

ADAPTIVE SHAPE FITTING FOR LIDAR OBJECT DETECTION AND TRACKING IN MARITIME APPLICATIONS

JIAYING LIN, GIOVANNI CAMPA, CHRISTIAN-EIKE FRAMING, JAN-JÖRAN GEHRT, RENÉ ZWEIGEL
& DIRK ABEL

Institute of Automatic Control, RWTH Aachen University, Germany

ABSTRACT

Environmental perception and monitoring play essential roles in maritime automation. Besides radar, the use of LiDAR for maritime surveillance has been increasing in recent years thanks to its high accuracy, high data density and good robustness against varying lighting conditions. This paper presents a novel approach for an adaptive shape-fitting technique using LiDAR point clouds in maritime applications, improving the object-tracking performance. The clustered LiDAR point clouds are fitted into bounding boxes or elliptic cylinders depending on their geometric shapes. A fitting score based on mean squared error is used for the shape decision. Afterwards, the extracted objects are associated with those in the past frames and tracked using an adaptive extended Kalman filter. The proposed algorithm is validated in simulation and post-processing using real-world test data. In simulations, the proposed adaptive shape-fitting technique shows a high object positioning and heading accuracy and guarantees a good object-tracking behaviour with a positioning error of 1.5 m. The proposed algorithm's efficiency and robustness are further validated using test data recorded in the real-world using an unmanned surface vehicle equipped with LiDAR and GNSS in Rostock harbour, Germany. Test results show that the proposed adaptive shape-fitting technique helps the multi-object tracker reach a 2D position error of approximately 2 m with an update rate of 10 Hz, which is sufficient for object tracking in maritime applications. The size accuracy is improved by 10%, and heading accuracy is improved by 16% compared with multi-object tracking approaches only using L-shape fitting.

Keywords: maritime surveillance, multi-object tracking, object detection, shape fitting

1 INTRODUCTION

Multi-object tracking is one of the essential tasks in maritime automation. In cooperative manoeuvring or path crossing, an unmanned surface vehicle (USV) is required to detect and track objects in its environment for trajectory planning and collision avoidance. Object-detection and object-tracking frameworks provide an estimate of the object's pose (e.g., position, heading) and dynamic (e.g., linear velocity, angular velocity) over time. Therefore, object detection and tracking are critical aspects of our project GALILEOnautic 2 [1], aiming to elevate the automation level and enable cooperative interaction among maritime participants.

The standard devices used for environment perception and surveillance in maritime applications are radar sensors and the automatic identification system (AIS). Radar allows the perception of objects at kilometre distances. But radar measurements are limited with their low accuracy and resolution. Marine radars have blind areas in the near field (<500m), in which the object detection accuracy is low. Furthermore, noise due to sea clutter complicates the detection of small profile objects [2], [3]. AIS allows vessels to broadcast their position and route information to other vessels and be localized and possibly avoided. Nowadays, ships 300 gross tonnages and upwards are required to use AIS equipment. Nevertheless, critical issues still exist with AIS-only solutions, such as low update rate, unpredictable faulty messages due to human error and insufficient availability for small vessels.

Due to radar and AIS's limitations, different authors [3], [4], [5] proposed the environment perception using LiDAR as an extension for near field detection. Compared with radar, LiDAR can perceive the near field environment in a higher resolution and accuracy with height information for more accurate object detection and tracking. In these approaches, the LiDAR point clouds are first divided into clusters belonging to separate objects. In [3] and [4], the authors have treated each object as a mass point, approximating the objects' position with the centre of the cluster itself without considering the object's geometry. Neglecting size information may lead to a collision when the objects are closed to the ego vehicle. To improve the positioning accuracy and avoid potential collisions, we should consider each object's geometry into the positioning, which means the LiDAR point clusters should be fitted into geometric shapes, so that the size and heading of the objects can also be determined. The authors in [5] have used elliptical cylinders to fit only one object, assuming that this object has elliptical shapes and no ground truth is available for the positioning accuracy. To improve the positioning accuracy and avoid oversized fitting, we should decide if the obstacle has a rectangular or elliptical shape since the maritime participants' shape is various. Even for the same vessel, different parts, e.g., the elliptical bow and the rectangular stern, are scanned while driving. Wrongly fitting an elliptical object into rectangular bounding boxes or the other way round may also lead to wrong size or heading determination.

Addressing the limitations mentioned above, we propose a novel shape-fitting technique for multi-object detection and tracking in maritime applications. Depending on its geometry, each LiDAR point cluster is fitted either into a bounding box using L-shape fitting or an elliptical cylinder using ellipse fitting. To find the best L-shape-fitting method for maritime application, we also summarize and compare three different L-shape-fitting methods, [6], [7] and [8], which are commonly used in the automotive sector. The shape-fitting results are then forwarded into adaptive extended Kalman filter (AEKF) respectively for multi-object tracking (MOT). The framework considers the unique requirements for maritime use – including various object shapes, dimensions, and motion profiles – and is flexible for an extension with radar or AIS to extend the perception scope. The proposed algorithm is tested both in simulation and using test data recorded in the real-world. With the new adaptive shape-fitting technique, the subsequent mean object-tracking accuracy should reach 2m at positioning, $\pm 20^\circ$ at heading. At the same time, the computational efficiency should be kept high with a 10 Hz repetition rate.

