

## RESEARCH ARTICLE

# A featureless approach for object detection and tracking in dynamic environments

Mohammad Zohaib<sup>1</sup>, Muhammad Ahsan<sup>1</sup>, Mudassir Khan<sup>2</sup>, Jamshed Iqbal<sup>3\*</sup>

**1** National Center of Robotics and Automation (NCRA), Department of Mechatronics and Control Engineering, University of Engineering and Technology Lahore, Lahore, Pakistan, **2** Syed Babar Ali School of Science and Engineering, Lahore University of Management Sciences, Lahore, Pakistan, **3** School of Computer Science, Faculty of Science and Engineering, University of Hull, Hull, United Kingdom

\* [j.iqbal@hull.ac.uk](mailto:j.iqbal@hull.ac.uk)



## Abstract

One of the challenging problems in mobile robotics is mapping a dynamic environment for navigating robots. In order to disambiguate multiple moving obstacles, state-of-art techniques often solve some form of dynamic SLAM (Simultaneous Localization and Mapping) problem. Unfortunately, their higher computational complexity press the need for simpler and more efficient approaches suitable for real-time embedded systems. In this paper, we present a ROS-based efficient algorithm for constructing dynamic maps, which exploits the spatial-temporal locality for detecting and tracking moving objects without relying on prior knowledge of their geometrical features. A two-prong contribution of this work is as follows: first, an efficient scheme for decoding sensory data into an estimated time-varying object boundary that ultimately decides its orientation and trajectory based on the iteratively updated robot Field of View (FoV); second, lower time-complexity of updating the dynamic environment through manipulating spatial-temporal locality available in the object motion profile. Unlike existing approaches, the snapshots of the environment remain constant in the number of moving objects. We validate the efficacy of our algorithm on both V-Rep simulations and real-life experiments with a wide array of dynamic environments. We show that the algorithm accurately detects and tracks objects with a high probability as long as sensor noise is low and the speed of moving objects remains within acceptable limits.

## OPEN ACCESS

**Citation:** Zohaib M, Ahsan M, Khan M, Iqbal J (2023) A featureless approach for object detection and tracking in dynamic environments. PLoS ONE 18(1): e0280476. <https://doi.org/10.1371/journal.pone.0280476>

**Editor:** Mohd Nadhir Ab Wahab, Universiti Sains Malaysia, MALAYSIA

**Received:** January 9, 2022

**Accepted:** January 3, 2023

**Published:** January 17, 2023

**Copyright:** © 2023 Zohaib et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** All relevant data are within the paper.

**Funding:** The author(s) received no specific funding for this work.

**Competing interests:** The authors have declared that no competing interests exist.

## 1 Introduction

Recent technological advancements in several multi-disciplinary domains engage mobile robots [1–3] to accomplish diversified tasks in various complex and challenging operational scenarios, thus highlighting the importance of accurate contextual characterization of both static and dynamic parts of the environment [4]. Typical applications of robots include executing pick and place tasks in warehouses [5–7], serving in restaurants [8], agriculture [9, 10], medical [11, 12], autonomous driving [13–16], Space [17, 18] and various industrial applications [19–22]. Most of these applications require environmental information to plan and follow the desired trajectories. Efficient path planning and collision avoidance essentially depend

upon the surrounding information [23]. The critical role of environmental information can be exemplified by the unfortunate collision of the Google/Tesla self-driving car. Other self-driving car companies including Volvo, Bosch, Uber, Ford, and FiveAI have also reported similar navigation problems. By critically prioritizing human safety, future developments in mobile robotics are anticipated to minimize fatal accidents and encounters [24–26]. This is to be achieved by making the vehicle intelligent so that it can sense and handle uncertainties inherently present in the dynamic environment.

### 1.1 The challenge

The inevitable involvement of obstacles related to humans significantly elevates the complexity of profiling the dynamic environment for the navigating robot. However, if a robot has environmental information, including the position of static and dynamic obstacles, it can generate a safe trajectory. Global path planning allows the robot to generate a time-invariant path based on the position of static objects. However, the presence of dynamic objects requires run-time obstacle avoidance strategies through local path planning [27]. Moreover, with experience, the robot learns the nature of environmental changes and generates several possible local paths with respect to uncertainties, thus highlighting the significance of static and dynamic information [28]. Based on accurate surrounding information for autonomous navigation in unknown environments, the ultimate goal is to efficiently use sensory percepts to separate static and dynamic objects.

The critical problem is to extract the dynamic part of the environment from sensory data, similar to tracking objects in an unknown environment. In computer vision, object identification methods are classified into four categories; motion-based, model-based, appearance-based, and feature-based [29–32]. However, in the case of laser data, the appearance-based method may not be applied directly due to limited information. Model and feature-based approaches require prior knowledge, i.e., features and dimensions of the objects present in an environment. The dimensions can be predicted by estimating keypoints on the objects representing the geometrical features [33–35]. On the other hand, no prior information is needed for motion-based methods [36, 37]; these find application in urban environments wherein model-based techniques may fail against unforeseen scenarios. This may be considered as an advantage of the motion-based methods in comparison with the model-based counterparts.

### 1.2 Direction of the research

Considering the pivotal role of environmental information in autonomous navigation and the shortcomings mentioned above in related schemes, the present research aims to propose an algorithm for obstacle detection and tracking [23]. It should be able to identify static and dynamic obstacles separately while ensuring reliability. We adopt a motion-based approach that deploys a LiDAR sensor to track moving objects. Several dynamic environments are created and simulated in V-Rep. The algorithm is implemented in Robot Operating System (ROS) Melodic Morenia in Ubuntu version 18.04 environment on i7 laptop with 16 GB RAM. A 2D Sweep LiDAR sensor is used for real-world experiments.

The remaining paper is structured as follows. Section 2 presents the literature on existing approaches for object detection and tracking. The proposed algorithm for the separation and tracking of static and dynamic objects is described in Section 3. For validation, various scenarios are tested to verify the accuracy of the proposed algorithm. Simulation and experimental results demonstrating successful tracking of objects are presented in Section 4. Section 5 concludes the manuscript with potential directions for future work.

## 2 Literature review

The scientific community reports several techniques to solve the problem highlighted in Section 1.1. Schulz et al. presented an approach for human movement detection in an office environment using feature-based techniques [38]. The approach is suitable for the indoor environment. In contrast, an outdoor environment poses difficulty in feature extraction due to several similar objects in the same surroundings, i.e., a tree can be recognized as a pedestrian. Also, the proposed technique can only be used for objects with specific shapes. Some of the other detection methods generate an occupancy grid map and then classify the nature of cells as static and dynamic [39]. Generally, such techniques suffer from problems like wrong velocity estimation, occlusions, and corrupted object silhouettes. Wang et al. also utilized occupancy grid maps for moving objects detection [40]. However, they adopted the background subtraction method to suitably monitor the motion of an object. However, the proposed method is incapable of detecting the static or dynamic nature of the first detected obstacle. This technique is also known as scan-matching that compares two consecutive scans after best alignment [41, 42]. Another work characterizing the nature of objects as a separate static map and a dynamic map is reported in [43]. However, it is not obvious how the objects have been extracted from the dynamic maps. Petrovskaya and Thrun proposed a model-based approach to detect moving objects. This research uses pre-modelled objects and detects these using the polar coordinate and scan differencing method. The presented method detects vehicles; however, it does not recognize unknown objects like pedestrians, bicyclists, etc. [44]. A similar model-based approach is proposed by Vu & Aycard in [45]. They introduced several fixed models that can be used to classify pedestrians, bikes, cars, or buses. Simultaneously Localization And Mapping (SLAM) is considered a solved problem for static environment [46]. However, it has not been fully explored for dynamic environments.

Several motion-based tracking algorithms have been under consideration that utilize Artificial Intelligence (AI) inspired techniques. For example, the Kalman or particle filter can be used in order to estimate an obstacle's position [47]. A probabilistic method, the Bayesian network can be used to generate two occupancy grid maps for presenting static ( $p(S^t|o^1 \dots o^t)$ ) and dynamic ( $p(D^t|o^1 \dots o^t)$ ) objects separately [48]. Where  $S^t$  and  $D^t$  represent static and dynamic maps at time  $t$ , respectively. The  $o^i$  is the  $i^{th}$  observation. The method is suitable for object separation; however, it does not maintain any history of the previous occupancy of the cells in the dynamic map ( $D$ ). The information in the map only needs to be set in order to represent the current occupancy of each cell.

## 3 Research methodology

Keeping in view the aforementioned literature, we are interested in a motion-based approach for object detection using only 2D laser data containing the depth of the obstacles. Inspired from the Bayesian method [48], the presented research introduces an approach to separate and track moving objects. In order to address the limitations of the method [48], our methodology keeps a history of the objects from the beginning. An object detected by the LiDAR sensor consists of several data points. The monitoring (comparison, detection, or identification) of the object using data points is an intensive task [49]. Therefore, a suitable option is to create a cluster of nearest data points, such that each cluster represents a separate object. Consequently, we need to monitor only a cluster in order to track an object. It significantly reduces the complexity and increases the efficiency of the algorithm. For this purpose, sensory information is transformed from laser data (in the local frame) to Point Cloud XYZ (PC-XYZ) in the global frame. The conversion is achieved by calling different ROS nodes. The converted data points are then clustered using a Euclidean distance based approach. The approach, with the help of

parameters (minimum/maximum points for clustering and tolerance), defines the environmental objects. Where, the tolerance is the radius for the k-NN searching, used to find the candidate points for a cluster. All the points within the radius are clustered and considered as a single object. We selected the radius based on several trials. We observed that in the case of a large radius, multiple objects are considered as a single object, whereas, for a small radius, a single object is considered as multiple objects. We found that 5cm is an optimal radius for our experiments. It can be updated with respect to the environment—the size of the objects, how close they are to each other, etc. Moreover, We consider minimum 15 and maximum 5000 data points. Data points fewer than 15 are considered as noise and are discarded. However, the upper limit can be updated corresponding to the dimension of an object.

