IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 February 2026, accepted 7 March 2026, date of publication 11 March 2026, date of current version 24 March 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3672927

RESEARCH ARTICLE

Self-Alignment-by-Tracking With a Mobile Single Robot

JAEUK BAEK[1], SEOKHO JEONG[2], DONGGYU CHOI[1], AND CHANGEUN LEE[1,3]
[1]Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea
[2]Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea
[3]University of Science and Technology, Daejeon 34113, Republic of Korea

Corresponding author: Changeun Lee (celee@etri.re.kr)

This work was supported by Korea Research Institute for Defense Technology Planning and Advancement (KRIT) grant funded by Korean Government (Defense Acquisition Program Administration (DAPA), ''Development of the Situation/Environment Recognition Technology for Micro-Swarm Robot'' under Grant KRIT-CT-22-006-002.

ABSTRACT In this paper, we address multiple object tracking (MOT) using a single mobile robot without any prior knowledge of its location or orientation. In scenarios where both objects and the robot are moving, continuous object tracking becomes challenging due to the lack of a consistent coordinate system. To overcome this, we propose a self-alignment-by-tracking mechanism to maintain spatio-temporal consistency. We formulate an optimization problem aimed at maximizing multiple object tracking accuracy (MOTA), but its combinatorial nature makes it difficult to solve directly. To address this, we decompose the problem into two subproblems: 1) consecutive spatial alignment of objects over time sequences to a common reference coordinate system and 2) object tracking within the same coordinate system for temporal consistency. Compared to conventional tracking processes, our proposed self-alignment-by-tracking mechanism requires an additional step of spatial alignment before the data association step, which may introduce inter-frame delays. However, we mitigate this issue by employing a geometry-based sorting mechanism, allowing us to search only a subset of the sorted neighboring objects. In addition, robust spatial alignment can be achieved by generating pseudo-environment objects through a closed-form expression, ensuring reliability even under challenging matching conditions. We analyze theoretical computational complexity, along with Monte Carlo simulations and real-world experiments, which validate the superiority of our proposed MOT process in terms of both accuracy and computational efficiency.

INDEX TERMS Multiple object tracking (MOT), mobile robots, spatio-temporal consistency, correspondence matching.

I. INTRODUCTION

Although many humanoid robots have been developed to replicate human abilities, small-sized robots have also been increasingly developed due to their mobility, covert nature, and versatility in various applications. However, many challenges have arisen in the widespread use of small-sized robots [1], [2], [3], [4]. First, limited battery life requires the robot to be an energy-efficient system, where multiple sensors that could waste energy cannot be implemented. Second, a small-sized sensor capable of providing sufficient environmental information needs to be used rather than a large one, such as Light Detection and Ranging (LiDAR). In such cases, a depth camera sensor may be the only viable option, as images and depth information can be used for object detection and localization, respectively [5], [6], [7].

Multiple object tracking (MOT) using a single depth camera mounted on a mobile robot offers several advantages over a stationary one, as it enables continuous tracking of moving objects, improves occlusion handling, and reduces ID switching caused by limited viewpoints. Many object tracking algorithms assume that only objects are moving, while the tracking source remains stationary, or the position and orientation of the moving robot are known [8], [9].

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

However, this is a strong assumption, especially in indoor environments, as it relies on position-aware sensors, such as LiDAR and gyroscope sensors, which are used in Simultaneous Localization and Mapping (SLAM) algorithms to create maps and determine position and orientation in a global coordinate system [10], [11]. When these sensors are not feasible due to size and power constraints, object tracking must operate with partial information about objects, such as location information, in a relative coordinate system rather than in a global one.

II. RELATED WORKS AND CONTRIBUTIONS

A. TWO APPROACHES FOR MOT IN MOBILE ROBOT

MOT in a mobile robot can be closely related to the tracking and localization [12]. When the robot or camera remains stationary, the locations of all detected objects are mapped into a fixed coordinate frame, allowing the system to focus primarily on object tracking. However, when the robot is in motion, its local (or global) position changes, and the tracking results must be continuously mapped to the corresponding local or global coordinate system to maintain robustness and consistency in object tracking.

There are two primary approaches for MOT in a mobile robot: (1) tracking-then-projection and (2) projection-then-tracking. In the first approach, objects are tracked in the image (pixel) coordinate space. The bounding boxes and corresponding image features are utilized by the tracking algorithm, and each tracked object is subsequently projected into the world coordinate system. This method has been extensively developed, benefiting from advances in AI-driven filtering algorithms [13], but the relative movements of robots and target objects introduce significant challenges such as out-of-scene phenomena, occlusion, and partial detection. These issues arise because the temporal continuity of the image stream is disrupted by variations in the scale and field of view (FoV) of the camera during robot motion.

However, the projection-then-tracking approach can effectively address these challenges. By first projecting all perceptual data into the world coordinate system, the consistency and robustness of MOT can be achieved. For instance, the location of stationary environment objects in a previous frame can be used to predict and associate the location of target objects in the current frame by analyzing their geometric relationships. Therefore, the projection-then-tracking paradigm represents a promising solution for achieving reliable MOT in a mobile robot.

B. MOT WITH UNKNOWN GLOBAL POSITION

The literature in this research area can be interpreted as 'Local Coordinate-based Object Tracking' and 'Object Tracking Without Initial Correspondence' [14]. Since no global coordinate information is available, spatial, temporal, or spatio-temporal alignment is essential for consistent object tracking. In multi-robot systems, spatial alignment can be achieved by designating one robot as a [15]. For instance, [16] optimizes relative transformations in a global manner rather than processing each robot's information independently. Similarly, [17] proposes a transformed EKF (T-EKF) to ensure consistency in multi-robot cooperative localization. Graph-theoretic approaches have been proposed for spatial alignment [18], where a consistency graph is constructed, and subgraphs or cliques are identified [19], [20], [21]. Reference [19] identifies the maximal clique while managing outliers to ensure consistent spatial correspondence. CLIPPER in [20] has been validated to achieve near-optimal performance in correspondence matching for noisy data by constructing a spatial consistency graph. TCAFF in [21] extends CLIPPER to maintain spatio-temporal consistency for MOT. Specifically, in each frame, CLIPPER is executed multiple times to generate putative associations, while Multiple Hypothesis Tracking (MHT) [22] is applied to identify the best one.

In a single-robot system, however, spatially distributed information from other robots is unavailable, necessitating a self-contained spatio-temporal alignment mechanism. Specifically, to address the time-varying nature of local coordinates due to the robot's mobility, a self-alignment-by-tracking mechanism is the only viable option, where the current information is transformed into the reference frame accordingly. In this mechanism, the performance of MOT relies on accurate alignment for improved data association, while computational complexity must be minimized, as inter-frame delays can degrade MOT performance. Both accuracy and time complexity need to be considered simultaneously, but rarely studied due to its difficulty.

C. DENSE FEATURE UTILIZATION FOR MOT

Numerous studies have explored leveraging rich features through end-to-end deep learning frameworks for SLAM, and these methodologies have been extended to MOT. Reference [23] adopted a hypergraph-based spatio-temporal alignment to associate consecutive frames, [24] constructed factor graphs to integrate multi-modal features, and [25], [26], [27] utilized re-identification mechanism for robustness matching. These approaches generally operate under the assumption that dense features are available, allowing deep neural networks to effectively extract discriminative representations.

However, in scenarios where sparse features are prevalent and data association is restricted to a limited number of points, relying solely on heavy deep learning architectures can be inefficient. In such cases, motion aided-probabilistic approaches can serve as a more robust and lightweight alternative. Furthermore, as small mobile robots frequently operate on Single-Board Computers (SBCs) with limited computational resources, the utilization of deep learning frameworks should be strategically minimized. Rather than an end-to-end deep learning pipeline, a hybrid approach [13] is required, where a deep learning network is selectively employed and core tracking and alignment processes are governed by computationally efficient probabilistic models.

D. MAIN CONTRIBUTIONS

In this paper, we adopt a projection-then-tracking process and propose a MOT process using a mobile single robot, assuming no prior information about its location and orientation. To achieve this, we formulate an optimization problem that aims to maximize multiple object tracking accuracy (MOTA) based on a self-alignment-by-tracking mechanism. However, solving this problem directly is challenging due to its combinatorial nature. To address this, we decompose the original problem into two subproblems: (1) consecutive coordinate transformation of objects over time sequences and (2) object tracking within the same coordinate system. Since environment objects, such as obstacles and background elements, remain stationary, they can serve as landmark points for alignment, particularly in the absence of location and orientation information for the robot. Therefore, we implement a self-alignment-by-tracking mechanism, where stationary environment objects between the reference and current frames are matched to obtain alignment information, followed by moving objects alignment for data association in the tracking process.