2 METHODOLOGY

2.1 System design

Figure 1 presents the algorithm structure for object detection and tracking using LiDAR. In the pre-processing module, the point clouds are grouped into clusters based on position density using the modified Euclidean clustering method. Afterwards, the shape-fitting module

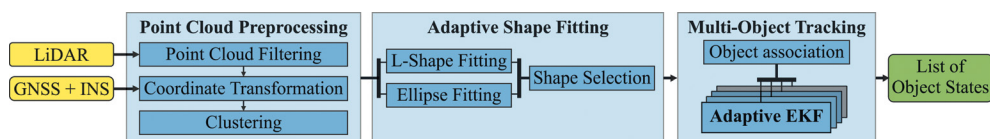


Figure 1: Proposed algorithm structure for LiDAR object detection and tracking.

converts the clusters corresponding to separate objects to an object list containing position, orientation and object size. Each object description is extracted from the shape that best fits the cluster point cloud to generate reliable values for the geometrical properties. The current object detections are subsequently associated with previously tracked objects using object position and size information. In the MOT part, we use a bank of AEKF to track the associated objects. Every AEKF tracks a single object such that the total number of tracked objects can be kept flexible. We adapt the noise covariance matrix according to the object’s visibility in sensor view so that temporarily erroneous measurements do not impact the estimation performance. This paper focuses on the module adaptive shape fitting and MOT.

2.2 Summary of shape-fitting methods

Figure 2 generalizes the problem statement of both L-shape and ellipse fitting: each of these algorithms takes as input a list of points in $\mathbb{R}^{n \times 2}$ in x and y axis, where n is the number of points in the cluster. The output of these algorithms is the shape’s central point $P_0(x_0, y_0)$, heading θ (along the longest side of the fitted shape) and dimension l and w . Since the vessels’ shapes are usually incomplete ellipse, the length of the shape and the central point P_0 are then adjusted by finding the minimal and maximal point along the major axis, such that the incomplete elliptic boundaries can be determined. The box or elliptic cylinder’s height h is defined as the vertical distance between the highest and lowest points. We neglect the pitch and roll angles due to their low relevance for trajectory planning and collision avoidance.

2.2.1 L-shape fitting

Given a set of 2D points $S = \{(x_i, y_i) | i = 1, 2, \dots, n\}$, the process of fitting an L-shape can be described as an optimization problem:

$$\min_{P, \theta, c_1, c_2} \sum_{i \in A} (x_i \cos \theta + y_i \sin \theta - c_1)^2 + \sum_{i \in Q} (-x_i \sin \theta + y_i \cos \theta - c_2)^2, \quad (1)$$

where the disjoint point sets P and Q split the n cluster points $P \cup Q = S, P \cap Q = \emptyset$. Each set is associated with one out of two perpendicular straight lines l_1 and l_2 :

$$\begin{aligned} l_1 : x_i \cos \theta + y_i \sin \theta &= c_1 \\ l_2 : -x_i \sin \theta + y_i \cos \theta &= c_2 \end{aligned} \quad (2)$$

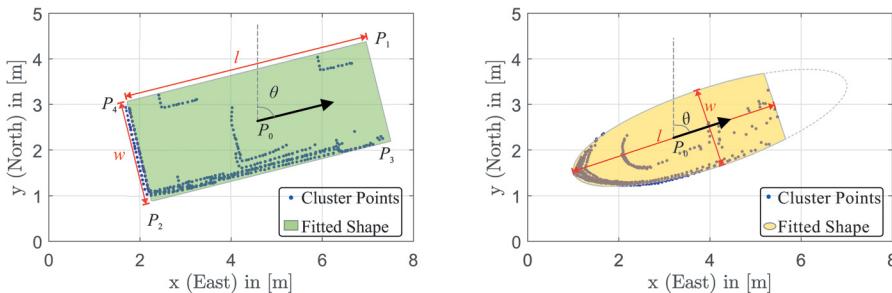


Figure 2: General geometry definitions of the shape fittings: Left: L-shape fitting for a container. Right: ellipse fitting for a small boat.

The solution of the optimization problem thus parameterizes two straight lines that approximate the object’s visible edges. The four corner points of the rectangle $\{P_i(x_i, y_i) | i = 1, 2, 3, 4\}$ in Fig. 2 can be determined by finding the furthest point associated with each line. While the linear regression of edge points with two straights represents a quadratic optimization problem, the additional problem of set association to the appropriate lines does not. Hence, the optimization problem does not have a direct analytic solution. In this section, we summarize and compare three methods to solve the L-shape-fitting problem:

(1) Brute Force Method: The Authors in [8] formulated the fitting problem by finding the optimized vector \mathbf{n} and \mathbf{c} : $\mathbf{n} = [\cos \theta_{opt} \sin \theta_{opt}]^T$, $\mathbf{c} = [c_{1,opt} c_{2,opt}]^T$ so that the sum of the Euclidean distance of all the points to their related l_1 or l_2 can be kept as minimal. According to our problem statement, we adapt and summarize the algorithm in [8], as shown in Algorithm 1. The splitting of \mathbf{S} to \mathbf{P} and \mathbf{Q} is conducted using a brute force method: first, the cluster points are sorted according to their position, and then all the possible point combinations are tried. Finally, the optimized \mathbf{P} and \mathbf{Q} that produce the lowest sum of Euclidean distance, projected with eigenvalue, are chosen. This method guarantees a robust way of choosing optimal set combinations but requires the point cloud to be sorted and is computationally expensive for a large point cloud set

