios, allowing the elimination of some of the drawbacks of human intervention.

However, ROVs still presented downsides such as its hard and error-prone piloting due to complex water dynamics, requiring trained operators; or the need for a support vessel, leading to expensive operational costs. To ease these drawbacks, there has been

increasing research towards intervention Autonomous Underwater Vehicles (AUVs) [6–8] and Underwater Vehicle Manipulator Systems (UVMS) [9,10].

Other challenges faced in underwater environments are presented regarding sensing in general and object perception in particular. Underwater sensing presents several challenges such as distortion in signals, light absorption and scattering, water turbidity changes or depth-depending colour distortion.

Intervention ROVs and AUVs are often equipped with a variety of sensing systems. When operating in unknown underwater environments, sonar systems are usually preferred as they are able to obtain bathymetric maps of large areas in a short time. Even though sonar is mostly used to provide general information about the environment or used in a first-stage approach to the area of interest, it has also been used to perform object detection by itself. Nonetheless, the preferred sensing modalities to obtain detailed, short-distance information with higher resolution are laser and video. These modalities are often used during the approach, object recognition and intervention phases. The use of the presented sensing systems towards object detection and, specifically, pipe and valve recognition is reviewed in Section 2.1.

To execute manipulation tasks, the sensing systems of a ROV or AUV must be able to provide enough information to perform accurate and robust scene understanding, including object detection, target recognition and pose estimation, under different experimental conditions.

This paper is a continuation of our previous work [11] where we proposed a deep learning-based approach to perform a pixel-wise segmentation of underwater pipes and valves from 3D RGB point cloud information.

In this paper, we make use of an improved evolution of the deep neural network used in our previous work to perform pixel-wise 3D segmentation. Additionally, we implement object detection over the segmented pixels, grouping them and being able to detect diverse pipe and valve instances in a point cloud. We develop an algorithm to extract information from the detected instances providing pipe vectors, gripping points, structural elements such as elbows or connections, and valve type (2-way or 3-way) and orientation. Furthermore, if the point clouds are spatially referenced, their information can be unified, forming an information map of an inspected area. Finally, the 3D segmentation, along with the information extraction and mapping algorithms, are executed online on an AUV, performing real-time underwater pipe and valve recognition, characterisation and mapping for inspection and manipulation tasks.

The remainder of this paper is structured as follows: Section 2 reviews the related work on underwater perception and pipe and valve identification, and highlights the main contributions of this work. Section 3 describes the methodology and materials adopted in this study. The experimental results are presented and discussed in Section 4. Section 5 details the network and information algorithms online implementation. Finally, Section 6 outlines the main conclusions and future work.

2. Related Work and Contributions

2.1. State of the Art

This section reviews the use of diverse sensing systems for underwater inspection, object detection and, specifically, pipe and valve recognition. The three most used sensing systems in underwater environments are sonar, laser and vision.

Sonar sensing is the preferred method when working in large, unknown environments, providing broad information in a quick manner [12,13]. It has also been used for object localisation in underwater scenarios [14,15]. Object detection deep learning techniques have been applied to underwater sonar imaging for diverse applications such as the detection of human bodies [16] or war mines [17]. Some of the drawbacks presented by sonar imaging are the noisy nature of the images, which generates texture information losses, and the fact that it is not able to capture colour information, which is useful in object recognition tasks.

Underwater laser scans can provide high-resolution 3D data that can be used for environment inspection and object recognition. Some studies on underwater pipeline detection include the works of Palomer et al. [18] where a laser scanner is integrated on an AUV for object detection and manipulation, or the works of Himri et al. [19,20] and Villacrosa et al. [21], where a recognition and pose estimation pipeline based on point cloud matching is built. As for downsides, laser systems tend to have a very high initial cost, are affected by light transmission problems and cannot provide colour information.

Vision is one of the most complete and used perception modalities in robotics and object recognition tasks thanks to its accessibility, ease of use and the fact that produces RGB high-quality information. It also has disadvantages as the obtained images are affected by light transmission problems, colouring distortions or environmental factors such as water turbidity. Nonetheless, some of these weaknesses can be alleviated by adapting the acquisition system to the environmental conditions, adjusting the operation range, calibrating the cameras or colour correcting the obtained images.

In the past, traditional computer vision approaches have been used to detect and track multiple submerged objects such as artifacts [22–25], cables [26–28] or pipelines [28–31]. Some works rely on texture and shape descriptors [28,31], others on template matching [32,33] or use colour segmentation to find and process regions of interest in the images [25,34].

Other works use a combination of multiple sources of information, Kallasi et al. in [35] and Razzini et al. in [7,36] present traditional computer vision methods that combine texture, shape and colour information to detect underwater pipelines and project them into point clouds obtained from stereo vision. In these works, the point cloud information is not used to assist the pipe recognition process.

Rekik et al. [37] developed the first trainable system to detect underwater pipelines, extracting several features and using a Support Vector Machine to classify between positive and negative underwater pipe image samples. Convolutional Neural Networks were introduced by Nunes et al. [38] to classify diverse underwater objects, including a pipeline. None of these works determined the position of the object within the image, only a binary classification of the object's presence was given.

Some studies introduced deep learning solutions applied to underwater computer vision, but are limited to the detection and pose estimation of 3D-printed objects [39] or living organisms such as fishes [40] or jellyfishes [41]. Few research studies involving pipelines are restricted to damage evaluation [42,43] or pipeline navigation from the inside [44]. Guerra et al. in [45] present one of the most advanced works on pipeline recognition using deep learning, where a drone equipped with a monocular camera is used to perform 2D detection of pipelines in industrial environments.

Therefore, with the exception of the later works of Himri et al. [20] and Villacrosa et al. [21], which will be later discussed in Section 4.1, the remaining works suffer from crucial drawbacks when tackling pipe and valve recognition for inspection and manipulation tasks. The most significant drawbacks from previous implementations, which are solved in our work, are listed below:

- Only recognising pipes, no valves, connections or elbows are detected.
- Not being able to detect multiple elements simultaneously, due to the nature of its data processing, only isolated objects can be detected.
- Not gathering information from the detected objects, such as pipe length, gripping points, orientation or valve type and position.
- Not being able, or no demonstration, of being able to be executed online on an inspecting or manipulating robot.

Finally, to the best knowledge of the authors, the only prior known research on underwater pipeline and valve 3D recognition using deep learning is our previous work presented in [11], upon which we build the research introduced in this paper.

2.2. Main Contributions

The main contributions of this paper are composed of:

1. Expansion of our novel point cloud dataset of underwater pipe and valve structures, adding point clouds obtained with a new pair of cameras mounted on an AUV. This dataset is used to train and test the selected deep neural network and information algorithms.
2. Development of novel algorithms to extract information from detected pipe and valve instances, providing data on pipe vectors, gripping points, structural elements such as elbows or connections, and valve type and orientation. The information from spatially referenced point clouds can be unified to create information maps of inspected areas.
3. Neural network and information algorithms validation by conducting underwater experiments where the point cloud segmentation, the information extraction algorithm and mapping algorithm are executed online in an AUV, performing real-time underwater pipe and valve recognition, characterisation and mapping for inspection and manipulation tasks.
4. The updated dataset (point clouds and corresponding ground truths) along with a trained model and the code of the algorithms used to perform the information extraction and mapping are provided to the scientific community in [46].

3. Methodology

This section presents an overview of the selected network; explains the acquisition, labelling and organisation of the data; details the information extraction and mapping algorithms; and exposes the validation process and evaluation metrics.

3.1. Deep Learning Network and Training Deatils

Even though most applications in the field work with 2D information, which some later project to the 3D space, we decided to use a 3D segmentation network using point clouds as input for diverse reasons. First of all, the introduction of depth data provides extra information to work with, allowing us to extract more features, helping the segmentation. Secondly, as we extract information from the segmented point clouds for inspection and manipulation tasks, 3D positioning would be a must. Thus, it would not make sense to use 2D segmentation to avoid possible matching failures in the 3D point cloud generation if the extracted information could not be projected into a 3D space.

In this work, we select the Dynamic Graph Convolutional Neural Network (DGCNN) [47] to perform the pipe and valve 3D segmentation. This network is an evolution of the PointNet deep neural network [48] that we used in our previous work, surpassing its performance on several benchmark datasets [47]. Like its predecessor, this network has a unified architecture that allows it to perform multiple tasks, ranging from object classification and part segmentation to scene semantic segmentation.

The novelty of the DGCNN architecture is the introduction of the named EdgeConv modules, which can be integrated into existing deep learning models such as PointNet. These modules capture local geometric structure information by generating edge features that describe relations between a point and its neighbours while being invariant to input permutations. Since proximity in the feature space differs from proximity in the input point cloud, the set of neighbours of a point changes from layer to layer. This results in a network graph that is updated after each network layer, leading to non-local diffusion of information over the whole point cloud. This allows the EdgeConv modules to capture global shape information.

Furthermore, to make the prediction invariant to the point cloud geometric transformations, all input sets are aligned to a canonical space before feature extraction. To achieve this, a 3×3 matrix is applied, obtained from a tensor concatenating the coordinates of each point and the coordinate difference between its neighbours.

