

Article

# Consensus-Based Information Filtering in Distributed LiDAR Sensor Network for Tracking Mobile Robots

Isabella Luppi <sup>†</sup> , Neel Pratik Bhatt  and Ehsan Hashemi <sup>\*</sup> 

Department of Mechanical Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada; luppi.isabella@gmail.com (I.L.); npbhatt@ualberta.ca (N.P.B.)

<sup>\*</sup> Correspondence: ehashemi@ualberta.ca

<sup>†</sup> Current address: The Charles Stark Draper Laboratory, Inc., Cambridge, MA 02139, USA.

**Abstract:** A distributed state observer is designed for state estimation and tracking of mobile robots amidst dynamic environments and occlusions within distributed LiDAR sensor networks. The proposed novel framework enhances three-dimensional bounding box detection and tracking utilizing a consensus-based information filter and a region of interest for state estimation of mobile robots. The framework enables the identification of the input to the dynamic process using remote sensing, enhancing the state prediction accuracy for low-visibility and occlusion scenarios in dynamic scenes. Experimental evaluations in indoor settings confirm the effectiveness of the framework in terms of accuracy and computational efficiency. These results highlight the benefit of integrating stationary LiDAR sensors' state estimates into a switching consensus information filter to enhance the reliability of tracking and to reduce estimation error in the sense of mean square and covariance.

**Keywords:** distributed sensor networks; information filters; consensus filters; LiDAR-based state estimation; perception



**Citation:** Luppi, I.; Bhatt, N.P.; Hashemi, E. Consensus-Based Information Filtering in Distributed LiDAR Sensor Network for Tracking Mobile Robots. *Sensors* **2024**, *24*, 2927. <https://doi.org/10.3390/s24092927>

Academic Editors: Daniel Galan, Ramon A. Suarez Fernandez and Francisco Javier Badesa

Received: 9 April 2024  
Revised: 30 April 2024  
Accepted: 30 April 2024  
Published: 4 May 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The evolution of distributed sensor networks (DSNs) and remote sensing has led to their application in motion planning and for the control of mobile autonomous systems for surveillance, environmental monitoring, warehouse management, and intelligent transportation [1–6].

Comprising spatially distributed sensors installed on infrastructures or mobile robots, DSNs can process local environmental data and generate a comprehensive scene overview through centralized or distributed state estimation [7–11], while maintaining asymptotic stability of the estimation error dynamics. As beneficial as these networks may be, the challenge is reliable estimation/sensing at the node (e.g., visual inference, point cloud clustering), particularly when it comes to target tracking in dynamic environments subject to partial or full occlusion [12–15]. Additionally, emerging research suggests that radar-based and UWB (Ultra-Wideband)-based distributed sensor networks offer promising alternatives for the localization and state estimation of mobile robots/vehicles [16,17], potentially enhancing accuracy and robustness in environments where LiDAR may encounter limitations.

The complexity and time-sensitivity of visual and LiDAR-based tracking of moving targets demand distributed observer design using continuous communication or event-triggered architecture [18–22], as accurate detection and pose estimation are integral for the robust and *scalable* tracking which is required for distributed motion planning and control of multiple mobile robots. In this regard, computationally efficient LiDAR-based state estimation at the edge (i.e., stationary sensing units), which provides accurate depth measurements provided by point clouds (removing the complexity of depth estimation and disparity map generation with a stereo/monocular camera), and maintains privacy (due to not processing of any images in the scene), has been the focus of the recent literature on remote estimation in DSNs [23–26].

However, processing the LiDAR point cloud data to obtain robots' three-dimensional bounding boxes, including their headings, and removing outliers are challenging and have been addressed through geometrical (i.e., model-based), filter-based, and end-to-end learning methods [27–29]. Despite this, the inherent variability in mobile robots'/ground vehicles' sizes and occlusion in dynamic scenes are the main challenges for reliability in detection and heading estimation in both model- and learning-based clustering and pose estimation approaches [30–33]. After reliable inference and pose estimation at the sensor node level, distributed estimation based on the Kalman consensus filter provides accurate estimates and convergence [34,35], depending on the topology of the networked sensors, guaranteeing the optimality of the variance and the stability of the error dynamics. Existing estimation methods in sensor networks, which centralize processing or use micro Kalman filters at each sensor node, impose considerable computational loads and require extensive inter-node communication during tracking and state estimation. These challenges are exacerbated in dynamic environments with multiple mobile robots and occlusion cases, impacting scalability and efficiency. Additionally, conventional L-shape fitting techniques for LiDAR-based tracking of mobile robots/vehicles are restricted to 2D data and necessitate sensors at the same vehicle height, limiting sensor placement flexibility and adaptability.

This paper develops a computationally efficient remote state estimation using a distributed architecture over LiDAR sensor networks, to address the limitations mentioned above and occlusion and limited visibility for the detection and tracking of mobile robots, through two key contributions:

- A generic LiDAR-based 3D bounding box detection and tracking method is designed to accommodate a wider range of sensor locations within DSNs;
- A distributed switching observer is designed to handle dynamic and occluded scenarios, to reduce overall computational cost, and to introduce short-term predictive capacity.

The remainder of the paper is organized as follows: Section 2 presents the background and point cloud processing. Section 3 introduces the designed distributed observer. Section 4 evaluates the performance and computational efficiency of the developed remote sensing framework in several experiments including occluded and dynamic scenes. Section 5 concludes this paper.

## 2. Background and Point Cloud Processing

The position vector  $\mathbf{p}_r$  of a mobile robot centroid and the orientation  $\mathbf{o}_r$  of the bounding box of the tracked robot are defined by

$$\mathbf{p}_r = [x_r \ y_r \ z_r]^\top, \mathbf{o}_r = [\phi_r \ \psi_r \ \theta_r]^\top, \quad (1)$$

where  $x_r$ ,  $y_r$ , and  $z_r$  represent the coordinates of the centroid in the fixed world frame  $\{G\}$ , while  $\phi_r$ ,  $\psi_r$ , and  $\theta_r$  are the bounding box (roll, pitch, and yaw) orientation angles. In order to develop the state observer and point cloud clustering, the following assumptions are made in this paper: **(i)** any variations in the  $z$ -value of the robot position are negligible (i.e., our case study is for the pose estimation and tracking of wheeled robots as shown in Figure 1); and **(ii)** the position  $\mathbf{p}_s$  and orientation  $\mathbf{o}_s$  of the sensor nodes are known in the global frame  $\{G\}$ . There is also no prior knowledge of the environment and dimensions of the robot. The computational challenge posed by high-dimensional point clouds is mitigated through voxelization (for dimensionality reduction). Given a point cloud  $\mathbf{C} \in \mathbb{R}^{N \times 3}$ , where  $N$  is the number of points, each represented by  $(x, y, z)$ , the application of the Voxel Grid Filter can be represented as

$$\mathbf{C}^v = \text{VoxelGrid}(\mathbf{C}, \alpha), \quad (2)$$

in which  $\alpha \in \mathbb{R}$  is the parameter controlling the size of the voxels, and  $\mathbf{C}^v \in \mathbb{R}^{N^v \times 3}$  is the resulting filtered point cloud with  $N^v < N$  points. This divides the point cloud into a three-dimensional grid and selects a single representative point from each grid cell,

discarding the rest. Each voxel  $\mathbf{v}_i$  is a cubic region of volume  $\alpha^3$ , and each point  $\mathbf{p} \in \mathbf{C}$  is assigned to a voxel based on its coordinates, determined by  $(i_x, i_y, i_z) = (\lfloor \frac{x}{\alpha} \rfloor, \lfloor \frac{y}{\alpha} \rfloor, \lfloor \frac{z}{\alpha} \rfloor)$ , where  $(i_x, i_y, i_z)$  are the voxel indices in the grid corresponding to each point  $\mathbf{p}$ , defining the voxel  $\mathbf{v}_i$  to which the point is assigned. All points  $\mathbf{p}$  within each voxel  $\mathbf{v}_i$  are then represented by the centroid  $\mathbf{c}_i$ , which is the arithmetic mean of the points in the voxel calculated by  $\mathbf{c}_i = \frac{1}{\mathcal{N}(\mathbf{v}_i)} \sum_{p \in \mathbf{v}_i} p$ , where  $\mathcal{N}(\mathbf{v}_i)$  is the number of points in the voxel. This approach effectively reduces the data volume while preserving the essential structural features of the point cloud.