Algorithm 1 L-Shape Fitting using Least Square Method (adapted from [8])

Input: Sorted 2D point cloud $\mathbf{S} = \{(x_i, y_i) | i = 1, 2, \dots, n\}$

Initialize: $\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ 0 & \mathbf{M}_{22} \end{bmatrix}$, with

$$\mathbf{M}_{11} = \begin{bmatrix} 1 & 0 \\ 0 & n-1 \end{bmatrix}, \mathbf{M}_{12} = \begin{bmatrix} x_1 & y_1 \\ \sum_{i=2}^n y_i & -\sum_{i=2}^n x_i \end{bmatrix}, \mathbf{M}_{22} = \begin{bmatrix} x_1^2 + \sum_{i=2}^n y_i & x_1 y_1 - \sum_{i=2}^n x_i y_i \\ x_1 y_1 - \sum_{i=2}^n x_i y_i & y_1^2 + \sum_{i=2}^n x_i \end{bmatrix}$$

for $i = 2 \rightarrow n$ **do** $\Delta \mathbf{M} = \begin{bmatrix} \Delta \mathbf{M}_{11} & \Delta \mathbf{M}_{12} \\ 0 & \Delta \mathbf{M}_{22} \end{bmatrix}$, with

$$\Delta \mathbf{M}_{11} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \Delta \mathbf{M}_{12} = \begin{bmatrix} x_i & y_i \\ -y_i & x_i \end{bmatrix}, \Delta \mathbf{M}_{22} = \begin{bmatrix} x_i^2 - y_i^2 & 2x_i y_i \\ 2x_i y_i & y_i^2 - x_i^2 \end{bmatrix}$$

$$\mathbf{M} = \mathbf{M} + \Delta \mathbf{M}, \lambda_i, \mathbf{v}_i = \text{eigen}(\mathbf{M}_{22} - \mathbf{M}_{12}^T \cdot \mathbf{M}_{11}^{-1} \mathbf{M}_{12})$$

end for

$$\lambda_{i_{min}} = \min_{i=1 \rightarrow n} \lambda_i$$

Output: $\mathbf{n} = \mathbf{v}_{i_{min}}, \mathbf{c} = (\mathbf{M}_{11, i_{min}}^{-1} \mathbf{M}_{12, i_{min}} \mathbf{n})$

(2) Rotation Method: To avoid the large for-loop of going through each cluster point, [7] proposed a method that searches for the best fitting rectangle among a list of incrementally rotated rectangles. As shown in Algorithm 2, by defining a searching resolution δ , this method iteratively tries every possible heading θ from , and finds the best fitting heading θ

by comparing a fitting score β . This score β considers the distance of the points to the related line l_1 or l_2 . Compared with the least squares method, the rotation method could avoid the high computational load but need to go through a for-loop with all the possible heading. The computational load is proportional to the desired heading resolution δ .

Algorithm 2 L-Shape Fitting using Rotation Method (Adapted from [7])

Input: 2D point cloud $\mathbf{S} = \{(x_i, y_i) | i = 1, 2, \dots, n\}$

for $k = 0 \rightarrow 90 / \delta$, **step 1, do**

$$\theta_k = k\delta$$

$$\mathbf{e}_1 = [\cos\theta_k \quad \sin\theta_k], \mathbf{e}_2 = [-\sin\theta_k \quad \cos\theta_k], \mathbf{C}_{1,k} = \mathbf{S} \cdot \mathbf{e}_1^T, \mathbf{C}_{2,k} = \mathbf{S} \cdot \mathbf{e}_2^T$$

$$\mathbf{D} = [\mathbf{C}_{1,k} - \min\{\mathbf{C}_{1,k}\}, \max\{\mathbf{C}_{1,k}\} - \mathbf{C}_{1,k}, \mathbf{C}_{2,k} - \min\{\mathbf{C}_{2,k}\}, \max\{\mathbf{C}_{2,k}\} - \mathbf{C}_{2,k}]$$

$$\mathbf{D}_{min} = \min_{rowwise} \mathbf{D}, \beta_k = \sum_{i=1}^n \mathbf{D}_{min}(i)^{-1}$$

end for

$$\beta_{k_{max}} = \max_{k=0 \rightarrow 90/\delta} \{\beta_k\}$$

Output: $\mathbf{n} = [\cos\theta_{k_{max}} \quad \sin\theta_{k_{max}}], \mathbf{C} = [\min\{\mathbf{C}_{1,k_{max}}\} \quad \min\{\mathbf{C}_{2,k_{max}}\}]$

(3) Corner Point Determination: To further reduce the computational load, the authors in [6] proposed an L-shape-fitting method based on corner point searching. As shown in Algorithm 3, the two diagonal corner points of the L-shaped cluster $P_1(x_1, y_1), P_2(x_2, y_2)$ are firstly determined as the ones located at the minimum and maximum x or y values. Afterwards, the third corner point is determined by finding the furthest point to the diagonal line drawn by P_1 and P_2 , thus a rectangle is constructed with them. This method executes much faster than the other methods; nevertheless, this method's accuracy should be proven, especially by dealing with complex cluster points projected from 3D to 2D.

