Bollard Segmentation and Position Estimation from Lidar Point Cloud for autonomous mooring

Journal:	*Transactions on Geoscience and Remote Sensing*
Manuscript ID	TGRS-2021-00531.R1
Manuscript Type:	Regular paper
Date Submitted by the Author:	n/a
Complete List of Authors:	Jindal, Mehak; University of Agder, Information and Communication Technology JHA, AJIT; University of Agder, Engineering Sciences; Cenkeramaddi, Linga; University of Agder, Information and Communication Technology
Keywords:	Lidar Data, Sensing Platforms, Oceans and Water

Bollard Segmentation and Position Estimation from Lidar Point Cloud for autonomous mooring

Mehak Jindal, Ajit Jha, and Linga Reddy Cenkeramaddi, *Senior Member, IEEE*

***Abstract*—This paper presents a computer-aided object detection and localization method from lidar 3D point cloud data. This topic of interest is in the framework of autonomous mooring, where the ship is tied to the rigid structure onshore (bollard) for autonomous maritime navigation. Using shape and features priors, unlike matching the whole object template to the experimental 3D point cloud representation of scene, two customized algorithms - (a) 3D feature matching (3DFM) and (b) Mixed feature-correspondence matching (MFCM) are presented. The proposed algorithms discriminate and extract the 3D points corresponding to the non-cooperative bollard's surface from background thus capable of classification, localization, and representing it using a unique co-ordinate in the 3D world. The proposed algorithms are tested and validated by implementing upon an experimental data set of 105 scenes where the bollard is at different positions and orientations with respect to lidar mounted on robotic arm. Statistical and probabilistic based approaches are taken into account to determine the performance of proposed algorithms. Model parameters estimation imply that errors resulting from 3DFM algorithm follow homoscedastic bimodal Gaussian distribution with individual Gaussian components having mean 0.03 m, and 0.09 m and both have equal standard deviation of 0.01 m. Further, posteriori component assignment probability is used to identify and cluster the scenes that contribute to relatively larger errors. Finally, an improved algorithm, MFCM is proposed whose errors follow unimodal Gaussian distribution with mean and standard deviation of 0.03 m and 0.01 m respectively thus mitigating the shortcomings of the former.**

***Index Terms*—Lidar 3D Point Cloud, Object Detection, Parameter Estimation, Autonomous Mooring.**

I. Introduction

AUTONOMUS systems such as (partial) self-driving cars, robots capable of routing in the industrial environment, drones and unmanned vehicles capable of continuous navigation require real time object detection and its position estimation. However, in the aforementioned use cases additional challenges may arise. First, the target may be uncooperative, meaning that there may not be easily recognizable markers or signs (such as reflector) on its surface. Second, it may be physically damaged and suffer optical degradation due to long exposure to the space environment. Hence, algorithms to detect the non-cooperative target is challenging and its position needs to be investigated.

This paper lies in the framework of autonomous maritime navigation, in particular, focusing on autonomous mooring where the ships can govern itself to navigate in the marine environment and moore itself onshore without human intervention. It addresses the problem of detecting the moore and localizing its position in 3D such that a robotic arm can place the mooring rope around the moore for autonomous and safe mooring (Fig. 1). This requires dealing with three different tasks - (a) detecting the moore (b) determining its unique position parameter such as (x,y,z) in 3D coordinate system and (c) tracking. While the former two are used to detect the moore and find its position in 3D when no prior knowledge of the scene is available, the latter is used to update the position parameters on the basis of previous measurements as soon as the new measurement is available or in case the relative position of the moore and the robotic arm is changed due to unseen conditions such as tide and wave.

Normally, ships or vessels use GPS to locate the location of bollard and mooring is done manually with human involvement. The use of GPS for bollard localization have serious implications. First, they have to rely on third part. Second, the measurement or localization from GPS are not accurate compared to local measurement using active sensors on-board in Ship or vessel. Third, the update rate of the GPS is lower as compared to lidar (proposed here). Fourth, the received signal from GPS might under go several reflections before reaching the receiver on board, causing multipath errors, delay. Further, the GPS receiver on the ship (for e.g.) uses multiple set of satellites in order to find the position. Changes in satellite configuration, orientation or position will affect the measurement. Based on the faulty position, if we guide the robot to that location with uncertainties, there is every chance that the manipulator or end-effector on the robot and do not successfully moore or even collide with bollard or nearby structure (in worst case) [1].

To overcome these drawbacks, it is desirable to have a local sensors that is capable of determining the bollard's position effectively. For that we propose lidar based sensor that lies on-board connected to the robot for mooring operations. The on-board sensor system for localization and segmentation had added advantages - (a) Independence: The driver of the ship or vessel do not have to rely on the third party (b) Redundancy: Since the proposed methodology is based on only lidar, redundant sensors can be kept as backup in case one fails (c) Local effect: Unlike the GPS based prior, where the signal can be delayed or poor estimation (because of factors mentioned above), the lidar based system is only affected by local environment unlike the thousands of kilometer between GPS satellite and GPS receiver (d) better accuracy in measurement - since the lidar is close to bollard, the disturbances

M. Jindal and L.R. Cenkeramaddi are with the Department of Information and Communication Technology, University of Agder, Grimstad, Norway (e-mail: mehakjindal1998@gmail.com, linga.cenkeramaddi@uia.no)

A. Jha is with the Department of Engineering Sciences, University of Agder, Grimstad, Norway (e-mail: ajit.jha@uia.no).

in measurement is local unlike the thousands of kilometer between GPS satellite and GPS receiver.

These problems are dealt by designing innovative algorithms capable of processing 3D point cloud data from active lidar. To overcome this limitation, lidars are used to sense the environment exploiting the fact they work relatively well independently of the illumination conditions, and the ability to easily discriminate the target from background. This research focuses on the use of lidar for autonomous mooring. Hence we propose algorithms capable of detecting, and localizing the bollard by means of experiment consisting of a robotic arm equipped with scanning lidar and its relative motion with the bollard carried out in a purposely developed realistic laboratory environment. Critical aspects include the robustness explained in terms of mean error and standard deviation of the error of the proposed algorithm to detect, and localize bollard at different relative orientations with respect to lidar mounted on the robotic arm. The novelties proposed here are listed below.

1) Propose custom algorithm for non-cooperative object (bollard) detection and localization - 3D feature matching (3DFM) algorithm. It extracts the features of the bollard's surface (unlike the whole object) from the computer aided model and matches those features to discriminate the experimental 3D lidar volumetric raw point clouds data corresponding to the bollard's surface from the background in the scene.
2) Propose Mixed feature-corresponding matching (MFCM) algorithm. It is based on the principle to uses 3D lidar volumetric raw point clouds data corresponding to the bollard's surface extractedin 1 using the features to identify to bollard's surface in the scene. Since it mixes two methods - feature matching as in (1) augmented by the correspondence matching in rest of the experimental acquired scene, hence the name Mixed feature-corresponding matching (MFCM).
3) Both the algorithms are implemented upon upon 105 scenes where bollard is at different orientation relative to lidar. Detailed performance evaluation of both algorithms 3DFM and MFCM qualitatively and quantitatively based on probabilistic parameter estimation.

Hence, we present an end-to-end pipeline - (a) proposing algorithms to detect and segment bollard from scene, (b)validating them upon 105 experiment data and (c) determining the performance of each of proposed algorithms.

The paper is organized as follows. Section II presents the current state of the art for non-cooperative object detection, localization and tracking, focusing on autonomous maritime navigation. Section III describes the experimental setup to mimic the realistic scenario where the lidar mounted on the robotic arm is used to capture and perceive the scene including bollard with various relative orientations between the lidar and the bollard. Section IV explains the the customized algorithms for non-cooperative bollard detection, and localization in the sparse 3D point cloud representation of the scenes. Further, section V presents the results followed by performance evaluation of proposed algorithms by the analyzing the critical aspect such as errors, and reliability. Finally, discussion and conclusion in section VI concludes the paper.

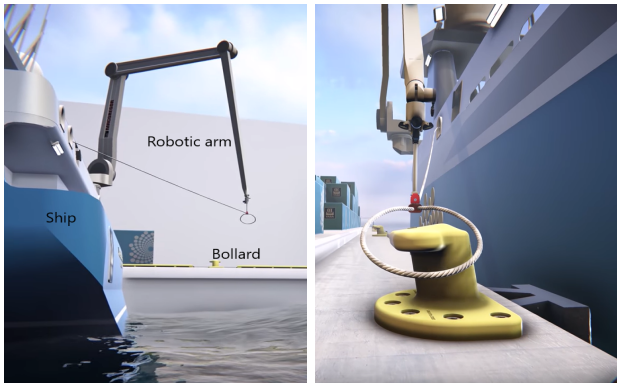

Fig. 1. Autonomous mooring. (left) conceptual presentation showing the relative position of ship, robotic arm and bollard; (right) exploded view (with permission from MacGregor Norway AS).

II. Related work

The lidar points are basically a range scan in 3D (x,y,z). It contains the spatial coordinates of the object in 3D making it easier to obtain the shape and pose of the object. In particular, in the case of autonomous vehicle where the perception and control both lie on the same platform, and accurately localizing the object in 3D coordinate system is crucial for subsequent path planning and control. To estimate the depth information in addition to the dimension of the object, 3D bounding box in point cloud in lidar such as PointNet [2] has been implemented, however it requires extensive computing resource to train. Exploiting the sparsity of 3D lidar point cloud, their density and resolution algorithms based on neural network such as Frustum PointNet [3] Multi-View 3D networks (MV3D) [4], RoarNet [5], AVOD [6] have been implemented. [7] used multi view convolution neural network with self-attention to classify roof from lidar point cloud. However, they are computationally expensive and require loads of data for training. Further, the model is highly data dependent that prevents its wide spread application.