**Figure 1.** Experimental setup for evaluation of the LiDAR-based distributed state observer: unmanned ground vehicle (UGV) setup (left), and solid-state LiDAR used for remote sensing (right).

The point cloud data (obtained from the solid-state *Robosense* LiDAR shown in Figure 1) require transformation for alignment with the global frame  $\{G\}$ . The transformation of all points  $\mathbf{p} \in \mathbf{C}^v$  involves rotation performed using a rotation matrix  $\mathbf{R} \in \mathbb{R}^{3 \times 3}$  derived from the Euler angles

$$\mathbf{R} = \mathbf{R}_z(\theta_s) \cdot \mathbf{R}_y(\psi_s) \cdot \mathbf{R}_x(\phi_s), \quad (3)$$

where  $\mathbf{R}_x(\phi_s)$ ,  $\mathbf{R}_y(\psi_s)$ , and  $\mathbf{R}_z(\theta_s)$  represent the rotation matrices around the  $x$ -,  $y$ -, and  $z$ -axes, respectively. The point cloud in the global frame is then expressed by

$$\mathbf{C}^G = \mathbf{R} \cdot \mathbf{C}^v + \mathbf{t}^G, \quad (4)$$

in which  $\mathbf{t} = [x_s \ y_s \ z_s]^\top$  is the translation vector. Considering the first assumption, we employ a thresholding technique to identify and eliminate ground points from the point cloud as

$$\mathbf{C}' = \{\mathbf{p} \in \mathbf{C}^G \mid z > h_f + \epsilon\}, \quad (5)$$

in which  $\mathbf{C}' \in \mathbb{R}^{N' \times 3}$  includes the ground-less point cloud,  $h_f$  is the elevation of the ground floor in the frame  $G$ , and  $\epsilon$  is a slack variable to deal with potential noise or deviations from the ideal floor plane. While this approach effectively removes the majority of the ground points, it may also remove low-elevation non-ground points. Since the geometry of other operating robots in the environment is known and the aim of this research is accurate clustering and reliable distributed robot state estimation (not generating a cost/occupancy map), even the unexpected exclusion of some low-elevation points due to thresholding will not significantly affect the performance of the proposed framework.

#### Clustering and Principle Component Analysis

To partition the point cloud into different groups, Euclidean clustering is employed. Given the point cloud  $\mathbf{C}'$ , Euclidean clustering involves finding the set of clusters  $\mathbf{C}_c = \{C_1, C_2, \dots, C_k\}$  that best represents the underlying structure of points  $\mathbf{p} \in \mathbf{C}'$ , where

each cluster  $C_i$  is defined as a set of neighboring points that are close to each other in the 3D space, as in

$$C_i = \{\mathbf{p}_j \mid \mathbf{p}_j \in \mathbf{C}' \text{ and } d_{i,j} \leq \rho\}, \quad (6)$$

where  $i = \{1, \dots, k\}$  denotes the cluster index,  $j = \{1, \dots, N'\}$  represents individual points within the point cloud, and  $\rho$  is a predefined distance threshold. It is important to note that the value of  $\rho$  must be at least as large as the leaf  $\alpha$  of the voxels. If  $\rho \leq \alpha$ , no clusters will be formed due to the inability of points to meet the proximity criteria. Each cluster  $C_i$  is characterized by computing the spans between the maximum and minimum values for height and length along the  $x$ - and  $y$ -axes. The selection of the clusters corresponding to the robot is achieved by comparing these dimensions to the prior geometry/size information of the robot operating in the environment. This comparison specifically considers the robot's shortest side and its longest diagonal to ensure the dimensions fall within predefined thresholds.

By comparing the current clusters with the previous estimates, we associate each selected cluster (mentioned above) with the corresponding object through maximum likelihood data association. For each cluster  $C_i \in \mathbf{C}_c$ , the association likelihood is calculated by evaluating the distance  $d_i$  between the centroid of the cluster and the prior estimated position  $\mathbf{p}_r$ . The cluster with the centroid closest to the past position estimate has the highest likelihood considering the bounds on the robot's speed and acceleration. The chosen association is where  $L_i$  is maximal, as in

$$i^* = \arg \max_i L_i, \quad L_i = \frac{1}{d_i}. \quad (7)$$

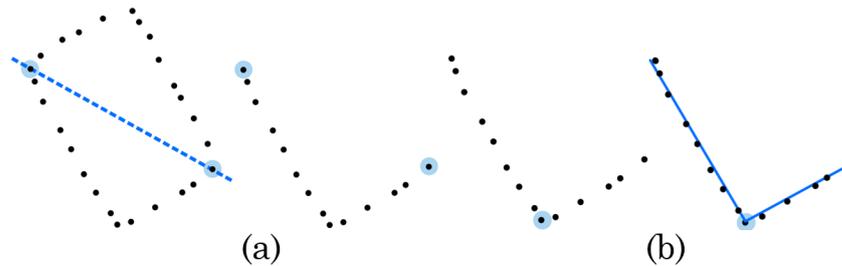
The aim is to fit a 3D bounding box around each cluster of points obtained, accurately representing the shape and dimensions of the robot in the scene. The proposed method modifies the conventional L-shape fitting [36,37], which is effective when the robot's point cloud consists of points belonging to the primary two edges of the moving robot. The LiDAR's bird's-eye-view estimated pose reveals the robot's top in the point cloud.

To isolate the robot's edges, the cluster  $C_i$  is projected onto the  $xy$ -plane and encapsulated by a convex hull,  $\text{CH}(C_i)$ , which is the smallest convex polygon encompassing all points in cluster  $C_i$  as shown in Figure 2. This method both identifies the mobile robot's edges and regularizes its shape. If the robot shape is not a complete box (due to protruding wheels or a manipulator), the convex hull algorithm normalizes the estimated cluster edges into rectangles. Let  $\mathbf{p}_h$  denote each point on the convex hull, where  $h = \{1, 2, \dots, N\}$ , and  $N$  is the total number of points constituting the convex hull  $\text{CH}(C_i)$ . To classify the remaining points, we analyze the angle  $\beta_h$  for each point  $\mathbf{p}_h \in \text{CH}(C_i)$  with respect to the infrastructure-mounted LiDAR position  $\mathbf{p}_s$ . By finding the lower bound  $\beta_{\min} = \min\{\beta_h\}$  where  $\beta_h = \tan^{-1}(\frac{y_h}{x_h})$ , the corresponding point  $\mathbf{p}_{\beta_{\min}}$  can be identified. Similarly, by finding the upper bound,  $\beta_{\max} = \max\{\beta_h\}$ , the point  $\mathbf{p}_{\beta_{\max}}$  can be determined. Hence, the points  $\mathbf{p}_{\beta_{\min}}$  and  $\mathbf{p}_{\beta_{\max}}$  correspond to the two primary edges of the mobile robot, as they represent the most laterally external points within the point cloud based on their angular positions.