To overcome the critical scenario with insufficient landmarks, we leverage probabilistic models to generate pseudo-environment objects for robust transformation, enabling reliable tracking in resource-constrained and feature sparse environments.

The main contributions of this paper are as follows.

- We propose a self-alignment-by-tracking mechanism to enable MOT in a mobile robot without any knowledge of its location and orientation. To maintain spatio-temporal consistency, the environment objects from the reference and the current frame are spatially matched, then the alignment information is applied to moving objects for temporal consistency.
- Compared to the object tracking within the same coordinate system, spatial alignment in a time-varying coordinate system introduces an additional step before the conventional data association, potentially causing inter-frame delays. To mitigate this, we reduce the complexity of spatial alignment using a geometry-based sorting mechanism, thereby avoiding the bottleneck of constructing and searching the entire consistency graph in [20].
- To address feature-deficient environments, we generate pseudo-environment objects to achieve robust spatial alignment through a closed-form expression. This approach ensures high computational efficiency, making it suitable for real-time processing on resource-constrained hardware.
- We analyze the theoretical computational complexity, and conduct Monte Carlo simulations and real-world experiments. The results validate the superiority of our proposed MOT process in terms of both accuracy and computational complexity.

The rest of this paper is organized as follows. Section III presents the preliminaries of the self-alignment-by-tracking mechanism. Section IV provides our system model and formulates the optimization problem. Section V introduces our proposed MOT process. Section VI presents the simulation results, followed by the conclusion in Section VII.

III. PRELIMINARIES

A. MATCHING PROBLEM

Given the object lists in Source $\boldsymbol{O_s}$ and Target $\boldsymbol{O_t}$, matching between these lists can be obtained by solving the following optimization problem:

$$\mathcal{Z}_{\boldsymbol{M}}(\boldsymbol{O_s}, \boldsymbol{O_t}) = \max_{\forall \boldsymbol{M}} \sum_{(i,j)\in\boldsymbol{M}} z_{ij}, \tag{1}$$

$$\text{s.t.} \sum_{j:(i,j)\in\boldsymbol{M}} z_{ij} \le 1, \quad \forall i \in \boldsymbol{O_s}, \tag{1a}$$

$$\sum_{i:(i,j)\in\boldsymbol{M}} z_{ij} \le 1, \quad \forall j \in \boldsymbol{O_t}, \tag{1b}$$

$$\boldsymbol{D}(\boldsymbol{M}) \le \zeta, \tag{1c}$$

where $\boldsymbol{M} \triangleq \{(i,j) | i \in \boldsymbol{O_s}, j \in \boldsymbol{O_t}\}$ is the set of matched index pairs. z_{ij} is a binary variable representing:

$$z_{ij} = \begin{cases} 1, & \text{if } \boldsymbol{O_s}(i) \text{ and } \boldsymbol{O_t}(j) \text{ are matched,} \\ 0, & \text{otherwise.} \end{cases}$$

(1a) and (1b) are binary constraints, while (1c) represents the constraint of local structure consistency with an upper bound ζ, where

$$\boldsymbol{D}(\boldsymbol{M}) = \max_{\forall (i,j)\in\boldsymbol{M}} ||\boldsymbol{O_s}(i) - \boldsymbol{O_s}(i')|| - ||\boldsymbol{O_t}(j) - \boldsymbol{O_t}(j')||, \tag{2}$$

$$\text{s.t. } (i', j') \in \boldsymbol{M} \setminus \{(i,j)\}. \tag{2a}$$

The matching problem $\mathcal{Z}_{\boldsymbol{M}}(\boldsymbol{O_s}, \boldsymbol{O_t})$ in (1) is a combinatorial problem, and its complexity increases exponentially with the size of $\boldsymbol{O_s}$ and $\boldsymbol{O_t}$ [28]. In addition, the upper bound of local structure consistency, ζ, determines the existence of a global optimal solution. No solution exists as ζ is close to zero, while multiple suboptimal solutions may exist as ζ increases. Note that ζ generally depends on system capabilities and must be derived from applications or test environments.

Remark 1: The objective function in (1) can be replaced by the constraint (1c) to formulate a new optimization problem, which minimizes the error of local structure consistency while satisfying the matching constraints. However, this optimization problem can yield results with no constraints on the number of matching pairs, which fails to address the uncertainty (or multi-modality) that arises when only two matching pairs exist.

Remark 2: In CLIPPER, the matching problem $\mathcal{Z}_{\boldsymbol{M}}(\boldsymbol{O_s}, \boldsymbol{O_t})$ is solved to optimize the maximum spectral radius clique (MSRC), where the consistency graph of $\boldsymbol{O_s}$ and $\boldsymbol{O_t}$ is constructed to obtain the affinity function and a non-convex continuous relaxation is applied to derive optimal

matching. This algorithm is effective for searching for a near-optimal solution, but constructing the entire consistency graph and its affinity function can be a bottleneck in real-time object tracking.

B. COORDINATE TRANSFORMATION

Once optimal matching $\boldsymbol{M}^*$ between $\boldsymbol{O_s}$ and $\boldsymbol{O_t}$ is obtained from (1), the coordinates of objects in $\boldsymbol{O_t}$ can be transformed into the coordinate system of $\boldsymbol{O_s}$ by deriving the corresponding rotation and translation matrices. Let $\boldsymbol{X}_{\boldsymbol{s}|\boldsymbol{M}^*}$ and $\boldsymbol{X}_{\boldsymbol{t}|\boldsymbol{M}^*}$ denote the coordinates of the source and target objects corresponding to the matching $\boldsymbol{M}^*$, respectively. Then, $\boldsymbol{X}_{\boldsymbol{t}|\boldsymbol{M}^*}$ can be transformed into $\boldsymbol{X}_{\boldsymbol{t}\to\boldsymbol{s}|\boldsymbol{M}^*}$, within the source coordinate system using Algorithm 1.

Algorithm 1 $\boldsymbol{X}_{\boldsymbol{t}\to\boldsymbol{s}|\boldsymbol{M}^*} = \mathcal{F}(\boldsymbol{X}_{\boldsymbol{t}|\boldsymbol{M}^*}, \boldsymbol{X}_{\boldsymbol{s}|\boldsymbol{M}^*})$

Input: $\boldsymbol{X}_{\boldsymbol{s}|\boldsymbol{M}^*}, \boldsymbol{X}_{\boldsymbol{t}|\boldsymbol{M}^*}$
Output: $\boldsymbol{X}_{\boldsymbol{t}\to\boldsymbol{s}|\boldsymbol{M}^*}$
Procedure
1: compute the average of the coordinates
: $\mu_{\boldsymbol{s}|\boldsymbol{M}^*} = \text{mean}(\boldsymbol{X}_{\boldsymbol{s}|\boldsymbol{M}^*})$
: $\mu_{\boldsymbol{t}|\boldsymbol{M}^*} = \text{mean}(\boldsymbol{X}_{\boldsymbol{t}|\boldsymbol{M}^*})$
2: compute cross covariance
: $\mathbf{cov} = \mathbb{E}\big[(\boldsymbol{X}_{\boldsymbol{s}|\boldsymbol{M}^*} - \mu_{\boldsymbol{s}|\boldsymbol{M}^*})(\boldsymbol{X}_{\boldsymbol{t}|\boldsymbol{M}^*} - \mu_{\boldsymbol{t}|\boldsymbol{M}^*})^T\big]$
3: apply singular value decomposition (SVD)
: $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \mathbf{S}VD(\mathbf{cov})$
4: calculate rotation matrix
: $\mathbf{R} = \mathbf{U}\mathbf{V}^T$
5: calculate translation matrix
: $\mathbf{T} = (\boldsymbol{X}_{\boldsymbol{s}|\boldsymbol{M}^*} - \mu_{\boldsymbol{s}|\boldsymbol{M}^*})^T - \mathbf{R}(\boldsymbol{X}_{\boldsymbol{t}|\boldsymbol{M}^*} - \mu_{\boldsymbol{t}|\boldsymbol{M}^*})^T$
6: obtain coordinate transformation results
: $\boldsymbol{X}_{\boldsymbol{t}\to\boldsymbol{s}|\boldsymbol{M}^*} = \mathbf{R}\boldsymbol{X}_{\boldsymbol{t}|\boldsymbol{M}^*}^T + \mathbf{T}$

C. MULTIPLE OBJECT TRACKING

In MOT, resolving the uncertainty in data association between measurements and tracking states is critical. When measurements from other objects are incorporated, false positive (FP), false negative (FN), or identity switch (IDSW) can occur, which degrades the accuracy of MOT [29]. In many studies and applications, the association of data is resolved using the well-known Hungarian Algorithm with various cost functions, such as the minimization of Euclidean distance and the maximization of likelihood probability [30]. Even with such efforts, measurements contaminated with severe errors create confusion about whether to begin new tracks or associate with existing track states [31].