With the objective is to overcome the limitations posed by individual methods - detect and localize the bollard in 3D space avoiding the use extensive processing and without a need to train the 3D convolution model with loads of test data, numerous approaches are proposed for object detection and classification. Authors in [8] proposed filtering of lidar point clouds using spectral graph based approach to separate the objects from ground. Ghamisi proposed classification of the objects from raw 3D lidar data by extracting several features such as - aggregated local point neighborhoods, laser echo ratio, variance of point elevation, plane fitting residuals, and echo intensity in addition to lidar digital surface model (DSM) [9]. Further, Authors in [10] presented methods to generate lidar points to meet certain level of details while keeping balance between the computation cost to process 3D lidar data, the storage space required to store sparse 3D lidar data while optimizing the coverage volumes. Similarly [11] proposed algorithm based on 3D shape context by defining the 3D point

cloud object. This method included several intermediate steps such as - road segmentation, ground segmentation, clustering similar points based on Euclidean distance, and determining Histogram. They contain several modules and the performance of the algorithm depended upon the efficiency of each of the intermediate steps. [23] proposed to detect the road crub based on two step method - first determine the potential points corresponding to crub based on energy function and then refine them using least cost path model.

In addition to the above mentioned techniques to process the sensory data and extract the object, model based algorithms has considered serious attraction [12]–[14]. It is based on the assumption that the lidar measurement data contain the features and related information related to the objects to be detected. For e.g. the lidar based 3D point cloud data contain the range of the objects in 3D space and hence by processing them, the 3D surface of the object can be extracted. Thus the template matching problem can be broken down to identifying the computer aided designed (CAD) generated model (of the object, in this case bollard) with that in the experimental lidar acquired data points. Approaches based on point tuple matching require the relative orientation point tuple has been proposed by authors in [15]. Voting based approach [16] requires additional intensity data. Hinterstoisser et al. in [17] presented template matching based on colour and depth from Kinect sensors. They propose using color information in order to prune invalid candidate transformations while depth data is used to improve pose estimation using Iterative closest point (ICP). Li et al. in [18] devised a method for detection and placement of CAD models in scanned scenes. This method has a drawback as it is used to match the entire object in the scene and the computation cost increases with the number of points in the scene and/or increasing the number of templates in the database. Authors in [19] used template matching together with kernel density estimation clustering to detect the pedestrian. Here, the raw 3D point cloud needed to be pre-processed such as - ground segmentation, grid filtering, hierarchical segmentation and projection of 3D data points to 2D plane using the principle component analysis (PCA). This projection of information to lower dimension results in loss of information. In addition, it is not always easy to separate the ground and hierarchical segmentation in itself is computationally expensive as it requires comparison of each data point with neighbouring points.

There are numerous work done related to mooring operations, for example, [20]–[22]. However, they are focused on manual operation operations and how the structural integrity can be maintained, the effect of mooring angle on mooring operations and so forth. In regards to the autonomous mooring where the moore is detected by sensors, and mooring operation is carried autonomously, very limited work has been done. Authors in [24] proposed laser based berthing and mooring of ship with out any towing assistance. Further, authors in [25] presented sensor fusion based approach for autonomous docking operation of surface vehicles (SV). However the approach was based on fusing data from lidar, IMU and GPS to determine the geometric features of floating platform and then determine the relative position and orientation of moore relative to SV.

Continuing the similar trend, we present novel methods to detect and localize the bollard in the scene from raw 3D point cloud. Overcoming the above mentioned drawbacks, our algorithm (3DFM) first extracts the features in the model data and then uses it to filter the data points in the experimental scene. This, on one hand lowers the computational cost, as the feature extraction in the model has to be done only once, while lowering the number of data points for comparisons in the experimental data. Further, an improved algorithm (MFCM) is presented by extending 3DFM algorithm to extract the point clouds and features of bollard from experimental scene and use it as template to detect and localize the bollard in rest of the scenes.

III. Problem Formulation

The concept behind the autonomous mooring is shown in Fig. 1. It consists of a robot (with manipulator) equipped with sensors (in this case lidar) and perception algorithm to perceive the scene. It does this in discrete steps of time in which it moves some predefined vertical distance from the bollard. The robot repeats this process until it localizes, and finds the unique coordinate of the bollard in 3D space. Once the position of bollard is found, the robotic arm holding the moore navigates to the bollard and lays off the moore around the bollard to dock the ship. The conceptual diagram in Fig. 1 is replicated in the laboratory (shown in Fig. 2). The experimental setup consists of Ouster lidar fitted on ABB robot placed at a vertical distance of 1.4 m from the bollard. The bollard is manufactured in-house[1] in the way that it meets the requirement of uncooperative target - (a) it is built with foam material, (b) it is not tagged with any kind of markers (c) it is not equipped with any kind of reflecting material, (d) it is not painted with bright colours instead painted in yellow to have low-moderate reflection. Then robotic arm is moved at different orientations relative to the bollard and the point cloud is acquired at each point (represented by blue dot in Fig. 3). A total of scenes, N_s = 105 were acquired[2]. During each measurement, the lidar is configured to have vertical and horizontal resolution of 64 and 1024 respectively, thus giving the measurement points of $64 \times 1024 = 65536$ for each scene. Then, each of these acquired scenes is tested upon algorithms to detect the bollard (represented by N_b points) and represent it by a unique coordinate in 3D, $\boldsymbol{p} \in R^3$.

The notations is as follows. Vectors and point clouds are defined by small, italics bold face, constant quantities by capital normal font, variables by small italics normal font and set of objects enclosed in curly braces. Let $\mathcal{S} = \{\boldsymbol{s}_i\}_{i=1}^{N_s}$, where $N_s \in Z$ be the set of all scenes captured during the experiment. Each element of $\mathcal{S}$ (i.e $\boldsymbol{s}_i$) is coordinate of 3D sparse

[1]The CAD model and the measuremet of dimenstion of template bollard is available at - https://github.com/ajitjha14/Bollard-Segmentation-and-Position-Estimation-fromLidar-Point-Cloud-for-autonomous-mooring/upload/main/Bollard_model_dimension

[2]The dataset is available at git: https://github.com/ajitjha14/Bollard-Segmentation-and-Position-Estimation-fromLidar-Point-Cloud-for-autonomous-mooring/tree/main/Data. At the time of uploading, Git only allows 100 files to upload, so 100 out of 105 files can be found here.

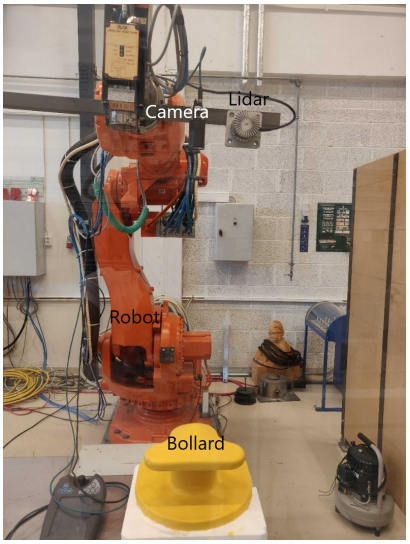

Fig. 2. Experimental setup for detection and classification of bollard for autonomous mooring. In this setup, the vertical distance of the lidar relative to bollard is 1.4 m. The lidar frame is as follows: +x is right, +y is up and +z is out of plane.

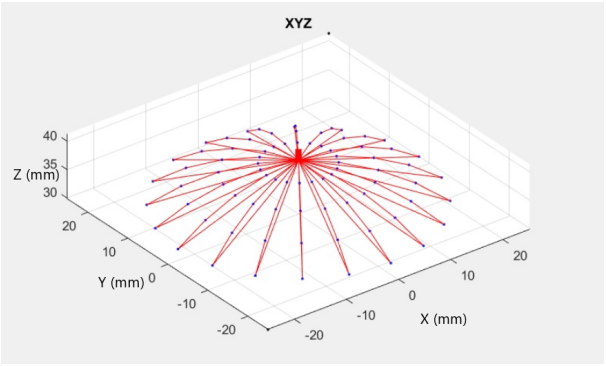

Fig. 3. Data acquisition at different relative position of robot with respect to the bollard.

point containing information about the scene. Mathematically, $\boldsymbol{s}_i = \{\boldsymbol{p}_j\}_{j=1}^{N_p}$, where $\boldsymbol{p}_j \in R^3 \quad \forall i$. The objective is to detect the bollard $b_i = \{\boldsymbol{p}_k\}_{k=1}^{N_b}$ where $\boldsymbol{p}_k \in R^3, N_b < N_p \quad \forall i$ from the scene s_i and finally presenting it with a unique coordinate $o_i = \{\boldsymbol{p}_i\}$, where $\boldsymbol{p}_i \in R^3 \quad \forall i$. For the ease of understanding the detailed relationships among the scene data set, ($\mathcal{S}$), individual scene (e.g. $\boldsymbol{s}_i$) and the co-ordinates of each lidar point ($\boldsymbol{p}_j$) is described in Fig. 4.

IV. Algorithms

To detect the bollard in the scene and determine its unique coordinate in 3D from the sparse point cloud captured by lidar, we employ two methods - (a) 3D feature matching (3DFM) and (b) Mixed feature-correspondence matching (MFCM).