**Figure 2.** The  $xy$ -projection of the UGV robot's clustered point cloud in real time for  $z_s \leq z_r$  (left), for  $z_s > z_r$  (center), and the convex hull of the cluster  $xy$ -projection (right).

To classify whether each  $\mathbf{p}_h \in \text{CH}(C_i)$  belongs to the main edges, we examine the relative position of each  $\mathbf{p}_h$  with respect to the LiDAR position  $\mathbf{p}_s$  and the line segment defined by the two points  $\mathbf{p}_{\beta_{\min}}$  and  $\mathbf{p}_{\beta_{\max}}$ . Initially, the vector  $\mathbf{v}$  that links the two points is computed by  $\mathbf{v} = \mathbf{p}_{\beta_{\max}} - \mathbf{p}_{\beta_{\min}}$ . Subsequently, the scalar product  $w_j$  of the vector  $\mathbf{v}$  and the vector from the sensor position  $\mathbf{p}_s$  to each point  $\mathbf{p}_j \in C_i$  is defined by  $w_j = \mathbf{v} \cdot (\mathbf{p}_j - \mathbf{p}_s)$ . As schematically represented in Figure 3a, the scalar dot product  $w_j$  is then used as an indicator of the position of  $\mathbf{p}_j$  with respect to  $\mathbf{p}_s$ .



**Figure 3.** (a) Lateral-external points and L-shape reduction: (i) if  $w_j \geq 0$ , the point  $\mathbf{p}_j$  lies on the same side of the decision boundary (dashed blue line) as the sensor, or is collinear with the line segment defined by  $\mathbf{p}_{\beta_{\min}}$  and  $\mathbf{p}_{\beta_{\max}}$  (it is considered to be part of the main edges and is retained), and (ii) if  $w_j < 0$ , the point  $\mathbf{p}_j$  is removed from  $C_i$ ; (b) closest point and RANSAC line fitting.

The nearest corner point  $\mathbf{p}_n$ , which corresponds to the point with the minimum Euclidean distance between all points in the target robot and the sensor position  $\mathbf{p}_s$ , is identified by

$$\mathbf{p}_n = \arg \min_{\mathbf{p}_j \in C_i} \|\mathbf{p}_j - \mathbf{p}_s\|. \quad (8)$$

The cluster  $C_i$  is then divided into two distinct subsets, namely  $C_i^A$  and  $C_i^B$ , based on the angle each point forms with respect to the corner point  $\mathbf{p}_n$ . The angle  $\gamma_j$  between point  $\mathbf{p}_j$  and  $\mathbf{p}_n$  is computed by  $\gamma_j = \tan^{-1}\left(\frac{y_j - y_n}{x_j - x_n}\right)$ . Consequently, the points with  $\gamma_j \leq \pi/2$  are included in the set  $C_i^A$ , while the remaining points are included in the set  $C_i^B$ , as in

$$C_i^A = \{\mathbf{p}_j \in C_i | \gamma_j \leq \frac{\pi}{2}\}, C_i^B = \{\mathbf{p}_j \in C_i | \gamma_j > \frac{\pi}{2}\}. \quad (9)$$

The lines  $l_i^A$  and  $l_i^B$  are then fitted to the point sets  $C_i^A$  and  $C_i^B$  using Random Sample Consensus (RANSAC) as shown in Figure 3b. The orientation difference between the two lines is denoted by

$$\delta\theta_i := \left| \theta_i^A - \theta_i^B \right|, \quad (10)$$

where  $\theta_i^A$  and  $\theta_i^B$  are the orientations of lines  $l_i^A$  and  $l_i^B$ , respectively. If the difference  $\delta\theta$  exceeds a predetermined threshold  $\bar{\delta}\theta$ , the lines  $l_i^A$  and  $l_i^B$  are accepted as the two edges of the target mobile robot. Consequently, the orientation of the robot,  $\theta_r$ , is determined by the longer of the two fitted lines,  $l_i^A$  or  $l_i^B$ . The scale of the bounding box is then determined by

$$l_r = \max\{\|l_i^A\|, \|l_i^B\|\}, w_r = \min\{\|l_i^A\|, \|l_i^B\|\}. \quad (11)$$

If the angular difference  $\delta\theta$  is within the threshold  $\bar{\delta}\theta$ , it suggests that points align along a single robot edge. This results in inaccurate bounding box identification using conventional L-shape detection methods. Afterwards, in the augmented LiDAR-based state clustering module (of the proposed estimation framework) in this paper, principal component analysis (PCA) is used to identify the bounding box's major axis based on the first principal component of the point cloud transformed to a new coordinate system; the



acceleration is valid between two consecutive frames (within the sample time  $T_s = 50$  ms, which is sufficiently small for the robot maximum speed of  $v_x = 1.2$  m/s to warrant this approximation) and the acceleration is updated at every time step using a moving horizon forecasting method described at the end of this subsection. The state and input matrices then yield

$$A = \begin{bmatrix} \mathbb{I}_3 & T_s \mathbb{I}_3 \\ \mathbf{0}_3 & \mathbb{I}_3 \end{bmatrix}, B = \begin{bmatrix} \frac{T_s^2}{2} \mathbb{I}_3 \\ T_s \mathbb{I}_3 \end{bmatrix}, \quad (16)$$

in which  $T_s$  is the sample time between two consecutive frames, and  $\mathbb{I}$  is the identity matrix. The sensor model of the stationary sensing unit  $i$  to track the state of the target/robot within the LiDAR sensor network with communication topology  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  is

$$\mathbf{y}_i(k) = H_i(k)\mathbf{x}(k) + w(k), \quad (17)$$

where  $w(k)$  is the measurement noise of the sensor  $i$  and  $H_i(k)$  is the observation matrix of the sensor  $i$ . Process and measurement noises within the distributed sensor setting are assumed to be uncorrelated, i.e.,  $\mathbb{E}\{\vartheta_k w_k^\top\} = 0$ . The input  $u(k) = [a_x(k), a_y(k), \ddot{\theta}_r(k)]^\top$  has the perceived longitudinal/lateral translational and rotational accelerations of the clustered point cloud centroid, and is obtained over a moving horizon using position and speed gradients. In this regard, the acceleration input is estimated remotely using a moving horizon  $n_h \in \mathbb{N}$  over past estimated states, as in  $u(k) \triangleq \mathbb{E}\{\mathbf{a}_k\}$ , where

$$\mathbf{a}_k = \left\{ \frac{d}{dt} \dot{\mathbf{p}}_{r,q} : \dot{\mathbf{p}}_{r,k} \in \mathbb{R}^3, k - n_h + 1 \leq q \leq k \right\}, \quad (18)$$

where the finite difference of  $\dot{\mathbf{p}}_{r,k} = [v_{r,x}, v_{r,y}, \psi_k]^\top$  over sample time  $T_s$  is used to approximate the time derivative  $\frac{d}{dt} \dot{\mathbf{p}}_{r,k}$ . For ease of notation, the updated values of the state variable and information matrices, which will be introduced in the next subsection, are denoted by  $\mathbf{x}^+$ , and subscripts  $k$  are dropped. As a result, the process dynamics (15) and the sensor model of the stationary sensing unit  $i$  of the network are written as

$$\mathbf{x}^+ = A\mathbf{x} + Bu + \vartheta, \quad \mathbf{y}_i = H\mathbf{x} + w_i. \quad (19)$$

The motion model dynamically incorporates changes in the system response over time by continuously updating the control inputs for the consensus filter, and enables reliable tracking of the robot in the designed consensus filter in case of occlusion in dynamic environments.