Algorithm 3 Corner Point Determination

Input: 2D point cloud $\mathbf{S} = \{(x_i, y_i) | i = 1, 2, \dots, n\}$

$(P_1(x_1, y_1), P_2(x_2, y_2)) = \text{most distant}(\mathbf{S}) // \text{find diagonal points in } \mathbf{S}$

for $i = 1 \rightarrow n$ **do**

$$d(i) = \frac{|(x_2 - x_1)(y_1 - y_i) - (x_1 - x_i)(y_2 - y_1)|}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}$$

end for

$$d_{i_{max}} = \max_{i=1 \rightarrow n} d(i), P_3(x_3, y_3) = (x_{i_{max}}, y_{i_{max}})$$

Output: $P_1(x_1, y_1), P_2(x_2, y_2), P_3(x_3, y_3)$

2.2.2 Ellipse fitting

Due to diverse vessels' shapes, ellipses can better fit those vessels with elliptical hulls. The ellipse-fitting problem can be formulated as finding the optimal 2D ellipse to fit the cluster points. If an ellipse is described as

$$E(x, y) = ax^2 + bxy + cy^2 + dx + ey + f = 0$$

$$\text{with the constraint } b^2 - 4ac < 0, \quad (4)$$

where the coefficients a to f describe the ellipse. The optimization problem can be defined as finding the optimal coefficient with the minimal sum of squared error (SSE).

$$SSE_{\text{ellipse}} = \min_{a,b,c,d,e,f} \sum_{i \in S} (ax_i^2 + bx_i y_i + cy_i^2 + dx_i + ey_i + f)^2 \quad (5)$$

The optimization is solved using the method proposed in [9]. Unlike the ellipse-fitting method proposed in [5], which assumes all the objects have a complete elliptical shape, we cut the ellipse based on the maximal distance of the point clouds projected onto the major axis, as shown in Fig. 2 (b). This method avoids the problem of fitting oversized shapes in the point cloud as in [5].

2.3 Selection of fitting geometry

As shown in Fig. 1, the shape-fitting step generates the best fitting shapes (box or elliptic cylinder) using the clusters as input. Tests have shown that these two shapes can be fitted reliably to all relevant surface objects. Rectangular vessels and floating containers can be fitted in a box using L-shape fitting, whereas boats or ferries can better be represented using elliptic cylinders. Therefore, we fit the point clouds using both L-shape and ellipse fitting and choose the better-fitted shape based on the mean squared error (MSE).

We propose a method based on the MSE to choose the geometry best fitting the point cloud shape for a clustered point set. As shown in Algorithm 4, the cluster points are fitted using both L-shape fitting and ellipse fitting, in which the MSE of both methods are compared. The fitted shape with less MSE is chosen. The MSE of the chosen shape is then used in the AEKF in MOT.

Algorithm 4 Shape decision

Input: 2D point cloud $\mathcal{S} = \{(x_i, y_i) | i = 1, 2, \dots, n\}$

$$MSE_{\text{Box}} = \frac{1}{n} \sum_{i=1}^n \min \left[|x_i \cos \theta_{\text{Box}} + y_i \sin \theta_{\text{Box}} - c_{1\text{Box}}|, |-x_i \sin \theta_{\text{Box}} + y_i \cos \theta_{\text{Box}} - c_{2\text{Box}}| \right]$$

$$MSE_{\text{Ellipse}} = \frac{1}{n} SSE_{\text{ellipse}}$$

if $MSE_{\text{Ellipse}} < MSE_{\text{Rectangle}}$

Shape = ellipse

else

Shape = rectangle

end if

Output: $P_0(x_0, y_0), \theta, l, w, MSE, \text{shape}$

2.4 Multi-object tracker

We proposed a similarity comparison method based on Bhattacharyya distance in our early work for the object association [10]. After the object association, each object is registered with a unique ID and subsequently tracked using its associated AEKF. We use one AEKF for each object so that the number of tracked objects is flexible. This also allows for a more straightforward implementation as compared to one size-adaptive filter that tracks all objects. Like a standard Kalman filter, each AEKF consists of prediction and correction parts. The state vector is first predicted based on a generally rigid body model and then corrected by the most recent measurement. The measurement covariance matrix is adapted based on the current measurement quality, affected by the object's pose towards the central sensor point. The state vector \mathbf{x} consists of 9 elements:

$$\mathbf{x} = [p_x \ p_y \ v_x \ v_y \ \theta \ \dot{\theta} \ l \ w \ h]^T. \quad (6)$$

where P_x and P_y denote the object's 2D centre position, corresponding to P_0 in shape fitting; v_x and v_y denote the 2D velocity; θ and $\dot{\theta}$ denote the heading and the rate of turn; l , w and h denote the length, width and height of the objects.