Let $\{\Theta_k\}$ and $\{\boldsymbol{r}_k\}$ denote a set of trackers and measurements at frame k, respectively. Also, denote β_{age} and β_{hit} as the minimum requirements of the age and streak of the trackers. Then, overall Kalman-filter based object tracking is described in Algorithm 2.

Remark 3: In the same coordinate system, measurement errors, arising from limitations of sensing devices, are the dominant factors affecting the performance of MOT.

Algorithm 2 $\{\hat{\boldsymbol{P}}_{\boldsymbol{k+1}|\boldsymbol{k}}\} = \mathcal{T}(\{\Theta_k\}, \{\boldsymbol{r}_k\})$

Input: $\{\Theta_k\}, \{\boldsymbol{r}_k\}$
Output: $\{\hat{\boldsymbol{P}}_{\boldsymbol{k}|\boldsymbol{k}}\}$
Procedure
1: associate tracks and measurements.
: $\{\boldsymbol{r}_k^{\text{o}}, \Theta_k^{\text{o}}\}, \{\boldsymbol{r}_k^{\text{x}}, \Theta_k^{\text{x}}\} = \text{associate}(\{\Theta_k\}, \{\boldsymbol{r}_k\})$
2: update the state of all the associated trackers Θ_k^{o}.
: $\Theta_{k+1} = \text{update}(\{\boldsymbol{r}_k^{\text{o}}, \Theta_k^{\text{o}}\})$
3: update the age of all the unassociated trackers Θ_k^{x}.
: $\Theta_k^{\text{x}}(\text{age}) += 1$
4: create new trackers using all the unassociated measurements $\boldsymbol{r}_k^{\text{x}}$.
: $\Theta_{k+1} \leftarrow \boldsymbol{r}_k^{\text{x}}$
5: **for** each $\Theta \in \{\Theta_{k+1} \cup \Theta_k^{\text{x}}\}$ **do**
6: **if** $\Theta(\text{age}) < 1$ **or** $\Theta(\text{hit\_streak}) > \beta_{hit}$ **then**
7: extract object information $\hat{\boldsymbol{P}}_{\boldsymbol{k}|\boldsymbol{k}}$ from the state of Θ
8: $\Theta(\text{age}) \leftarrow 0$
9: **end if**
10: **if** $\Theta(\text{age}) > \beta_{age}$ **then**
11: delete Θ
12: **end if**
13: **end for**

However, when different coordinate systems need to be considered, additional errors may arise, primarily due to mismatched correspondence in the alignment process into a common coordinate system. Furthermore, as described in Remark 2, it is time consuming to solve the matching problem in (1) for coordinate transformation, leading to intermittent inputs to the tracking algorithm. This phenomenon presents another challenge for MOT in different coordinate systems, as objects appear to jump from one location to another (although this not the case).

IV. SELF-ALIGNMENT-BY-TRACKING WITH A MOBILE SINGLE ROBOT

We consider a single sensor system for MOT without knowledge of location and orientation over time. In other words, both objects and sensors have no restriction on their mobility, resulting in time-varying relative positions between them. For continuous MOT in such scenarios, we consider a self-alignment-by-tracking mechanism.

Without loss of generality, objects can be classified based on their mobility: moving objects are free to move along any trajectories without limitation, while environment objects remain stationary. The system leverages pre-trained object detection models to provide the information about classes and bounding boxes of objects [32]. In addition, depth measurements are used to calculate the two-dimensional locations of the detected objects.

A. PROBLEM FORMULATION

Let $\boldsymbol{X}_k = \boldsymbol{X}_k^m + \boldsymbol{X}_k^e$ denote a set of measurements at kth frame, where $\boldsymbol{X}_k^m$ and $\boldsymbol{X}_k^e$ represent measurements from moving and

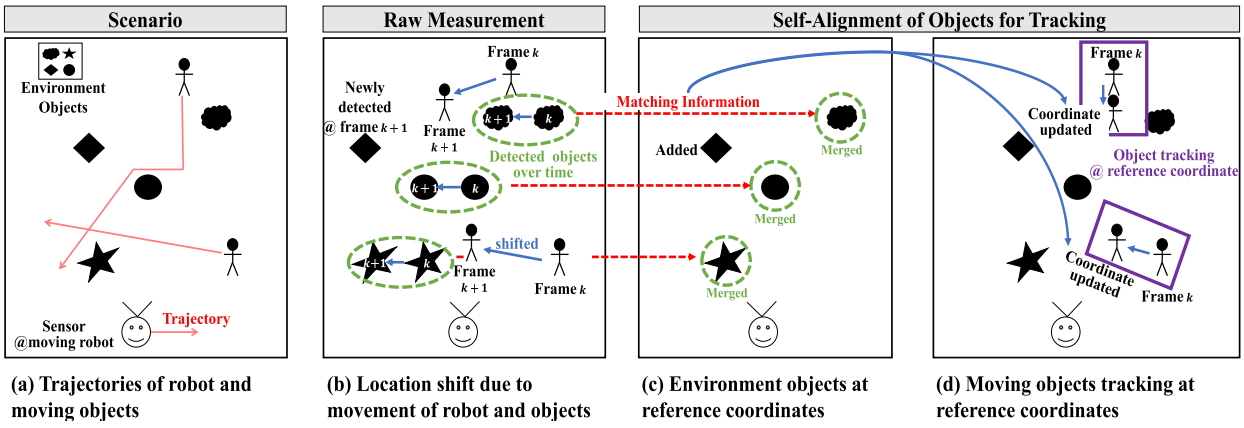


FIGURE 1. The overall MOT process in a single sensor mounted on moving agents, without any information about the sensor's location and orientation. (a, b) The robot's movement affects the perceived locations of all objects. In particular, the position of the moving object changes significantly due to the combination of its own motion and the robot-induced displacement, resulting in increased relative positional variation. (c) Spatio-temporal alignment of environment objects enables the extraction of correspondence information. (d) The extracted information is subsequently used to align moving objects, thereby facilitating object tracking in a unified reference coordinate system.

environment objects, respectively. Then, MOT problem under a self-alignment-by-tracking mechanism can be represented for each frame k as

$$\max_{\boldsymbol{M}^*, \boldsymbol{X}_{ref}} \quad \text{MOTA}(\hat{\boldsymbol{P}}_{k|k}), \tag{3}$$

$$\text{s.t. } \hat{\boldsymbol{P}}_{k|k} = \mathcal{T}\Big(\{\Theta_k\}, \{\boldsymbol{X}_{k\to ref|\boldsymbol{M}^*}\}\Big), \tag{3a}$$

$$\boldsymbol{X}_{k\to ref|\boldsymbol{M}^*} = \mathcal{F}(\boldsymbol{X}_{k|\boldsymbol{M}^*}, \boldsymbol{X}_{ref|\boldsymbol{M}^*}), \tag{3b}$$

$$\boldsymbol{M}^* = \mathcal{Z}_{\boldsymbol{M}}(\boldsymbol{X}_{ref}, \boldsymbol{X}_k), \tag{3c}$$

where MOTA represents accuracy of MOT [29]. The functions $\mathcal{T}$ and $\mathcal{F}$ are described in detail in Algorithms 1 and 2, respectively. (3c) represents the optimal matching in problem (1) between the current measurements, $\boldsymbol{X}_k$, and the measurements at the reference frame $\boldsymbol{X}_{ref}$.

In order to maximize the objective function in (3), we need to carefully determine $\boldsymbol{X}_{ref}$, as its coordinate system is applied to all the object trackings. In addition, (3c) has a critical impact on data association in tracking process, therefore, it must be solved accurately in a real-time manner.

B. PROBLEM REFORMULATION

In (3), $\boldsymbol{M}^*$ plays a key role in two ways: 1) mismatching objects over time leads to incorrect translation and rotation matrices to the reference coordinate system, which can result in anomaly points during the tracking process, and 2) a time-consuming matching algorithm causes the transformed measurements to become intermittent, potentially leading to missed objects.

While the constraint (3c) ensures the consistency of local structure in matching, maximizing its objective function is not ideal for MOT due to the time-consuming process. Instead, finding the minimum number of matchings can be beneficial as it allows all transformed measurements to be utilized in each frame.