The DGCNN architecture takes point clouds as input and outputs a class label for each point. While training the network, it is also fed with ground truth labels, indicating the real class of each point from the point cloud. The labelling process is further detailed in Section 3.2.2.

As the original DGCNN implementation, we use a softmax cross-entropy loss along with an Adam optimiser. The decay rate for batch normalisation starts with 0.5 and is gradually increased to 0.99. In addition, we apply a dropout with a keep ratio of 0.7 on the last fully connected layer, before class score prediction.

Other hyperparameters are selected based on the experiments conducted in our previous work [11]. For the training, we use stochastic gradient descent with a learning rate of 0.001, a batch size of 16; block and stride distances of 1 m; and, finally, a maximum number of allowed points per block of 128. These hyperparameters have proven to offer very good metrics in terms of point cloud segmentation while greatly reducing inference time, a key factor in the online execution of this network in an AUV to perform real-time inspection and manipulation tasks. Details on the network and algorithms online execution in an AUV are given in Section 5.

To improve the network performance, we implement an early stopping strategy based on the work of L. Prechelt in [49], ensuring that the network training process stops when the divergence between validation and training losses is minimum. This technique allows us to obtain more general and broad training, avoiding overfitting. The DGCNN architecture is presented in Figure 1.

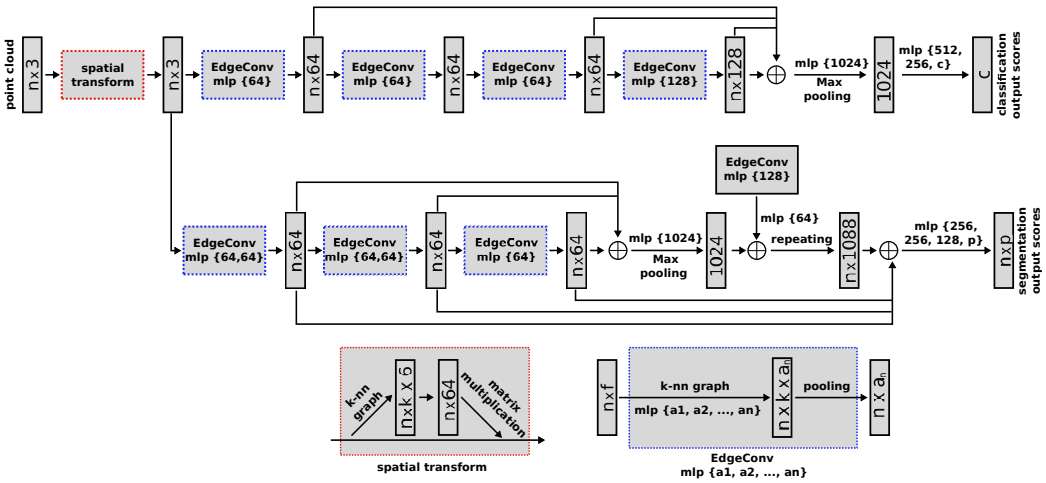


Figure 1. DGCNN architecture. Taken from [47], with permission from author Yue Wang, 2021.

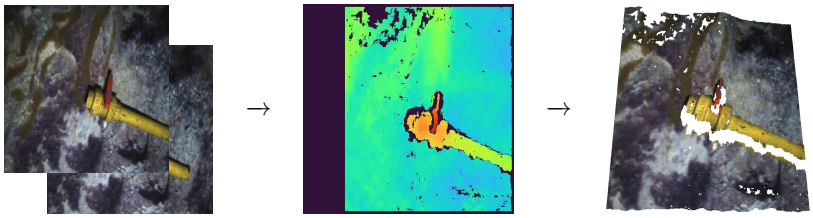
3.2. Data

This subsection explains the acquisition, labelling and management of the data used to train and test the deep neural network.

3.2.1. Acquisition

The used point clouds come from two different sources. First, we reuse our previous dataset from [11], consisting of 192 underwater point clouds containing diverse pipe and valve structures and connections. This dataset was gathered by performing diverse surveys on an artificial pool with an Autonomous Surface Vehicle equipped with a Bumblebee2 Firewire stereo rig and using the Robot Operating System (ROS) middleware [50].

The second source of point clouds is another stereo pair rig, composed of two Manta G283 cameras mounted on an AUV, gathering point clouds through ROS once again. We performed up to six immersions with the AUV to record different pipe structures and valve connections, two of them at an artificial pool, and the remaining four at different locations at the sea. The acquisition process is pictured in Figure 2.

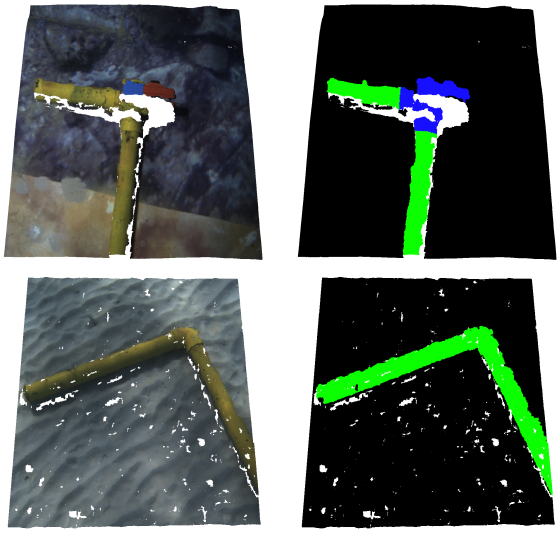


(**a**) (**b**) (**c**)

Figure 2. Data acquisition process: (**a**) Left and right stereo images from a calibrated stereo rig; (**b**) Disparity depth image obtained using ROS stereo processing [51]; (**c**) Point cloud generated by merging depth and colour information (Blank spaces correspond to areas where no matching between stereo images could be found or to covered areas).

3.2.2. Ground Truth Labelling

In order to train and test the network, ground truth label maps are manually built from the obtained point clouds. The points corresponding to each class are marked with a different label. The studied classes and their RGB labels are: *Pipe* (Green: 0, 255, 0), *Valve* (Blue: 0, 0, 255) and *Background* (Black: 0, 0, 0). Figure 3 shows a couple of point clouds along with their corresponding ground truth label maps.



(**a**) (**b**)

Figure 3. Point cloud labelling: (**a**) Original point cloud; (**b**) Ground truth annotations. Points corresponding to pipes, valves and background, are marked in green, blue and black, respectively.

3.2.3. Dataset Management

To configure our dataset, we gather the point clouds obtained from the two previously mentioned sources in Section 3.2.1. First, we take 192 point clouds from our previous dataset (from now on, referred to as set S_{ASV}). Second, we extract point clouds from the AUV immersions. From the two pool immersions we extract 104 and 51 point clouds (from now on, referred to as sets $S_{POOL\text{-}1/2}$, respectively). From the four sea immersions, we extract 45, 56, 36 and 30 point clouds (from now on referred as sets $S_{SEA\text{-}1/2/3/4}$, respectively).

In order to build the dataset used for the training, validation and test, we decided to gather the point clouds from the sets S_{ASV}, $S_{POOL\text{-}1}$ and $S_{SEA\text{-}1/2}$, conforming a total of 397 point clouds along their corresponding ground truth label maps.

This dataset contains point clouds gathered using two different pairs of stereo cameras, in fresh and salt water, under different environmental conditions and showcasing a wide variety of pipe structures and valve connections over different backgrounds, such as a plastic lone, sand or rocks. This represents a broad spectrum of scenarios to assure

robustness in the network training and reduce its overfitting. The dataset is split into a train-validation set (90% of the data, 357 point clouds) and a test set (10% of the data, 40 point clouds), which will be referred to as T_{BASE}. Additionally, sets $S_{POOL\text{-}2}$ and $S_{SEA\text{-}3}$ are used to perform a secondary test, from now on referred as T_{EXTRA}, containing a total of 87 point clouds. The point clouds from this test are from immersions whose data has not been used for the network training or validation, and contain different, unseen environmental conditions, pipe structures, valve connections and backgrounds. Hence, this test allows us to assess how well the network generalises its training to new data.

Finally, set $S_{SEA\text{-}4}$ contains point clouds gathered by the AUV navigating over a larger structure containing multiple pipes and valves. This set will be used to test the mapping algorithm presented in Section 3.3. This test set will be referred to as T_{MAP}.

Figure 4 illustrates the dataset management, while in Figure 5 some examples of point clouds are shown.

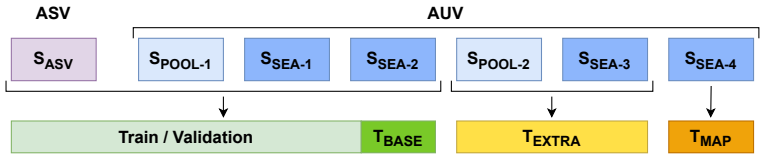


Figure 4. Dataset management.