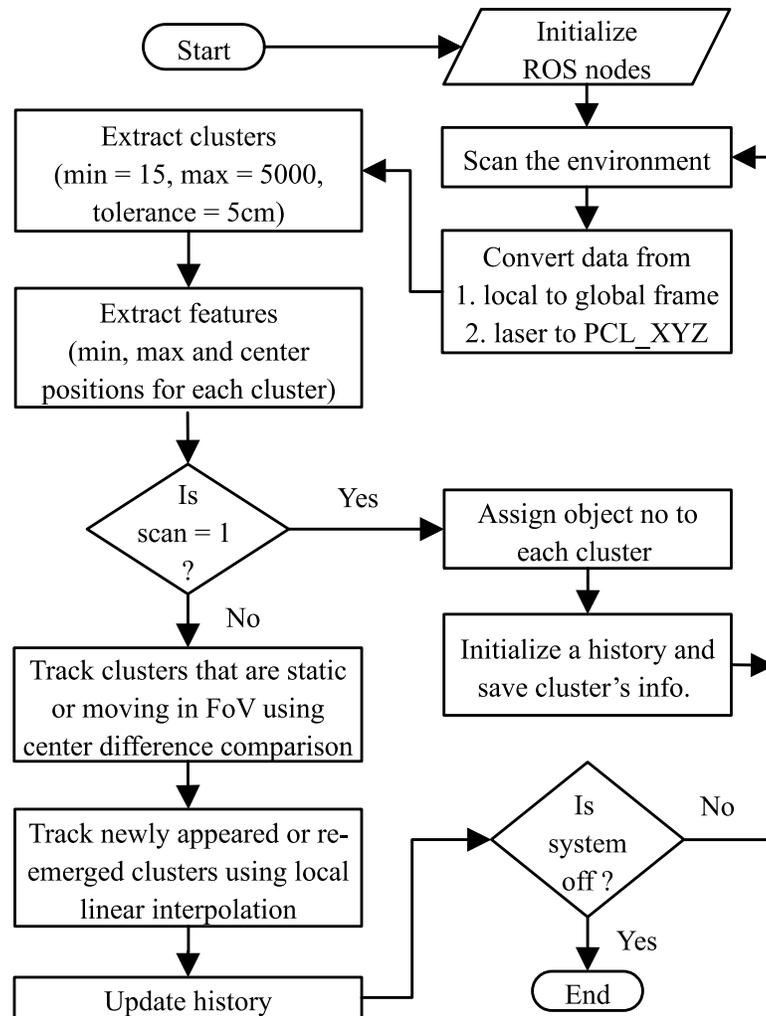
After object detection, the next vital step is to identify these and track their movement. Object identification is achieved by finding the dimensions (center, minimum, and maximum points) of each cluster. We utilize the cluster's parameters for calculating the average in order to get the required dimensions with the help of the Point Cloud Library (PCL) feature extractor. We create a boundary box around every cluster for ease in visualization. The parameters of the cluster are stored in the history along with the cluster number and its status (i.e., moving, static, or outside the Field of View (FoV)). Initially, all the detected objects are considered as static and are stored in the history with random object numbers. However, in corresponding scans, the objects are tracked as static or dynamic using center difference comparison. Whereas the objects re-emerging from outside FoV are estimated by incorporating local linear regression. The process of objects detection, identification, and tracking in a dynamic environment is illustrated in Fig 1.

The first step is related to the identification of the objects that are static or moving in FoV. This is achieved by defining two thresholds (static and dynamic) around each detected object. Fig 2(a) illustrates the thresholds for a square-shaped object. It can be observed that the static threshold is too small with respect to the size of the object. Therefore, it minimizes the incorrect estimation, i.e., two objects cannot exist inside the static threshold. Similarly, it also endures the sensory noise, i.e., the center point of a cluster can vary within the region. The dynamic threshold is selected to be greater than the rotational radius of objects with an assumption that the objects are of the same size. However, it can be adjusted by considering the object's size and speed. So, it is large in size for speedy objects and short for slow-moving objects. If  $R_{obj}$  is the object's rotational radius,  $V_{obj}$  is its velocity and  $W_{obj}$  is the object's width, the radius of the dynamic threshold ( $R_{dy_{threshold}}$ ) can be determined by (Eq 1).

$$R_{dy_{threshold}} = \frac{V_{obj} \times 3W_{obj} + R_{obj}}{3W_{obj}}. \quad (1)$$

Considering the limitations of a physical environment, it is assumed that  $R_{dy_{threshold}}$  can vary from static threshold to  $3 \times W_{obj}$ . The objects detected in the new scan are considered one by one for comparison with all the objects available in the history, as shown in Fig 2(b). The top and bottom rows, presenting three objects, correspond to the new scan and history, respectively. The  $d_{ij}$  in the figure is the measured distance of the detected object  $i$  from object  $j$  of history.

If the distance  $d_{ij}$  lies within the static threshold, the object  $i$  is considered as static object  $j$ . This presents a slight change in the object's position. Likewise, if the calculated minimum distance is observed between static and dynamic thresholds, the object is considered as dynamic.



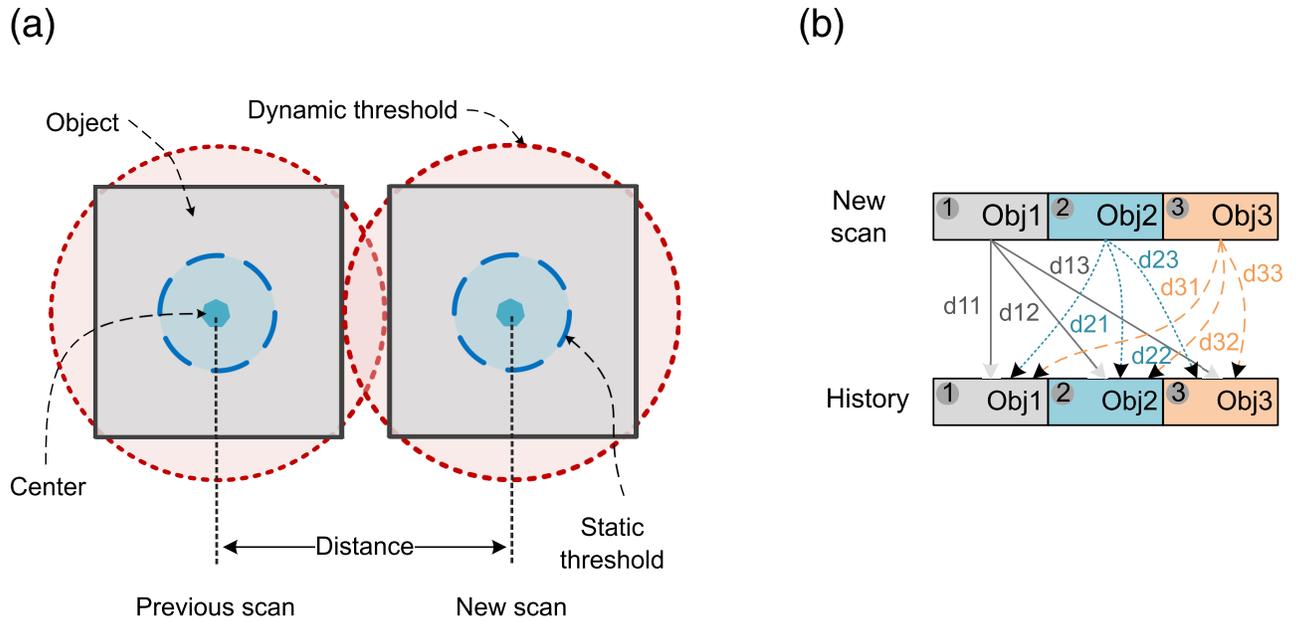
**Fig 1. Flowchart of the proposed approach presenting a step-by-step process of tracking the objects by maintaining the history.**

<https://doi.org/10.1371/journal.pone.0280476.g001>

These detected static and dynamic objects are saved in the history with appropriate original objects. At this stage, history is updated for the tracked static and moving objects only.

In the second step, we consider the objects that disappeared in the last scan(s) and are re-emerged in the FoV (occlusion scenario). These objects and any other newly available untracked objects are estimated using local linear regression as illustrated in Fig 3.

The arrows emerging from the robot represent the sensor's FoV. A wall in front of the robot limits the FoV. The tiny dots on the robot's left side symbolize the positions of an object saved in history. The dots enclosed by the circle shown on the left side of the wall are the recently observed positions of the object. The scenario illustrates that the object is moving from left to right towards the wall. After some scans, the object is observed very close to the wall, and finally, it moves behind the wall. In these scans, the robot cannot detect the object as it is out of its vicinity. The status of the object in the history is set to an unobservable object (hidden state). After some scans, the same object appears on the right side of the wall. At this stage, the approach uses the previous ten histories of the object and calculates the local linear



**Fig 2. Identification of static and moving objects inside FoV.**

<https://doi.org/10.1371/journal.pone.0280476.g002>

regression line using (Eq 2).

$$[m, c] = (x^T \cdot x)^{-1} \times x^T \cdot y$$

$$x = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ \vdots & \vdots \\ x_{10} & 1 \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{10} \end{bmatrix} \tag{2}$$

Where  $x$  and  $y$  represent the positions of the object as per the last ten histories. The  $m$  and  $c$  are the slope and  $y$ -intercept of the regression line, respectively. After approximation, a tangent distance of the object's new position from the line is measured. If the distance is within the error threshold, the object is considered as the corrected estimated object and is mapped to the previous original object in the history. The error threshold is set equal to the static threshold. If multiple objects move outside the FoV in a single scan, then a separate regression line is calculated corresponding to each object. On reappearance, the distance of each object is measured with every regression line one by one. If an object is found comparatively closest to the regression line with a distance below the error threshold, it is considered as the corresponding line object. However, if an object corresponds to more than one line, it is skipped and the next scans are awaited since if the object is dynamic, it will move and consequently will provide a better clue for its estimation. However, if an object appears to be static in several consecutive scans, it is considered as a newly appeared static object.