1) 3D feature matching (3DFM): In the case of 3DFM, first the CAD model of the bollard (T) is converted to point cloud ($\boldsymbol{t_c}$). As explained above, we have a custom built 3D CAD model of bollard that was manufactured in-house. While manufacturing the CAD model, the parameters were chosen in a way that it fits with the real world bollard (especially

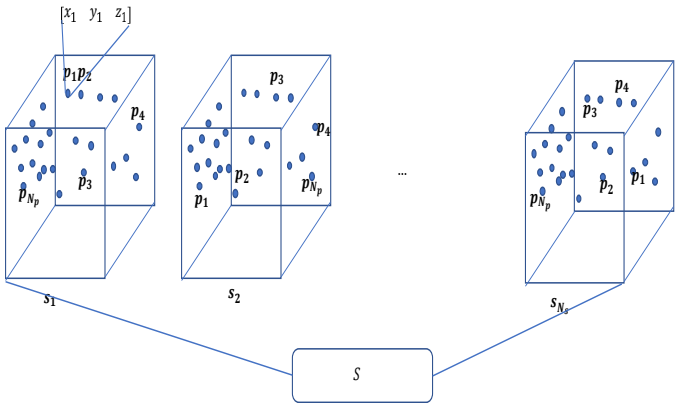

Fig. 4. Relationship between scene set, ($\mathcal{S}$), scene ($\boldsymbol{s}_i$) and the co-ordinates of points in the scene ($\boldsymbol{p}_i$)). The dimensions of $\mathcal{S}$ is $[1 \times N_s]$; $\boldsymbol{s}_i$ is $[1 \times N_p]$; $\boldsymbol{p}_i$) is $[3 \times 1]$. For the experiment $N_S = 105$, $N_p = 65536$.

in Norway). Since the actual dimension of the bollard e.g. its length, width and height is known, using these geometric parameters, a model of plane is formulated using RANSAC algorithm [26]. Then from this model, the points lying on the plane is differentiated with other. In this way the points corresponding to the top surface of bollard ($\boldsymbol{t}$) is discriminated from rest of the point cloud. Once the point corresponding to the plane of top surface of bollard is estimated, it is matched with the experimental data set ($\mathcal{S}$, defined above in Fig. 2 and Fig. 3) obtained from lidar for registration, and to identify the bollard under consideration. Further, the plane of top surface of bollard ($\boldsymbol{t_p}$), and its centroid ($\boldsymbol{o}$) is extracted representing the bollard by a unique coordinate. The entire procedure for template matching based extraction from scene is described in Algorithm 1.

2) Mixed feature-correspondence matching (MFCM): In the case of MFCM, first the bollard and the lidar is calibrated such that the latter lie vertically at distance of 1.4 [3] m (for e.g.) above the former. Under these conditions, the scene is acquired (s_1, shown in Fig. 2 and described by the center point in Fig. 3). Using Algorithm 1, the point cloud corresponding to the top surface of bollard in the experimental scene ($\boldsymbol{s}_1$) is obtained and is used as template ($\boldsymbol{t_t}$) to locate the bollard, find the centroid of the top surface of bollard to represent it by a unique co-ordinate in rest of the scenes $\{\boldsymbol{s}_i\}_{i=2}^{N_s-1}$. All the steps associated with template extraction from calibrated scene and its use in correspondence matching is explained in Algorithm 2 and 3 respectively.

V. Results

The detailed investigation on the efficiency of the methodologies present is further explained.

A. 3D feature matching (3DFM)

Figure 5 summarizes the results for the 3DFM method (Algorithm 1) - starting from the acquired scene (Fig. 5 (a)), 3D CAD template of bollard (Fig. 5 (b)), the point cloud

[3]The choice of vertical distance sets the distance where the lidar starts scanning

Algorithm 1 Bollard detection and position estimation using 3DFM.

Input: Scenes, $\mathcal{S} = \{\boldsymbol{s}_i\}_{i=1}^{N_s}$; Template, T
Output: Surface of bollard, $\boldsymbol{t_{p'}}_i$; centroid of surface of bollard, $\boldsymbol{o}_i$; error, e_i $\quad \forall i$

1: $\boldsymbol{t_c} \leftarrow$ convert T to point cloud
2: $\boldsymbol{t} \leftarrow$ point cloud representing the surface of bollard obtained from geometric dimension and features of surface of bollard from $\boldsymbol{t_c}$
3: $c \leftarrow$ centroid of the surface of bollard obtained from $\boldsymbol{t}$
4: **for** i=1 to N_s **do**
5: $\quad \boldsymbol{s}_i \leftarrow$ remove all the points in $\boldsymbol{s}_i$ with +ve y coordinate ▷ +y represents the point above the lidar
6: $\quad \boldsymbol{s}_i \leftarrow$ denoise $\boldsymbol{s}_i$
7: $\quad \boldsymbol{t_p}_i \leftarrow$ plane of the bollard in $\boldsymbol{s}_i$ using the features in $\boldsymbol{t}$ and distance 1.4 m ▷ the robot starts looking for bollard as the distance is 1.4 m from lidar (for e.g.)
8: $\quad \theta_i \leftarrow$ angle between the $\boldsymbol{t_p}_i$ and $\boldsymbol{t}$ using Iterative closest point algorithm (ICP)
9: $\quad \boldsymbol{t_p}_i \leftarrow$ rotated $\boldsymbol{t_p}_i$ by angle θ_i to align extracted plane with the template, $\boldsymbol{t}$
10: $\quad \boldsymbol{o}_i \leftarrow$ centroid of the surface of bollard point clouds obtained from $\boldsymbol{t_p}_i$
11: $\quad e_i \leftarrow$ error, distance between c and $\boldsymbol{o}_i$ calculated using Euclidean norm
12: $\quad$ **return** $\boldsymbol{t_p}_i$, $\boldsymbol{o}_i$, e_i
13: **end for**

Algorithm 2 Template for MFCM

Input: Calibrating scene, $\boldsymbol{s}_1$; surface of the bollard, $\boldsymbol{t}$ (from Algorithm 1, step 2)
Output: Surface of bollard, $\boldsymbol{t_t}$ ▷ returns the surface of bollard extracted from calibrating scene ($\boldsymbol{s}_1$)

1: $\boldsymbol{s}_1 \leftarrow$ remove all the denoised points in $\boldsymbol{s}_1$ with +ve y coordinate ▷ +y represents the point above the lidar
2: $\boldsymbol{t_t} \leftarrow$ plane of the bollard in $\boldsymbol{s}_1$ using $\boldsymbol{t}$ ▷ using RANSAC and ICP algorithm
3: **return** $\boldsymbol{t_t}$

corresponding to the top surface of the bollard (Fig. 5 (c)) and finally the points clouds corresponding to the bollard's top surface is extracted using Algorithm 1 (Fig. 5 (d)). Algorithm 1 is applied upon all scenes $\mathcal{S}$ with different orientation as shown in Fig. 3 using setup described in Fig. 2. The errors

Algorithm 3 Bollard detection and position estimation using MFCM

Input: Scenes, $\mathcal{S} = \{\boldsymbol{s}_i\}_{i=2}^{N_s}$; Template, $\boldsymbol{t_t}$ (obtained from Algorithm 2)
Output: Surface of bollard, $\boldsymbol{t_p}$; centroid of surface of bollard, $\boldsymbol{o}_i$; error, e_i $\quad \forall i$

1: **for** i=1 to N_s **do**
2: $\quad$ repeat step 5 to 12 (described in Algorithm 1)
3: $\quad$ **return** $\boldsymbol{t_p}_i$, $\boldsymbol{o}_i$, e_i
4: **end for**

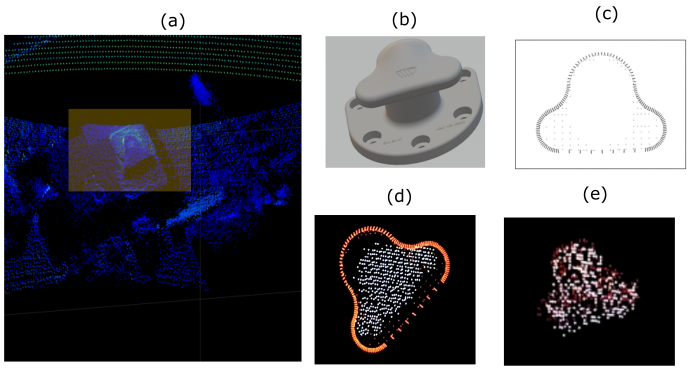

Fig. 5. Testing and validation of the proposed algorithms. (a) Experimental scene acquired from the setup defined in Fig. 2 and the bollard highlighted by shaded rectangle. (b) 3D template of bollard. (c) top surface of bollard extracted (d) extraction of the bollard's surface from the scene using 3DFM using Algorithm 1 (white) and compared against the template (orange) (e) using MFCM, Algorithm 2 and 3 (white) and compared against the template (red).

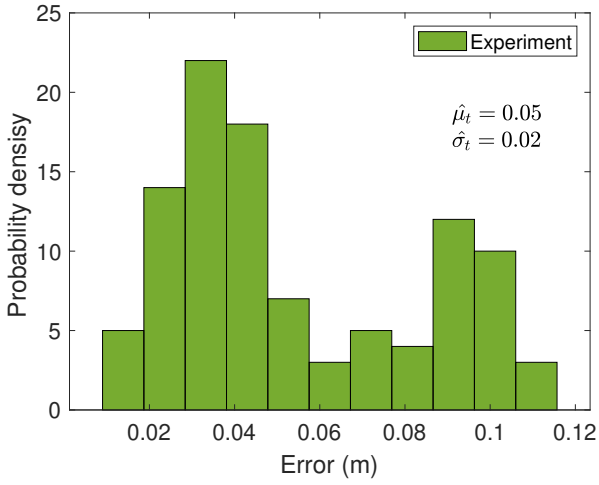

Fig. 6. Performance evaluation of 3DFM (Algorithm 2). The errors incurred by processing the set of scenes in $\mathcal{S}$ follow univariate bi-modal Gaussian distribution.

defined as the Eculidean Norm (see Algorithm 1, step 11) is summarized in Fig. 6. It is basically the histogram of the errors defined in terms of probability density. It is observed that the probability density function of the error incurred from the 3DFM algorithm is bi-modal as it has two dominant modes. Thus, this case can be classified as univariate bi-modal Gaussian mixture model (GMM). Hence the problem now translates to (a) finding the optimal Gaussian components (cluster), K that make this distribution (b) estimation of parameters of individual Gaussian components $\hat{\theta}^t = [\hat{\mu}^t \quad \hat{\sigma}^t]$ and (c) probability that a given value from the error data set fall with in given Gaussian cluster.