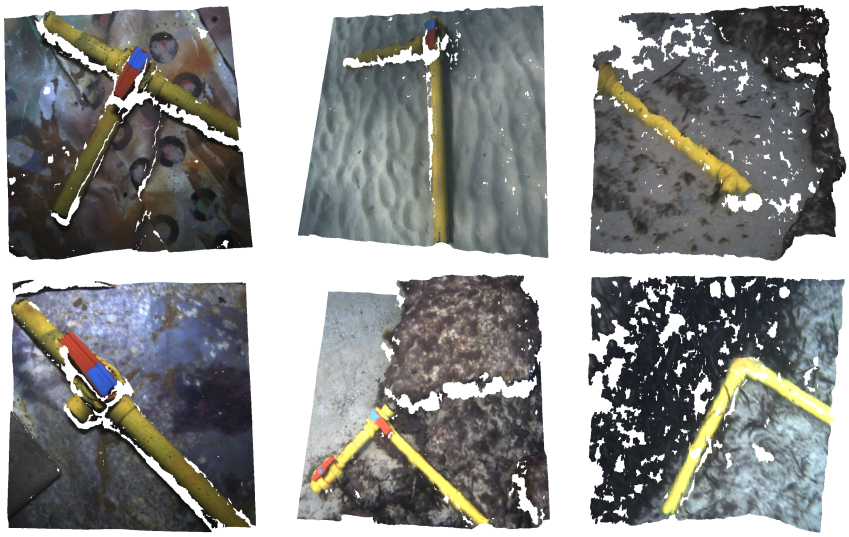


Figure 5. Point clouds from the dataset showcasing diverse pipe structures and valve connections over different backgrounds.

3.3. Segmentation Understanding Algorithms

Once the deep neural network has processed a point cloud and generated its semantic segmentation, we need to further process this segmentation and extract information to use in inspection and manipulation tasks. To do that, we develop two algorithms. The first algorithm, referred to as Information Extraction Algorithm (IEA), takes the network output and extracts information such as the number of pipes and valves present in the point cloud, their position, orientation or even pipe connections and valve type (2-way or 3-way). The second algorithm, referred to as Information Unification Algorithm (IUA), unifies the information extracted from multiple localised point clouds taken on a studied area and generates a global information map. Next, both IEA and IUA algorithms are detailed.

3.3.1. Information Extraction Algorithm

The starting point of this algorithm is the semantic segmentation outputted by the deep neural network. Its first step is to transform the pixel-wise segmentation into an instance-

based one using a Density-Based Spatial Clustering of Applications with Noise (DBSCAN), clustering pixels of the same class that are closer than a distance threshold, clusters that do not contain enough points to be considered as instances are deleted. This way, the different pipe and valve instances present in a point cloud are detected. Additionally, when a cluster belonging to a valve instance is found, it is set to "steal" the points belonging to pipe instances that are within a determined radius of the valve instance central point. This is due to the fact that, as the main body of the valve is very similar to the actual pipes, it sometimes gets misclassified as a pipe and only the handle of the valve is correctly classified, this way, the body of the valve is reclassified. Figure 6 shows the instance clustering and valve reclassification on a segmented point cloud.

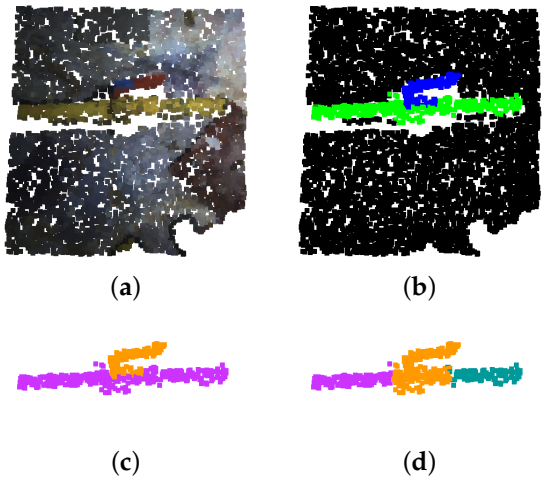


Figure 6. Point cloud clustering and valve reclassification. (**a**) Original point cloud. (**b**) Deep neural network segmentation. (**c**) Instance clustering. (**d**) Valve reclassification. Instances are represented following the purple, orange, teal and garnet colour sequence.

The following step of the algorithm is to extract information from the detected valve instances. First, the central point of the valve instances is calculated as the average XYZ coordinate values of all its belonging points, noting the valve position. Second, the algorithm performs a point cloud registration between each valve instance and five-point cloud models of 2-way and 3-way valves, obtaining the rotation matrix and model that provides a maximum registration score, inferring its pose and type. Using the valve registration data and its pre-known shape, a vector is generated to indicate each valve size and orientation.

Valve instances that do not reach a certain registration score threshold with any point cloud model are discarded. Consequently, the previously mentioned "stolen" points corresponding to discarded valve instances are returned to their corresponding original pipe instances.

Figure 7 showcases the described valve information extraction process.

The next step is to extract information from the detected pipe instances. First, the point cloud instances are voxelized and flattened into a 2D matrix, where closing and opening morphology operations are performed to consolidate the instance as a unique object.

At this point, the skeleton of the instance is computed, obtaining a chain of linked matrix coordinates and depicting the pipe shape. Moreover, coordinates with up to three neighbours are marked as connection points between different pipes. Once the smaller chains are discarded, the remaining chains and connection points are reprojected into the original instance 3D points.

Then, the algorithm calculates the 3D vector between each chain point and its linked points, providing information about the chain curvature. From there, pipe elbows can be established on point sequences with greater curvature than a selected threshold. Moreover, these vectors provide information on the chain length, allowing us to locate the position of a determined percentage of pipe length, this information is very useful to provide grabbing

points for pipe manipulation. Finally, a vector describing each straight portion of a chain is calculated between its first and last point, giving information on the corresponding pipe orientation and length.

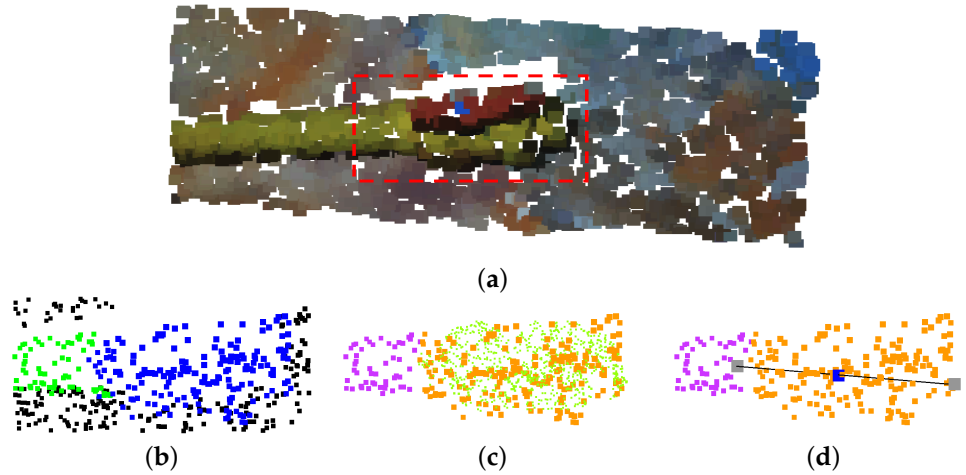


Figure 7. Valve information extraction process: (**a**) Original point cloud (area of interest highlighted in red square); (**b**) Deep neural network segmentation; (**c**) Instance clustering and best valve model registration (light green points); (**d**) Instance clustering along resulting valve central point (blue point) and vector (black line between the two gray points).

Figure 8 showcases the described pipe information extraction process.

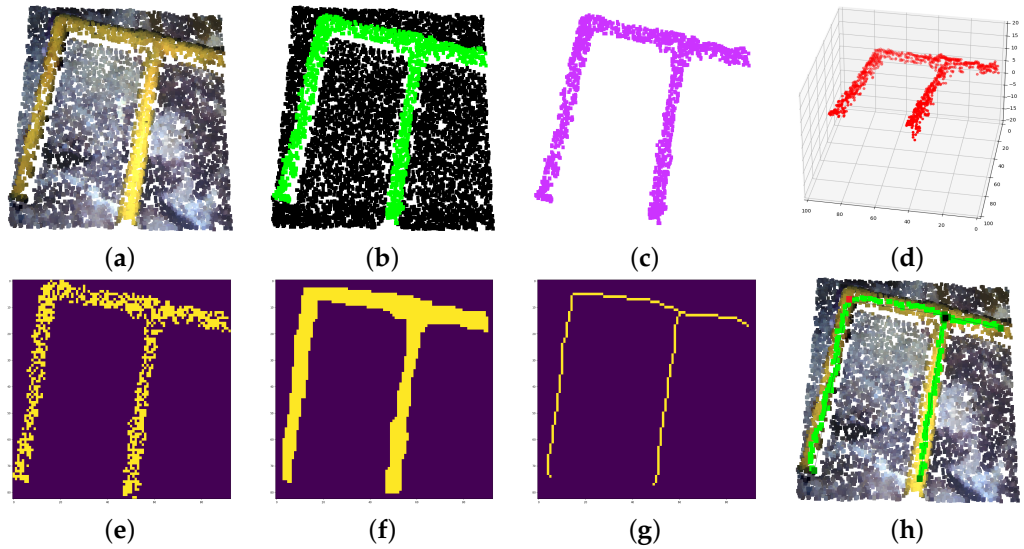
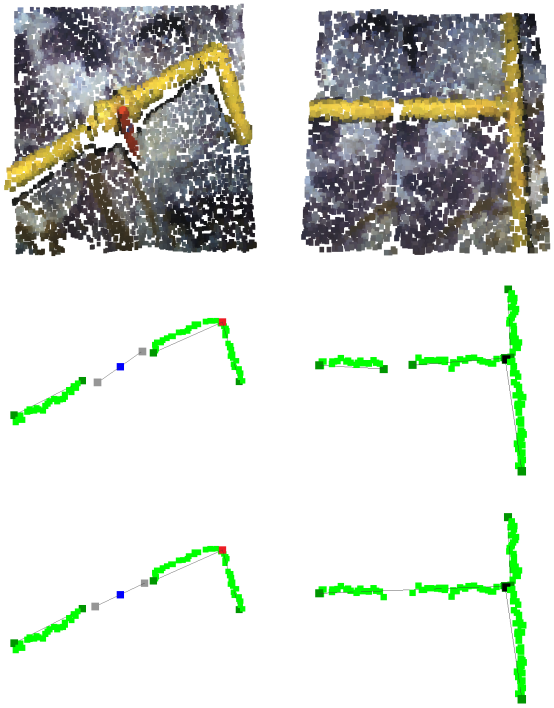