### 3.1. Consensus Information Filter

The observer tracks the robot over the distributed sensor network, and the aim is for the network to provide a set of state estimates  $\hat{\mathbf{x}}_i(k)$  of the robot through the local exchange of messages among close/neighbor sensing units. At each time instance  $k$ , the state observer allocates the resource to only one stationary sensing unit denoted by the base node  $\mathcal{B}(k)$  for state estimation in the global frame using a consensus information filter, in which the measurement  $\mathbf{y}_i$  (with covariance  $\mathbf{R}_i$ ) of the stationary sensing unit  $\mathbf{s}_i$  is used to calculate the information vector  $\mathbf{z}_i = H_i^\top \mathbf{R}_i^{-1} \mathbf{y}_i$  and the information matrix  $\mathbf{I}_i = H_i^\top \mathbf{R}_i^{-1} H_i$ . If the stationary sensing unit  $\mathbf{s}_i$ , which has the robot in its field of view, matches the base node allocation  $\mathcal{B}(k)$  for time  $k$ , it checks for any available measurements  $\mathbf{y}_j$  (with covariances  $\mathbf{R}_j$ ) from the following set:

$$\mathcal{S}_i = \{j | j \in \mathcal{N}_i, l_{r,j} \geq \bar{l}_r \wedge w_{r,j} \geq \bar{w}_r\}, \quad (20)$$

where  $\mathcal{N}_i$  denotes the neighbors of node  $i$  on the network (with topology  $\mathcal{G}(k)$ , which could be a dynamic/switching graph). If found, the sensing unit  $i$  receives the pair from the other sensing unit  $j \in \mathcal{S}_i$ , and transforms them into the following information form:

$$\bar{\mathbf{I}}_i = \sum_{j \in \mathcal{S}_i} \mathbf{I}_j, \quad \bar{\mathbf{y}}_i = \sum_{j \in \mathcal{S}_i} \mathbf{z}_j, \quad \bar{\mathcal{S}}_i = \mathcal{S}_i \cup \{i\}. \quad (21)$$

Then, the following distributed observer scalable in  $n$  (developed in [35,38]) is used to estimate the robot pose with  $\check{\mathbf{x}}_i$  as the set of prior estimates (predictions) of the state  $\mathbf{x}(k)$ :

$$\begin{aligned} \hat{\mathbf{x}}_i &= \check{\mathbf{x}}_i + \mathcal{L}_i(\bar{\mathbf{y}}_i - \bar{\mathbf{I}}_i\check{\mathbf{x}}_i) + \zeta P_i \sum_{j \in \mathcal{S}_i} (\check{\mathbf{x}}_j - \check{\mathbf{x}}_i), \\ \check{\mathbf{x}}_i^+ &= A\hat{\mathbf{x}}_i + Bu_i, \quad P_i^+ = A\mathcal{L}_iA^\top + BQB^\top, \end{aligned} \quad (22)$$

in which  $\mathcal{L}_i = (P_i^{-1} + \bar{\mathbf{I}}_i)^{-1}$  is the consensus information filter gain,  $\check{\mathbf{x}}_i^+$  is the state update of the local information filter, and  $\zeta = \frac{\iota}{1 + \|\bar{P}_i\|}$ . The small constant  $\iota > 0$  is chosen in the order of the (integration) time interval  $T_s$  used for discretization of the continuous-time constant-acceleration motion model.  $\|\bar{P}_i\| = \sqrt{\text{tr}(P_i^\top P_i)}$  denotes the Frobenius norm of the matrix  $P_i$ . It should be mentioned that the distributed state observer in (22) is an information filter form of the Kalman consensus filter provided in the following:

$$\begin{aligned} \hat{\mathbf{x}}_i &= \check{\mathbf{x}}_i + \mathcal{K}_i(\mathbf{y}_i - H_i\check{\mathbf{x}}_i) + \eta_i \sum_{j \in \mathcal{S}_i} (\check{\mathbf{x}}_j - \check{\mathbf{x}}_i), \\ P_i^+ &= AE_iA^\top + BQB^\top, \end{aligned} \quad (23)$$

with the update of the prior estimates  $\check{\mathbf{x}}_i^+ = A\hat{\mathbf{x}}_i + Bu_i$ , and the Kalman gain  $\mathcal{K}_i = P_iH_i^\top (H_iP_iH_i^\top + \mathbf{R}_i)^{-1}$ . The error covariance matrix (associated with  $\hat{\mathbf{x}}_i$ ) is denoted by  $E_i = S_iP_iS_i^\top + \mathcal{K}_i\mathbf{R}_i\mathcal{K}_i^\top$  with  $S_i = \mathbb{I} - \mathcal{K}_iH_i$ , where  $\mathbb{I}$  is the identity matrix with proper dimensions.

**Remark 1.** The (global) asymptotic stability of the error dynamics of the Kalman consensus filter (23) with the choice of consensus gain  $\eta_i = \zeta S_i$  is proved in [35], Theorem 2, where all sensor node estimators asymptotically reach a consensus  $\hat{\mathbf{x}}_1 = \hat{\mathbf{x}}_2 = \dots = \hat{\mathbf{x}}_n = \mathbf{x}$ .

In the case where no measurement is available through stationary sensors (i.e.,  $\bar{\mathcal{S}} = \emptyset$ ), the state estimate is the predicted pose by the motion model (15), with  $u(k)$  as the behaviour-based input obtained over a horizon  $n_h$  using (18). The base node  $\mathcal{B}(k)$  for the next time step is selected based on a distance function  $d(\mathbf{s}_j)$  that measures the Euclidean distance between the estimated state and the positions of the stationary sensing units:

$$d(\mathbf{s}_j) = \|\hat{\mathbf{x}}_j^{(1:2)} - \mathbf{p}_{\mathbf{s}_j}^{2D}\|_2, \quad \forall j \in \mathcal{S} \quad (24)$$

where  $\hat{\mathbf{x}}_j^{(1:2)} \triangleq [\hat{x}_{r,j}(k), \hat{y}_{r,j}(k)]^\top$  includes the first two elements of the estimated state  $\hat{\mathbf{x}}_j$  in the 2D plane at time  $k$ . Then, the sensor node closest to the predicted state is chosen as the next allocated resource  $\mathcal{B}(k+1) \triangleq \mathbf{s}_j$ , where  $j = \arg \min d(\mathbf{s}_j)$ . The stationary sensor  $\mathbf{s}_j$  broadcasts the updated and predicted state estimates, the updated covariance, and the base node  $\mathcal{B}(k+1)$  for the next time step. The decision to allocate resources based primarily on spatial distance arises from the inherent spatial distribution observed among both the robot and the sensor nodes in the scene. This spatial distribution emerges as a result of factors such as the physical dimensions of robots and environmental constraints in dynamic scenes, which collectively ensure a dispersion of robots across the operational environment.