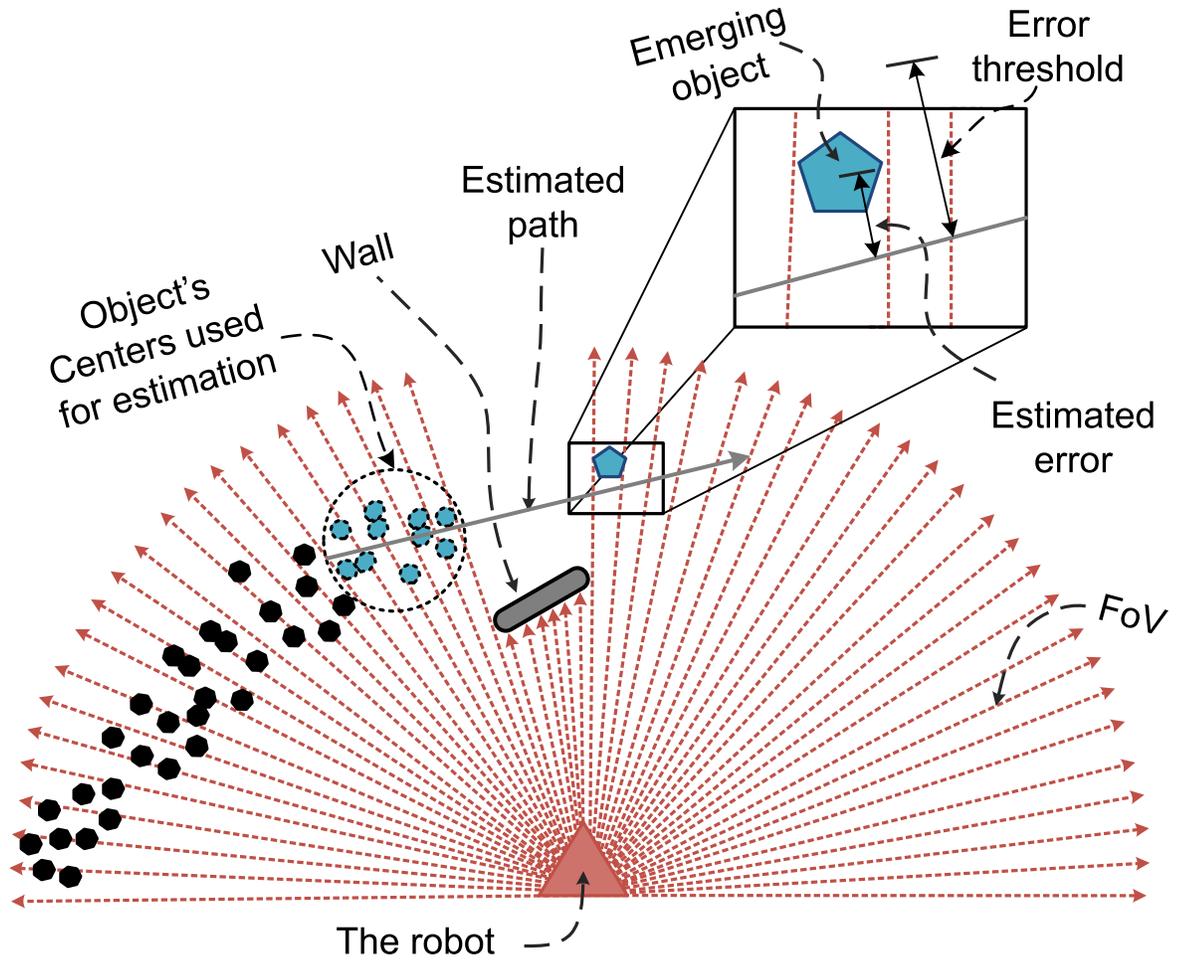


Fig 3. Estimation of objects using local linear regression.

<https://doi.org/10.1371/journal.pone.0280476.g003>

In the third step, we focus on the objects that do not lie in the region of any regression line. Such objects are also considered as newly detected objects and are added directly to the history as new entries. Table 1 presents the process of tracking those objects that are observed outside the FoV in some scans using estimated regression lines. A and B represent the number of nearer and far lines, respectively.

The history contains objects ( $Obj_i$ ) with respect to scan numbers (scan  $j$ ). Where each object has its unique number, status (static or dynamic), center, minimum and maximum point. Although the objects are not tracked in sequence, they are appropriately mapped in the history. The simulated results of the algorithm are discussed in Section 4.

Table 1. Object tracking using local linear regression.

| A  | B        | Action for objects                   | Remarks  |
|----|----------|--------------------------------------|--|
| 0  | 0        | Skip                                 | The object must be either near or far from any line          |
| 0  | 1 or > 1 | Add in history as new entry          | Newly appeared object is found                               |
| 1  | x        | Add in history with estimated object | Object is correctly found and updated in history as original |
| >1 | x        | Skip                                 | Wait for next scans  |

<https://doi.org/10.1371/journal.pone.0280476.t001>

## 4 Results

This section validates the proposed approach by presenting accurate results in different scenarios. One of the test cases is inspired from a real-world example of an agent tracking a flock of birds. The goal is to achieve approximately zero relative velocity of the agent w.r.t. the flock. Analogously, the agent robot detects, tracks and follows the remaining robots in the environment. To validate the approach in a real-world environment, experiments are conducted in which a human is moving in a room in an irregular path. Finally, by considering the selected thresholds, boundaries of the proposed approach are discussed w.r.t. the speed and turns taken by the objects inside or outside FoV.

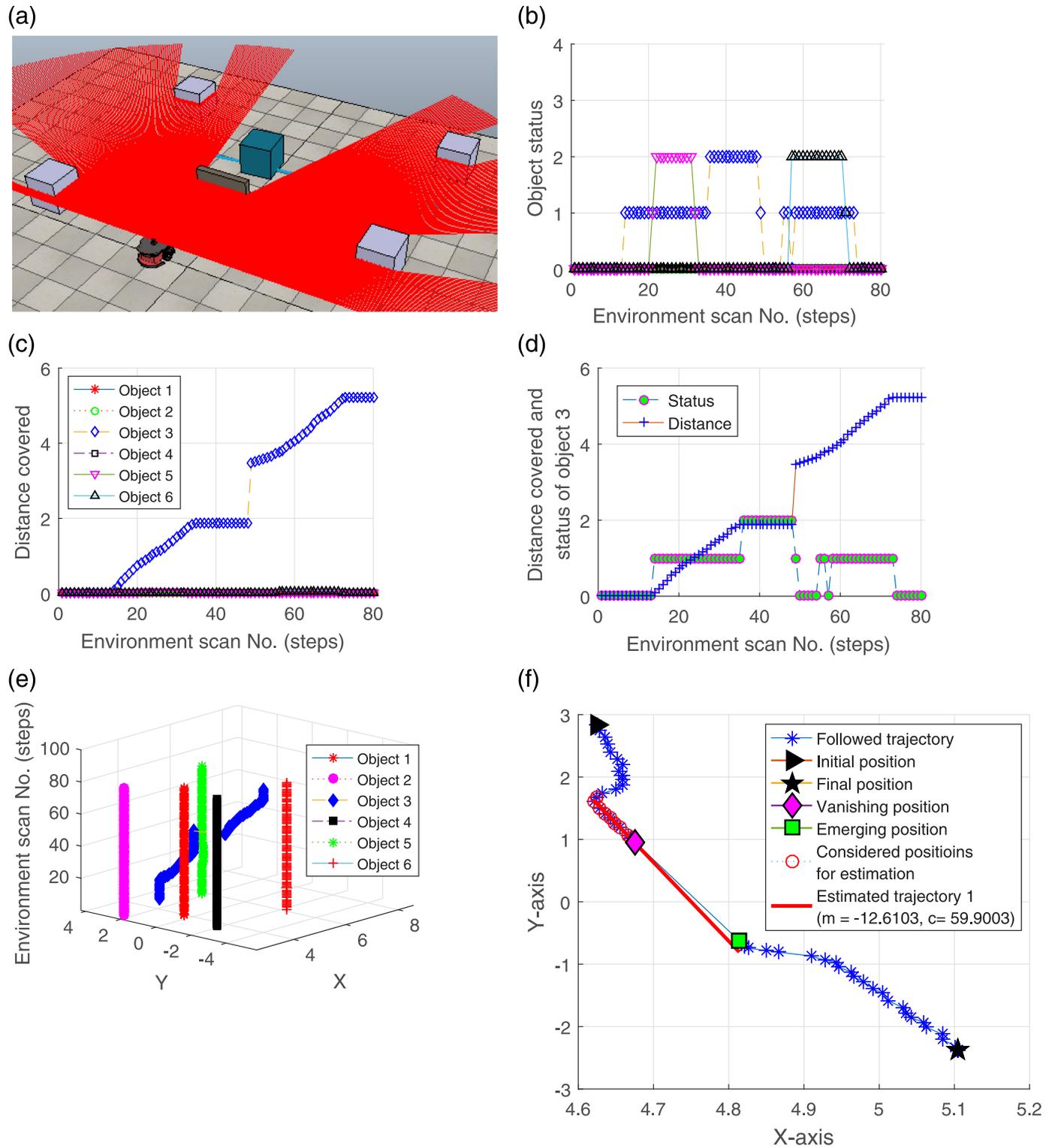
### 4.1 Simulation results

The simulated environments with different configurations are created by considering static and dynamic objects where the dynamic objects follow linear, angular or irregular trajectories. The results show that the hidden objects are correctly tracked after being visible to the robot.