Figure 8. Pipe information extraction process. (**a**) Original point cloud. (**b**) Deep neural network segmentation. (**c**) Instance clustering. (**d**) 3D voxelization. (**e**) 2D matrix flattening. (**f**) Morphological operations: closing and opening. (**g**) Morphological operation: skeletonization. (**h**) Information reprojection overlapped onto original point cloud (light green: skeleton, dark green: pipe start-end, red: elbow, black: connection, lines: pipe vectors).

Finally, the valve and pipe information is refined. For the valves, its vector direction is recalculated taking into account the presence of pipes near the valve's central point. If only one pipe is near, the valve vector is aligned to the pipe vector, if two or three pipes are present and two of them have parallel vectors, the valve pipe is aligned with that vector. Additionally, if three pipes are found near its central point, the valve type is set automatically to 3-way. For the pipes, vectors belonging to pipes of different instances that are near and parallel, are unified.

Figure 9 shows examples of valve and pipe refinement.



(**a**) (**b**)

Figure 9. Information refinement: (**a**) Valve vector reorientation; (**b**) Pipe vector unification. Top: Original point cloud. Middle: information before refinement. Bottom: information after refinement.

3.3.2. Information Unification Algorithm

This algorithm is built on top of the information provided by the previously detailed IEA and its end is to generate unified information maps by merging information from different point clouds. It is strictly necessary that the point clouds are referenced to a localised frame, whether it is an absolute frame such as geolocalisation or a relative one such as odometry.

Different methods are used to merge the pipe and valve information from new upcoming point clouds to the one already present in the information map. For the pipes, the algorithm checks if upcoming pipe chains are near chains present in the information map. Near chains are merged and new vectors and elbows are computed as explained in the IEA description. Moreover, a validation count is assigned to each pipe chain, indicating the number of times it has appeared in different point cloud information extractions, merged chains add up their validation counts. This validation count is used further into the algorithm to decide whether or not detection is a spurious false positive or a certain true positive.

For the connection points, the algorithm checks if new upcoming connections are situated near prior registered connections. Near connections are merged, averaging their 3D position. A validation count is also assigned to each connection point.

For the valves, the procedure is the same as for the connection points, with the addition that the valve vector direction is also averaged between the prior valve vector and the upcoming one.

Finally, for each K processed point cloud, the algorithm runs a validation count check, where pipes, connection points and valves with lower validation count than a determined threshold are discarded.

Figure 10 presents the IUA output when implemented over a series of point clouds containing two pipes, a valve and an elbow. It can be seen how a false positive valve

appears on the original information extracted near the elbow, but it is later eliminated by the algorithm count check as it is not found in any other point cloud information.

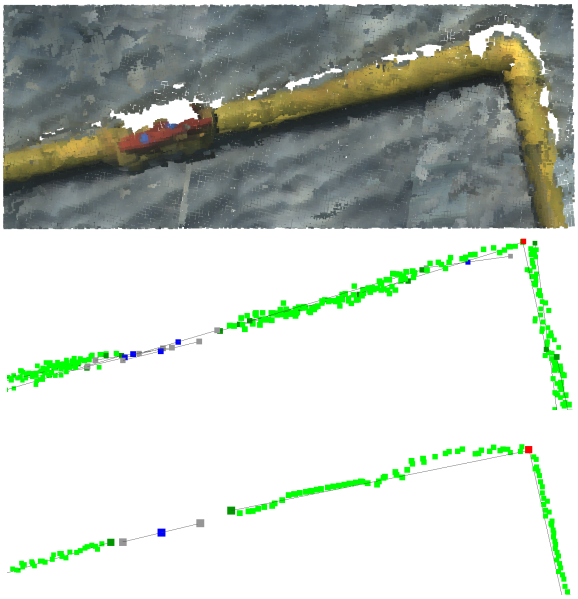


Figure 10. IUA implementation. **Top**: Overlapped point clouds. **Middle**: Overlapped extracted information from IEA. **Bottom**: Unified information from IUA.

Figure 11 presents a flowchart of both IEA and IUA algorithms, describing their workflow and interrelation. Additionally, the commented code of both algorithm implementations is provided in [46], which offers a deeper insight into the diverse algorithm steps and their numerous parameters that can be tweaked.

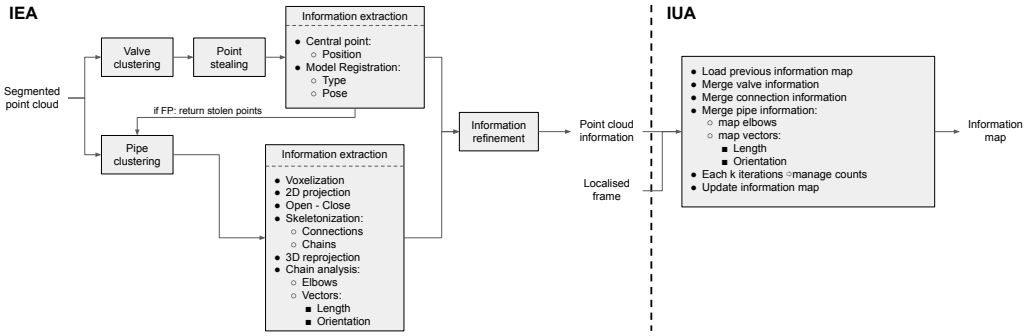


Figure 11. IEA and IUA algorithms flowchart, describing their workflow and interrelation.

3.4. Validation and Evaluation Metrics

DGCNN is a highly efficient and effective network, obtaining great metrics in both object classification and segmentation tasks in indoor and outdoor scenarios [47]. However, it has never been tested and validated in underwater scenarios.

In order to validate the DGCNN, we use the 10k-fold cross-validation method [52]. With it, the train-validation partition of our dataset (see Figure 4) is split into ten equally sized subsets. Next, the network is trained ten times, each one using a different subset as validation and the nine remaining as training, generating ten models, which are then tested against the T_{BASE} and T_{EXTRA} test sets. The final performance is computed as the average results for the ten models. This method reduces the variability of the results, making them less dependent on the selected training and validation data, and therefore obtaining a more accurate performance estimation.

To evaluate a model performance, we make a point-wise comparison between the network predictions and their corresponding ground truth annotations. For each class, the number of correctly segmented points, True Positives (*TP*); and the number of incorrectly segmented points, False Positives (*FP*) or False Negatives (*FN*) is computed. The number of *TP*, *FP* and *FN* are used to calculate the *Precision*, *Recall* and *F1-score* for each class, following Equations (1)–(3).

$$Precision = \frac{TP}{TP + FP} \ , \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \ , \tag{2}$$

$$F1\text{-}score = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision} \ . \tag{3}$$

The goal of this work is not to work on a pixel-level segmentation, but to group pixels into instances from which to extract information. Therefore, it makes sense to evaluate the network performance on an instance level as well. In order to achieve that, we apply the clustering method explained in Section 3.3.2 to the network segmentation output and ground truth annotations. From there, we make use of the *Intersection over Union* (IoU) metric, which provides the similarity between two instances. The IoU value between two instances is calculated following Equation (4).

$$\text{IoU} = \frac{inst1 \cap inst2}{inst1 \cup inst2} = \frac{P_{shared}}{P_{inst1} + P_{inst2} - P_{shared}} \ . \tag{4}$$

where $P_{inst1/2}$ denotes the number of points forming instance one or two and P_{shared} the number of points shared by both instances.