Mathematically, a univarite multimodal GMM is defined in terms of individual Gaussian component weights (ϕ_k), mean (μ_k) and variances (σ_k)[4]. For e.g. a GMM with K Gaussian components, the k^{th} Gaussian component has mean μ_k and

[4]since the model here is univariate, we stick with the notation corresponding to univariate, bimodal GMM.

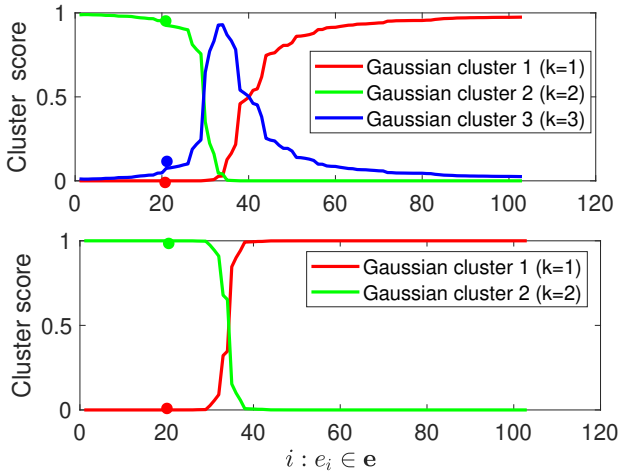

Fig. 7. Optimal choice of Gaussian components.

variance σ_k and has weight ϕ_k. Since we have the data related to errors, it is desirable to calculate the posterior probability, defined as the probability that the error e_i ($e_i \in \boldsymbol{e}$) belongs to a particular Gaussian component k with probability p. Mathematically, it is defined as

$$p(e_i) = \sum_{k=1}^{K} \phi_k \mathcal{N}(x \mid \mu_k, \sigma_k), \quad \text{where} \tag{1}$$

$$\mathcal{N}(e_i \mid \mu_k, \sigma_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(e_i-\mu_k)^2}{2\sigma_k^2}}, \quad \text{and} \tag{2}$$

$$\sum_{k=1}^{K} \phi_k = 1 \tag{3}$$

a) Gaussian component: Though, two dominant modes in the probability density of the error (Fig. 6) is clearly visible, quantitative analysis is performed to ascertain the optimal number of Gaussian cluster (modes). For this, mathematical induction based approach is taken. First, the error data is fitted with three Gaussian components, $k = 3$ using $k-$means++ algorithm [27]. The cluster score, defined as the probability that a given error sample from $\{e_i\}_{i=2}^{N_s}$, lies in k^{th} cluster is plotted against the scene ($i \in [2..N_s]$)[5] Fig. 7 (up) shows the results for $k = 3$. It is observed that for most of the cases, the Gaussian component with $k = 2$ (C_2) is almost zero for most of the scenes and only the other Gaussian components, $k = 1$ (C_1) and $k = 3$ (C_2) are dominant. For e.g. consider the case when $i = 20$ (demonstrated by the solid filled circle) the cluster score of components $C_1(k = 1)$, $C_2(k = 2)$ and $C_3(k = 3)$ are 0, 0.1 and 0.9 respectively. Similarly, Fig. 7 (down) shows the results for $k = 2$. In this case, the Gaussian clusters are well defined with minimum, overlap. For e.g. for scene $i = 20$, the (demonstrated by the solid filled circle) the cluster score of components $C_1(k = 1)$, and $C_2(k = 2)$ are 0, and 1 respectively. Thus, it is binary clustering for most of the cases (except at the transitions) with one cluster having dominant score over other.

[5]It should be noted that each scene number, i corresponds to different orientation of lidar with respect to bollard, described in Fig. 3. Indexing of scene, i starts from 2 as the first index ($i = 1$) is used as template for rest of the scenes.

b) Parameter estimation of individual Gaussian component: The algorithm to estimate the parameters and the probability is explained in Algorithm 4. The results are

Algorithm 4 Parameter estimation of the GMM resulting from errors in 3DFM (Algorithm 1)

Input: $\mathbf{e} = \{e_i\}_{i=2}^{N_s}$ ▷ error data
$K = 2$ ▷ number of Gaussian components (please see section V(a) titled *Gaussian component*)
Output: $\hat{\boldsymbol{\mu}}^t, \hat{\boldsymbol{\sigma}}^t, \hat{\boldsymbol{\phi}}^t,$ ▷ mean, standard deviation and mixture weight of individual Gaussian component respectively (see (1), (2) and (3)) such that $\hat{\boldsymbol{\mu}}^t, \hat{\boldsymbol{\sigma}}^t, \hat{\boldsymbol{\phi}}^t \in R^K$

$\mathbf{r} \leftarrow K$ random integers between 1 and N ▷ $\mathbf{r} \in Z^K$
$\hat{\boldsymbol{\mu}}^t \leftarrow \mathbf{e}(\mathbf{r})$ ▷ assign random values from error data to the estimated mean of each Gaussian component
$\hat{\boldsymbol{\sigma}}^t \leftarrow \frac{1}{N}\sum_{i=1}^{N}(e_i - \bar{\mathbf{e}})^2$ ▷ assign sample variance to the estimated variance of each Gaussian component
$\hat{\boldsymbol{\phi}}^t \leftarrow \frac{1}{K}$ ▷ Set all Gaussian components distribution prior estimates to the uniform distribution
for $iter = 1$ to 500 **do**
  $\hat{\boldsymbol{\gamma}}^t \leftarrow \frac{\hat{\boldsymbol{\phi}}^t \mathcal{N}(\mathbf{e} \mid \hat{\boldsymbol{\mu}}^t, \hat{\boldsymbol{\sigma}}^t)}{\sum_{\forall i}(\hat{\boldsymbol{\phi}}^t \mathcal{N}(\mathbf{e} \mid \hat{\boldsymbol{\mu}}^t, \hat{\boldsymbol{\sigma}}^t)}$ ▷ $\hat{\boldsymbol{\gamma}}^t \in R^{N\times K}$. $\hat{\gamma}_{ik}^t$ is the probability that i^{th} value of the error data, e_i is generated by Gaussian component C_k. Thus $\hat{\gamma}_{ik}^t = p(C_k \mid e_i, \hat{\boldsymbol{\mu}}^t, \hat{\boldsymbol{\sigma}}^t, \hat{\boldsymbol{\phi}}^t)$
  $\hat{\boldsymbol{\phi}}^t \leftarrow \frac{\sum_{\forall k} \hat{\boldsymbol{\gamma}}^t}{N}$ ▷ update Gaussian component weight
  $\hat{\boldsymbol{\mu}}^t \leftarrow \frac{\sum_{\forall k} \hat{\boldsymbol{\gamma}}^t \mathbf{e}}{\sum_{\forall k} \hat{\boldsymbol{\gamma}}^t}$ ▷ update mean
  $\hat{\boldsymbol{\sigma}}^2 \leftarrow \frac{\sum_{\forall k} \hat{\boldsymbol{\gamma}}^t (\mathbf{e} - \hat{\boldsymbol{\mu}}^t)^2}{\sum_{\forall k} \hat{\boldsymbol{\gamma}}^t}$ ▷ update variance
  return $\hat{\boldsymbol{\mu}}^t, \hat{\boldsymbol{\sigma}}^t, \hat{\boldsymbol{\phi}}^t$
end for

described in Fig. 8 and Fig. 9. It is observed that the numerical method based approach to determine the parameters works effectively for all the parameters. We summarize the parameters by $\hat{\boldsymbol{\theta}}^t = [\hat{\boldsymbol{\mu}}^t; \hat{\boldsymbol{\sigma}}^t; \hat{\boldsymbol{\phi}}^t]$, such that $\hat{\boldsymbol{\theta}}^t \in R^{3\times 2}$, as the univariate bimodal GMM is defined by three parameters and each of parameters has two values as determined above in Algorithm 4. To be very specific , the parameters are as follows - $\hat{\theta}_1^t = [\hat{\mu}_1^t \;\; \hat{\sigma}_1^t \;\; \hat{\phi}_1^t] = [0.03 \;\; 0.01 \;\; 0.67]$ and $\hat{\theta}_2^t = [\hat{\mu}_2^t \;\; \hat{\sigma}_2^t \;\; \hat{\phi}_2^t] = [0.09 \;\; 0.01 \;\; 0.33]$. Further, the observed experimental error approximated by a Gaussian components is shown in Fig. 10 for probabilistic description and performance estimation. It can be clearly observed that the parameters estimated from the Algorithm 4 fits well with the experimental observed error from template matching. Hence the error distribution resulting from the template matching algorithm follows univariate bimodal GMM.

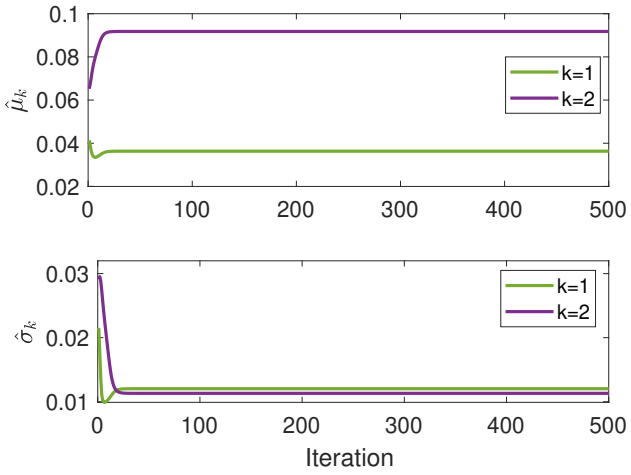

Fig. 8. Error parameter estimation resulting from 3DFM described in Algorithm 4. The errors follow univariate, bimodal GMM. The mean (up) and standard deviation (down) of individual Gaussian components are - $\hat{\boldsymbol{\mu}}^t = [\hat{\mu_1}, \hat{\mu_2}] = [0.03, 0.09]$ m and $\hat{\boldsymbol{\sigma}}^t = [\hat{\sigma_1}, \hat{\sigma_2}] = [0.01, 0.01]$ m respectively.