**4.1.1 Environment having a dynamic object hidden from the robot's FoV.** An environment is created by introducing a wall between the robot and the dynamic object. The object starts moving inside FoV before being occluded by the wall. The robot successfully tracks it using the local linear regression on its reappearance. The environment highlighting the occlusion scenario of a dynamic object in the presence of multiple static objects is illustrated in Fig 4(a). The status of the objects and the distance covered by them from their emerging initial positions are illustrated in Fig 4(b) and 4(c), respectively. In Fig 4(b), 0 on the y-axis indicates static, 1 represents moving, and 2 shows the invisible state of the objects. It can be visualized from the status plot that three objects (3, 5 and 6) are invisible to the robot in different scans. However, Fig 4(c) validates that only object 3 covers some distance and objects 5 and 6 remain stationary throughout the scans. They are hidden in some scans due to the movement of the dynamic object (Obj 3). The rest of the objects (1, 2 and 4) are detected as static. The tracking performance of object 3 is illustrated in Fig 4(d). It is clear that after being invisible in scan 35, the object re-emerges in scan 48. A 3D view of the trajectories followed by the objects is depicted in Fig 4(e). The gap in the trajectory of object 3 highlights that the object information is not present in the history profile and it is in the invisible states (i.e. 35 to 48). Similarly, the connectivity line also validates the tracking of object 3. The approximation of the line and tracking of the dynamic object (as detailed in Section 3) is depicted in Fig 4(f). The vanishing and emerging positions of the object are illustrated. The approach estimates object 3 by considering its previous history and tracks it correctly. After reaching the final position, the object becomes static, which can be noticed in the figure.

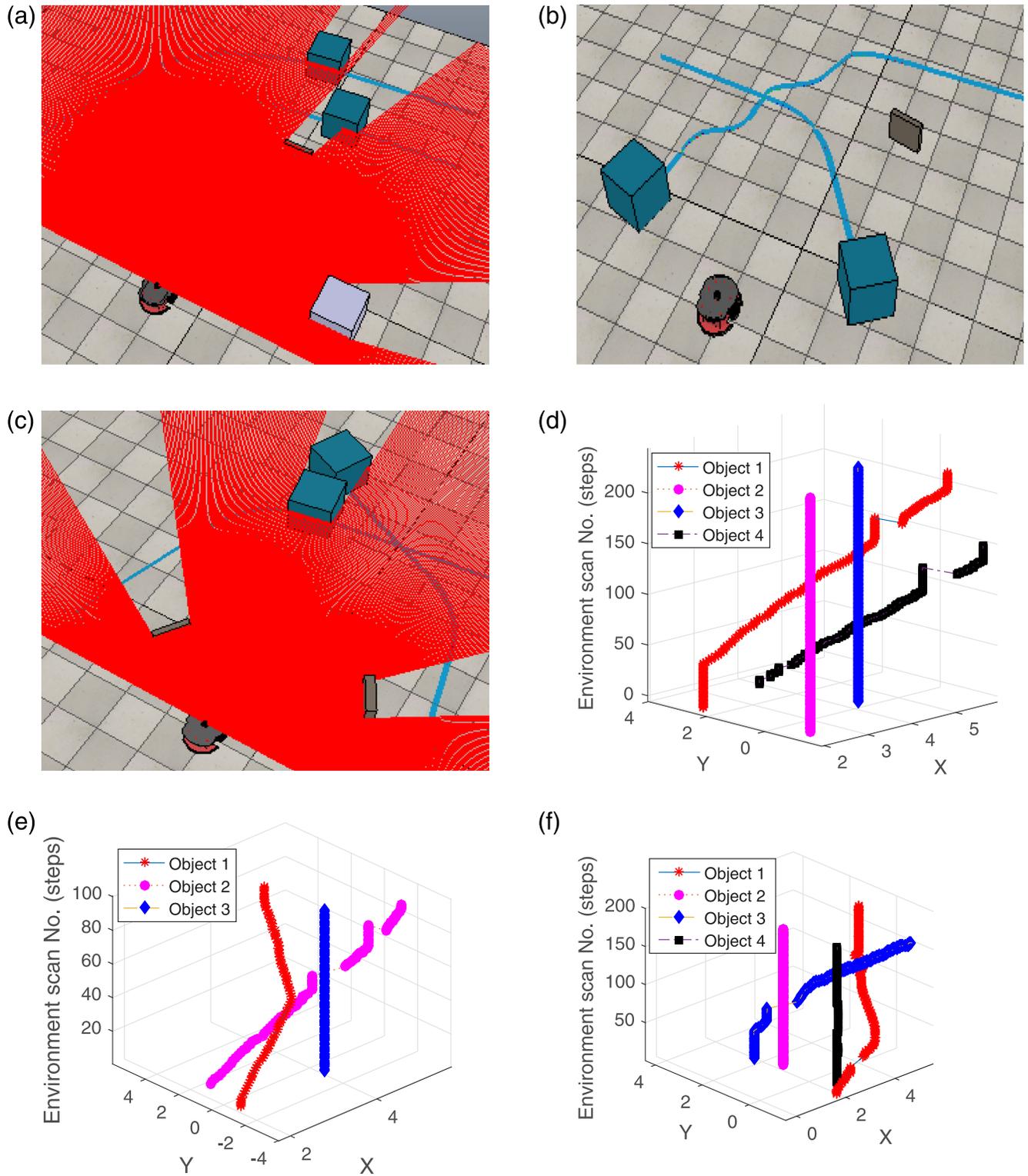
The results verify that the dynamic object is successfully separated and tracked on its re-emergence in FoV after being hidden in the presence of multiple stationary objects. Although, due to occlusion, some objects are detected at different positions after some time, however, the proposed approach follows the objects previous history instead of considering these to be new objects, thus highlighting its novelty.

**4.1.2 Environments containing objects with nonlinear, irregular and parallel trajectories.** Three different scenarios are discussed here. Fig 5(a) illustrates a nonlinear parallel motion of the two objects in the presence of a static object and a wall. Initially, three objects are detected, i.e., a dynamic object (Obj 1), a wall and a static object. However, a second dynamic object goes hiding behind the first dynamic object. After some time, due to the long trajectory, the second dynamic object becomes visible to the robot and is considered as object 4. During the locomotion, both the dynamic objects disappeared due to the wall and are then re-emerged in a very close span. By considering the previous history of the objects, object 1