If sensor node  $\mathbf{s}_i$  is not assigned as the base node  $\mathcal{B}(k)$ , it functions as a contributing node within the network. Non-base nodes with available measurements transmit the measurement vectors  $\mathbf{y}_j$  and the corresponding measurement covariance matrices  $\mathbf{R}_j$  to  $\mathcal{B}(k)$ . Independently of their measurement status, all nodes subsequently receive the most

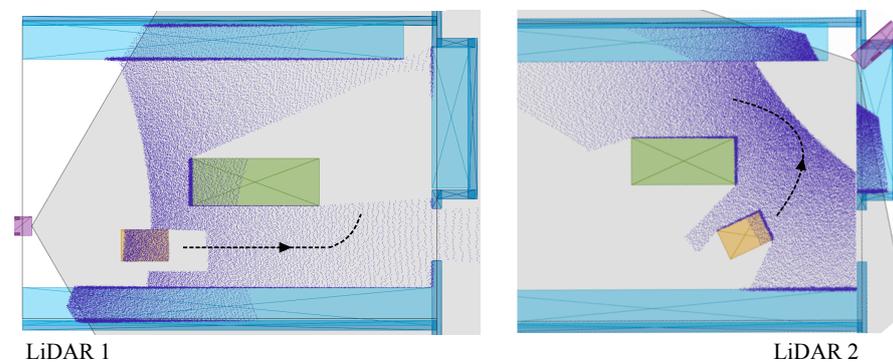
recent state estimates  $\hat{\mathbf{x}}$ , the prediction state  $\check{\mathbf{x}}$ , the covariance  $\mathbf{P}$ , and the identifier  $\mathcal{B}_{k+1}$  of the next node scheduled for base status from  $\mathcal{B}_k$ .

### 3.2. Covariance Dual-Rate Self-Tuning

The need for a dual-rate tuning method is that the noise in the proposed distributed state observer (due to LiDAR-based sensing and point cloud clustering) is time-varying and changes depending on the distance of the target vehicle/robot from the stationary sensing unit (i.e., the solid-state LiDAR sensor). Therefore, two initial covariances are used for the information filter:  $\Sigma^c$ , associated with  $d^c$ , which is the closest Euclidean distance (for close-range measurements); and  $\Sigma^f$ , associated with the farthest Euclidean distance  $d^f$  from the sensing unit (for far-range measurements). The covariance allocation during the measurement update in the consensus filter is then a linear interpolation  $\mathbf{R}_k = a_1 \Sigma^f + a_2 \Sigma^c$ , using the measured distance  $d$  (obtained by clustering and  $l_r, w_r$ ) with  $\tilde{d} = d^f - d^c$ .

## 4. Experimental Results and Discussion

Several experiments were conducted in an indoor setting comprising various static and dynamic objects (with the operator walking around the robot) to evaluate the performance and computational efficiency of the developed LiDAR-based remote sensing and distributed estimation framework. The experimental methodology included testing in (i) full occlusion and (ii) partial occlusion in dynamic environments to evaluate the proposed distributed estimator's accuracy, convergence, and computational efficiency using one and two LiDAR remote sensing units. Furthermore, high-fidelity simulations using Matlab (2023b) *Automated Driving Toolbox* were conducted for the (i) state estimation of multiple skid-steer mobile robot models using a network of solid-state LiDAR sensors (as shown in Figure 5), and (ii) identification of optimal coverage areas. The global frame of reference  $\{\mathbf{G}\}$ , which located its origin at one of the corners of the testing hall/room, was established for all test scenarios.



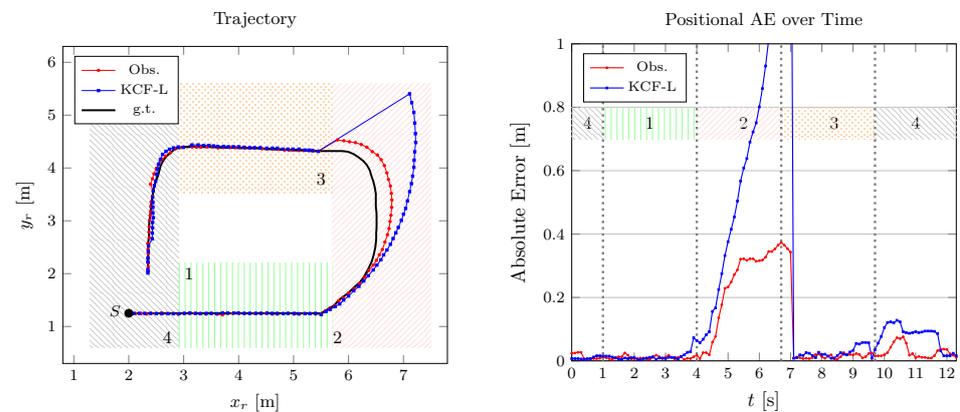
**Figure 5.** High-fidelity simulation of LiDAR coverage in indoor setting.

The mobile test platform was a *Clearpath's Husky* skid-steer UGV, with a maximum speed of 1.2 m/s for the conducted indoor tests in dynamic environments. The experiments employed a dual solid-state LiDAR setup (with horizontal and vertical fields of view of 120° and 25°, respectively, a range of 150 m, a region of interest (ROI) feature for better resolution around fields of view limits, and accurate line separation, i.e., 3–4 cm detection range), each paired with an embedded *Jetson Xavier NX*. The first LiDAR was oriented at a pitch of 25° and a yaw of −90°; the second LiDAR had a pitch of 15° and a yaw of 170° degrees. This arrangement ensured comprehensive coverage of the area. The experimental methodology was designed to investigate two distinct scenarios (mentioned above), each tested five times in dynamic scenes under similar nominal control inputs to the UGV during navigation with obstacle avoidance. Throughout all testing scenarios, the UGV performed dynamic motion planning, as well as followed a consistent navigational path, and operated without any communication with the distributed sensor network.

The qualitative/quantitative experimental results are provided in the following subsections for two main scenarios: with only one stationary LiDAR sensor (to evaluate the clustering, tracking, and pose estimation), and with two remote sensing units. In this section, (i) “KCF-L” denotes the Kalman consensus filter architecture [35,39] which utilizes the proposed point cloud processing and pose estimation from Section 2; and (ii) “Obs.” denotes the developed consensus information filter, including point cloud processing and LiDAR-based distributed pose estimation (i.e., the whole pipeline proposed in this paper).

#### 4.1. Occlusion with One Remote Sensing Unit

In the first test scenario, only LiDAR 1 was used for state estimation. The estimation results for this scenario are graphically illustrated in Figure 6. The results are consistent throughout the entire set of runs; therefore, only one run is displayed in the figure. Subsequently, the quantitative data presented are mean values computed from all runs.