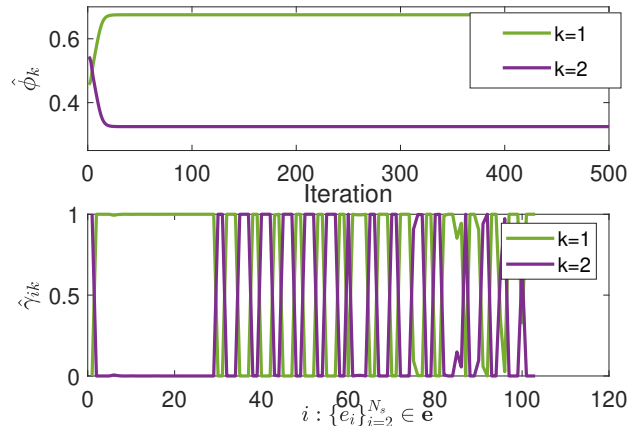

Fig. 9. Error parameter estimation resulting from 3DFM described in Algorithm 4. The errors follow univariate, bimodal GMM. (up) The weight of component C_1 and C_2 is 0.67 and 0.33 respectively. (down) demonstrates the probability that error sample e_i comes form k^{th} component with parameters $\hat{\boldsymbol{\mu}}^t$, $\hat{\boldsymbol{\sigma}}^t$ and $\hat{\boldsymbol{\phi}}^t$ explained above.

c) Clustering Gaussian components: Having determined the model parameters and using Bayes' theorem, the probability that the data points belongs to cluster k is determined as

$$p(C_k \mid e_i) = \frac{\hat{\phi}_k^t \mathcal{N}(e_i \mid \hat{\mu}_k^t, \hat{\sigma}_k^t)}{\sum_{j=1}^{j=K} \hat{\phi}_j^t \mathcal{N}(e_i \mid \hat{\mu}_j^t, \hat{\sigma}_j^t)}, \tag{4}$$

where $k \in [1..K]$. All the related parameters are obtained and described above (please see section V(a) and V(b)). Putting the values of those parameters in 4, the probability that individual error values (e_i) lies to a particular Gaussian component (cluster) is determined and shown in Fig. 11. The scenes i and hence the orientation of lidar relative to the bollard that cause larger error (purple) are well separated by their counter part that cause relatively lower errors (green).

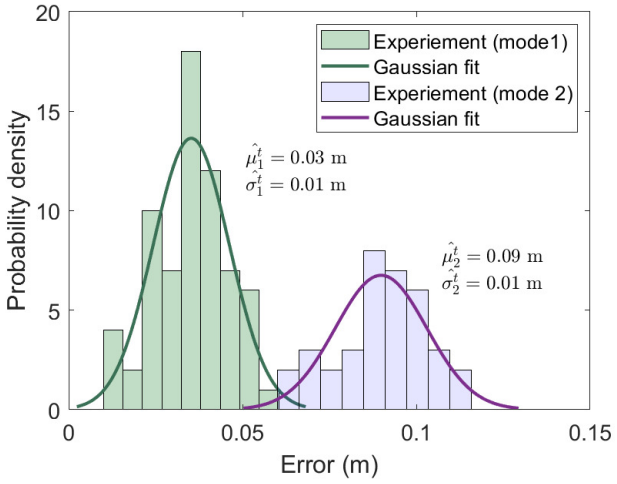

Fig. 10. Performacne evaluation of 3DFM algorithm. From the probability density (histogram) it is observed that the distribution has two modes hence the errors follow univariate bimodal Gaussian mixture model (GMM). Parameters of Gaussian components are obtained from Algorithm 4 with the parameters summarized in Fig. 8 and Fig. 9 is compared with the experimental error. It is observed that the experimental error well fits univariate bimodal GMM.

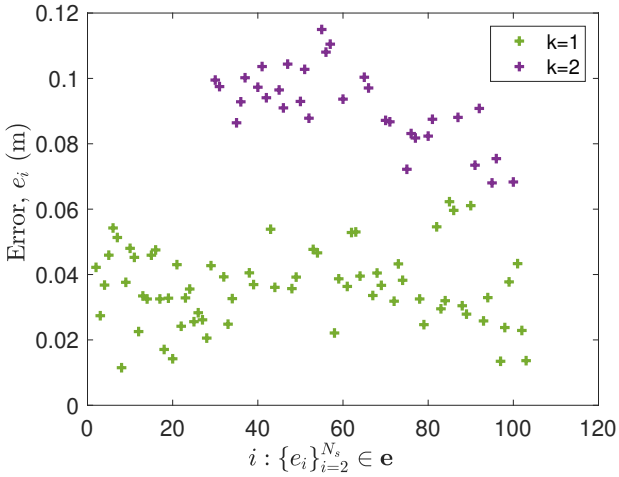

Fig. 11. Clustering the error samples resulting from template matching algorithm described in Algorithm 4. Putting the parameters obtained from Fig. 8 and 9 in (4), we are able to separate the samples that corresponds to Gaussian component having lower error values ($k = 1, C_1$) from the relatively higher error values ($k = 2, C_2$).

B. Mixed feature-correspondence matching (MFCM)

Figure 5 (e) summarizes the results for the MFCM method (Algorithm 1 and 2). Similar to 3DFM, histogram in terms of probability density as a function of error is shown in Fig. 12 (blue). It is observed that the probability density function of the error incurred from the MFCM algorithm is uni-modal as it has one dominant mode. Thus, this case can be classified as univariate uni-modal Gaussian model (unlike the 3DFM case, where it was GMM). Further, it is desirable to estimate the parameters of statistical model that best describes the probability of occurrence of observed error such that statistical inference can be made. Mathematically, this translates the problem to finding an optimal parameter θ that maximizes

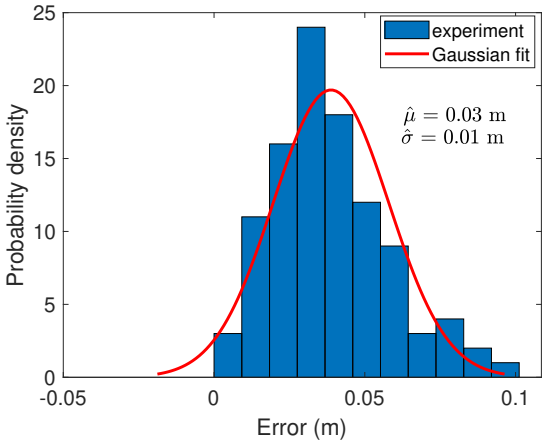

Fig. 12. Errors resulting from the correspondence matching algorithm (3). The error follows a Gaussian probability density function (pdf) with mean ($\mu^c = 0.03$ m) and standard deviation ($\sigma^c = 0.01$ m).

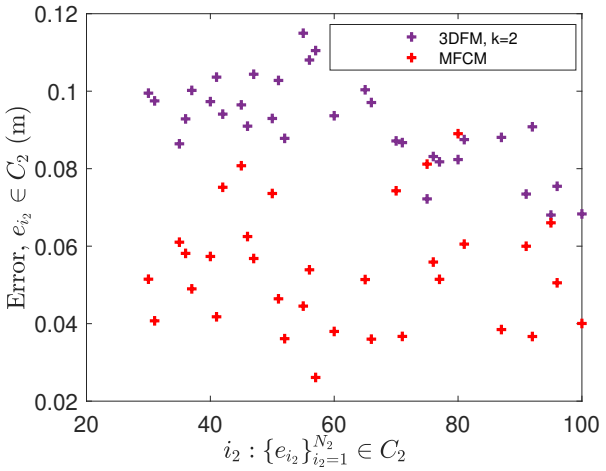

Fig. 13. Performance comparison of 3DFM (Algorithm 1) with MFCM algorithm (Algorithm 2 and 3). The errors introduced by the former is compensated by the latter. This result can be attributed to the absence of extra mode with mean $\hat{\mu}_2^t$ in latter.

the statistical model. Mathematically,

$$\hat{\mu^c} = \arg\max_{\mu} \quad \mathcal{N}(\boldsymbol{e} \mid \mu, \sigma) \tag{5}$$

$$\hat{\sigma^c} = \arg\max_{\sigma} \quad \mathcal{N}(\boldsymbol{e} \mid \mu, \sigma) \tag{6}$$

It is found that the Gaussian model has parameters - estimated mean, $\hat{\mu^c} = 0.03$ m and estimated deviation, $\hat{\sigma^c} = 0.01$ m. It is well described in Fig. 12 (red). Further, t-test, with significance level ($\alpha = 0.05$) is carried to perform the statistical hypothesis testing to validate the estimated parameter. Here, we define the null hypothesis (H_0) and alternate hypothesis (H_1) as - H_0 : error comes from Gaussian distribution having estimated mean, $\hat{\mu} = 0.03$ m and H_1 : error comes from Gaussian distribution having different estimated mean, $\hat{\mu} \neq 0.03$ m. From the test, the p-value (p), the probability of observing significant deviations from estimated mean ($\hat{\mu} = 0.03$) assuming that null hypothesis is true approximately equal to 1. Since $p > \alpha$, we fail to reject the null hypothesis and thus the errors follow Gaussian distribution with mean $\hat{\mu^c} = 0.03$ m.

Detailed modelling of the error statistics of both algorithms - 3DFM and MFCM, confirm that while the errors incurred from former follow univariate bimodel Gaussian distribution with two components (GMM) having means 0.03 and 0.09 m and standard deviation 0.01 m, the latter follow univarite unimodal Gaussian distribution with mean 0.03 and standard deviation 0.01 m. Thus, MFCM algorithm performs better in terms of lowered mean error, while both the algorithm are equally precise (standard deviation). To make a better illustration of this conclusion, referring to Fig. 11, the scene i corresponding to the component $C_2, (k = 2)$ resulting from 3DFM is compared with MFCM algorithm (shown in Fig. 13). It is observed that the MFCM algorithm has smaller errors as compared to the 3DFM, hence giving the improved performance. It is also evident that about 95% of the error lies in the range [0.01 m, 0.05 m] ($\hat{\mu^c} \pm 2\hat{\sigma^c} = 0.03 \pm 0.02$). Since the diameter of moore rope is much larger (approximately 2x) than the diameter of bollard (usually in terms of meters), the maximum error estimated from the correspondence matching is with in the allowable error.