The LiDAR measurements \mathbf{y} can be defined as

$$\mathbf{y} = [p_x \ p_y \ \theta \ l \ w \ h]^T. \quad (7)$$

The model used in this paper is a general nonlinear rigid body model:

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{G}_k \mathbf{w}_k, \mathbf{w}_k \in \mathbb{N}(0, \mathbf{Q}_k), \quad (8)$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k, \mathbf{v}_k \in \mathbb{N}(0, \mathbf{R}_k), \quad (9)$$

where k is the time index, \mathbf{x}_k denotes the state vector defined in (6). \mathbf{F}_k and \mathbf{G}_k are the linear matrices for state transitions and noise according to a point mass model. \mathbf{y}_k is the measurement vector and \mathbf{h}_k is the nonlinear measurement equation of \mathbf{x}_k . \mathbf{w}_k and \mathbf{v}_k are additive Gaussian noise, where \mathbf{Q}_k and \mathbf{R}_k denote the process and measurement noise covariance matrices, respectively.

As discussed in shape fitting, the quality of the object's measurement should be considered. Here, we define a 5×1 visibility vector of the object \mathbf{v} to denote the trust in the measurement \mathbf{y}_k , where the elements in \mathbf{p}_{vis} are within $[0, 1]$. Aside from the *MSE* determined by shape fitting, the visibility vector of the object \mathbf{p}_{vis} also depends on the object's relative pose:

$$\mathbf{p}_{vis} = f(p_x - p_{x_{sensor}}, p_y - p_{y_{sensor}}, \theta - \theta_{sensor}, MSE) \quad (10)$$

where $p_{x_{sensor}}$, $p_{y_{sensor}}$ and θ_{sensor} describe the ego sensor pose. The parameters $p_x - p_{x_{sensor}}$, $p_y - p_{y_{sensor}}$ and *MSE* have negative correlations with \mathbf{p}_{vis} : the visibility would decrease if the object is far away from the ego sensor, or the *MSE* is high. For specific relative heading, e.g., $\theta - \theta_{sensor} = 0^\circ$ or 90° , the visibility is also low.

$$\mathbf{R}_k = \mathbf{R}_{k0} + r(1 - \mathbf{p}_{vis})(1 - \mathbf{p}_{vis})^T \quad (11)$$

where \mathbf{R}_{k_0} is the noise coefficient matrix at full visibility, and $r \geq 0$ denotes a positive pre-defined constant factor. In the case of poor object visibility, \mathbf{R}_k is increased, and the AEKF puts more weight on the state prediction, and the measurement's impact on the estimation is consequently reduced. According to the measured noise covariance, the \mathbf{R}_{k_0} is defined as follows.

$$\mathbf{R}_{k_0} = \text{diag}\left([0.1 \ 0.1 \ 0.05 \ 0.5 \ 0.5]^2\right) \quad (12)$$

3 RESULTS

3.1 Simulation results

To validate the proposed algorithm, we have first constructed simulation scenarios using the ROS GAZEBO toolbox. Figure 3 (a) demonstrates a simulation scenario where the local USV carries a simulated Velodyne Puck VLP-16 LiDAR – the same type as in the real-world test – that can perceive the environment at . As USV model, we have used the Virtual RobotX model [11] and evaluated the performance of the presented shape-fitting methods (refer to section 2.2) with both a container and a crew boat as target objects (see Fig. 3 (a)). Figure 3 (b) shows a real-time visualization of the shape-fitting results. The point clouds are first clustered and then fitted into rectangular cubes or elliptical cylinder after conducting the selection algorithm presented in section 2.3. L-shape fitting is selected for container and ellipse fitting for the crew boat and the fisher boat. The ellipse fitting of those vessels shows a qualitative improvement than only using L-shape fitting for all the objects. In the following, we analyse the performance of the adaptive shape fitting in detail. It is expected that the positioning, size and heading accuracy should be improved using the proposed algorithms.

We have conducted the same test drive with each shape-fitting method separately to compare the performance. For evaluating the three L-shape-fitting methods, we use the container with a base area of $8.6 \text{ m} \times 3.6 \text{ m}$ as the target object since it has a rectangular shape. Based on the MSE, the automatic geometry selection (section 2.3) selects L-shape fitting as the proper method, as shown in Table 1, which presents the resulting performance. Here, we first use the intersection over union (IoU) – which is the overlapped area divided by the unioned area – to evaluate the fitted shape's alignment with the reference. To further evaluate the

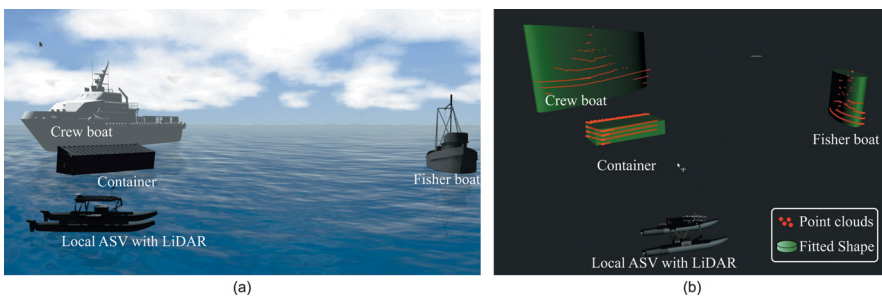


Figure 3: A simulation scenario in ROS GAZEBO. (a): the constructed scenario in GAZEBO. (b): the sampled point clouds and the corresponding fitted shapes visualized in ROS Rviz.

critical metrics, we compare the 2D positioning error, the absolute heading error, the relative size error ($\frac{|l_{measured} - l_{ref}|}{l_{ref}}$ and $\frac{|w_{measured} - w_{ref}|}{w_{ref}}$) and the runtime for one execution.