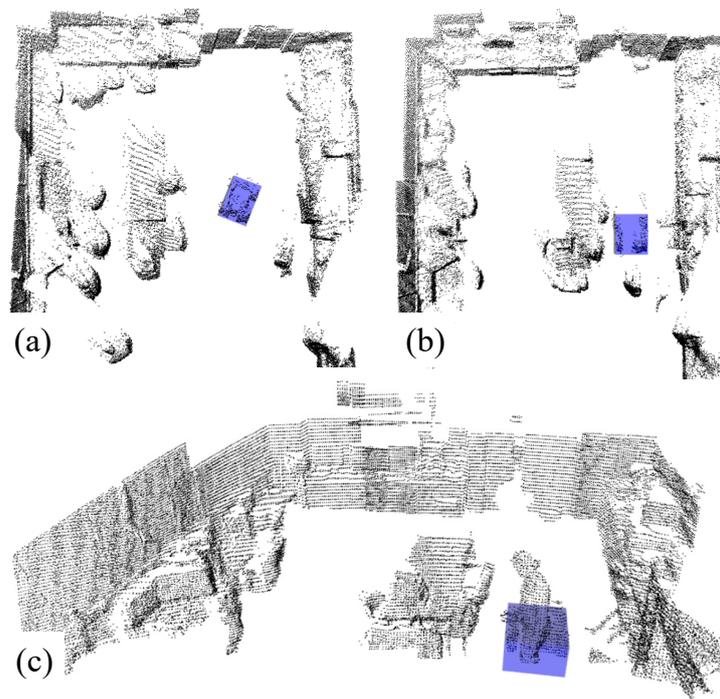


**Figure 6.** Tracked position and its absolute error over time in Scenario 1, where S is the starting point for the trajectory.

In the event of measurement unavailability, both the Kalman consensus filter (KCF-L) and the designed distributed observer use the predicted state as the estimation until the updated measurements for the subsequent step. However, KCF-L uses a general dynamics model without input estimation, resulting in constant velocities throughout the occluded area. Contrarily, the proposed framework estimates the input acceleration (for the dynamical process) using a moving horizon and the velocity magnitude right before the occlusion instance. This allows the designed framework to make accurate predictions of changes in the robot heading angle, thus reducing the estimation error after being updated by measurements from LiDAR 1. The point cloud representation of the scene partially occluded by a moving obstacle (i.e., operator) is shown in Figure 7, which confirms consistent clustering and state estimation using one remote sensing unit.

Position and orientation estimation errors are in Table 1, where MAE, MSE, and RMSE denote mean absolute error, mean squared error, and root mean square error, respectively. These values are mean quantitative results derived from the entire set of runs, which were five consecutive tests in the environment with dynamic objects and similar nominal control inputs (for the purpose of obstacle avoidance). The experimental studies confirm that as long as the robot trajectory does not change drastically between sample times for occluded scenes, the constant-acceleration motion model can be used for *full* occlusion.

Furthermore, it is important to note that the original bounding box L-shape fitting method fails to function under bird’s-eye-view conditions. This method was designed for sensors positioned at the same height as the vehicle; it effectively utilizes angled views to estimate dimensions and orientation.



**Figure 7.** Point cloud clustering in (a) a dynamic scene with moving objects, (b) partial occlusion, and (c) full occlusion by an operator moving alongside the robot (in blue).

**Table 1.** Position/heading estimation errors in Scenario 1.

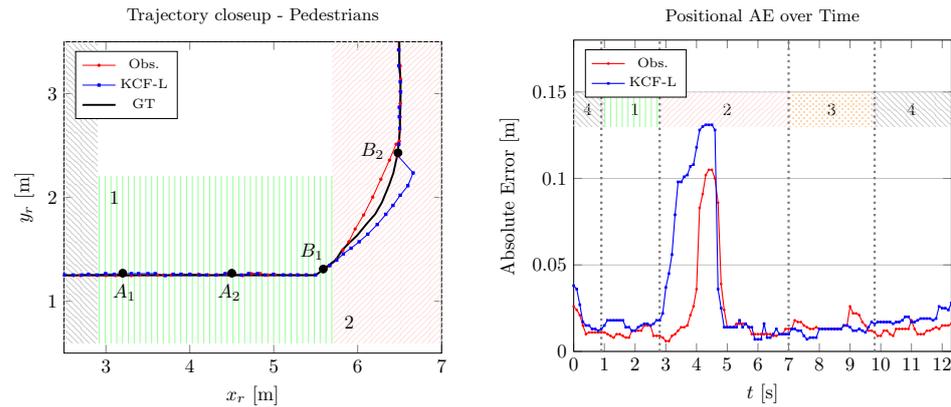
Method	MAE (m)	MSE	RMSE
Obs.	0.074	0.019	0.138
KCF-L	0.199	0.192	0.438
Method	MAE (°)	MSE	RMSE
Obs.	5.2	64.3	8.0
KCF-L	12.5	509.4	22.6

#### 4.2. Two Sensor Nodes

In Scenario 2, the performance of the developed framework is evaluated using two sensor nodes. Initially, the environment does not present any significant occluding obstacles. Subsequently, the presence of the moving operator in Areas 1 and 2 leads to the formation of two occluded areas, discernible in Figure 8 between points  $A_1$  and  $A_2$  and between  $B_1$  and  $B_2$ .

For occlusion A instances, the alteration is not significant as the robot maintains a straight path throughout the occlusion. In contrast, occlusion B presents more noticeable differences as the robot follows a curve and transitions into the second area, necessitating a shift in the sensors' field of view (from  $L_1$  to  $L_2$ ). Although occlusion B incites an increase in the absolute error, the maximum estimation error by the proposed method does not exceed 11 cm for the position estimation, as shown in Figure 8.

The position and heading angle estimation results are compared in Table 2, which highlights improvements in the MSE, MAE, and error covariance using the proposed framework. Even though the performances of both state observers are very close for this scenario, under full robot visibility, Obs. demonstrates superior performance under pedestrian occlusion. This is attributed to the input estimation, which is a feature not employed by the KCF-L.



**Figure 8.** State estimation results and absolute error comparison with KCF-L for single run of Scenario 2, with full and partial occlusion.

**Table 2.** Position/heading estimation errors in Scenario 2 with 2 remote sensing units and occlusion scenarios.

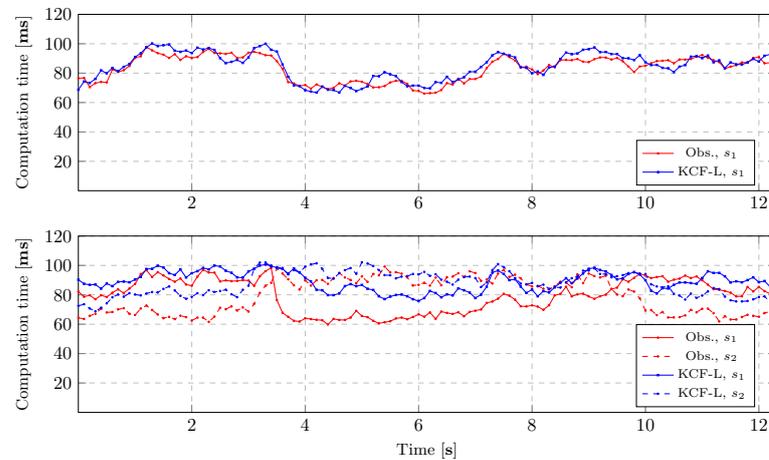
Method	Pedestrians	MAE (m)	MSE	RMSE
Obs.	No	0.018	0.001	0.032
KCF-L	No	0.018	0.001	0.031
Obs.	Yes	0.023	0.002	0.042
KCF-L	Yes	0.031	0.003	0.052
Method	Pedestrians	MAE (°)	MSE	RMSE
Obs.	No	1.9	5.9	2.4
KCF-L	No	2.0	5.8	2.4
Obs.	Yes	3.4	10.2	3.2
KCF-L	Yes	4.9	33.8	5.8

Several experimental tests and error convergence analyses in various high-/low-excitation maneuvers with occlusion provide insightful observation on the impact of perceptually degraded conditions on the performance of the designed LiDAR-based distributed switching state observer in this paper: **(i)** When the robot is fully visible by SSUs, the developed observer and the KCF-L (which uses the developed computationally efficient clustering approach, but utilizes the conventional Kalman consensus filter) perform reliably for position estimation; however, less heading estimation error for “Obs.” offers an advantage in terms of computational cost on a network level, making it a more efficient choice in scenarios where computational resources are a limiting factor. **(ii)** The variance in the estimation error by the proposed framework is lower than that of KCF-L in almost all experiments. This is attributed to the utilization of the region of interest for outlier identification/rejection in the new distributed estimation framework.