VI. Conclusion and Discussion

Two algorithms - 3D feature matching (3DFM) and Mixed feature-corresponding matching (MFCM) are presented to detect and localize a non-cooperative target (bollard) by processing raw 3D point cloud data from lidar aimed at autonomous mooring. The performance of proposed algorithm is tested and validated upon 105 scenes where the relative position of lidar and bollard is different. Based on detailed statistical and probabilistic theory, the model parameters of the errors resulting from both algorithm are estimated. Further, using Bayes' theorem and the estimated parameters, the errors are clustered in a way to find out the suitability of the one algorithm over the other. Finally, it was concluded though both the methods had equal mean deviation error of 0.01 m, the MFCM algorithm having mean error 0.03 m out performs 3DFM algorithm.

Further, it would also be desirable to study the effect of different lighting conditions (e.g. limited vision), and different weather conditions replicating urban sea. Secondly, the proposed algorithm can be extended to the bollards of different shape, size, structure and morphology. Another aspect is to evaluate the performance of algorithms at different distance at various lidar resolution. The slight increase in vertical distance of lidar relative to bollard would not affect the result much. But if the distance is increased significantly, the bollard would contain relatively a smaller fraction of total points and the detection could be challenge. But this can be compensated by choosing the lidar operating mode with greater resolution, that in turn would increse the cost. Further, a balance between lidar operating resolution (64x1024 or other configuration depending upon lidar model and cost) and its distance from bollard should be optimized and it remains the work for future.

ACKNOWLEDGMENT

Authors would like to thank Dr. Geir Hovland for making the bollard template available and his useful suggestions. Authors also thank Norwegian Reserach Council of Norway for supporting this research project number - 261647/O20, under the BIA Program and INCAPS, project number - 287918 under INTPART program.

REFERENCES

[1] S. Sukkarieh, E. M. Nebot and H. F. Durrant-Whyte, "A high integrity IMU/GPS navigation loop for autonomous land vehicle applications," in IEEE Transactions on Robotics and Automation, vol. 15, no. 3, pp. 572-578, June 1999

[2] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017, July 2017.

[3] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D Object Detection from RGB-D Data," in IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2018, June 2018.

[4] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "MultiView 3D Object Detection Network for Autonomous Driving," in IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017, July 2017.

[5] K. Shin, Y. P. Kwon, and M. Tomizuka, "RoarNet: A Robust 3D Object Detection based on Region Approximation Refinement," CoRR, vol. abs/1811.03818, 2018.

[6] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D Proposal Generation and Object Detection from View Aggregation," in IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS) 2018. IEEE, 2018.

[7] D. A. Shajahan, V. Nayel and R. Muthuganapathy, "Roof Classification From 3-D LiDAR Point Clouds Using Multiview CNN With Self-Attention," in IEEE Geoscience and Remote Sensing Letters, vol. 17, no. 8, pp. 1465-1469, Aug. 2020

[8] E. Bayram, P. Frossard, E. Vural and A. Alatan, "Analysis of Airborne LiDAR Point Clouds With Spectral Graph Filtering," in IEEE Geoscience and Remote Sensing Letters, vol. 15, no. 8, pp. 1284-1288, Aug. 2018.

[9] P. Ghamisi and B. Höfle, "LiDAR Data Classification Using Extinction Profiles and a Composite Kernel Support Vector Machine," in IEEE Geoscience and Remote Sensing Letters, vol. 14, no. 5, pp. 659-663, May 2017.

[10] K. L. Damkjer and H. Foroosh, "Lattice-Constrained Stratified Sampling for Point Cloud Levels of Detail," in IEEE Transactions on Geoscience and Remote Sensing, vol. 58, no. 8, pp. 5627-5641, Aug. 2020, doi: 10.1109/TGRS.2020.2967880.

[11] Y. Yu, J. Li, H. Guan, C. Wang and J. Yu, "Semiautomated Extraction of Street Light Poles From Mobile LiDAR Point-Clouds," in IEEE Transactions on Geoscience and Remote Sensing, vol. 53, no. 3, pp. 1374-1386, March 2015.

[12] Y. Yu, J. Li, H. Guan, F. Jia and C. Wang, "Three-Dimensional Object Matching in Mobile Laser Scanning Point Clouds," IEEE GeoScience and Remote Sensing Letters, Vol. 12, no. 3, pp. 492-496, March 2015.

[13] K. Williams, M. J. Olsen, G. V. Roe and C. Glennie, "Synthesis of transportation applications of mobile LiDAR", Remote Sens., vol. 5, no. 9, pp. 4652-4692, Sep. 2013.

[14] Y. Yu, J. Li, J. Yu, H. Guan and C. Wang, "Pairwise three-dimensional shape context for partial object matching and retrieval on mobile laser scanning data", IEEE Geosci. Remote Sens. Lett., vol. 11, no. 5, pp. 1019-1023, May 2014.

[15] B. Drost, M. Ulrich, N. Navab, and S. Ilic,"Model globally, match locally: efficient and robust 3D object recognition,"Proceedings of the 2010 IEEE conference on computer vision and pattern recognition (CVPR), 2010.

[16] B. Drost, S. Ilic, "3D object detection and localization using multimodal point pair features," IEEE Proceedings of the 2012 second international conference on 3D imaging, modeling, processing, visualization and transmission (3DIMPVT)," 2012. pp. 9–16 .

[17] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, et al, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," Proceedings of the Asian conference on computer vision. Springer; 2012. pp. 548–62.

[18] Y. Li, A. Dai, L. Guibas, M. Nießner, "Database-assisted object retrieval for real–time 3D reconstruction,"Proceedings of the computer graphics forum, 34. Wiley Online Library; 2015. pp. 435–46.

[19] Kaiqi Liu, Wenguang Wang and Jun Wang, "Pedestrian Detection with Lidar Point Clouds Based on Single Template Matching," Electronics, Vol. 8, pp. 780, 2019

[20] S. Ji, M. Choi and Y. Kim, "A Study on Position Mooring System Design for the Vessel Moored by Mooring Lines," in IEEE/ASME Transactions on Mechatronics, vol. 20, no. 6, pp. 2824-2831, Dec. 2015.

[21] S. Han, B. Jia and Y. Gu, "Study on optimizing influence of mooring angle in multi-point mooring positioning system," 2018 Chinese Control And Decision Conference (CCDC), pp. 3777-3782, 2018.

[22] M. Dissanayake et al., "Automated Application of Full Matrix Capture to Assess the Structural Integrity of Mooring Chains," in IEEE Access, vol. 6, pp. 75560-75571, 2018.

[23] S. Xu, R. Wang and H. Zheng, "Road Curb Extraction From Mobile LiDAR Point Clouds," in IEEE Transactions on Geoscience and Remote Sensing, vol. 55, no. 2, pp. 996-1009, Feb. 2017.

[24] Marko Perkovic et. al,"Laser-Based Aid Systems for Berthing and Docking," J. Mar. Sci. Eng. 8, 346, 2020.

[25] P. Leite, R. Silva, A. Matos and A. M. Pinto, "An Hierarchical Architecture for Docking Autonomous Surface Vehicles," 2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 1-6, 2019.

[26] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography", Commun. ACM, vol. 24, no. 6, pp. 381-395, 1981.

[27] D. Arthur, and S. Vassilvitskii, "k-means++: the advantages of careful seeding," Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics Philadelphia, USA, 2007, pp. 1027–1035.

Mehak Jindal completed bachelors degree in Computer Science and Engineering in 2020. She was a visiting student at the department of ICT, University of Agder, Norway during January 2020 - June 2020. She is currently pursuing Masters in Remote Sensing and Geographic Information Systems at Indian Institute of Remote Sensing, Dehradun, India. Her primary interests areas include the application of machine learning for satellite images and in the field of remote sensing.

Ajit Jha was born in Nepal in 1984. He received B.Sc. in Electronics and Communication Engineering (Bangladesh), European Masters in Photonic Networks from Aston University (England) and Scuola Superiore Sant Anna (Italy) and PhD from Technical University of Catalunya (Spain) and Karlsruhe Institute of Technology (Germany) in 2007, 2012 and 2016 respectively. From 2016-2019 he worked he worked at various industries related to Autonomous vehicle working on innovative technologies such as Automotive ethernet, ADAS, Surround view system , camera mirror system, Blind sport warning to name a few. Currently, he is an Associate Professor of Mechatronics at Department of Engineering Sciences at University of Agder, Norway. He is actively involved in research focused on sensors, sensor fusion, image/signal processing, ML, ADAS functionalities towards Autonomous systems, and IoT. He has (co) authored more than 20 articles and two patents. In addition, he has been active reviewer, and member of techincal program committee of numerous international peer-review journals and conferences. Dr. Jha is the recipient of Erasmus Mundus Masters Course (EMMC) and Erasmus Mundus Joint Doctorate (EMJD) both funded by European Union (EU).

Linga Reddy Cenkeramaddi received the master's degree in electrical engineering from the Indian Institute of Technology, New Delhi, India, in 2004, and the Ph.D. degree in electrical engineering from the Norwegian University of Science and Technology, Trondheim, Norway, in 2011. He worked for Texas Instruments in mixed signal circuit design before joining the PhD program at NTNU. After finishing his PhD, he worked in radiation imaging for an atmosphere space interaction monitor (ASIM mission to International Space Station) at University of Bergen, Norway from 2010 to 2012. At present, he is working as an associate professor at University of Agder, Campus Grimstad, Norway. His main scientific interests are in Cyber-Physical Systems, Autonomous Systems and Wireless Embedded Systems.