The brute force method shows the lowest IoU among L-shape methods since it requires sorted point clouds as input and is sensitive to noise. By projecting the 3D point clouds to 2D, more noises are introduced, leading to worse sorting performance and lousy brute force method performance. The rotation method has the highest IoU and best size estimation but suffers from a large heading error and a high execution. The high heading error is due to 3D projection, which can generate more than 2 lines instead of an L-shape, breaking this method’s assumption. The corner point method has the shortest execution time and shows good positioning and heading accuracy close to the rotation method’s results. The projection from 3D to 2D will not affect this method since the corner points can be kept the same in this projected input. Considering all these metrics, we chose to implement the corner point method for L-shape fitting in our algorithm because it shows good overall accuracy with a comparably low computational load.

In addition to the container, we evaluate the L-shape and ellipse fitting (section 2.2.2), and the geometry selection (section 2.3) methods on the crew boat. As opposed to the container, the 43 m × 10.4 m crew boat exhibits an elliptical shape again confirmed by the geometry selection method based on MSE. In the comparison in Table 1, the ellipse fitting overall shows the highest accuracy in all metrics combined with a low execution time. Choosing ellipses as fitting shapes for crew ships or other elliptical vessels helps improve the overall shape-fitting performance.

The shape-fitting results are used as input for the tracking algorithm (see section 2.4). To validate its performance, Fig. 4 illustrates the fisher boat’s detailed tracking result. In comparison, the performance of the AEKF with only L-shape fitting is also presented. It shows histograms of the 2D position error, absolute velocity error, heading error and size error. The mean 2D position error is around 1,5 m. More than 90 % of the positioning results have accuracy within 2 m which is higher than the intermediate result after shape fitting. From the estimated 2D velocity v_x and v_y by the AEKF, the absolute velocity is compared with the reference. The MOT overall shows a good motion profile with low velocity error. To compare with the AEKF using only L-shape fitting, the proposed solution shows a noticeable

Table 1: Performance comparison of L-shape- and ellipse-fitting methods for the container

Target object	Algorithm	IoU	Position error	Heading error	Length error	Width error	MSE	Runtime
Container	Ellipse fitting	20%	2 m	10 °	20.4 %	35.4 %	0.05 m	0.465 ms
	Brute Force	27%	1.6 m	9 °	16.3 %	52.8 %	0.025 m	0.882 ms
	Rotation	39%	1.6 m	15 °	5.8 %	22 %	0.022 m	2.36 ms
	Corner Point	35%	1.5 m	7 °	7 %	33 %	0.021 m	0.065 ms
Crew boat	Ellipse fitting	61%	2.1 m	1.8°	6%	27%	0.02 m	0.918 ms
	Brute Force	53%	2.4 m	2.1 °	13.5 %	35.6 %	0.045 m	4.722 ms
	Rotation	53%	2.4 m	19.1 °	13 %	60 %	0.053 m	8.612 ms
	Corner Point	54%	2.7 m	7 °	10.2 %	27.8 %	0.043 m	0.284 ms

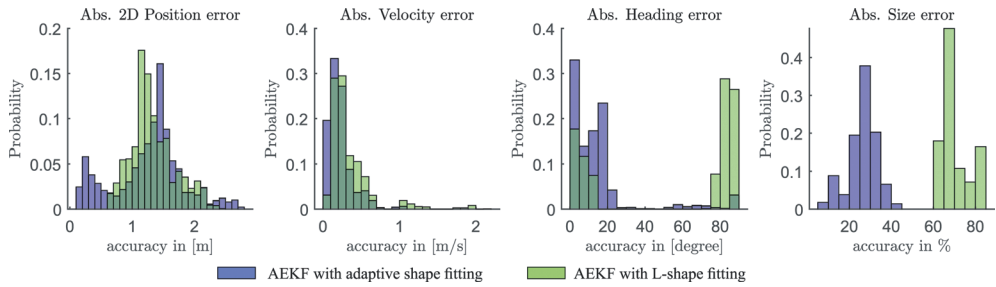


Figure 4: Detailed simulation results of the fisher boat. The tracking errors, including absolute 2D position, velocity, heading and size error, during a test drive in the simulation are determined with ground truth and plotted in a normalized histogram.

improvement in heading and size estimation. In this case, the ellipse fitting determines a stable heading using the vessel's curve hull information, whereas the heading generated from L-shape is switched orthogonally. However, the size (length \times width) is underestimated by 30 % due to the angular-discrete LiDAR measurements.

3.2 Post-processing using Real-world test data

As shown in Fig. 5, we have tested the algorithms using an USV, which is equipped with a Velodyne Puck VLP-16 LiDAR sensor at the front, a Septentrio AsteRx-3 GNSS receiver and a Litef LFK-100 inertial navigation system. The non-parallelized MOT algorithm is implemented and executed on a laptop with a 2.6 GHz Intel® Core i5 Processor. For reference purposes, we have used an elliptical rubber dinghy equipped with the Septentrio AsteRx-3 GNSS receiver. The real-time kinematic solution of the GNSS receiver with an accuracy of around 5 cm is only used as a position reference. To generate a qualitative heading reference, we assume that the rubber boat has not drifted, based on which the course over ground (COG) is equal to the heading. The COG can then be determined using the speed in x and y directions.