To determine whether a predicted instance is a *TP* or a *FP*, an IoU threshold value needs to be established. Following the criteria applied in the PASCAL VOC challenge [53], we set this threshold at $thr_{iou} = 0.5$. A predicted instance is classified as *TP* if the IoU value with any ground truth instance is greater than the thr_{iou} and the prediction class (C_{pred}) is the same as the ground truth instance class (C_{gt}). Otherwise, the predicted instance is classified as a *FP* (Equation (5)).

$$Inst. = \begin{cases} TP, & \text{if IoU} >= thr_{iou} \ \& \ C_{pred} == C_{gt} \ , \\ FP, & \text{otherwise} \ . \end{cases} \tag{5}$$

Ground truth instances that do not have an IoU > thr_{iou} with any predicted instance are counted as undetected instances, *FN*.

Once each prediction instance is classified as either *TP* or *FP*, and the number of *FN* is obtained, the instance-level *Precision*, *Recall* and *F1-score* metrics are computed following the previous Equations (1)–(3).

Finally, to evaluate the information provided by the IEA and IUA, information ground truths are built, manually, over the T_{BASE}, T_{EXTRA} and T_{MAP} test sets point cloud network segmentations, annotating the same information generated by the IEA for its comparison. The extracted information is compared to the ground truth annotations using different metrics for each type of information.

For the pipe information, the vectors are compared in terms of magnitude and direction, for the elbows and connections points, the distance difference from their ground truth annotation counterparts is measured.

For the valve information, the describing vector is compared only in terms of direction, since its magnitude is fixed by parameter, central valve point diversion is also measured. Lastly, the correct valve type classification is checked.

4. Experimental Results and Discussion

This section reports the results obtained by the DGCNN segmentation at pixel and instance levels. The IEA and IUA are also evaluated, checking the validity of the provided information.

4.1. DGCNN Segmentation Results

The results presented in this section are the average values obtained from the ten models generated when using the 10k-fold cross-validation method previously explained in Section 3.4.

Table 1 presents the pixel-level metrics when evaluating the DGCNN over the T_{BASE} test set. The presented metrics are the per-class *F1-score (F1)* and its mean value *(mF1)*. Additionally, since the background class is not converted into instances nor used by the IEA or IUA, it makes sense to only focus on the pipe and valve classes when conducting the per-pixel evaluation of the network. The *2mF1* metric measures the mean *F1-score* only taking into account those two classes.

Table 1. DGCNN pixel-level metrics on T_{BASE} test set.

Class	*F1 (%)*	*mF1 (%)*	*2mF1 (%)*
Pipe	92.4		
Valve	84.9	92.3	**88.7**
Background	99.7		

The DGCNN reaches a *F1-score* value of 92.4% for the pipe class, an 84.9% for the less represented and harder to identify valve class and a 99.7% for the prevailing background class, resulting in a mean *F1-score* of 92.3%. When only taking into account the pipe and valve classes, the network scores an *2mF1* of 88.7%.

Table 2 presents the instance-level metrics when evaluating the DGCNN over the T_{BASE} test set. The presented metrics are the per-class Intersection over Union (*IoU*) and its mean value (*mIoU*) along the per-class *F1-score (F1)* and its mean value *(mF1)*.

Table 2. DGCNN instance-level metrics on T_{BASE} test set.

Class	*IoU (%)*	*mIoU (%)*	*F1 (%)*	*mF1 (%)*
Pipe	83.4	80.4	96.7	**95.1**
Valve	77.4		93.5	

The achieved *mIoU* value is 80.4%, which indicates a high overlap between predicted and ground truth instances. This similarity is reflected in the *mF1* score, reaching a value of 95.1%. The reached instance-level *mF1* score is higher than the obtained pixel-level *2mF1* score, this indicates that the applied pixel clustering allows us to match predicted and ground truth instances even when there exist pixel differences between them.

Tables 3 and 4 present the pixel-level and instance-level metrics, respectively, when evaluating the DGCNN over the T_{EXTRA} test set.

Table 3. DGCNN pixel-level metrics on T_{EXTRA} test set.

Class	*F1 (%)*	*mF1 (%)*	*2mF1 (%)*
Pipe	91.4		
Valve	83.0	91.4	**87.2**
Background	99.7		

Table 4. DGCNN instance-level metrics on T_{EXTRA} test set.

Class	*IoU (%)*	*mIoU (%)*	*F1 (%)*	*mF1 (%)*
Pipe	82.7	79.7	96.3	**95.4**
Valve	76.8		94.6	

The results for the T_{EXTRA} test set at both pixel-level and instance-level evaluations are equally good as the ones obtained for the T_{BASE} test set, reaching a pixel-level *2mF1* of 87.2% and an instance-level *mF1* of 95.4%. This means that the DGCNN is able to generalise its training and avoid overfitting, being able to correctly segment more challenging point clouds with unseen pipe and valve connections and environment conditions like the ones present in the T_{EXTRA} test set.

The metrics presented in this section outperform other state-of-the-art methods for pipe recognition: [35]-traditional computer vision algorithms over 2D underwater images achieving an F1-score of 94.1%, [7]-traditional computer vision algorithms over 2D underwater images achieving a mean F1-score over three datasets of 88.0% and [45]-deep learning approach for 2D drone imagery achieving a pixel-wise accuracy of 73.1%.

For the valve recognition, the only works that consider this class are [20,21], which use Bayesian techniques to perform 3D object recognition and classification of different types of valves and structural elements such as elbows or 'T' junctions. Vallicrosa et al. [21] report an average F1-score of 78% in the object recognition and classification task, this metric is not directly comparable to our results since we do not distinguish between different types of valves and structural elements such as elbows or 'T' junctions are extracted from pipe information, not identified as elements per se.

An issue of Vallicrosa et al. [21] work is the fact that one of the first steps of the proposed algorithm is to perform a normal estimation and region growing grouping over the analysed point cloud and classify each region into two groups, flat areas or points belonging to the pipeline structure. This method for selecting the pipeline structure points would produce critical failures in real-world implementations, as any non-flat area would be analysed as part of the pipeline system.

4.2. IEA Results

Table 5 shows the evaluation metrics obtained by the IEA when applied over the network segmentation of the T_{BASE} and T_{EXTRA} test sets altogether.

Table 5. IEA metrics on T_{BASE} and T_{EXTRA} test sets.

Info.	ΔC_{Point} **(cm)**	$\Delta V_{Magnitude}$ **(cm)**	$\Delta V_{Direction}$ **(°)**
Pipe	-	1.94	4.20
Elbow	2.70	-	-
Conn.	1.69	-	-
Valve	1.63	-	13.31

The IEA was able to detect all pipes, elbows, connections and valves present in the network segmentation, without generating any false positives.

For the pipe information, the vector magnitude (pipe length) and direction (pipe orientation) are evaluated. The difference between IEA output and ground truth for the magnitude is 1.94 cm, and 4.2° for the direction. For the elbow and connection information, only the central point position is evaluated, for which the differences are 2.7 cm and 1.69 cm, respectively. With these metrics, it can be determined that the IEA is able to determine pipe length, orientation and its different structural elements with high accuracy.

For the valve information, the central point (valve position) and its vector direction (valve orientation) are evaluated, obtaining a divergence of 1.63 cm and 13.31°, respectively. Furthermore, the valve type is correctly identified 73.6% of the time. Even though the valve

position is detected with high accuracy, there exists a small error when determining its orientation and a higher one when classifying its type.

Qualitative results of the neural network segmentation and IEA output over diverse point clouds are shown in Figure 12.

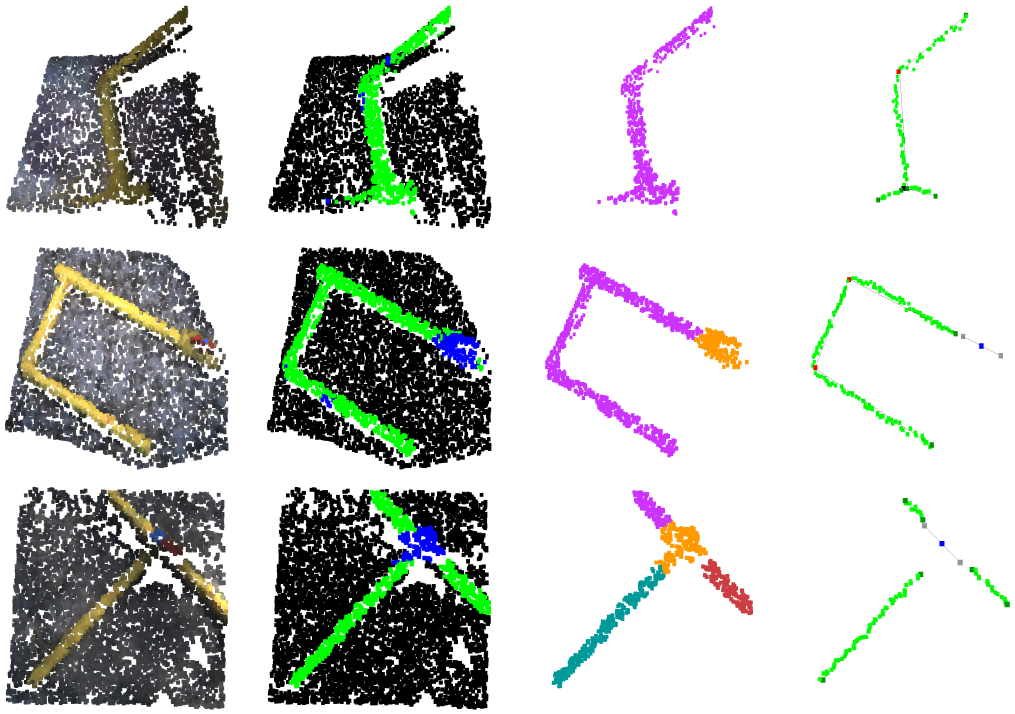


Figure 12. Neural network segmentation and IEA qualitative results. (**a**) Original point cloud. (**b**) Deep neural network segmentation. (**c**) Instance clustering. (**d**) Information extracted from IEA.