**Fig 4. An experiment presenting tracking of a dynamic object in occlusion case.**

<https://doi.org/10.1371/journal.pone.0280476.g004>

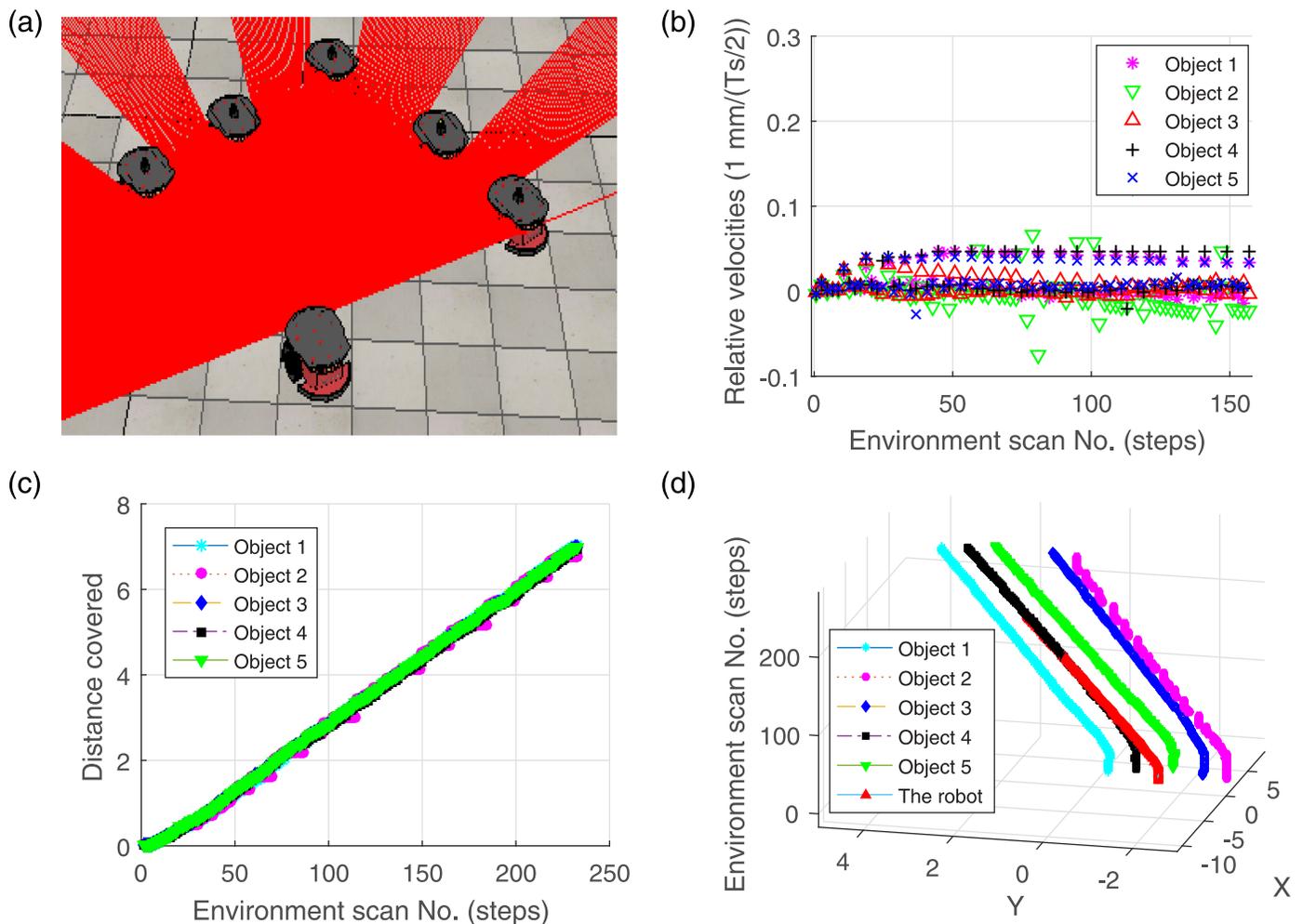


**Fig 5. Accurate tracking of objects following nonlinear, irregular, parallel and crossing trajectories.**

<https://doi.org/10.1371/journal.pone.0280476.g005>

and 4 are correctly estimated after their re-emergence in the robot FoV as highlighted by the trajectories in the 3D plot (Fig 5(d)). Fig 5(b) shows objects exhibiting irregular and intersecting trajectories. The corresponding result of the objects tracking is illustrated in Fig 5(e). Both objects are visible to the robot. One object moves in a nonlinear fashion while the other follows an irregular trajectory in the beginning. Both objects are detected and tracked accurately. Another similar scenario is illustrated in Fig 5(c), where two objects pass each other within close proximity. Both the objects are detected only once. Object 1 disappears twice; however, it is found to be stationary after the second emergence. Therefore, the second estimation is not required. A 3D view of the followed trajectories presenting accurate tracking is depicted in Fig 5(f). The objects in all the environments are successfully tracked accurately, demonstrating the algorithm’s robustness.

**4.1.3 Special scenario—Following birds flock.** A scenario “following the birds flock” further highlights the accuracy of the approach by considering zero relative velocity between bird robots and an agent robot. In Fig 6(a), the agent robot moves but remains stationary with respect to the birds flock. The velocities are compared in Fig 6(b). The estimated distance covered by the bird robots and trajectories followed by the birds and the agent robot are presented in Fig 6(c) and 6(d), respectively.



**Fig 6. Validation results in the case of birds flock scenario.**

<https://doi.org/10.1371/journal.pone.0280476.g006>

The above results demonstrate that the proposed algorithm is potentially helpful in the tracking of parallel moving robots, swarm robots, collaborative robots [50] etc.

## 4.2 Experimental results

After successfully demonstrating the efficacy of our algorithm in the simulation environment, we now validate its real-life applications through experimental results. The experiment setup contains a Scansweep LiDAR sensor integrated with a four-wheel robot. The sensor is capable of scanning in  $360^\circ$  with a 1cm resolution and has a default sampling rate of 500Hz (which can be increased to 1075Hz). It can detect objects within the 40m range. The detected walls and a human walking randomly in a room are shown in Fig 7(a)–7(l). It can be observed that in most cases, the human is detected by the LiDAR sensor. However, due to hardware imperfections and sensor performance limitations, it occasionally fails to capture human being full motion (Fig 7(k) and 7(l)). Overall, experimental results establish that the algorithm not only correctly distinguishes between the stationary objects i.e., walls and the moving object i.e., the walking person, but also accurately tracks the latter.

The status and distance covered by the human are illustrated in Fig 8(a) and 8(b), respectively. The static position is indicated with '0' status on the y-axis. The status '2' shows that the object is invisible to the robot. Most of the time, the status remains '1', indicating human locomotion. Another performance metric is an estimated location. The Fig 8(c) attributes motion to walking human only; the walls are treated as stationary objects. The conceived trajectories of the human and the walls are depicted in Fig 8(d). Note that at certain locations (Fig 7(e) and 7(j)), the algorithm may fail to unhide occluded portions of the black wall due to concealing foreground (e.g. the human) and sensor noise.

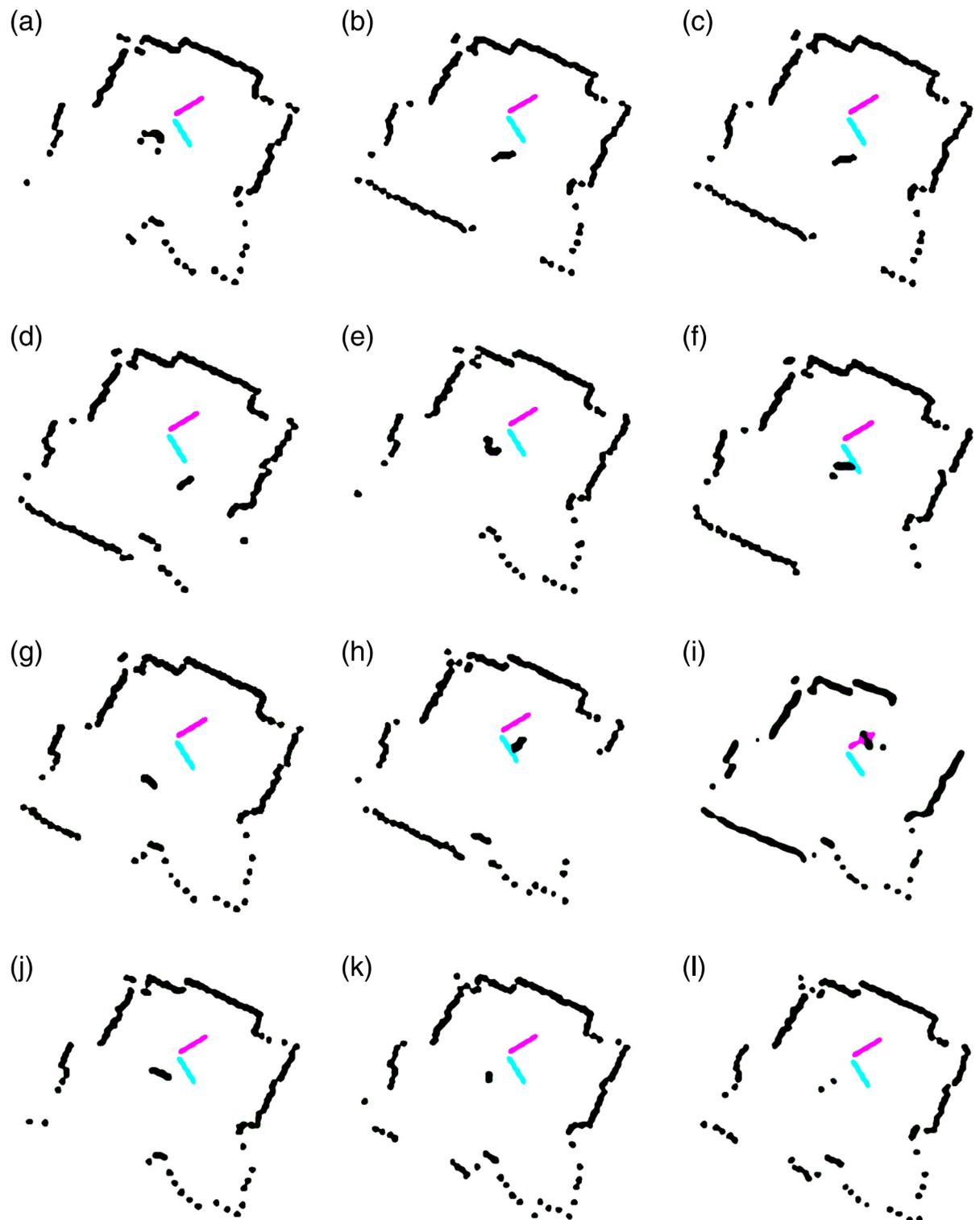
Within the constraints of real environment, experiments were conducted in several indoor arenas (dark rooms, lounges, corridors) as well as in appropriate outdoor spaces. The algorithm remains successful in detecting, separating and tracking the environmental objects, as evident in Figs 7 and 8. These results exhibit the validity of the proposed algorithm in real-world scenarios.

## 4.3 Algorithm optimal performance

The results described in Sec. 4.1 and 4.2 consider different scenarios, environments, motion patterns of objects, trajectories to be followed, etc. This section explores the limiting cases of the proposed approach. We select suitable parameter values for the simulations; the static threshold is 0.02m, the dynamic threshold is 0.4m, and the size of objects is tested from 0.2m to 2m with the sensor scanning interval of 0.002 sec.