During the real-world test, the USV moves close to one of the main fairways in Rostock harbour. The left plot in Fig. 5 shows the positioning results of a test drive, where the USV and the reference boat are moved randomly but with high dynamic. During the test, the LiDAR can detect all the objects in view simultaneously, and the positioning results of the adaptive shape fitting are shown in green dots. Afterwards, the currently detected objects are associated with those in the past, assigned with unique IDs (presented in different colours) and tracked using AEKF.

Figure 6 presents the tracking result of the rubber dinghy compared with the reference. To evaluate the proposed adaptive filtering technique's performance, we compare it with the approach, which only uses the brute force L-shape-fitting method with AEKF. During the test, the ellipse is selected to fit the reference boat most of the time. The AEKF with the new adaptive fitting method has a good 2D positioning accuracy of around 2 m and have a better position and velocity estimation even when the LiDAR measurements are missing (at 65 sec to 75 sec). The less noisy positioning results from the adaptive shape fitting stabilize the AEKF estimation. The proposed method also shows a better performance on size and heading estimation by producing fewer errors since a better-fitted shape is chosen (mostly

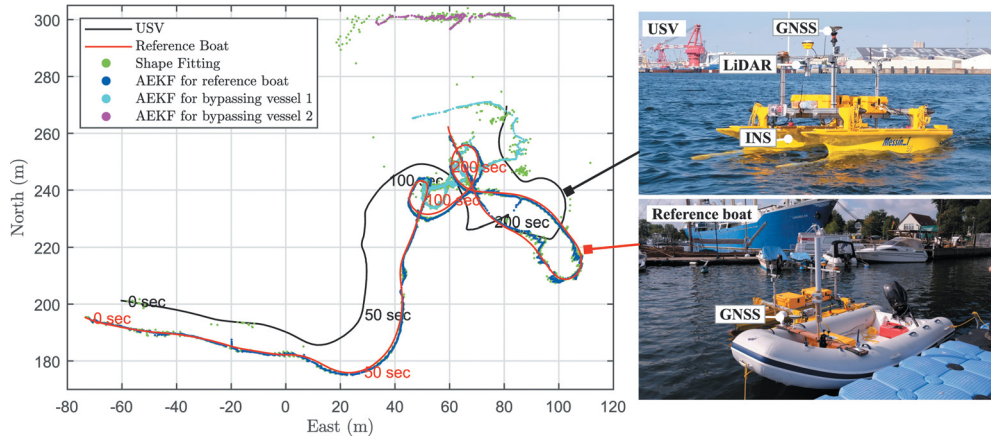


Figure 5: Experimental setup (right) and test results overview (left). The USV is equipped with LiDAR and running the MOT; a rubber dinghy is taken as a reference target for the MOT test. Both of the vessels are property of the University of Rostock [12].

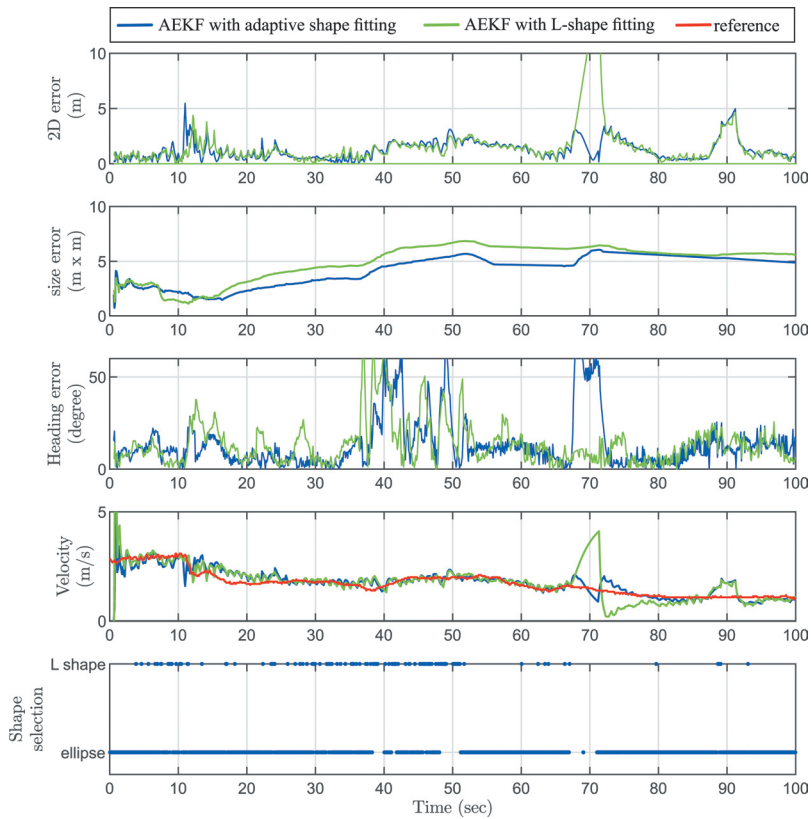


Figure 6: Results using the real-world test data: the AEKF with the adaptive shape fitting for the reference boat is shown in blue. The AEKF with L-shape fitting is shown in green.