4.3. IUA Results

Table 6 shows the evaluation metrics obtained after applying the IUA on the information extracted by the IEA from the network segmentation of the T_{MAP} test set. For this execution, the validation count check was executed for five analysed point cloud information ($K = 5$) with a count threshold of 2.

Table 6. IUA metrics on T_{MAP} test set.

Info.	ΔC_{Point} (cm)	$\Delta V_{Magnitude}$ (cm)	$\Delta V_{Direction}$ (°)
Pipe	-	3.12	2.42
Elbow	3.40	-	-
Conn.	-	-	-
Valve	3.59	-	13.92

All pipes, elbows and valves were detected without generating any false positives. Additionally, the presented metrics are very similar to the ones obtained by the IEA execution on the T_{BASE} and T_{EXTRA} test sets, which implies that the IUA is able to merge the information from diverse point clouds while preserving its quality.

Figure 13 shows the T_{MAP} test set original point clouds along the IEA information extraction output and the final unified information by the IUA.

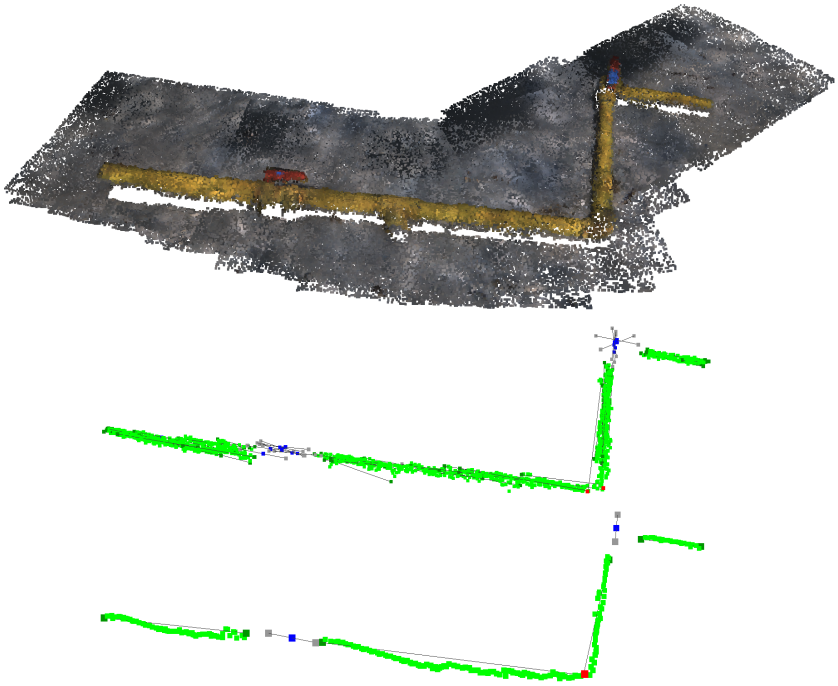


Figure 13. IUA implementation over T_{MAP} test set. **Top**: Overlapped point clouds. **Middle**: Overlapped extracted information from IEA. **Bottom**: Unified information from IUA.

5. AUV Online Implementation

An objective of this work is to implement the semantic segmentation network and information algorithms on an AUV and execute them online during manipulation and inspection tasks. This section describes the used AUV characteristics, the online implementation of the neural network and information algorithms, and its validation.

5.1. AUV Description

The used AUV is a SPARUS II model unit [54] (Figure 14) equipped with three motors, granting it three degrees of mobility (surge, heave and yaw). Its navigation payload is composed of: (1) a Doppler Velocity Logger (DVL) to obtain linear and angular speeds and altitude; (2) a pressure sensor which provides depth measurements; (3) an Inertial Measurement Unit (IMU) to measure accelerations and angular speeds; (4) a Compass for heading; (5) a GPS to be georeferenced during surface navigation; and (6) a Short Baseline acoustic Link (USBL) used for localisation and data exchange between the robot and a remote station. Additionally, it is equipped with a stereo pair of Manta G283 cameras facing downwards.

The robot has two computers. One is dedicated to receiving and managing the navigation sensor data and running the main robot architecture developed under ROS (Intel i7 processor at 2.2 GHz, Intel HD Graphics 3000 engine and 4 GB of RAM). The second computer is used to capture the images from the stereo cameras and execute the online semantic segmentation and information algorithms (Intel i7 processor at 2.5 GHz, Intel Iris Graphics 6100 and 16 GB of RAM).

The localisation of the vehicle is obtained through the fusion of multiple state estimations produced by the DVL, IMU, Compass, GPS, USBL, visual odometry and a navigation filter [55]. This localisation can be integrated into the point clouds generated from the images captured by the stereo pair of cameras to spatially reference them, which is a requirement to execute the IUA.

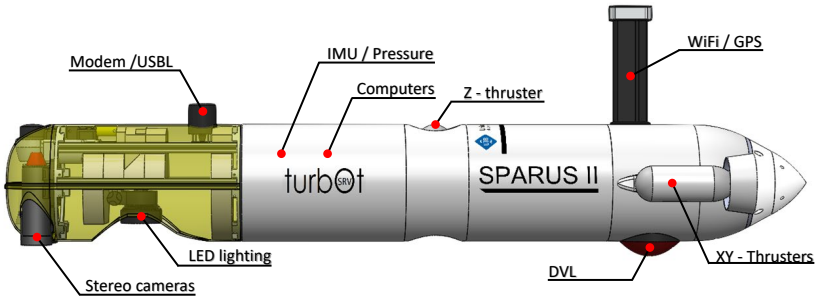


Figure 14. SPARUS II AUV.

5.2. Implementation

To perform the online implementation we design a pipeline based on ROS.

First, the images published by the stereo pair are transformed into point clouds to be processed by the neural network. To do so, diverse C++ ROS nodes are set up to: (1) rectify the raw images using the camera calibration parameters; (2) decimate the rectified images from their original size (1920 × 1440 pixels) to 960 × 720 pixels; (3) calculate the disparity map and generate the point clouds; and (4) downsample the point clouds using a voxel grid. Additionally, a python ROS node is set up to subscribe to the downsampled point clouds.

Following this, the point cloud is fed into a previously loaded inference graph of a DGCNN trained model, performing the semantic segmentation. From there, the IEA and IUA are executed. Finally, a publishing python ROS node is set up to publish the extracted information back into ROS to be accessed by other robots, sensors or actuators.

This pipeline achieves the implementation of the semantic segmentation network and information algorithms on an AUV and allows its execution online during manipulation and inspection tasks.

5.3. Validation

To validate the online execution, the frame rate of the output information stream is evaluated. An online execution was performed during the immersions conforming the $S_{POOL\text{-}2}$ and $S_{SEA\text{-}3}$ sets. In total, the online workflow was tested for 15′23″. For each execution the achieved output information stream frame rate and the time that each online execution step described in Section 5.2 takes are calculated.

For each immersion, the inspected pipe and valve configuration are different, making the IEA and IUA algorithms execution time vary, as the number and shape of pipes and valves are different, making the time analysis more robust as it covers a wider variety of scenarios.

The average output information stream frame rate and times for each online execution step are calculated as the mean value from both executions. Figure 15 presents a breakdown of the total average online execution time into its different steps.

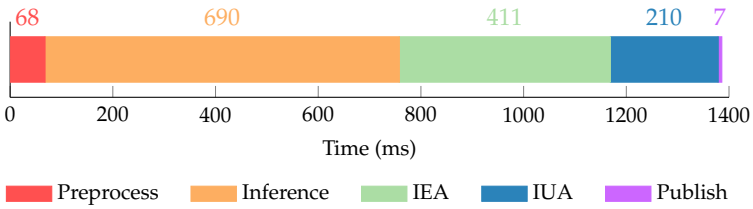


Figure 15. Online execution time breakdown.

The total average online execution time is 1.39 s, which results in an output information stream frame rate of 0.72 fps. The preprocessing step takes a mean of 68 ms (4.9% of the total time) and includes all operations to transform the images published from the stereo pair into point clouds to be processed by the neural network. The network inference takes the biggest amount of time with a mean of 690 ms (49.8% of the total time). Following,

the IEA and IUA take a mean of 411 ms and 210 ms, accounting for 29.7% and 15.1% of the total time, respectively. Finally, the information publication takes a mean of 7 ms (0.5% of the total time).

The achieved output information stream frame rate is more than enough to perform manipulation tasks, as these kinds of operations in underwater scenarios tend to have slow and controlled dynamics. Additionally, for most manipulation tasks the IUA may not be executed, lowering the overall online execution time to 1.18 s, and thus increasing the achieved frame rate to 0.85 fps.