By relaxing the constraint (3c) into $|\boldsymbol{M}'| \geq N$, where

$$\boldsymbol{M}' = \left\{ \begin{array}{l} ||\boldsymbol{O}_s(i) - \boldsymbol{O}_s(i')|| - ||\boldsymbol{O}_t(j) - \boldsymbol{O}_t(j')|| \leq \zeta, \\ (i,j) \Big| \text{s.t.} \quad i, i' \in \boldsymbol{O}_s, i \neq i', \\ \qquad\qquad j, j' \in \boldsymbol{O}_t, j \neq j' \end{array} \right\},$$

we can reformulate the constraint (3c) as the task of finding at least N matching elements. Then, the problem in (3) at frame k can be reformulated as

$$\max_{\boldsymbol{M}', \boldsymbol{X}_{ref}} \quad \text{MOTA}(\hat{\boldsymbol{P}}_{k|k}), \tag{4}$$

$$\text{s.t. } \hat{\boldsymbol{P}}_{k|k} = \mathcal{T}\Big(\{\Theta_k\}, \{\boldsymbol{X}_{k\to ref|\boldsymbol{M}'}\}\Big), \tag{4a}$$

$$\boldsymbol{X}_{k\to ref|\boldsymbol{M}'} = \mathcal{F}(\boldsymbol{X}_{k|\boldsymbol{M}'}, \boldsymbol{X}_{ref|\boldsymbol{M}'}), \tag{4b}$$

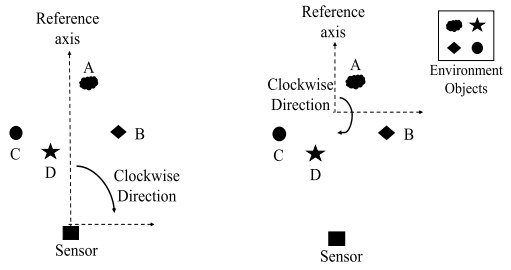
$$|\boldsymbol{M}'| \geq N. \tag{4c}$$

Remark 4: $N(\geq 3)$ needs to be set as the minimum requirement for obtaining unique translated location because $N = 2$ may lead to multiple possible solutions [31]. Although many matching pairs (i, j) can exist in $|\boldsymbol{M}'| = N$, they do not pose a problem as long as the matching preserves a consistent local structure. Furthermore, since we do not need to search through all matching pairs, we can reduce the search time for finding the matching pairs, thereby avoiding intermittent measurements as input in MOT.

V. PROPOSED MOT PROCESS

Since the problem in (4) cannot be solved optimally, we propose the MOT process as a suboptimal approach, as illustrated in Fig. 1. This process consists of three main components: 1) selection and updating of the reference frame, 2) measurement processing for coordinate unification, and 3) Kalman filter-based object tracking.

In the following subsections, we first describe the fast matching algorithm for a given $\boldsymbol{X}_{ref}$, followed by the method for determining $\boldsymbol{X}_{ref}$. Then, the complexity and optimality of the proposed MOT process are analyzed.



(a) Sensor location. (b) Center location of the objects.

FIGURE 2. Two different axis origins. Environment objects sorted in a clockwise direction based on the axis origin.

A. FAST MATCHING ALGORITHM FOR A GIVEN X_{REF}

When $\boldsymbol{X_{ref}}$ is given, the problem in (4) can be solved by finding $|\boldsymbol{M}'| = N$, and then applying the functions $\mathcal{F}$ and $\mathcal{T}$, sequentially. However, the matching $\boldsymbol{M}'$ of all measurements between $\boldsymbol{X_{ref}}$ and $\boldsymbol{X_k}$ can provide incorrect transformation information, because both sensors and moving objects are moving. In other words, the current location of moving objects at frame k needs to be tracked based on the matching information, rather than being used as a prior information for the matching process. Since the environment objects remain stationary, these locations (i.e., $\boldsymbol{X}^e_{\boldsymbol{ref}}$ and $\boldsymbol{X_k}^e$) can be used to determine $\boldsymbol{M}'$.

Let $\mathcal{G}(\boldsymbol{X}_k)$ represent a local structure graph, where each node contains its own class information and the edge between two nodes is weighted by the distance. Once $\mathcal{G}(\boldsymbol{X_{ref}}^e)$ and $\mathcal{G}(\boldsymbol{X_k}^e)$ are constructed, $|\boldsymbol{M}'| = N$ can be solved iteratively by searching for pairs of nodes with the same class information, along with the adjacent nodes that have similar weights until N pairs of $\boldsymbol{X}^e_{\boldsymbol{ref}}$ and $\boldsymbol{X_k}^e$ are found.

To speed up the search process, it is not necessary to construct the entire of $\mathcal{G}(\boldsymbol{X}^e_k)$ and $\mathcal{G}(\boldsymbol{X_k}^e)$. Instead, we can sort the environment objects based on their geometry and search for them using partial information. Fig. 2 illustrates two different axis origins that can be used for sorting the environment objects in clockwise (or counterclockwise) direction. Fig. 2a represents the sensor location as the axis origin, where the objects are sorted based on their angles relative to the sensor. However, ambiguity can arise when multiple objects are located close to the reference axis, since the angle to the axis origin is the only parameter used to distinguish them. For instance, while we might expect the objects to be sorted in a clockwise order, such as [A, B, D, C], Fig. 2a could result in an order like [A, B, C, D]. To address this problem, the axis origin can be set at the center location of the objects, rather than at the sensor location (as shown in Fig. 2b). Once the environment objects are sorted, search process for satisfying $|\boldsymbol{M}'| = N$ can be performed efficiently, as adjacent objects are already aligned.

Remark 5: In extreme cases where the environment objects exhibit a highly symmetric geometry, ambiguity may arise in solving (4c) to estimate $\boldsymbol{M}'$. Although Section VI demonstrates that accurate matching can generally be achieved across various experimental settings, further improvement can be possible by introducing a cosine similarity constraint with a threshold γ into (4c) to mitigate such uncertainty:

$$\boldsymbol{M}' = \left\{ \begin{array}{l} ||\boldsymbol{O_s}(i) - \boldsymbol{O_s}(i')|| - ||\boldsymbol{O_t}(j) - \boldsymbol{O_t}(j')|| \leq \zeta, \\ (i,j) \Big| \text{s.t.} \quad \frac{\mathbf{v}_s(i)^T \mathbf{v}_t(j)}{\|\mathbf{v}_s(i)\| \|\mathbf{v}_t(j)\|} \geq \gamma, \\ \qquad i, i' \in \boldsymbol{O_s}, i \neq i', \\ \qquad j, j' \in \boldsymbol{O_t}, j \neq j' \end{array} \right\},$$

where $\mathbf{v}_s(i)$ and $\mathbf{v}_t(j)$ represent the feature embeddings corresponding to the matched pair (i, j) from the source and target objects, respectively [13]. While image-space geometric features, constructed by concatenating the center point and shape of bounding boxes, can be utilized in a computationally efficient manner due to their invariance across consecutive frames, deep features, obtained by encoding the image region within the bounding box, offer higher representational capacity in resource-rich platforms. In addition, the cosine similarity threshold γ can be set empirically to suit different operational environment.

B. PSEUDO ENVIRONMENT OBJECT GENERATION FOR CHALLENGING MATCHING CONDITIONS

In this subsection, we elaborate to resolve the challenging problem of $N = 2$, as identified in Remarks 4, by generating pseudo environment object. For notational simplicity and without loss of generality, let $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ represent the environment objects in $\boldsymbol{X_{ref}}$, while $\boldsymbol{B'}$ and $\boldsymbol{C'}$ denote the corresponding objects in $\boldsymbol{X}_k$. Our object is to synthesize a pseudo object $\boldsymbol{A'}$ to establish a complete set of a matching pairs: $\{(\boldsymbol{A}, \boldsymbol{A'}), (\boldsymbol{B}, \boldsymbol{B'}), (\boldsymbol{C}, \boldsymbol{C'})\}$. To achieve this, we first evaluate the validity of potential correspondence by utilizing a likelihood-based correspondence score, followed by a refinement process to derive a robust pseudo object $\boldsymbol{A'}$.