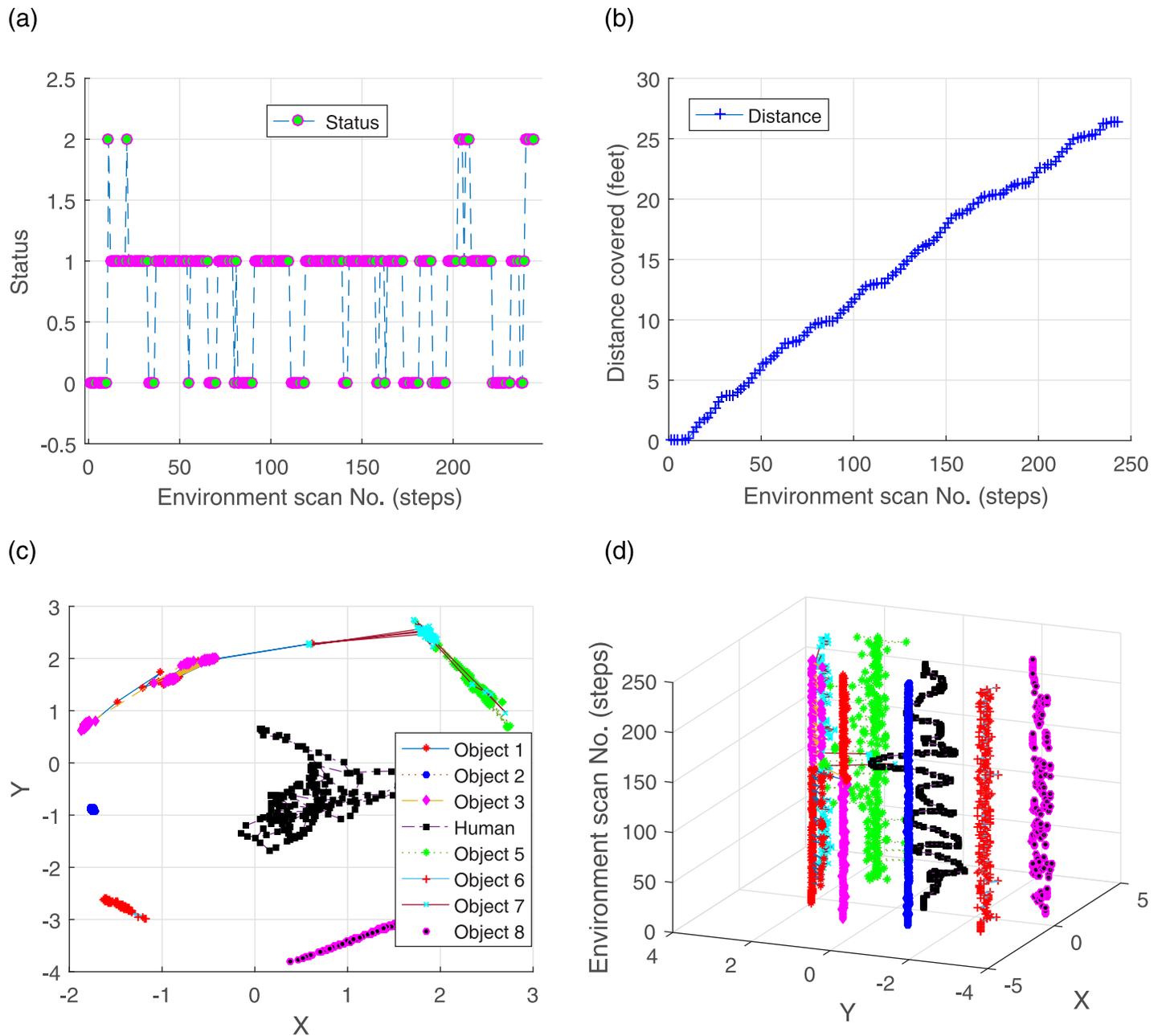
Due to the predefined thresholds, the algorithm better tracks objects moving in FoV with speed falling in a specific range. The accuracy may decrease for objects turning away from FoV. It is because our approach is feature agnostic, to mean that it does not recognize objects by their dimensions or other geometrical features. The size of objects is also restricted by the predefined clusters. An object containing less than 15 data points is simply discounted as sensor noise. On the other hand, 5000 data points are decomposed into multiple objects. Nevertheless, these thresholds are set according to the given environment.

Fig 9(a) illustrates the best detection accuracy w.r.t. the object speed during its motion inside FoV. The speed of the objects is presented in  $1 \text{ mm}/(T_s/2)$ , where  $T_s$  represents the sensor sampling time set to 0.002 sec. The figure shows that when the speed of an object is less than  $1 \text{ mm}/(T_s/2)$ , it is correctly detected as an original object, but incorrectly detected as static. For speed range 1 to  $2.9 \text{ mm}/(T_s/2)$ , the approach successfully detects as well as accurately tracks the speed of an object. Therefore, this region is named as the sweet speed region.



**Fig 7. Human moving in a room (a to l).**

<https://doi.org/10.1371/journal.pone.0280476.g007>



**Fig 8. Tracking of a human moving inside a room.**

<https://doi.org/10.1371/journal.pone.0280476.g008>

When the speed exceeds  $2.9 \text{ mm}/(T_s/2)$ , the dynamic threshold is eclipsed and the object is treated as multiple objects. Note that tracking is only possible as long as an object remains identifiable.

Fig 9(b) deals with the objects returning to the FoV after a brief absence. In this case, detection and tracking are also speed-dependent. Therefore, if an object moves with a speed lower than  $0.4 \text{ mm}/(T_s/2)$ , the algorithm may mistake it for a new object or fail to treat it as a dynamic object. For speed range  $0.4$  to  $1 \text{ mm}/(T_s/2)$ , the approach considers dynamic objects as static upon reappearance, unless the object label was initialized to *dynamic*. Based on these

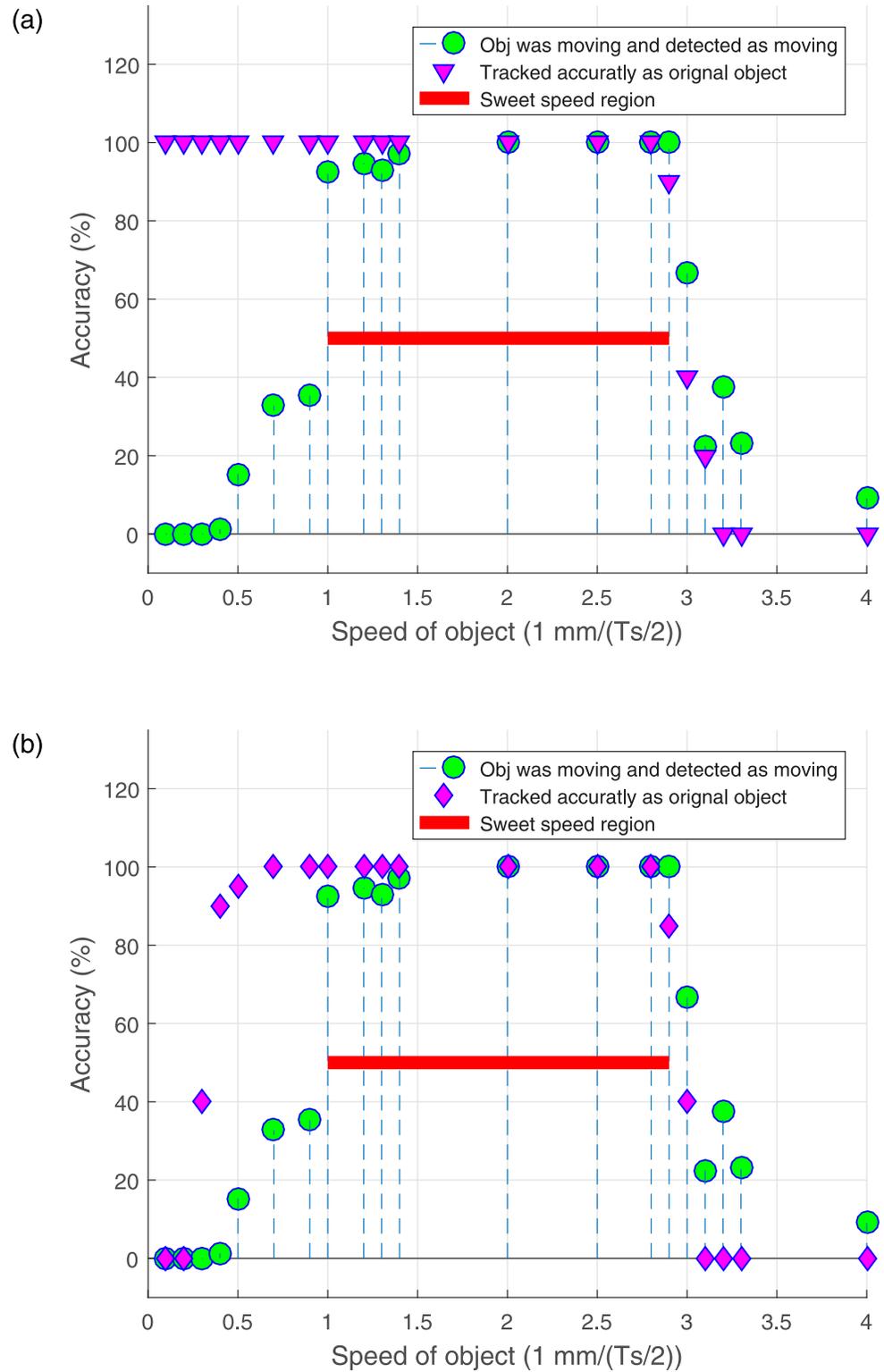


Fig 9. Boundaries of the approach.

<https://doi.org/10.1371/journal.pone.0280476.g009>

observations, we establish that the sweet speed region is 1 to 2.9 mm/(Ts/2), as highlighted in Fig 9(b) with a solid line. At speed greater or equal to 3 mm/(Ts/2), the objects are not detectable, therefore untraceable.