ellipse) during the test drive. However, a large heading error is shown between 35 sec to 50 sec, where the reference boat is very close to the USV, stirring up waves and causing noisy LiDAR detection due to the spray. In such a scenario, fast boats and their spray cannot be discerned from one another by our LiDAR system. Therefore, the adaptive fitting selection frequently switches between ellipse and L-shape since both methods generate similar MSE. Since the positioning, size and velocity estimation can still be kept in high accuracy during this period, a dangerous collision can still be avoided using the detection technique. To sum up, the proposed algorithm has an overall high accuracy, achieving a positioning accuracy of around 2 m, improving the size and heading estimation accuracy by and respectively, compared with the MOT with only L-shape fitting.

4 CONCLUSION

This paper proposes a new adaptive shape-fitting method for MOT using LiDAR for maritime applications. After clustering the LiDAR point clouds, these clusters are fitted into two common geometric shapes. We have compared three L-shape-fitting methods and chose the corner point determination method for L-shape fitting due to the high accuracy and computational efficiency. To extend the shape fitting for maritime use, we have implemented ellipse and L-shape fitting and selected, for each object, the method yields the lowest MSE for each object. Afterwards, the objects are tracked using a bank of AEKF, in which the noise covariance matrix is adapted based on the measurement quality.

The proposed algorithms are validated both in simulation and real-world tests. The simulation in ROS GAZEBO verifies the algorithm regarding position, size and velocity accuracy. To further verify the real-time performance, we conducted a test drive using a USV in Rostock, Germany. In post-processing these real-world data, the algorithm can identify multiple object shapes within sensor view. The mean positioning accuracy of the tracking is around with a repetition rate. Compared with our early work, which uses only L-shape fitting for MOT, the new approach shows higher accuracy in size (improved by 16%) and heading estimation (improved by 10%). In summary, the proposed LiDAR MOT with adaptive shape fitting meets the close-field object-tracking requirements, which compensate for the drawbacks of AIS and radar solutions. In the future, we integrate the LiDAR MOT with further sensor modules for maritime application to extend the perception scope, compensating for the limited LiDAR detection range.

ACKNOWLEDGEMENTS

The joint research project GALILEOnautic 2 is supported by the German Federal Ministry for Economic Affairs and Energy (grant 50NA1808). The basis for the support is a decision by the German Bundestag.

REFERENCES

- [1] Gehrt J.-J., Zweigel R., Roy S., Bueskens C., Kurowski M., Jeinsch T., Schubert A., Gluch M., Simanski O., Pairet-Garcia E., Bruhn W., Diegel F. & D. Abel, Optimal maneuvering and control of cooperative vessels within harbors. *Journal of Physics: Conference Series*, **1357**, pp. 1-12, 2019.
- [2] Bole A.G., Wall A.D. & Norris A., Radar and ARPA manual: radar, AIS and target tracking for marine radar users, 3 ed., Elsevier, 2014.

- [3] Elkins L., Sellers D. & Monach W.R., The autonomous maritime navigation (AMN) project: field tests, autonomous and cooperative behaviors, data fusion, sensors, and vehicles. *Journal of Field Robotics*, **27**, pp. 790-818, 2010.
- [4] Han J., Cho Y., Kim J., Kim J., Son N.-S. & Kim S.Y., Autonomous collision detection and avoidance for ARAGON USV: development and field tests. *Journal of Field Robotics*, **37**, pp. 987-1002, 2020.
- [5] Ruud K., Brekke E. & Eidsvik J., LIDAR extended object tracking of a maritime vessel using an ellipsoidal contour model. *2018 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pp. 1-6, 2018.
- [6] Ye Y., Fu L. & Li B., Object detection and tracking using multi-layer laser for autonomous urban driving. *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 259-264, 2016.
- [7] Zhang X., Xu W., Dong C. & Dolan J.M., Efficient L-shape fitting for vehicle detection using laser scanners. *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 54-59, 2017.
- [8] Shen X., Pendleton S. & Ang M.H., Efficient L-shape fitting of laser scanner data for vehicle pose estimation. *2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pp. 173-178, 2015.
- [9] Halir R. & Flusser J., Numerically stable direct least squares fitting of ellipses, pp. 1-8, 1998.
- [10] Lin J., Koch L., Kurowski M., Gehrt J.-J., Abel D. & Zweigel R., Environment perception and object tracking for autonomous vehicles in a harbor scenario. *23rd IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1513-1518, 2020.
- [11] Bingham B., Aguero C., McCarrin M., Klamo J., Malia J., Allen K., Lum T., Rawson M. & Waqar R., Toward maritime robotic simulation in Gazebo. *OCEANS 2019 MTS/IEEE SEATTLE*, pp. 1-10, 2019.
- [12] Kurowski M., Roy S., Gehrt J., Damerius R., Büskens C., Abel D. & Jeinsch T., Multi-vehicle guidance, navigation and control towards autonomous ship maneuvering in confined waters. *2019 18th European Control Conference (ECC)*, pp. 2559-2564, 2019.