1) LIKELIHOOD-BASED CORRESPONDENCE SELECTION

In the absence of ground-truth correspondence, it is necessary to evaluate all putative correspondence candidates. For a given pair set $\{(\boldsymbol{B}, \boldsymbol{B'}), (\boldsymbol{C}, \boldsymbol{C'})\}$, we identify the most reliable correspondence by maximizing the following score function

$$S_{int} = S_{disp} S_{cov} S_{dist}, \tag{5}$$

where

$$S_{disp} = \frac{1}{1 + \|\overrightarrow{BB'} - \overrightarrow{CC'}\|}, \tag{6}$$

$$S_{cov} = \det(\boldsymbol{\Sigma})^{-\zeta}, \tag{7}$$

$$S_{dist} = \exp\left(-\frac{\|\boldsymbol{A} - \boldsymbol{A}_{guess}\|^2}{2\sigma^2_{dist}} \right). \tag{8}$$

The integrated score function S_{int} serves as a stringent filtering criterion, which is maximized only when all constituent scores are simultaneously maximized. $S_{disp} \in (0, 1]$ is the displacement consistency score. $S_{cov} \in (0, 1]$ is the covariance-based score with a scale parameter ζ, where

Algorithm 3 Proposed MOT Process

Input: $\{X_k\}$, $N(\geq 3)$, $\eta(\geq 0.5)$
Output: $\{\hat{P}_{k|k}\}$
Procedure
Initialization and update of X^e_{ref}
for frame k **do**
  if $|X^e_k| \geq N$ and $|X^e_k| > |X^e_{ref}|$ **then**
    $X^e_{ref} \leftarrow X^e_k$
  end if
end for
align X^e_{ref} based on its center location
Matching
align X^e_k based on its center location.
if $|X^e_k| \leq 2$ **then**
  Find the best correspondence with $S_{int} > \eta$
  Generate pseudo environment object
else
  find M' until $|M'| = N$ based on their class label, angular consistency, and feature embedding sequentially.
end if
Coordinate Transformation
stack X^m_{ref} and $X^e_{ref|M'}$ to obtain $X_{ref|M'} = [X^m_{ref}, X^e_{ref|M'}]$
stack X^m_k and $X^e_{k|M'}$ to obtain $X_{k|M'} = [X^m_k, X^e_{k|M'}]$
$X_{k\to ref|M'} = \mathcal{F}(X_{k|M'}, X_{ref|M'})$
extract the coordinate of moving objects $X^m_{k\to ref|M'}$
Object Tracking
$\{\hat{P}_{k+1|k}\} = \mathcal{T}\left(\{\Theta_k\}, \{X^m_{k\to ref|M'}\}\right)$

$\boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$ represents the uncertainty along the principal axes. The columns of vectors $\boldsymbol{V} = [\boldsymbol{v}_\parallel, \boldsymbol{v}_\perp]$ denote the principal directions of uncertainty, where $\boldsymbol{v}_\parallel$ is the unit vector along $\overrightarrow{BC}$, and $\boldsymbol{v}_\perp$ denotes its orthogonal component. The diagonal matrix $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \lambda_2)$, with eigenvalues $\lambda_1 < \lambda_2$, defines the magnitude of variance along these respective axes. The distance-based score $S_{dist} \in (0, 1]$ evaluates the proximity between $\boldsymbol{A}$ and its initial guess $\boldsymbol{A}_{guess} = \boldsymbol{A} + \frac{\boldsymbol{B}+\boldsymbol{C}}{2}$, where $\sigma^2_{\text{dist}} = \max\left(\frac{\|\boldsymbol{B}\|+\|\boldsymbol{C}\|}{2}, \epsilon\right)$ represents an adaptive variance with lower bound ϵ.

After calculating (5) for all putative correspondences, the most likeliy $\boldsymbol{A}'$ is determined by selecting the candidate that yields the maximum score.

2) REFINEMENT OF A'

Based on the estimated correspondence, we can refine the position of $\boldsymbol{A}'$ to produce a robust virtual landmark (i.e., pseudo environment object) by minimizing the following objective function

$$\mathcal{J}(\boldsymbol{A}') = \alpha\|\boldsymbol{A}' - \boldsymbol{A}_{\text{guess}}\|^2 \tag{9a}$$
$$+ \beta(\boldsymbol{A}' - \boldsymbol{A}_{\text{guess}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{A}' - \boldsymbol{A}_{\text{guess}}) \tag{9b}$$
$$+ \gamma\|\boldsymbol{A}' - \boldsymbol{A}\|^2. \tag{9c}$$

(9a) enforces geometric consistency with the initial guess, (9b) incorporate statistical uncertainty via the inverse covariance matrix, and the final term in (9c) ensures stability of pseudo object $\boldsymbol{A}'$ relative to the reference object $\boldsymbol{A}$.

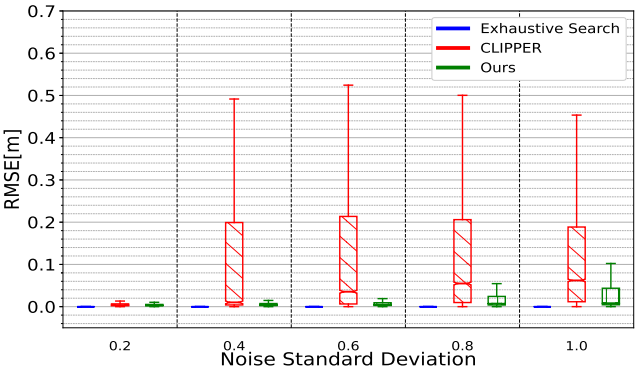


FIGURE 3. Comparisons of spatial alignment accuracy under different noise conditions.

The objective function $\mathcal{J}(\boldsymbol{A}')$ is quadratic convex, and its closed-form optimal solution can be obtained by setting the gradient $\nabla_{A'}\mathcal{J} = 0$, which yields

$$\boldsymbol{A}'_{opt} = \left(\sum_i \boldsymbol{W}_i\right)^{-1}\left(\sum_i \boldsymbol{W}_i\boldsymbol{\mu}_i\right), i \in \{1, 2, 3\}. \tag{10}$$

The weight matrices and their associated anchors are defined as $\boldsymbol{W}_1 = \alpha \cdot \mathbf{I}$, $\boldsymbol{W}_2 = \beta \cdot \boldsymbol{\Sigma}^{-1}$, $\boldsymbol{W}_3 = \gamma \cdot \mathbf{I}$ and $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \boldsymbol{A}_{\text{guess}}$ and $\boldsymbol{\mu_3} = \boldsymbol{A}$, respectively.

Once the optimal estimate $\boldsymbol{A}'_{opt}$ is obtained, a complete set of a matching pairs, $\{(\boldsymbol{A}, \boldsymbol{A}'), (\boldsymbol{B}, \boldsymbol{B}'), (\boldsymbol{C}, \boldsymbol{C}')\}$, can be established. These correspondences are utilized to resolve the $N = 2$ case, effectively serving as a relaxation of the geometric limitations stated in Remark 4. It should also be noted that the values of (α, β, γ) are critical in (10) and their evaluations are provided in Sec. VI.

C. DECISION OF X_{REF}

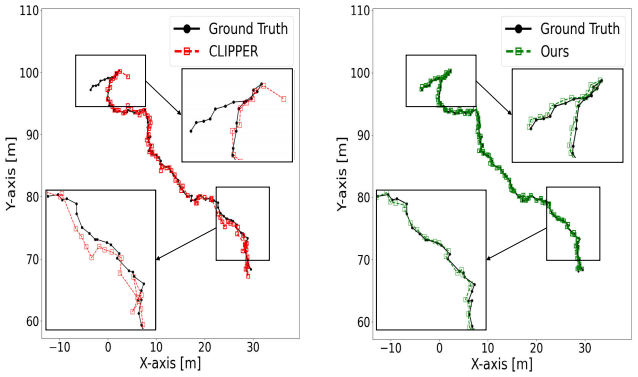
Since all the measurements are mapped into the space of $\boldsymbol{X}_{ref}$, it is crucial to determine this reference frame. As long as numerous environment objects are detected in a frame, this frame can serve as a reference. However, not all environment objects can consistently be detected throughout the entire surveillance period due to the limitations of sensor's viewpoint and the object detection model's capability.

Based on Remarks 4 and 5, the initial $\boldsymbol{X}_{ref}$ can only be set once the number of detected environment objects is at least three in the case of Remark 4, and two in the case of Remark 5 and Sec. V-B. If not possible, MOT cannot be initiated, and the measurements are stored in a buffer. Over time, more environment objects may be detected, allowing $\boldsymbol{X}_{ref}$ to be updated to enhance the robustness of the matching process. Then, the previous measurements stored in the buffer can be used for MOT within the updated reference coordinate system, enabling the tracking of all trajectories of moving objects. For example, in Fig. 1, $\boldsymbol{X}_{ref}$ is initialized at frame k, where three environment objects are detected. As time progresses, $\boldsymbol{X}_{ref}$ can be updated as more environment objects are detected, allowing additional objects to be used in determining $\boldsymbol{M}'$.

TABLE 1. Optimality gap of MOTA across various small-scale noise levels under different landmark configurations.

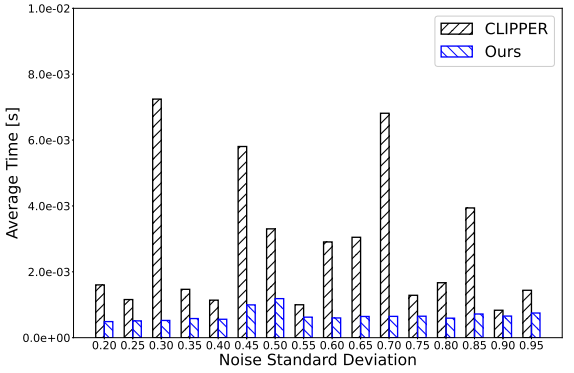
Configurations	$n^e = 5, n^m = 2$					$n^e = 7, n^m = 2$				
	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
CLIPPER (Delay)	20.05	70.23	84.11	89.67	92.16	2.00	39.02	71.63	86.19	93.44
CLIPPER	8.56	35.09	44.16	50.05	51.19	0.42	19.42	36.47	47.22	52.71
Naive nearest	12.12	16.64	22.89	27.65	32.53	14.05	22.37	29.14	37.36	42.86
Hungarian algorithm	0.00	0.22	2.20	3.35	5.88	0.00	0.11	0.47	3.07	4.26
Ours	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

* n^e and n^m denote the number of environment landmarks and moving objects, respectively.