Regarding inspection tasks, a method to validate the achieved output information stream frame rate is to check if exists an overlap between the analysed point clouds, ensuring full coverage of the inspected area. To do so, the overlap between the original images from analysed point clouds is checked. This overlap depends on the camera displacement between the images from two consecutive analysed point clouds (d_{KF}) and on the height of the image footprint (h_{FP}). Then, the *overlap* can be expressed as:

$$overlap = (h_{FP} - d_{KF}) \cdot h_{FP}^{-1} \, , \tag{6}$$

$$d_{KF} = v \cdot frame\_rate^{-1} \, , \tag{7}$$

$$h_{FP} = (a \cdot h_{image}) \cdot f^{-1} \, . \tag{8}$$

where v denotes the AUV velocity, a the navigation altitude, h_{image} the image height pixels and f the camera focal length.

During inspection tasks, an AUV such as the SPARUS II can achieve velocities up to $v = 0.4$ m/s and navigate at a minimum altitude of $a = 1.5$ m. Using these parameters along the Manta G283 camera intrinsic focal length of $f = 1505.5p$ and image height resolution of $h_{image} = 1440p$, the obtained overlap is 61.4%. Thus, the output information stream frame rate is high enough to get point clouds to overlap even when the AUV navigates at its maximum speed and minimum altitude.

6. Conclusions and Future Work

This paper presented the implementation of the DGCNN deep neural network to perform pixel-wise 3D segmentation of underwater pipes and valves from point clouds. To train the network, multiple immersions were conducted with an AUV to gather point clouds containing diverse pipe and valve configurations.

Two information algorithms were developed, the first one groups the segmented pixels into instances and implements object detection, detecting pipe and valve instances in a point cloud. Following this, it extracts information from the detected instances providing pipe vectors, gripping points, structural elements such as elbows or connections, and valve type and orientation. The second algorithm unifies information from spatially referenced point clouds, forming an information map of an inspected area.

Lastly, an ROS pipeline was built to execute the 3D segmentation and information extraction and unification algorithms online on an AUV, performing real-time underwater pipe and valve recognition, characterisation and mapping for inspection and manipulation tasks.

The neural network evaluation presented good results, reaching a mean *F1-score* value of 88.0% between the two conducted tests at a pixel-wise level and of 95.3% at an instance-level. Validating the use of the DGCNN deep neural network on underwater scenarios.

The information extraction algorithm results showcased excellent metrics when extracting information from pipe instances and their structural elements (elbows and connections), and good metrics when extracting valves position, orientation and type. The mapping algorithm was able to merge information from diverse point clouds, preserving its quality and deleting spurious false positive detections. This information can be used as first-stage inspection tools to provide layout maps of pipelines, or even directly detect pipe ruptures or deviations.

Finally, the online execution validation demonstrated that the output information stream frame rate is high enough to perform manipulation tasks and to get point clouds to overlap, permitting an adequate implementation of the information unification algorithm and ensuring full coverage of an inspected area. Additionally, the online implementation enables the use of real-time information to perform pipeline tracking and following during inspection tasks.

It is important to point out that the whole workflow presented in this work is executed using a simple point cloud as an input, no matter what its source is (i.e., stereo vision, sonar, laser). Thus, it can be implemented and utilised in multiple scenarios covering a wide range of applications.

Further developments will focus on studying the implementation of new deep neural networks to improve their segmentation performance and reduce the inference time, as this is the most time-consuming step of the online execution. Additionally, new ways of extracting pipe and valve information will be studied to improve pose and type detection using Bayesian techniques as presented by Vallicrosa et al. in [21], maybe even being able to detect the valve handle position, provide the valve state and allow the generation of pipeline flow diagrams.

Author Contributions: Conceptualization, G.O.-C. and Y.G.-C.; methodology, G.O.-C., Y.G.-C. and M.M.-A.; software, M.M.-A.; validation, M.M.-A.; formal analysis, M.M.-A.; investigation, M.M.-A.; resources, G.O.-C. and Y.G.-C.; data curation, M.M.-A.; writing—original draft preparation, M.M.-A.; writing—review and editing, G.O.-C., Y.G.-C. and M.M.-A.; visualization, M.M.-A.; supervision, G.O.-C. and Y.G.-C.; project administration, G.O.-C. and Y.G.-C.; funding acquisition, G.O.-C. and Y.G.-C. All authors have read and agreed to the published version of the manuscript.

Funding: Miguel Martin-Abadal was supported by Ministerio de Ciencia e Innovación, which is part of Agencia Estatal de Investigación (AEI), through the grant PRE2018-084117 (Cofunded by European Social Fund. ESF investing in your future). Gabriel Oliver-Codina was supported by Grant PID2020-115332RB-C33 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe". Yolanda Gonzalez-Cid was supported by Ministry of Economy and Competitiveness (AEI,FEDER,UE), under contract TIN2017-85572-P and by the Comunitat Autonoma de les Illes Balears through the Direcció General de Política Universitaria i Recerca with funds from the Tourist Stay Tax Law (PRD2018/34).

Data Availability Statement: The data presented in this study are openly available in Kaggle with DOI:10.34740/kaggle/dsv/2759939.

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

References

1. Yu, M.; Ariamuthu Venkidasalapathy, J.; Shen, Y.; Quddus, N.; Mannan, M.S. Bow-tie Analysis of Underwater Robots in Offshore Oil and Gas Operations. In Proceedings of the Offshore Technology Conference, Houston, TX, USA, 1–4 May 2017. [CrossRef]
2. Costa, M.; Pinto, J.; Ribeiro, M.; Lima, K.; Monteiro, A.; Kowalczyk, P.; Sousa, J. Underwater Archaeology with Light AUVs. In Proceedings of the OCEANS 2019-Marseille, Marseille, France, 17–20 June 2019; pp. 1–6. [CrossRef]
3. Asakawa, K.; Kojima, J.; Kato, Y.; Matsumoto, S.; Kato, N. Autonomous underwater vehicle AQUA EXPLORER 2 for inspection of underwater cables. In Proceedings of the 2000 International Symposium on Underwater Technology (Cat. No.00EX418), Tokyo, Japan, 26 May 2000; pp. 242–247. [CrossRef]
4. Jacobi, M.; Karimanzira, D. Underwater pipeline and cable inspection using autonomous underwater vehicles. In Proceedings of the 2013 MTS/IEEE OCEANS-Bergen, Bergen, Norway, 10–14 June 2013; pp. 1–6. [CrossRef]
5. Capocci, R.; Dooly, G.; Omerdić, E.; Coleman, J.; Newe, T.; Toal, D. Inspection-Class Remotely Operated Vehicles—A Review. *J. Mar. Sci. Eng.* **2017**, *5*, 13. [CrossRef]
6. Ridao, P.; Carreras, M.; Ribas, D.; Sanz, P.J.; Oliver, G. Intervention AUVs: The Next Challenge. *Annu. Rev. Control* **2015**, *40*, 227–241. [CrossRef]
7. Lodi Rizzini, D.; Kallasi, F.; Aleotti, J.; Oleari, F.; Caselli, S. Integration of a stereo vision system into an autonomous underwater vehicle for pipe manipulation tasks. *Comput. Electr. Eng.* **2017**, *58*, 560–571. [CrossRef]
8. Heshmati-Alamdari, S.; Nikou, A.; Dimarogonas, D.V. Robust Trajectory Tracking Control for Underactuated Autonomous Underwater Vehicles in Uncertain Environments. *IEEE Trans. Autom. Sci. Eng.* **2021**, *18*, 1288–1301. [CrossRef]