Turning angle also tends to limit the algorithm's capability of tracking objects following the curved paths. We observe that when the objects turn inside FoV, they are always accurately traceable. Accuracy declines when objects turn outside from FoV. In this experiment, the speed of the objects is set to 2 mm/(Ts/2). When objects take sharp turns angling 70° to 110° outside FoV, they are not correctly tracked. The geometry and dimensions of the obstacles, portion of an object falling outside FoV, irregular motion pattern before vanishing and other similar complex scenarios further hinder detection and tracking. The performance is restored when objects turn outside FoV at an angle range 0° to 79° or 111° to 180°.

## 5 Conclusions and discussion

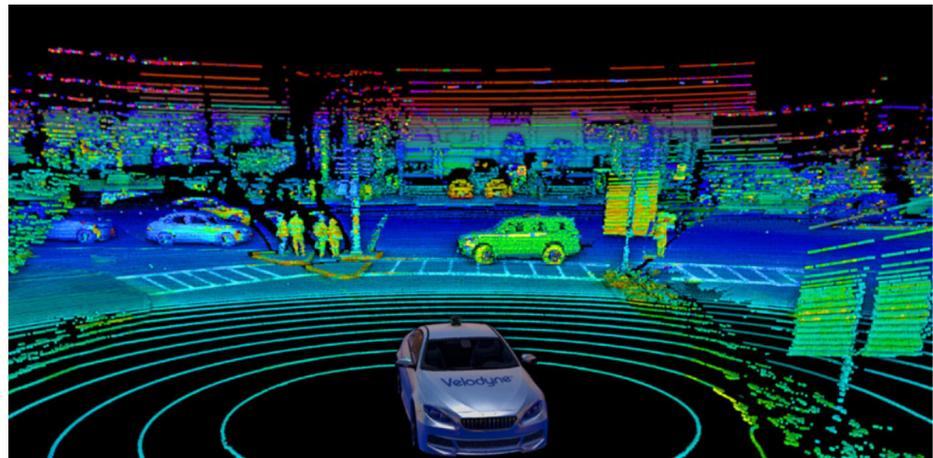
The manuscript presents an algorithm for the detection, separation, and tracking of static and dynamic objects in an unknown environment. Our approach adopts a featureless scheme based on a 2D LiDAR sensor that decodes sensory data into likely time-varying boundaries of objects for their detection and tracking. It identifies the objects those are disappeared from the robot's FoV and are reemerged at different locations. This is achieved by applying the local linear regression to the previously maintained histories of the objects. An overview of the proposed approach is summarized as follows.

Initially, sensory information is converted from data points to clusters using Euclidean distance based clustering. Each cluster represents a separate detected object. With the help of these clusters, the approach extracts the dimensions of each object by utilizing PCL. It includes center, minimum and maximum positions. The extracted parameters for each scan, saved in a history, play a vital role in tracking an object by iteratively updating its status (static or dynamic). Initially, the status of all objects is set to static while their identifiers are randomly chosen and the history is populated with the initial clusters. These variables are updated in the subsequent scans, wherein the objects are tracked in two steps. In the first step, objects which are static and moving inside FoV are tracked by comparing the Euclidean distance between the center points in consecutive scans. However, the objects which are newly detected or reappearing in FoV, are detected with the help of local linear regression in the second step. After some environmental scans, the approach accurately separates static and dynamic objects by tracing their history data.

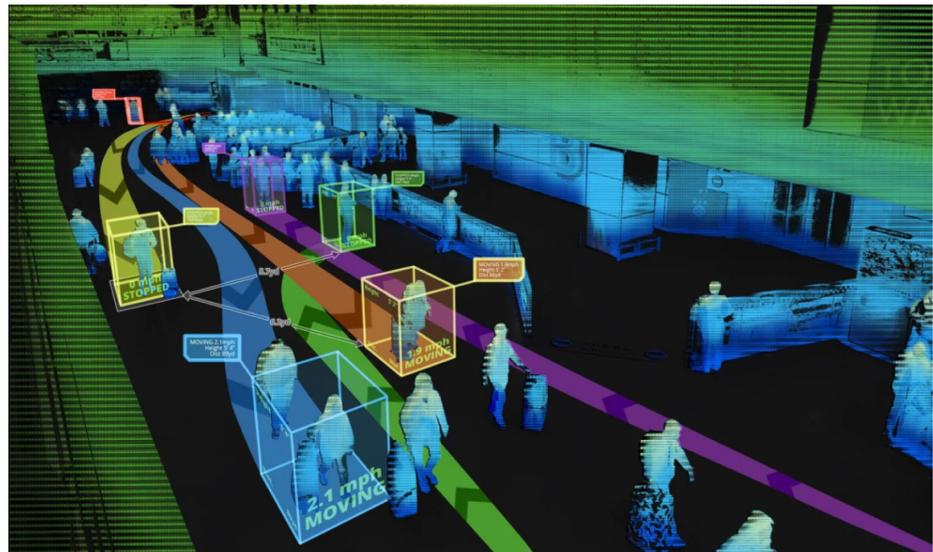
Various V-Rep simulated environments quantify the performance of the algorithm. We assume that the sensory information is accurate and the robot parameters are obtained from the V-Rep built-in functions. Sensory data is transformed to a global frame for calculations. Robot Operating System (ROS) implements control routines for the robot. We also test the algorithm in real-world experimental environments and find that it outputs an accurate estimation of moving objects in the indoor arenas.

An interesting avenue for future work lies in comparison with the probabilistic schemes for object detection and tracking. Our algorithm avoids lengthy computations of probabilistic methods by tracking threshold boundaries of the moving objects and consulting scan history. For what type of dynamic environments does the threshold boundaries approach becomes less effective, would seek further investigation. Another direction motivates the extension of the proposed scheme to the outdoor environments. After considering important parameters such as weather and lighting conditions and variations in the dimensions and speeds of the objects, we feel that a 3D LiDAR sensor providing an enhanced FoV, can improve detection accuracy which generally leads to higher tracking performance. Fig 10 shows two applications deploying

(a)



(b)



**Fig 10. Application in real-world environment; (a) autonomous driving, (b) people tracking for security purpose.**

<https://doi.org/10.1371/journal.pone.0280476.g010>

3D LiDAR sensors for objects (vehicle or human) detection. Lastly, the outputted map of the static environment can also be used for path planning and obstacle avoidance. Improved strategies of autonomous control can benefit industrial pick and place robots or collaborative surveillance agents [51–53].

## Author Contributions

**Conceptualization:** Muhammad Ahsan.

**Data curation:** Mohammad Zohaib.

**Formal analysis:** Mohammad Zohaib, Mudassir Khan.

**Investigation:** Mohammad Zohaib, Mudassir Khan.

**Methodology:** Mohammad Zohaib, Muhammad Ahsan.

**Project administration:** Jamshed Iqbal.

**Resources:** Jamshed Iqbal.

**Software:** Mudassir Khan.

**Supervision:** Muhammad Ahsan.

**Validation:** Jamshed Iqbal.

**Visualization:** Mudassir Khan.

**Writing – original draft:** Mohammad Zohaib, Muhammad Ahsan.

**Writing – review & editing:** Jamshed Iqbal.

## References

1. Masood K, Molfino R, Zoppi M. Simulated Sensor Based Strategies for Obstacle Avoidance Using Velocity Profiling for Autonomous Vehicle FURBOT. *Electronics*. 2020; 9(6): 883 <https://doi.org/10.3390/electronics9060883>
2. Khan H, Iqbal J, Baizid K, Zielinska T. Longitudinal and lateral slip control of autonomous wheeled mobile robot for trajectory tracking. *Frontiers of Information Technology & Electronic Engineering*. 2015; 16(2): 166–172. <https://doi.org/10.1631/FITEE.1400183>
3. Iqbal J, Nabi R U, Khan A A, Khan H. A novel track-drive mobile robotic framework for conducting projects on robotics and control systems, *Life Science Journal*. 2013; 10(3)
4. You Y, Luo K, Phoo C P, Chao W L, Sun W, Hariharan B, Weinberger K Q. Learning to Detect Mobile Objects from LiDAR Scans Without Labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2022*; 1130-1140
5. Masood K, Morales D P, Fremont V, Zoppi M, Molfino R. Parking Pose Generation for Autonomous Freight Collection by Pallet Handling Car-like Robot. *Energies*. 2021; 14(15): 4677 <https://doi.org/10.3390/en14154677>
6. Wermelinger M, Johns R, Gramazio F, Kohler M, Hutter M. Grasping and Object Reorientation for Autonomous Construction of Stone Structures. *IEEE Robotics and Automation Letters*. 2021; 6(3): 5105–5112 <https://doi.org/10.1109/LRA.2021.3070300>
7. Zohaib M, Pasha S M, Hassan Z, Iqbal J. A centralized architecture for inventory management using RFID. In *International Conference on Robotics and Artificial Intelligence (ICRAI)*. 2016; 118–123
8. Wong I A, Huang J, Lin Z, Jiao H. Smart dining, smart restaurant, and smart service quality (SSQ). *International Journal of Contemporary Hospitality Management*. 2022; 34(6):2272–2297 <https://doi.org/10.1108/IJCHM-10-2021-1207>
9. Ali A A, Zohaib M, Mehdi S A. An autonomous seeder for maize crop. In *Proceedings of 5th International Conference on Robotics and Artificial Intelligence*. 2019; 42-47.
10. Hassan M U, Ullah M, Iqbal J. Towards autonomy in agriculture: Design and prototyping of a robotic vehicle with seed selector. In *2nd International Conference on Robotics and Artificial Intelligence (ICRAI)*; Rawalpindi Pakistan. 2016. pp. 37-44.
11. Iqbal J, Tsagarakis N G, Caldwell D G. Four-fingered light-weight exoskeleton robotic device accommodating different hand sizes. *IET Electronics Letters*. 2015; 51(12): 888–890 <https://doi.org/10.1049/el.2015.0850>
12. Iqbal J, Tsagarakis N G, Fiorilla A E, Caldwell D G. Design requirements of a hand exoskeleton robotic device. *14th IASTED International Conference on Robotics and Applications*, Massachusetts US. 2019; 44-51
13. Mahjourian R, Kim J, Chai Y, Tan M, Sapp B, Anguelov D. Occupancy flow fields for motion forecasting in autonomous driving. *IEEE Robotics and Automation Letters*. 2022; 7(2): 5639–5646. <https://doi.org/10.1109/LRA.2022.3151613>
14. Abbasi R, Bashir A K, Alyamani H J, Amin F, Doh J, Chen J. Lidar point cloud compression, processing and learning for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*. 2022; 1–18 <https://doi.org/10.1109/TITS.2022.3167957>