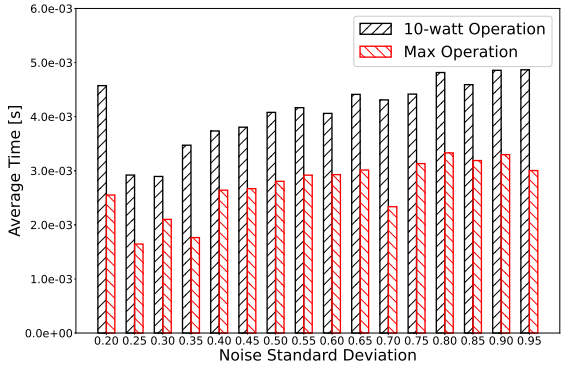


(a) CLIPPER + EKF (b) Our algorithm + EKF

FIGURE 4. Trajectories of a moving object in the reference coordinate system.



(a) Comparisons of CLIPPER and our algorithm running on a desktop system.



(b) Comparisons of our algorithms running on an edge device with different power modes.

FIGURE 5. Average time consumption to find M' (or M^*).

D. OVERALL MOT PROCESS

Details of the proposed MOT process are illustrated in Algorithm 3. It is worth noting that any Kalman filter (KF)-based tracking algorithm, such as extended Kalman filter (EKF), unscented Kalman filter (UKF), or interacting multiple model (IMM), can be incorporated into our algorithm [33]. In addition, data association problems in MOT can be effectively addressed using the Hungarian algorithm [30].

E. PERFORMANCE ANALYSIS

When $\boldsymbol{M}^*$ is given, MOTA is mainly affected by the object tracking process, specifically the data association between $\boldsymbol{X}_{\boldsymbol{k}\rightarrow\boldsymbol{ref}|\boldsymbol{M}^*}$ and the predicted states of all trackers. In our scenarios where there are few environment objects with error-prone measurements, $\boldsymbol{M}^*$ is not available, and exact matching of environment objects becomes a key factor. Our proposed MOT process can accommodate errors by adjusting ζ, and detailed experimental results are presented in Section VI to validate its performance.

The time complexity of our proposed MOT process is analyzed with a focus on the matching process of environment objects. In order to state more concisely, let n^e_{ref} and n^e_k denote the number of environment objects in the reference frame and k-th frame, respectively. Then, the complexity of sorting the environment objects as described in Fig. 2 is $\mathcal{O}(n^e_k \log n^e_k)$. Searching for the matching pairs to find $\boldsymbol{M'}$ until $|\boldsymbol{M'}| = N$, based on the bisection method, has a complexity of $\mathcal{O}(n^e_k \log n^e_{ref})$. Therefore, the time complexity of our proposed MOT process is $\mathcal{O}(n^e_k \log n^e_{ref})$. It is worth noting that constructing entire consistency graphs and finding M^* have an overall complexity of $\mathcal{O}(n^e_k n^e_{ref})$. To be specific, constructing the affinity matrix of the entire consistency graphs requires $\mathcal{O}(n^e_k n^e_{ref})$, with a complexity of $\mathcal{O}(n^e_k \log n^e_{ref})$ of the backtracking line search process to find the best matching. Therefore, searching only a subset of the sorted objects, instead of observing the entire consistency graph, can reduce computational complexity.

VI. EXPERIMENT RESULTS

In this section, numerical results are provided to verify our proposed MOT process. We provide simulation results

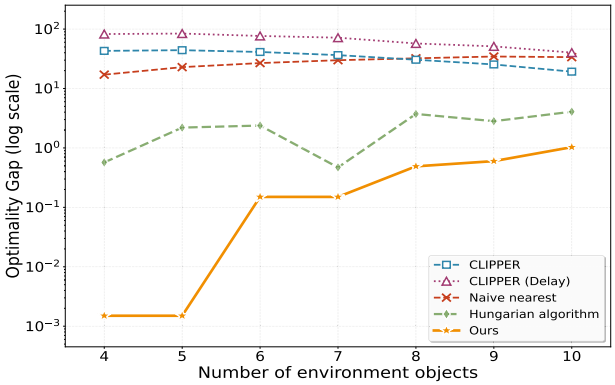


FIGURE 6. Optimality gap of MOTA as the number of environment objects increases.

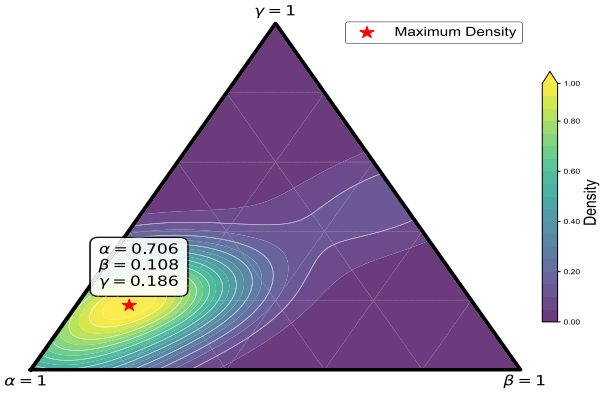


FIGURE 7. Composition analysis of (α, β, γ).

along with the real-world environment results to validate the performance. Monte Carlo simulations were designed to explore a broad range of scenarios, while real-world experiments aimed to evaluating the performance under the worst-case conditions of symmetry. Specifically, we considered environments exhibiting both semantic and spatial symmetry, where all the environment objects belong to the same classes and are arranged in geometrically symmetric configuration.

A. SIMULATION RESULTS

In the simulations, we assume that two different noise scales are applied to generate 2D location measurements. The large-scale noise, with a standard deviation 0.2, is assumed to reflect common errors from the sensor devices. Additionally, based on the fact that depth measurements from the sensor to each object can vary due to the object's shape, size, and orientations, we also consider a small-scale noise that can be applied for each object. [34], [35]

A Monte Carlo simulation is assumed, where each scenario consists of 100 frames, and the performance is averaged across the scenarios. For each scenario, it is assumed that two moving objects move at a random speed ranging from a minimum walking speed of 0.5 m/s to a running speed of 1.8 m/s in arbitrary directions, within an environment where eight environment objects are randomly distributed. A single robot also moves at half the speed range of moving objects in an arbitrary direction. Since no information about the sensor's location and orientation is assumed, the 2D location measurements are only valid in the relative coordinate systems, not in a global coordinate system.

For the basis of comparison, we consider three matching algorithms to find $\boldsymbol{M}'$ (or $\boldsymbol{M}^*$): (1) exhaustive search (to find $\boldsymbol{M}^*$), (2) CLIPPER (to find $\boldsymbol{M}'$), and (3) our algorithm (to find $\boldsymbol{M}'$). Additionally, the same value of $\zeta = 1.0$ are applied to all algorithms for a fair comparison.

Fig. 3 compares a matching performance of three different algorithms with various small-scale noise standard deviations. The root mean square error (RMSE) between ground truth (GT) locations and the transformed locations from three different algorithms is compared. It is observed from Fig. 3 that exhaustive search can find $\boldsymbol{M}^*$, meaning that the RMSE error is equal to zero. On the other hand, as noise standard deviations increase, both our algorithm and CLIPPER produce anomaly transformed results. However, by observing the median and 75 percentile RMSE values from the figure, we can demonstrate that our algorithm is robust to measurement noise.

Fig. 4 represents examples of trajectories where the tracking performance of two algorithms is compared. For both algorithms, the initial trajectories that do not have their ID switched as the frame progresses are illustrated. It is observed from Figs. 4a and 4b that when a moving object starts in the lower right parts, our algorithm successfully follows the object throughout the trajectory. In contrast, CLIPPER fails to track the object due to incorrect coordinate transformations, especially when the trajectory changes abruptly.