9. Nikou, A.; Verginis, C.K.; Dimarogonas, D.V. A Tube-based MPC Scheme for Interaction Control of Underwater Vehicle Manipulator Systems. In Proceedings of the 2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV), Porto, Portugal, 6–9 November 2018; pp. 1–6. [CrossRef]
10. Heshmati-Alamdari, S.; Bechlioulis, C.P.; Karras, G.C.; Nikou, A.; Dimarogonas, D.V.; Kyriakopoulos, K.J. A robust interaction control approach for underwater vehicle manipulator systems. *Annu. Rev. Control* **2018**, *46*, 315–325. [CrossRef]
11. 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.* **2021**, *9*, 5. [CrossRef]
12. Jonsson, P.; Sillitoe, I.; Dushaw, B.; Nystuen, J.; Heltne, J. Observing using sound and light – a short review of underwater acoustic and video-based methods. *Ocean Sci. Discuss.* **2009**, *6*, 819–870. [CrossRef]
13. Burguera, A.; Bonin-Font, F. On-Line Multi-Class Segmentation of Side-Scan Sonar Imagery Using an Autonomous Underwater Vehicle. *J. Mar. Sci. Eng.* **2020**, *8*, 557. [CrossRef]
14. Kim, B.; Yu, S. Imaging sonar based real-time underwater object detection utilizing AdaBoost method. In Proceedings of the 2017 IEEE Underwater Technology (UT), Busan, Korea, 21–24 February 2017; Volume 845, pp. 1–5. [CrossRef]
15. Wang, X.; Liu, S.; Liu, Z. Underwater sonar image detection: A combination of nonlocal spatial information and quantum-inspired shuoed frog leaping algorithm. *PLoS ONE* **2017**, *12*, e0177666. [CrossRef]
16. Lee, S.; Park, B.; Kim, A. A Deep Learning based Submerged Body Classification Using Underwater Imaging Sonar. In Proceedings of the 2019 16th International Conference on Ubiquitous Robots (UR), Jeju, Korea, 24–27 June 2019; pp. 106–112. [CrossRef]
17. Denos, K.; Ravaut, M.; Fagette, A.; Lim, H. Deep learning applied to underwater mine warfare. In Proceedings of the OCEANS 2017-Aberdeen, Aberdeen, UK, 19–22 June 2017; pp. 1–7. [CrossRef]
18. Palomer, A.; Ridao, P.; Youakim, D.; Ribas, D.; Forest, J.; Petillot, Y. 3D laser scanner for underwater manipulation. *Sensors* **2018**, *18*, 1086. [CrossRef]
19. Himri, K.; Pi, R.; Ridao, P.; Gracias, N.; Palomer, A.; Palomeras, N. Object Recognition and Pose Estimation using Laser scans For Advanced Underwater Manipulation. In Proceedings of the 2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV), Porto, Portugal, 6–9 November 2018; pp. 1–6. [CrossRef]
20. Himri, K.; Ridao, P.; Gracias, N. Underwater Object Recognition Using Point-Features, Bayesian Estimation and Semantic Information. *Sensors* **2021**, *21*, 1807. [CrossRef] [PubMed]
21. Vallicrosa, G.; Himri, K.; Ridao, P.; Gracias, N. Semantic Mapping for Autonomous Subsea Intervention. *Sensors* **2021**, *21*, 6740. [CrossRef] [PubMed]
22. Olmos, A.; Trucco, E. Detecting man-made objects in unconstrained subsea videos. In Proceedings of the British Machine Vision Conference, Cardiff, UK, 2–5 September 2002; pp. 50.1–50.10. [CrossRef]
23. Chen, Z.; Wang, H.; Xu, L.; Shen, J. Visual-adaptation-mechanism based underwater object extraction. *Opt. Laser Technol.* **2014**, *56*, 119–130. [CrossRef]
24. Ahmed, S.; Khan, M.F.R.; Labib, M.F.A.; Chowdhury, A.E. An Observation of Vision Based Underwater Object Detection and Tracking. In Proceedings of the 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE), Jaipur, India, 7–8 February 2020; pp. 117–122. [CrossRef]
25. Prats, M.; García, J.C.; Wirth, S.; Ribas, D.; Sanz, P.J.; Ridao, P.; Gracias, N.; Oliver, G. Multipurpose autonomous underwater intervention: A systems integration perspective. In Proceedings of the 2012 20th Mediterranean Conference on Control Automation (MED), Barcelona, Spain, 3–6 July 2012; pp. 1379–1384. [CrossRef]
26. Ortiz, A.; Simó, M.; Oliver, G. A vision system for an underwater cable tracker. *Mach. Vis. Appl.* **2002**, *13*, 129–140. [CrossRef]
27. Fatan, M.; Daliri, M.R.; Mohammad Shahri, A. Underwater cable detection in the images using edge classification based on texture information. *Meas. J. Int. Meas. Confed.* **2016**, *91*, 309–317. [CrossRef]
28. Narimani, M.; Nazem, S.; Loueipour, M. Robotics vision-based system for an underwater pipeline and cable tracker. In Proceedings of the OCEANS 2009-EUROPE, Bremen, Germany, 11–14 May 2009; pp. 1–6. [CrossRef]
29. Tascini, G.; Zingaretti, P.; Conte, G. Real-time inspection by submarine images. *J. Electron. Imaging* **1996**, *5*, 432– 442. [CrossRef]
30. Zingaretti, P.; Zanoli, S.M. Robust real-time detection of an underwater pipeline. *Eng. Appl. Artif. Intell.* **1998**, *11*, 257–268. [CrossRef]
31. Foresti, G.L.; Gentili, S. A hierarchical classification system for object recognition in underwater environments. *IEEE J. Ocean. Eng.* **2002**, *27*, 66–78. [CrossRef]
32. Kim, D.; Lee, D.; Myung, H.; Choi, H. Object detection and tracking for autonomous underwater robots using weighted template matching. In Proceedings of the 2012 Oceans-Yeosu, Yeosu, Korea, 21–24 May 2012; pp. 1–5. [CrossRef]
33. Lee, D.; Kim, G.; Kim, D.; Myung, H.; Choi, H.T. Vision-based object detection and tracking for autonomous navigation of underwater robots. *Ocean Eng.* **2012**, *48*, 59–68. [CrossRef]
34. Bazeille, S.; Quidu, I.; Jaulin, L. Color-based underwater object recognition using water light attenuation. *Intell. Serv. Robot.* **2012**, *5*, 109–118. [CrossRef]
35. Kallasi, F.; Oleari, F.; Bottioni, M.; Lodi Rizzini, D.; Caselli, S. Object Detection and Pose Estimation Algorithms for Underwater Manipulation. In Proceedings of the 2014 Conference on Advances in Marine Robotics Applications, Palermo, Italy, 16–19 June 2014.
36. Lodi Rizzini, D.; Kallasi, F.; Oleari, F.; Caselli, S. Investigation of Vision-based Underwater Object Detection with Multiple Datasets. *Int. J. Adv. Robot. Syst.* **2015**, *12*, 77. [CrossRef]

37. Rekik, F.; Ayedi, W.; Jallouli, M. A Trainable System for Underwater Pipe Detection. *Pattern Recognit. Image Anal.* **2018**, *28*, 525–536. [CrossRef]
38. Nunes, A.; Gaspar, A.R.; Matos, A. Critical object recognition in underwater environment. In Proceedings of the OCEANS 2019-Marseille, Marseille, France, 17–20 June 2019; pp. 1–6. [CrossRef]
39. Jeon, M.; Lee, Y.; Shin, Y.S.; Jang, H.; Kim, A. Underwater Object Detection and Pose Estimation using Deep Learning. *IFAC-PapersOnLine* **2019**, *52*, 78–81. [CrossRef]
40. Jalal, A.; Salman, A.; Mian, A.; Shortis, M.; Shafait, F. Fish detection and species classification in underwater environments using deep learning with temporal information. *Ecol. Inform.* **2020**, *57*, 101088. [CrossRef]
41. Martin-Abadal, M.; Ruiz-Frau, A.; Hinz, H.; Gonzalez-Cid, Y. Jellytoring: Real-time jellyfish monitoring based on deep learning object detection. *Sensors* **2020**, *20*, 1708. [CrossRef] [PubMed]
42. Kumar, S.S.; Abraham, D.M.; Jahanshahi, M.R.; Iseley, T.; Starr, J. Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks. *Autom. Constr.* **2018**, *91*, 273–283. [CrossRef]
43. Cheng, J.C.; Wang, M. Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques. *Autom. Constr.* **2018**, *95*, 155–171. [CrossRef]
44. Rayhana, R.; Jiao, Y.; Liu, Z.; Wu, A.; Kong, X. Water pipe valve detection by using deep neural networks. In *Proceedings of the Smart Structures and NDE for Industry 4.0, Smart Cities, and Energy Systems*; SPIE: Bellingham, WA, USA, 2020, ; Volume 11382, pp. 20–27. [CrossRef]
45. Guerra, E.; Palacin, J.; Wang, Z.; Grau, A. Deep Learning-Based Detection of Pipes in Industrial Environments. In *Industrial Robotics-New Paradigms*; IntechOpen: London, UK, 2020. [CrossRef]
46. Martin-Abadal, M.; Oliver-Codina, G.; Gonzalez-Cid, Y. Project Webpage for "Real-Time Pipe and Valve Characterisation and Mapping for Autonomous Underwater Intervention Tasks". 2021. Available online: http://srv.uib.es/3d-pipes-2/ (accessed on 24 October 2021).
47. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.* **2019**, *38*, 146. [CrossRef]
48. Charles, R.Q.; Su, H.; Kaichun, M.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 77–85. [CrossRef]
49. Prechelt, L. Early Stopping—But When? In *Neural Networks: Tricks of the Trade*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 53–67. [CrossRef]
50. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Montreal, Canada, 6–9 July 2009; Volume 3.
51. ROS-Stereo Image Proc. Available online: http://wiki.ros.org/stereo_image_proc (accessed on 18 May 2022).
52. Geisser, S. The predictive sample reuse method with applications. *J. Am. Stat. Assoc.* **1975**, *70*, 320–328. [CrossRef]
53. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
54. Carreras, M.; Hernández, J.D.; Vidal, E.; Palomeras, N.; Ribas, D.; Ridao, P. Sparus II AUV - A hovering vehicle for seabed inspection. *IEEE J. Ocean. Eng.* **2018**, *43*, 344–355. [CrossRef]
55. Font, E.G.; Bonin-Font, F.; Negre, P.L.; Massot, M.; Oliver, G. USBL Integration and Assessment in a Multisensor Navigation Approach for field AUVs. *Int. Fed. Autom. Control. (IFAC)* **2017**, *50*, 7905–7910. [CrossRef]