15. Zheng T, Huang Y, Liu Y, Tang W, Yang Z, Cai D, He X. CLRNet: Cross Layer Refinement Network for Lane Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022; 898-907
16. Sato T, Chen Q A. Towards Driving-Oriented Metric for Lane Detection Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022; 17153-17162
17. Iqbal J, Saad M R, Malik A, Tahir A M. State estimation technique for a planetary robotic rover. *Revista Facultad de Ingenieria-Universidad de Antioquia*. 2014; (73): 58–68
18. Shi L, Yao H, Shan M, Gao Q, Jin X. Robust control of a space robot based on an optimized adaptive variable structure control method. *Aerospace Science and Technology*. 2022; 120: 107267. <https://doi.org/10.1016/j.ast.2021.107267>
19. Masood K, Zoppi M, Molfino R. Mathematical Modelling for Performance Evaluation Using Velocity Control for Semi-autonomous Vehicle. In *International Workshop on Soft Computing Models in Industrial and Environmental Applications*. 2020; 617-626
20. Baizid K, Ćuković S, Iqbal J, Yousnadj A, Chellali A, Devedzic G, Ghionea I. IRoSim: Industrial Robotics Simulation Design Planning and Optimization Platform Based on CAD and Knowledgeware Technologies. *Robotics and Computer Integrated Manufacturing*. 2016; 42: 121–134 <https://doi.org/10.1016/j.rcim.2016.06.003>
21. Ajwad S A, Iqbal J, Khan A A, Mehmood A. Disturbance-observer-based robust control of a serial-link robotic manipulator using SMC and PBC techniques", *Studies in Informatics and Control*. 2015; 24(4): 401–408 <https://doi.org/10.24846/v24i4y201504>
22. Fan J, Wang S, Yu Q, Zhu Y. Swimming performance of the frog-inspired soft robot. *Soft Robot*. 2020; 7(5):615–626. <https://doi.org/10.1089/soro.2019.0094> PMID: 32401696
23. Hu Jw, Zheng By, Wang C, Zhao C, Hou X, et al. A survey on multi-sensor fusion based obstacle detection for intelligent ground vehicles in off-road environments. *Frontiers Inf Technol Electronic Engineering*. 2020; 21(1): 675–692. <https://doi.org/10.1631/FITEE.1900518>
24. Hu H, Liu Z, Chitlangia S, Agnihotri A, Zhao D. Investigating the Impact of Multi-LiDAR Placement on Object Detection for Autonomous Driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022; 2550-2559
25. Wei B, Ren M, Zeng W, Liang M, Yang B, Urtasun R. Perceive, attend, and drive: Learning spatial attention for safe self-driving. In *IEEE International Conference on Robotics and Automation (ICRA)*. 2021; 4875-4881
26. Peri N, Luiten J, Li M, Ošep A, Leal-Taixé L, Ramanan D. Forecasting from LiDAR via Future Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022; 17202-17211
27. Zohaib M, Iqbal J, Pasha S M. A novel goal-oriented strategy for mobile robot navigation without sub-goals constraint. *Revue Roumaine Sci. Techn.-Ser. Electrotechn. Energetique*. 2018; 63(1): 106–111.
28. Zohaib M, Pasha S M, Javaid N, Salaam A, Iqbal J. An Improved Algorithm for Collision Avoidance in Environments Having U and H Shaped Obstacles. *Studies in Informatics and Control*. 2014; 23 (1): 97–106. <https://doi.org/10.24846/v23i1y201410>
29. Mouawad I, Odone F. FasterVideo: Efficient Online Joint Object Detection and Tracking. In *International Conference on Image Analysis and Processing*. 2022; 375-387
30. Hanif MS, Ahmad S, Khurshid K. On the improvement of foreground–background model-based object tracker. *IET Computer Vision*. 2017; 11(6):488–96 <https://doi.org/10.1049/iet-cvi.2016.0487>
31. Zohaib M, Taiana M, Bue A D. Towards Reconstruction of 3D Shapes in a Realistic Environment. In *International Conference on Image Analysis and Processing*. 2022; 3–14
32. Held P, Steinhauser D, Koch A, Brandmeier T, Schwarz U T. A Novel Approach for Model-Based Pedestrian Tracking Using Automotive Radar. *IEEE Transactions on Intelligent Transportation Systems*. 2021; 1–14
33. Zohaib M, Taiana M, Padalkar MG, Del Bue A. 3D Key-Points Estimation from Single-View RGB Images. In *International Conference on Image Analysis and Processing*. 2022; 27-38
34. Gao T, Pan H, Gao H. Monocular 3D Object Detection with Sequential Feature Association and Depth Hint Augmentation. *IEEE Transactions on Intelligent Vehicles*. 2022; 81:5973–5988
35. Ji C, Liu G, Zhao D. Monocular 3D object detection via estimation of paired keypoints for autonomous driving. *Multimedia Tools and Applications*. 2022; 1–16
36. Sommer L, Krüger W, Teutsch M. Appearance and Motion Based Persistent Multiple Object Tracking in Wide Area Motion Imagery. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021; 3878-3888

37. Han S, Huang P, Wang H, Yu E, Liu D, Pan X. Mat: Motion-aware multi-object tracking. *Neurocomputing*. 2022; 75–86 <https://doi.org/10.1016/j.neucom.2021.12.104>
38. Schulz D, Burgard W, Fox D, Cremers A B. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. *IEEE International Conference on Robotics and Automation*. 2001. pp. 1665-1670.
39. Nuss D, Reuter S, Thom M, Yuan T, Krehl G, et al. A random finite set approach for dynamic occupancy grid maps with real-time application. *The International Journal of Robotics Research*. 2018; 37(8): 841–866. <https://doi.org/10.1177/0278364918775523>
40. Wang C C, Thorpe C, Thrun S. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. *IEEE International Conference on Robotics and Automation, Taipei, Taiwan*. 2003. pp. 842-849.
41. Chen P, Shi W, Fan W, Xiang H, Bao S. RectMatch: A novel scan matching method using the rectangle-flattening representation for mobile LiDAR systems. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2021; 180: 191–208. <https://doi.org/10.1016/j.isprsjprs.2021.08.011>
42. Mihálik M, Hruboš M, Vesterický P, Holečko P, Nemeč D, Malobický B, Mihálik J.A Method for Detecting Dynamic Objects Using 2D LiDAR Based on Scan Matching. *Applied Sciences*. 2022; 12(11): 5641.
43. Wolf D F, Sukhatme G S. Mobile robot simultaneous localization and mapping in dynamic environments. *Autonomous Robots*. 2005; 19(1), 53–65. <https://doi.org/10.1007/s10514-005-0606-4>
44. Petrovskaya A, Thrun S. Model based vehicle tracking for autonomous driving in urban environments. *Proceedings of Robotics: Science and Systems IV; Zurich, Switzerland*. 2008. 34.
45. Vu T D, Aycard O. Laser-based detection and tracking moving objects using data-driven markov chain monte carlo. *IEEE International Conference on Robotics and Automation*. 2009. pp. 3800-3806.
46. Iqbal J, Khan H, Chellali R. A unified SLAM solution using partial 3D structure. *Elektronika ir Elektrotehnika*. 2014; 20(9): 3–8. <https://doi.org/10.5755/j01.eee.20.9.8707>
47. Azimjonov J, Özmen A. A real-time vehicle detection and a novel vehicle tracking systems for estimating and monitoring traffic flow on highways. *Advanced Engineering Informatics*. 2021; 50: 101393 <https://doi.org/10.1016/j.aei.2021.101393>
48. Zhang Q, Rekleitis I, Dudek G. Uncertainty reduction via heuristic search planning on hybrid metric/topological map. *12th Conference on Computer and Robot Vision, Halifax, NS*. 2015. p. 222-229.
49. Fazlali H, Xu Y, Ren Y, Liu B. A versatile multi-view framework for lidar-based 3d object detection with guidance from panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022; 17192-17201
50. Ali S, Khan Z, Din A Hassan M U. Investigation on communication aspects of multiple swarm networked robotics. *Turkish Journal of Electrical Engineering & Computer Sciences*. 2019; 27(3): pp.2010–2020. <https://doi.org/10.3906/elk-1710-260>
51. Bisio I, Haleem H, Garibotto C, Lavagetto F, Sciarrone A. Performance Evaluation and Analysis of Drone-based Vehicle Detection Techniques From Deep Learning Perspective. *IEEE Internet of Things Journal*. 2021; 14(8): 1–8
52. Chen G, et al. VisDrone-MOT2021: The Vision Meets Drone Multiple Object Tracking Challenge Results. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021; 2839-2846
53. Bisio I, Garibotto C, Haleem H, Lavagetto F, Sciarrone A. On the localization of wireless targets: A drone surveillance perspective. *IEEE Network*. 2021; 35(5): 249–255 <https://doi.org/10.1109/MNET.011.2000648>