Fig. 5 represents average time consumption to find $\boldsymbol{M}'$ (or $\boldsymbol{M}^*$) in various noise standard deviations using two different devices. In Fig. 5a, a desktop equipped with an Intel i9-12900K CPU, an RTX 3090 Ti GPU, and 64GB of RAM is used to calculate the average time consumption. As observed in Fig. 5a, our algorithm requires no more than 2.0 ms to find $\boldsymbol{M}'$, as aligning environment objects and searching with partial information significantly reduce time consumption. On the other hand, CLIPPER requires more times up to 8.0 ms because constructing the affinity matrix and finding $\boldsymbol{M}^*$ are time-consuming process. Fig. 5b illustrates the time consumption of our algorithm operating on an edge-device, where two different power modes of a Jetson ORIN NX with 32GB RAM are applied. It is observed that more time is required as less power is used to operate, but in all the cases, the time consumption does not exceed 5.0 ms. The results indicate that the core of our proposed algorithm is not a computational bottleneck. In contrast, the object detection model used in Sec. VI-D is the primary consumer of system resources, requiring approximately 70 % usage of CPU capacity, 65 % of RAM and 75 % of GPU on average. It is worth noting that the time consumptions in Fig. 5 seem to be quite low, but MOT for each frame is completed when

TABLE 2. Optimality gap of MOTA across various dropout rates.

Droupout Rate	10%			30%			50%			70%			90%		
	0.1	0.3	0.5	0.1	0.3	0.5	0.1	0.3	0.5	0.1	0.3	0.5	0.1	0.3	0.5
CLIPPER (Delay)	68.30	96.10	97.90	74.20	95.70	98.40	81.90	96.90	98.60	89.30	98.30	98.80	93.30	98.50	98.70
CLIPPER	32.60	56.20	61.40	39.60	57.80	61.70	44.30	59.00	60.50	49.60	59.30	60.90	52.90	58.90	60.60
Naive nearest	12.20	21.20	35.40	25.60	33.60	36.30	31.00	34.40	36.30	31.00	32.30	35.10	30.50	31.90	34.90
Hungarian algorithm	1.30	5.40	14.60	2.10	7.90	23.20	2.60	12.30	28.70	2.90	14.10	32.90	3.80	13.70	31.90
Ours	2.60	4.10	6.90	6.19	6.70	8.60	8.50	9.30	10.80	9.90	10.40	11.80	10.90	11.40	12.30
Ours+Pseudo objects	1.30	2.20	4.30	2.50	3.50	5.60	3.30	4.40	7.10	3.70	4.80	7.90	3.80	5.20	7.90

* Three types of small-scale noise are applied for each drop rate.

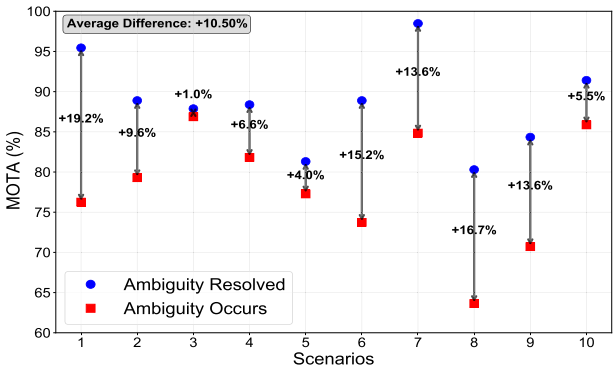


FIGURE 8. Performance comparisons of MOTA with respect to matching ambiguity.

$\boldsymbol{M}'$, $\boldsymbol{X}_{k\to ref|\boldsymbol{M}'}$, and $\{\hat{\boldsymbol{P}}_{k+1|k}\}$ are sequentially determined. Therefore, the time consumption of finding $\boldsymbol{M}^*$ needs to be minimized in order to avoid inter-frame delays.

B. OPTIMALITY GAP OF MOTA

Since our proposed algorithm is a suboptimal approach and we can obtain the exact matching results through Monte Carlo simulations, we adopt the optimality gap as a evaluation metric. A lower value indicates a performance closer to the optimal MOTA. In addition, we adopt two additional baselines for comparison: first, a naive nearest environment matching, and second, the Hugarian algorithm based on the cost of local structural differences.

Fig. 6 illustrates the impact of the number of environment objects on MOTA. The degradation in MOTA (i.e., the increase in the optimality gap) is mainly attributed to an increase in FP, which stems from incorrect coordinate transformations and inter-frame delays (as noted in Remarks 3). It is observed that as the number of environment objects increases, optimality gap also grows, reflecting the degradation in MOTA. While CLIPPER is designed for dense points matching, and thus its optimality gap decreases as the number of objects increases, executing it multiple times incurs significant inter-frame delays, leading to performance loss. It is also observed that both our algorithm and the Hungarian algorithm outperform the other baselines. However, the narrower optimality gap consistently demonstrates the superiority of our algorithms over the baselines.

TABLE 3. Performance comparisons of MOTA in real-world experiments. Average execution time is 0.207s for CLIPPER and 0.126s for ours.

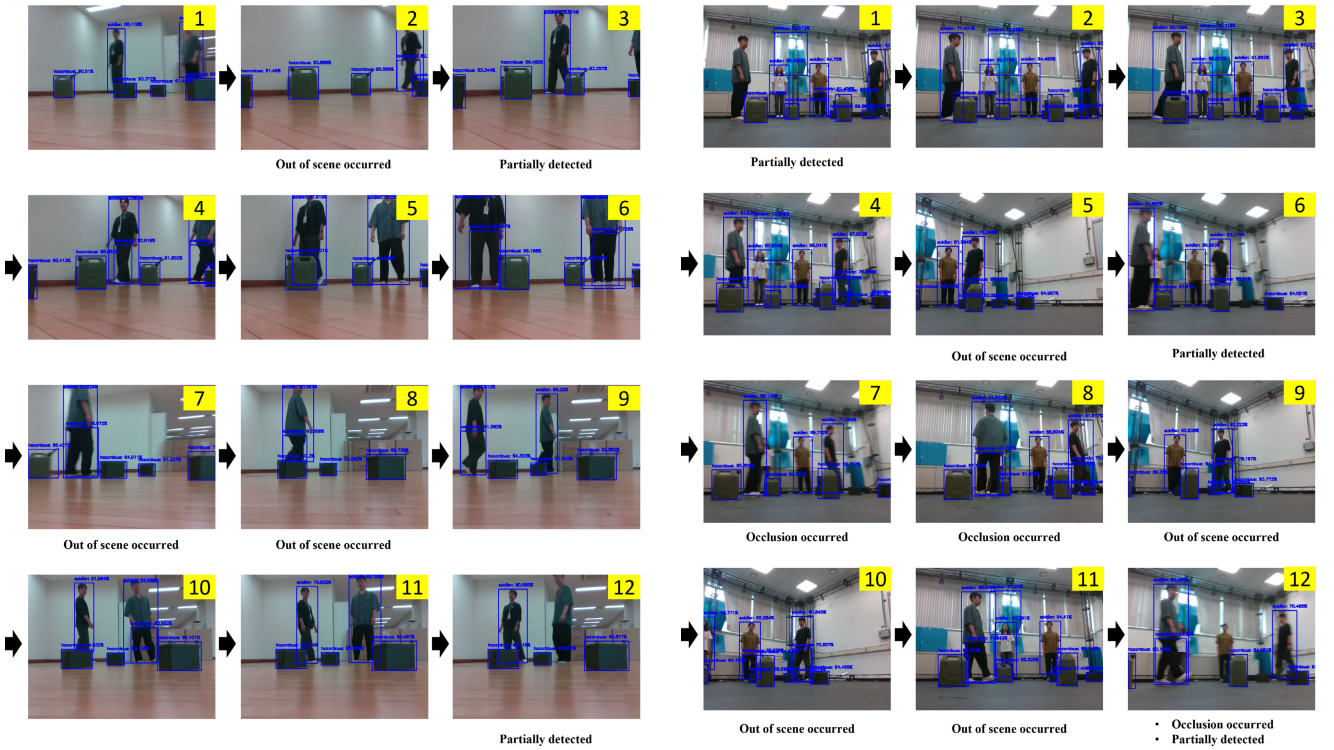
n^m	CLIPPER	Naive nearest	Hungarian algorithm	Ours	Ours+Pseudo objects
2	69.34	68.23	69.5	87.98	**91.06**
3	58.84	54.34	60.5	83.03	**88.89**
4	52.37	56.36	63.99	78.84	**87.98**

Table 1 presents the optimality gap of MOTA under various noise levels across different numbers of environment objects. It is observed that sparse object matching exhibits significantly different characteristics from dense matching, as CLIPPER underperforms compared to both the Hungarian algorithm and our proposed algorithm. Furthermore, while the optimality gaps of other baselines tend to widen as the noise level increases, our algorithm remains robust, ensuring higher matching accuracy and superior object tracking performance.

C. QUANTITATIVE EVALUATION OF SYNTHESIZED PSEUDO OBJECTS

We conducted 1000 Monte Carlo simulations to evaluate the sensitivity of the weighting parameters (α, β, γ) in generating the pseudo object. In these simulations, it is assumed that the reference frame contains three landmarks (i.e., $|\boldsymbol{X}^e_{\boldsymbol{ref}}| = 3$), but only two of three are observed in the current frame (i.e., $|\boldsymbol{X}^e_k| = 2$), representing a typical geometric degeneracy. Leveraging the known ground-truth correspondence in the simulation environment, we evaluated the fidelity of the generated pseudo objects by measuring the Euclidean distance between their positions and the corresponding ground-truth coordinates.