Dear Associate Editor and Reviewers,

Thank you for reviewing our manuscript comprehensively. Your suggestions and feedbacks have helped us improve the quality of the work. Please find the response to your queries below.

Associate Editor Comments:

Comments to the Author:

The scholarly contribution is limited, while it may provide an interesting solution(s) to a real application. However, the need, significance and effectiveness of the solution in practice is not convincingly demonstrated. You may want to clarify why you do not have the locations of the bollard as the prior, which is so easy to get from surveying. Plus, I believe you need it to guide the robot arm anyway. The absolute accuracy of 0.01m etc does not mean anything. It should be a function of your lidar ground spacing. This has to be generalized to a meaningful situation/statement. What is the expected accuracy anyway for this application ? The difference and need of 'two' methods are not clearly stated. Furthermore, I agree with the reviewers the article is hard to follow in many places.

Response: Thank you for reviewing our manuscript in great detail. We are sorry that some context was not clear. But now we have addressed your concerns.

"Regarding need of significance and effectiveness"

Response: We are again sorry that it was not clear in the manuscript. Now we have included the following the manuscript that explains the need of local sensing system.

Normally Ships or vessels use GPS to locate the location of bollard and mooring is done manually with human involvement. The use of GPS for bollard localization have serious implications. First, they have to rely on third part. Second, the measurement or localization from GPS are not accurate compared to local measurement using active sensors on-board in Ship or vessel. Third, the update rate of the GPS is lower as compared to lidar (proposed here). Fourth, the received signal from GPS might under go several reflections before reaching the receiver on board, causing multipath errors, delay. Further, the GPS receiver on the ship (for e.g.) uses multiple set of satellites in order to find the position. Changes in satellite configuration, orientation or position will affect the measurement. Based on the faulty position, if we guide the robot to that location with uncertainties, there is every chance that the manipulator or end-effector on the robot and do not successfully moore or even collide with bollard or nearby structure (in worst case)

To overcome these drawbacks, it is desirable to have a local sensors that is capable of determining the bollard's position effectively. For that we propose lidar based sensor that lies on-board connected to the robot for mooring operations. The on-board sensor system for localization and segmentation had added advantages

Accuracy: We demonstrated the accuracy (determined by the mean of the errors) of proposed algorithm is 3 cm as compared to 1.5 m for GPS based system [Ref. 1 in manuscript]

Precision: We demonstrated the precision (determined by the standard deviation of the errors) of proposed algorithm is 1 cm

Independence: The driver of the ship or vessel do not have to rely on the third party

Redundancy: Since the proposed methodology is based on only lidar, redundant sensors can be kept as backup in case one fails.

Local effect: Unlike the GPS based prior, where the signal can be delayed or poor estimation (because of factors mentioned above), the lidar based system is only affected by local environment unlike the thousands of kilometer between GPS satellite and GPS receiver.

All these points have bee added to the manuscript to justify the need of local sensor system.

"Regarding accuracy as function of distance"

Response: I agree with you that the accuracy could be a function of distance of lidar relative to bollard. This need to be studied thoroughly. This is the work of future and we have explained this in detail in the discussion part in the manuscript.

"Regarding accuracy of mooring"

Response: Normally the moore are 30-60 cm in wide, but this varies with geographic location. So accuracy in order of cm is required (unlike in meters from GPS as mentioned in [])

"Regarding the need of two methods"

Response: Initially we made an hypothesis – instead of matching the whole CAD template of bollard to the scene and finding the bollard in the scene as mentioned in [17, 18], why not first extract the particular segment from CAD template bollard (in this case top surface, also known as features) and then use this as template to find the bollard's surface in the scene. This has advantages in terms of computation complexity. The complexity increases with the number of points to be processed. Since we extract the key features from template bollard (reduced number of points) and use this to match with the bollard in scene. This reduced the computational cost dramatically. We named it 3DFM algorithm. With standard laptop as computing device, the results were obtained with in seconds.

But during the experiment and implementation, we came up with additional idea – First use the CAD template to find the bollard's surface in experimental scene, and use the latter as template to find the bollard's surface in experiment. We named it MFCM algorithm.

The working principle of both the algorithm is well described in section IV (1) and IV (2) respectively. After detailed evaluation of both the algorithms we concluded that MFCM has better KPIs in terms of accuracy and precision (again this is well elaborated in abstract and Results section)

"Regarding hard to follow"

Response: We are sorry that we did not present our work well such that the readers can understand it. However, following your suggestions, along with Reviewers suggestions, significant changes has been done. We are confident that now the manuscript will be easy follow.

Reviewer: 1

Comments to the Author
Bollard detection and segmentation is an important research topic in autonomous mooring. The authors utilize the advantages of lidar to accomplish this task. However, the proposed method has weak innovations and the superiority of proposed method is not demonstrated in the analysis section. The content of the manuscript also needs to be re-organized.

Response: Thank you going through the manuscript comprehensively. The novelty of our proposed method is now highlighted clearly in the manuscript. Please refer to last paragraph in section introduction. Following your suggestions and other Reviewers, the content has changed and re-organized.

"Regarding novelty"

Novelty 1: Two algorithms for bollard detection and segmentation

1a : Propose custom algorithm for non-cooperative object (bollard) detection and localization - 3D feature matching (3DFM) algorithm. It extracts the features of the bollard's surface (unlike the whole object) from the computer aided model and matches those features to discriminate the experimental 3D lidar volumetric raw point clouds data corresponding to the bollard's surface from the background in the scene.

1b : Propose Mixed feature-corresponding matching (MFCM) algorithm. It is based on the principle to uses 3D lidar volumetric raw point clouds data corresponding to the bollard's surface extracted in 1a using the features to identify to bollard's surface in the scene.

"superiority of proposed method"

Novelty 2: Comprehensive study of the proposed algorithm and determine the performance of each of the proposed algorithm. Based on statistics, we determined that MFCM performs better than 3DFM. Please refer to Figure 13, that clearly demonstrates that the larger error from 3DFM (in purple) is compensated by MFCM (in red). Also the KPIs of each of the algorithm is mentioned in abstract.

Following your suggestions and other Reviewers, the content has changed and re-organized. We are confident that it will enhance the ease of readability and easy to follow.

1. The title is too long, and the authors should shorten the title.

Response: I have changed the title to - "Bollard Segmentation and Position Estimation from Lidar Point Cloud for autonomous mooring"

2. The last sentence of novelty in page 1 is too long.

Response: It is paraphrased now.

3. The organization of related work section is not clear. In my opinion, the authors should re-organize this section.

Response: Thank you for the detailed review of our work. Following your suggestion, the un-related topics such as camera, sensor-fusion and other techniques has been removed. Only the works related to point cloud processing for object detection, segmentation and localization is included.

4. The title of section III can be changed to be "problem formulation", and delete the title of section III-A.

Response: Have made the changes as you suggested.

5. What is the "geometric dimension and features" in the second step of algorithm 1? How does the top surface of bollard t is extracted from the whole point cloud of bollard? What is the meaning of "+ve"?

Response: Thank you for pointing out that we missed to explain the details of extracting point clouds representing the top surface of bollard using dimensions of Template Bollard.

As explained in the manuscript that, we have a custom built 3D CAD model of bollard that was manufactured in-house. While manufacturing the CAD model, the parameters were chosen in a way that it fits with the real world bollard (Fig. 1 below). Since the actual dimension of the bollard e.g. its length, width, height is known, using these geometric parameters, a model of plane is formulated using RANSAC algorithm. Then from this model, the points lying on the plane is differentiated with other. In this way the points corresponding to the top surface of bollard ($\boldsymbol{t}$) is discriminated from rest of the point cloud. Once the point corresponding to the plane of top surface of bollard is estimated, it is matched with the experimental data set (S, defined above in Fig. 2 and Fig. 3) obtained from lidar for registration, and to identify the bollard under consideration. These are now included in the manuscript.

Further, the detail measurement of the bollard template (3D CAD) is uploaded in git (https://github.com/ajitjha14/Bollard-Segmentation-and-Position-Estimation-fromLidar-Point-Cloud-for-autonomous mooring/upload/main/Bollard_model_dimension) and is also included in the manuscript now. We tried to upload in the manuscript but the measurement was not clear. So we decided to upload to Git and provide the link in manuscript.

Regarding the '+ve'. The lidar is positioned in such a way that its -ve y co-ordinate falls towards the bollard and the +ve y co-ordinate falls towards the ceiling. So all the points corresponding to the +ve y co-ordinate is filtered out. This is also included in caption of Fig. 2

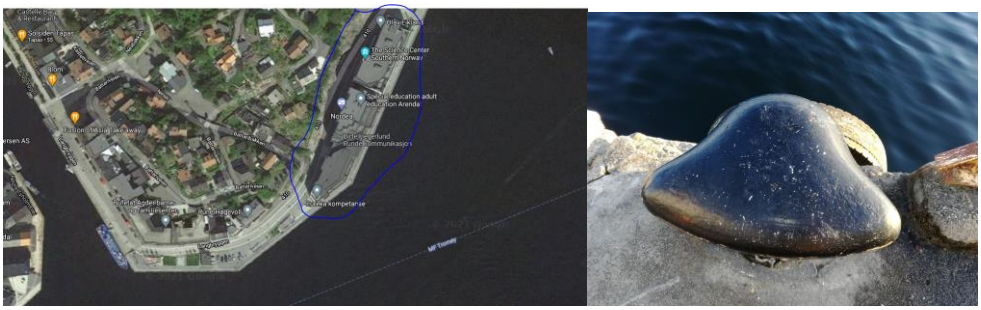

Fig. The real world example of bollard (right) located in Arendal, Norway (left). The CAD model of template bollard in manuscript (Fig. 2 in manuscript) is based on dimension of this real world bollard.

6. The goal of the proposed is to detect and segment the bollard from point cloud. However, the content in the section V pays lots of attentions on the analysis the statistical properties of errors, which has less meaning for demonstrating the superiority of proposed method.