The solid-state LiDAR used for experiments features accurate line separation at a 3–4 cm detection range, generating accurate point clouds with high frequencies of 50 Hz in indoor settings. The proposed distributed estimation framework enables processing and publishing the pose estimates with a sample time  $T_s = 50$  ms. Considering the maximum robot speed of  $v_x = 1.2$  m/s, the maximum range of 70 m for indoor operation, and the sampling time of 50 ms (including the process required for clustering/estimation), the utilized LiDAR enables accurate estimation (RMSE < 13 cm even with occlusion) with various robot speeds. For other outdoor settings with higher vehicle speeds, e.g., intelligent transportation, the correlation between the speed and accuracy of LiDAR point clouds is an important aspect and needs to be taken into account.

The computational times of the proposed distributed state observer are shown in Figure 9 for both scenarios (with occlusions) using a 6-core NVIDIA Carmel ARM 1.9 GHz

with 16 GB memory (available through *Jetson Xavier NX* embedded systems for edge computing purposes), and confirm the computational efficiency of the framework, i.e., <100 ms, which is sufficient for the accurate motion planning of mobile robots with maximum speeds of 2 m/s, and is important for scalability in larger sensor networks.



**Figure 9.** Computational time of the proposed LiDAR-based remote sensing framework for Scenario 1 (**top**) and Scenario 2 (**bottom**) subject to occlusion and dynamic objects moving around the robot.

## 5. Conclusions

In this paper, a remote LiDAR-based state estimation (and tracking) framework is developed and experimentally validated for mobile robots in distributed sensor networks by designing a consensus information filter and computationally efficient point cloud processing. As confirmed by experiments, the proposed approach hinges on two primary contributions: a LiDAR-based 3D bounding box detection and tracking method that expands the versatility of sensor placements in distributed sensor networks, and a node-switching distributed observer to address occlusion and uncertainties in dynamic environments, achieving average estimation errors as low as 2 cm and  $2^\circ$  under nominal conditions (<4 cm and  $<3.5^\circ$  under occlusions). These advancements significantly reduce computational demands (i.e., less than 100 ms) while incorporating short-term motion predictive capabilities, thereby enhancing the scalability and efficiency of mobile robot tracking in dynamic scenes with human presence while maintaining privacy. Through experimental evaluations and high-fidelity simulations in indoor settings, the effectiveness of the framework in terms of the accuracy of the estimation and the asymptotic stability of the error dynamics is demonstrated in normal and arduous scenarios (including occlusion, limited visibility, and dynamic obstacles) and compared with the Kalman consensus filter. Future work will focus on further refining the predictive modeling in large sensor networks and exploring its applications in unstructured operational settings with multi-robot tracking. Additionally, research will aim to move beyond the assumption of flat terrain and negligible z-axis variations in robot positions to handle more complex and varied terrain for field applications with stationary sensing units.

**Author Contributions:** Conceptualization, E.H. and I.L.; methodology, I.L., N.P.B. and E.H.; writing, I.L., N.P.B. and E.H.; simulation, I.L. and N.P.B.; data analysis, I.L. and E.H.; visualization, I.L.; review and editing, I.L., N.P.B. and E.H.; funding acquisition, E.H.; project administration, E.H.; supervision, E.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Natural Sciences and Engineering Research Council of Canada RGPIN-2020-05097, and in part by the New Frontiers in Research Fund Grant NFRFE-2022-00795.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

LiDAR	Light detection and ranging.
DSN	distributed sensor networks.
PCA	principal component analysis.

### References

1. Rodríguez-Canosa, G.; Del Cerro Giner, J.; Barrientos, A. Detection and Tracking of Dynamic Objects by Using a Multirobot System: Application to Critical Infrastructures Surveillance. *Sensors* **2014**, *14*, 2911–2943. [[CrossRef](#)] [[PubMed](#)]
2. Li, S.; Kong, R.; Guo, Y. Cooperative distributed source seeking by multiple robots: Algorithms and experiments. *IEEE/ASME Trans. Mechatron.* **2014**, *19*, 1810–1820. [[CrossRef](#)]
3. Kuutti, S.; Fallah, S.; Katsaros, K.; Dianati, M.; McCullough, F.; Mouzakitis, A. A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications. *IEEE Internet Things J.* **2018**, *5*, 829–846. [[CrossRef](#)]
4. Nazir, U.; Shahid, N.; Arshad, M.; Raza, S. Classification of localization algorithms for wireless sensor network: A survey. In Proceedings of the 2012 International Conference On Open Source Systems And Technologies, Lahore, Pakistan, 20–22 December 2012; pp. 1–5.
5. Khelifi, M.; Benyahia, I.; Moussaoui, S.; Naït-Abdesselam, F. An overview of localization algorithms in mobile wireless sensor networks. In Proceedings of the 2015 International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS), Paris, France, 22–24 July 2015; pp. 1–6.
6. Liu, J.; Chu, M.; Reich, J. Multitarget Tracking in Distributed Sensor Networks. *IEEE Signal Process. Mag.* **2007**, *24*, 36–46. [[CrossRef](#)]
7. Olfati-Saber, R.; Jalalkamali, P. Coupled distributed estimation and control for mobile sensor networks. *IEEE Trans. Autom. Control* **2012**, *57*, 2609–2614. [[CrossRef](#)]
8. Park, S.; Martins, N. Design of distributed LTI observers for state omniscience. *IEEE Trans. Autom. Control* **2016**, *62*, 561–576. [[CrossRef](#)]
9. Lederer, A.; Yang, Z.; Jiao, J.; Hirche, S. Cooperative Control of Uncertain Multiagent Systems via Distributed Gaussian Processes. *IEEE Trans. Autom. Control* **2023**, *68*, 3091–3098. [[CrossRef](#)]
10. Yao, Z.; Gupta, K. Distributed Roadmaps for Robot Navigation in Sensor Networks. *IEEE Trans. Robot.* **2011**, *27*, 997–1004. [[CrossRef](#)]
11. Cai, Z.; Liu, J.; Chi, W.; Zhang, B. A Low-Cost and Robust Multi-Sensor Data Fusion Scheme for Heterogeneous Multi-Robot Cooperative Positioning in Indoor Environments. *Remote Sens.* **2023**, *15*, 5584. [[CrossRef](#)]
12. Salimzadeh, A.; Bhatt, N.; Hashemi, E. Augmented Visual Localization Using a Monocular Camera for Autonomous Mobile Robots. In Proceedings of the 2022 IEEE 18th International Conference On Automation Science And Engineering (CASE), Mexico City, Mexico, 22–26 August 2022; pp. 1124–1131.
13. We, J.; Zhang, X. Multiple visual-targets tracking in decentralized wireless camera sensor networks. In Proceedings of the 2010—MILCOM 2010 Military Communications Conference, San Jose, CA, USA, 1 October–3 November 2010; pp. 1056–1061.
14. Karakaya, M.; Qi, H. Detection-based tracking for crowded targets in distributed visual sensor networks. In Proceedings of the 2011 Future of Instrumentation International Workshop (FIIW) Proceedings, Oak Ridge, TN, USA, 7–8 November 2011; pp. 118–121.
15. Fortin, B.; Lherbier, R.; Noyer, J. A Model-Based Joint Detection and Tracking Approach for Multi-Vehicle Tracking With Lidar Sensor. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 1883–1895. [[CrossRef](#)]
16. Liu, J.; Pu, J.; Sun, L.; He, Z. An Approach to Robust INS/UWB Integrated Positioning for Autonomous Indoor Mobile Robots. *Sensors* **2019**, *19*, 950. [[CrossRef](#)] [[PubMed](#)]
17. Yu, X.; Li, Q.; Queraltá, J.P.; Heikkonen, J.; Westerlund, T. Applications of UWB Networks and Positioning to Autonomous Robots and Industrial Systems. In Proceedings of the 2021 10th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 7–10 June 2021; pp. 1–6. [[CrossRef](#)]
18. Ge, X.; Han, Q.; Zhang, X.; Ding, L.; Yang, F. Distributed event-triggered estimation over sensor networks: A survey. *IEEE Trans. Cybern.* **2019**, *50*, 1306–1320. [[CrossRef](#)] [[PubMed](#)]
19. Maleki, S.; McDonald, A.; Hashemi, E. Visual Navigation for Autonomous Mobile Material Deposition Systems using Remote Sensing. In Proceedings of the ITSC 2023, Québec City, QC, Canada, 22–25 May 2023; pp. 22–29.
20. Battistelli, G.; Chisci, L. Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability. *Automatica* **2014**, *50*, 707–718. [[CrossRef](#)]
21. He, X.; Hashemi, E.; Johansson, K. Navigating a mobile robot using switching distributed sensor networks. In Proceedings of the 2023 European Control Conference (ECC), Bucharest, Romania, 13–16 June 2023; pp. 1–6.