Response: Yes, the goal is to propose algorithms that is capable to detect and segment bollard from the point cloud. It is desirable for the algorithm to work at various position and orientation of the lidar relative to the bollard. For this, we proposed two algorithms - (a) 3DFM (b) FMCM and the novelties is briefly mentioned above. Further the novelties are mentioned in the manuscript as well now.

With due respect, I would like to mention that this is one of the strong points for the manuscript that it performs the reliability check of the proposed algorithm. For the real-world implementation, it is very desirable to know in advance about the reliability of the algorithm. For this, both the algorithms were used to process 105 experimental point cloud with various positions and orientation of the lidar relative to bollard. Then based on the data backed by statistical processing, we concluded that FMCM was a better candidate then 3DFM. This is summarized in Fig. 12. Based on this result, for real world application, Fig. 12 presents a strong argument in favour of FMCM in real world application.

In this way we have presented an end-to-end pipeline - (a) proposing algorithms, (b)implementing them upon experiment data and (c) determining the performance of each of proposed algorithm. I am sorry that we could not point the importance of the performance evaluation. But now they have been included in the manuscript to strengthen the scope of work.

Reviewer: 2

Comments to the Author
This manuscript proposes a computer-aided object detection and localization method from lidar 3D point cloud data for autonomous maritime navigation. Comprehensive evaluations with an experimental data set of 105 scenes validate the superiority of the proposed method. However, there are still some problems in this manuscript.

Response: Thank you for reviewing our manuscript in details and pointing out the shortcomings. This will add value to our manuscript.

1)In this manuscript, Although v-a0a is mentioned in algorithm 4, its meaning is not explained, and v-a0b is not mentioned in the context., please add relevant explanations in the manuscript.

Response: We are sorry that we did not explain in details and that it has caused the confusion. In fact, v-a0a is a hyperlink that links to section 'V(a) Gaussian component' where we explained how to calculate the parameter K (number of Gaussian components). As per suggestion, the link has been removed and text has been added – ' number of Gaussian components (please see section V(a) titled *Gaussian component*)'

Similarly, v-a0b is replaced with – 'All the related parameters are obtained and described above (please see section V(a) and V(b))'

2)2.In the experimental part of the article, in algorithm 1, the scale and structure of data set S should be introduced in detail; in algorithm 2, the way of obtaining calibration scene s1 is one part of S？

Response: Thank you for pointing out that the relationship among different variables were not clear. To make it clearer, we have added Figure 4 in the manuscript.

Yes, in algorithm 2, s_1 is one of element of S, where the lidar is vertically above the bollard. This is already mentioned in section IV(2)

3)The goal of this paper is to solve the problem of non-cooperative target detection and location. It is better to display or analyze the non-cooperative characteristics of the data set used by s, and the experimental test data should include the data collected in the actual production.

Response: To address the non cooperative characteristic of the bollard, we have added the following sentence in the manuscript - 'The bollard is manufactured in-house meeting the shape, size and dimension of the standard real world bollard (especially in Norway). The bollard is designed and manufactured in the way that it meets the requirement of uncooperative target - (a) it is built with foam material, (b) it is not tagged with any kind of markers (c) it is not equipped with any kind of reflecting material, (d) it is not painted with bright colours instead painted in yellow to have low-moderate reflection.'

The point cloud representation of experimental data would not fit well with the A4 size paper. However, most of the relevant part of the scene is demonstrated in Figure 5 (a). Respecting your suggestions, the experimental test data is uploaded in git and link (https://github.com/ajitjha14/Bollard-Segmentation-and-Position-Estimation-fromLidar-Point-Cloud-for-autonomous-mooring/tree/main/Data) is provided in the manuscript. During upload, it was found that git only allows to upload 100 files. So 100 out of 105 files were uploaded.

Further, the detail measurement of the bollard template (3D CAD) is uploaded in git (https://github.com/ajitjha14/Bollard-Segmentation-and-Position-Estimation-fromLidar-Point-Cloud-for-autonomous mooring/upload/main/Bollard_model_dimension) and is also included in the manuscript now. We tried to upload in the manuscript but the measurement was not clear. So we decided to upload to Git and provide the link in manuscript.

Reviewer: 3

Comments to the Author
- Accept - Minor revision.

Comments:

The authors present two algorithms - (a) 3D feature matching (3DFM) and Mixed feature-correspondence matching (MFCM) to detect and localize bollard from the 3D scene. The algorithms are implemented on 105 acquired experimental scenes. The authors also evaluates the performance of algorithms using error data obtained during the implementation and makes statistical analysis to analyze their performance and demonstrates that latter is better than the former. In the case of 3DFM, there is novelty that the top surface of bollard (from CAD template) is extracted and used for extracting the surface of 3D experimental bollard, reducing the computational cost. Similary in case of MFCM, the 3D point cloud extracted from 3DFM is used as template to localize the bollard. The performance evaluation and comparisons including - probability distribution estimation of errors, parameters estimation - estimating mean and standard deviation error, clustering the scenes that cause relatively larger errors of each of the methods add value to the work.

The presentation, and flow of material is smooth and well established. The state of the art covering the camera, sensor fusion and CNN is well elobroated but can be improved. This work can be considered to be published, however following issues must be addressed.

Response: Thank you for your time in reviewing our manuscript. We are sure that your suggestions will improve its quality.

line 35 - These problems are dealt with -- omit with

Response: It is removed now.

line 40 - To overcome this limitation, lidars are used to sense the environment exploiting the fact they work well independently -- This is not true lidar data are effected by illumination. ..... relatively well would be a better fit here

Response: Thank you for reviewing the manuscript in detail. I have added them.

line 46 - tracking is not the scope here. The authors only does detection and localization.

Response: It is removed now

Algorithm 1 - Why Authors chooses 1.4 m and starts scanning the scene ? It could be well eloborated.

Response: The distance is chosen as proof of concept and as to match the real world problem . I agree that the algorithms could be tested upon various other distance and the optimal working distance could be found, but it remains the work for future.

Algorithm 4 - Why the upper limit of iter is set to 500 ? Although from Fig. 7, the error converges well below 500 iterations, is there any justification or it is obtained by hit and trial method ?

Response: Yes, you are right, it was chosen by trial. As you pointed out the error converges well below, but a higher number of iteration is chosen to be sure that the algorithm has enough time (through number of iterations) to converge, in case of some undesirable data comes across.

Reviewer: 4

Comments to the Author

The paper demonstrates a methodological workflow for automated maritime mooring using 3D LiDAR point cloud.
Though the work appears interesting and the paper is not well written. The sections are not organised properly and flow of text appears confusing to the reader

Response: Thank you for reviewing our manuscript comprehensively. Your suggestions had helped us improve the quality of our work. Following your suggestions and other Reviewers, the content has changed and re-organized now.

Specific comments:
How does each scene has exactly 65536 points? Was there any preprocessing applied to extract exactly the number of points mentioned?

Response: The operation mode of lidar is chosen to have vertical and horizontal resolution of 64 and 1024 respectively giving the data points – 64x1024=3556. Now this is explained in the manuscript.

Does the effect of the vertical distance(in this case 1.4 m) influence the point cloud captured?

Response: The distance is chosen as proof of concept and as to match the real world problem In my opinion, the slight increase in vertical distance would not affect much, but if the distance of the lidar to bollard is fairly large, the bollard would would contain relatively a smaller fraction of total points and the detection could be challenge. But this can be compensated by choosing the lidar operating mode with greater resolution. Further, a balance between lidar operating resolution (64x1024 or other configuration depending upon lidar model) and its distance from bollard should be optimized. I agree that the algorithms could be tested upon various other distance and the optimal working distance could be found, but it remains the work for future. Thank you for pointing out the insight. I have now added them in section Discussion and future work.

How many CAD templates were used for the study? Will changing the model used, affect the performance of the algorithm? Discussion of the results needs to be more comprehensive

Response: Only on CAD template were use. It was made in according with the standard shape and dimension of bollard in Norway. Respecting, other Reviewer's and your comment, CAD model and the measurement of the bollard template is uploaded in Git and the link is included in the manuscript. It can be found at - https://github.com/ajitjha14/Bollard-Segmentation-and-Position-Estimation-fromLidar-Point-Cloud-for-autonomous-mooring/upload/main/Bollard_model_dimension

Literature review appears to be incomplete. The state of the art work on using LidAR sensors for mooring is not present.

Response: We have added them now. Please refer to reference [19-24] in the manuscript.

Time complexity of the algorithm may be specified. The application needs to work in real time. What is the time complexity for different processes?

Response: Thank you raising your concern. Computation time is one of the main concerns in the real world application. This has been already stated in the novelty – the paragraph just before section – Problem formulation. I am again including for ease of reading.

"we present novel methods to detect and localize the bollard in the scene from raw 3D point cloud. Overcoming the above mentioned drawbacks, our algorithm (3DFM) first extracts the features in the model data and then uses it to filter the data points in the experimental scene. This, on one hand lowers the computational cost, as the feature extraction in the model has to be done only once, while lowering the number of data points for comparisons in the experimental data. Further, an

improved algorithm (MFCM) is presented by extending 3DFM algorithm to extract the point clouds and features of bollard from experimental scene and use it as template to detect and localize the bollard in rest of the scenes."

The computational cost depends upon the number of points that has to be compared in experiment scene with that in template. Unlike the methods for example in [17] where the authors had to compare the experimental data points with the entire CAD model, with the proposed method, we only need to compare with limited number of points (points corresponding to the surface of bollard), reducing the computational cost dramatically. With the standard laptop, our method could do the computation in order of seconds.

Thank you again for reviewing our manuscript and giving useful comments and suggestions. These suggestions have not only enriched our knowledge but also helped to strengthen our research. We are confident that we have addressed the concern that you have had.

Sincerely,

Ajit Jha
Mehak Jindal, and
Linga Reddy Cenkeramaddi