22. He, X.; Hashemi, E.; Johansson, K. Event-Triggered Task-Switching Control Based on Distributed Estimation. *IFAC-PapersOnLine* **2020**, *53*, 3198–3203. [[CrossRef](#)]
23. Song, Y.; Guan, M.; Tay, W.; Law, C.; Wen, C. Uwb/lidar fusion for cooperative range-only slam. In Proceedings of the 2019 International Conference On Robotics And Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 6568–6574.
24. Akiyama, K.; Azuma, K.; Shinkuma, R.; Shiomi, J. Real-time adaptive data transmission against various traffic load in multi-LIDAR sensor network for indoor monitoring. *IEEE Sens. J.* **2023**, *23*, 17676–17689. [[CrossRef](#)]
25. Akiyama, K.; Shinkuma, R.; Yamamoto, C.; Saito, M.; Ito, T.; Nihei, K.; Iwai, T. Edge computing system with multi-LIDAR sensor network for robustness of autonomous personal-mobility. In Proceedings of the 2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW), Bologna, Italy, 10–13 July 2022; pp. 290–295.
26. Rausch, M.; Feher, G. Stationary LIDAR Sensors for Indoor Quadcopter Localization. In Proceedings of the IEEE SENSORS, Rotterdam, The Netherlands, 25–28 October 2020; pp. 1–4.
27. Flögel, D.; Bhatt, N.; Hashemi, E. Infrastructure-Aided Localization and State Estimation for Autonomous Mobile Robots. *Robotics* **2022**, *11*, 82. [[CrossRef](#)]
28. Azim, A.; Aycard, O. Detection, classification and tracking of moving objects in a 3D environment. In Proceedings of the 2012 IEEE Intelligent Vehicles Symposium, Madrid, Spain, 3–7 June 2012; pp. 802–807.
29. Naujoks, B.; Wuensche, H. An Orientation Corrected Bounding Box Fit Based on the Convex Hull under Real Time Constraints. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Suzhou, China, 26–30 June 2018; pp. 1–6.
30. Najdataei, H.; Nikolakopoulos, Y.; Gulisano, V.; Papatriantafillou, M. Continuous and Parallel LiDAR Point-Cloud Clustering. In Proceedings of the 2018 IEEE 38th International Conference On Distributed Computing Systems (ICDCS), Vienna, Austria, 2–5 July 2018; pp. 671–684.
31. Zermas, D.; Izzat, I.; Papanikolopoulos, N. Fast segmentation of 3D point clouds: A paradigm on LiDAR data for autonomous vehicle applications. In Proceedings of the 2017 IEEE International Conference On Robotics And Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 5067–5073.
32. Saravananarajan, V.; Chen, R.; Chen, L. LiDAR Point Cloud Data Processing in Autonomous Vehicles. In Proceedings of the 2021 Fourth International Conference On Electrical, Computer And Communication Technologies (ICECCT), Erode, India, 15–17 September 2021; pp. 1–5.
33. Douillard, B.; Underwood, J.; Kuntz, N.; Vlaskine, V.; Quadros, A.; Morton, P.; Frenkel, A. On the segmentation of 3D LIDAR point clouds. In Proceedings of the 2011 IEEE International Conference on Robotics And Automation, Shanghai, China, 9–13 May 2011; pp. 2798–2805.
34. Liang, C.; Xue, W.; Fang, H.; He, X.; Gupta, V. On consistency and stability of distributed Kalman filter under mismatched noise covariance and uncertain dynamics. *Automatica* **2023**, *153*, 111022. [[CrossRef](#)]
35. Olfati-Saber, R. Kalman-consensus filter: Optimality, stability, and performance. In Proceedings of the 48th IEEE Conference on Decision And Control (CDC) Held Jointly with 2009 28th Chinese Control Conference, Shanghai, China, 15–18 December 2009; pp. 7036–7042.
36. Qu, S.; Chen, G.; Ye, C.; Lu, F.; Wang, F.; Xu, Z.; Gel, Y. An Efficient L-Shape Fitting Method for Vehicle Pose Detection with 2D LiDAR. In Proceedings of the 2018 IEEE International Conference on Robotics And Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 12–15 December 2018; pp. 1159–1164.
37. Zhao, C.; Fu, C.; Dolan, J.; Wang, J. L-Shape Fitting-Based Vehicle Pose Estimation and Tracking Using 3D-LiDAR. *IEEE Trans. Intell. Veh.* **2021**, *6*, 787–798. [[CrossRef](#)]
38. Olfati-Saber, R.; Jalalkamali, P. Collaborative target tracking using distributed Kalman filtering on mobile sensor networks. In Proceedings of the 2011 American Control Conference, San Francisco, CA, USA, 29 June 29–1 July 2011; pp. 1100–1105.
39. Kamal, A.T.; Ding, C.; Song, B.; Farrell, J.A.; Roy-Chowdhury, A.K. A Generalized Kalman Consensus Filter for wide-area video networks. In Proceedings of the 2011 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, FL, USA, 12–15 December 2011; pp. 7863–7869.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.