Fig. 7 represents a ternary plot of (α, β, γ), with the constraint of $\alpha + \beta + \gamma = 1$. It is observed that α should be closed to 1, indicating that the geometric consistency in (9a) is significantly more critical than other factors in ensuring the accuracy of pseudo object generation. In addition, it is observed that the optimal weighting triplet (α, β, γ) was empirically determined to be approximately (0.7, 0.1, 0.2). This configuration is subsequently employed in further experiments to evaluate the performance of generating pseudo objects in MOTA.



(a) Scenario of two moving objects

(b) Scenario of four moving objects

FIGURE 9. Representative sequences of consecutive frames from our real-world datasets. All the sample snapshots were extracted at a rate of 1 frame per second (fps).

Table 2 represents the optimality gap of MOTA in the challenging scenarios where the $N = 2$ constraint is applied to a predetermined percentage of total frames. It is observed that as the droprate increases, the optimality gap generally widens across all methods. However, our proposed algorithm consistently outperforms the baselines. Specifically, the results demonstrate that the proposed pseudo object generation provides significant resilience against intermittent landmark loss, effectively maintaining tracking continuity under data-sparsity conditions.

D. REAL WORLD ENVIRONMENT RESULTS

We conducted real-world experiments to validate our proposed MOT process. For this, we utilize a pre-trained YOLOX-small model [32] for object detection, achieving 92.8 mAP, and mount a Intel realsense D435i depth camera on top of small robots to acquire depth information. To make the spatial alignment problem more challenging, we considered environments exhibiting both semantic and spatial symmetry, where all the environment objects belong to the same classes and are arranged in geometrically symmetric configuration.

Fig. 8 represents the impact of matching ambiguity when $n^e = 2$. Because of the robot's movement, the objects were not continuously observable throughout the surveillance period. Such limited and intermittent observations introduce matching ambiguity, which in turn degrades MOTA due to incomplete coordinate transformation. Nonetheless, as stated in Remark 5, the incorporation of object features extracted from bounding boxes effectively mitigates this ambiguity and contributes to enhanced MOTA performance.

Table 3 presents a comparison of MOTA and time complexity. As n^m increases, spatio-temporal alignment becomes more challenging due to the presence of a larger number of moving objects, potentially resulting in more complex and unstable conditions in the matching and tracking process. It is observed from Table 3 that our proposed MOT process outperforms CLIPPER in terms of MOTA and time complexity in real-world environments, consistent with the findings from Monte Carlo simulations.

VII. CONCLUSION

In this paper, we have addressed multiple object tracking (MOT) using a single mobile robot without any prior knowledge of its location or orientation. The proposed self-alignment-by-tracking mechanism can maintain spatio-temporal consistency in MOT. In particular, our proposed geometry-based sorting mechanism enhances the search algorithm to obtain consecutive spatial alignment of objects over time sequences. In addition, robust spatial alignment is ensured by synthesizing pseudo-environment objects, even in challenging matching scenarios. Then, any Kalman-filter based tracking algorithm can be incorporated for temporal consistency. We have conducted theoretical analysis along with Monte Carlo simulations and real-world

experiments. The results demonstrate that our proposed MOT process outperforms both deep feature-based approach and matching-oriented methods across both simulated and real-world experiments. Furthermore, the deployment of our MOT process on a single board computer (SBC) confirms its ability to effectively balance tracking accuracy and computational efficiency, making it highly suitable for resource-constrained platforms. In future work, we plan to refine the reference frame selection process to better accommodate long-range temporal information and manage its associated latency. To this end, we will investigate the integration of attention mechanisms, replay buffers or hypergraph-based frameworks to enhance tracking continuity and robustness.

APPENDIX
EVALUATION ON REAL-WORLD PUBLIC DATASET

Many public datasets [12] focus on the tracking-then-projection approach, where only RGB images and corresponding detection results are provided. Consequently, these datasets do not capture the dynamic characteristics of MOT in mobile robot scenarios. In contrast, for projection-then-tracking approaches to evaluate our proposed algorithm, no public real-world datasets are available.

To ensure a fair and comprehensive evaluation of our proposed algorithm, we conducted both Monte Carlo simulations and real-world experiments. Monte Carlo simulations were designed to explore a broad range of scenarios, while real-world experiments aimed to evaluating the performance under the worst-case conditions of symmetry. Specifically, we considered environments exhibiting both semantic and spatial symmetry, where all the environment objects belong to the same classes and are arranged in geometrically symmetric configuration. In addition, we conducted further real-world experiments by introducing increased robot dynamics while varying the number of moving objects.

Fig. 9 illustrates representative examples from our real-world datasets. Due to the robot's movement and the limited field of view (FoV) of the onboard sensor, each sample may experience one or more of the following conditions: (1) out-of-scene targets, (2) partial detections, or (3) occlusions. These constraints lead to a degradation in MOTA performance and highlight the inherent challenges of MOT in mobile robotic platforms.

REFERENCES

[1] G. C. de Croon, J. Dupeyroux, S. B. Fuller, and J. A. Marshall, "Insect-inspired AI for autonomous robots," *Sci. Robot.*, vol. 7, no. 67, Jun. 2022, Art. no. eabl6334.

[2] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 2, Mar. 2019, Art. no. 1729881419839596.

[3] P. K. Panigrahi and S. K. Bisoy, "Localization strategies for autonomous mobile robots: A review," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 8, pp. 6019–6039, Sep. 2022.

[4] A. T. Abu-Jassar, H. Attar, A. Amer, V. Lyashenko, V. Yevsieiev, and A. Solyman, "Development and investigation of vision system for a small-sized mobile humanoid robot in a smart environment," *Int. J. Crowd Sci.*, vol. 9, no. 1, pp. 29–43, Jan. 2025.

[5] J. Biswas and M. Veloso, "Depth camera based indoor mobile robot localization and navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Sep. 2012, pp. 1697–1702.

[6] H. Xing, L. Shi, K. Tang, S. Guo, X. Hou, Y. Liu, H. Liu, and Y. Hu, "Robust RGB-D camera and IMU fusion-based cooperative and relative close-range localization for multiple turtle-inspired amphibious spherical robots," *J. Bionic Eng.*, vol. 16, no. 3, pp. 442–454, May 2019.

[7] Z. Xu, X. Zhan, Y. Xiu, C. Suzuki, and K. Shimada, "Onboard dynamic-object detection and tracking for autonomous robot navigation with RGB-D camera," *IEEE Robot. Autom. Lett.*, vol. 9, no. 1, pp. 651–658, Jan. 2024.

[8] I. Ullah, D. Adhikari, H. Khan, M. S. Anwar, S. Ahmad, and X. Bai, "Mobile robot localization: Current challenges and future prospective," *Comput. Sci. Rev.*, vol. 53, Aug. 2024, Art. no. 100651.

[9] N. Ganganath and H. Leung, "Mobile robot localization using odometry and Kinect sensor," in *Proc. IEEE Int. Conf. Emerg. Signal Process. Appl.*, Jan. 2012, pp. 91–94.

[10] H. Song, W. Choi, and H. Kim, "Robust vision-based relative-localization approach using an RGB-depth camera and LiDAR sensor fusion," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3725–3736, Jun. 2016.

[11] X. Xu, L. Zhang, J. Yang, C. Cao, W. Wang, Y. Ran, Z. Tan, and M. Luo, "A review of multi-sensor fusion SLAM systems based on 3D LiDAR," *Remote Sens.*, vol. 14, no. 12, p. 2835, Jun. 2022.

[12] C. Du, C. Lin, R. Jin, B. Chai, Y. Yao, and S. Su, "Exploring the state-of-the-art in multi-object tracking: A comprehensive survey, evaluation, challenges, and future directions," *Multimedia Tools Appl.*, vol. 83, no. 29, pp. 73151–73189, Feb. 2024.

[13] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.

[14] H. Yin, X. Xu, S. Lu, X. Chen, R. Xiong, S. Shen, C. Stachniss, and Y. Wang, "A survey on global LiDAR localization: Challenges, advances and open problems," *Int. J. Comput. Vis.*, vol. 132, no. 8, pp. 1–33, Aug. 2024.

[15] C. Fan, L. Li, M.-M. Zhao, and M. Zhao, "An extended joint spatial and temporal cooperation model for the range-based localization problem," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 12121–12134, Dec. 2019.

[16] F. Ghorbani, Y.-C. Chen, M. Hollaus, and N. Pfeifer, "A robust and automatic algorithm for TLS-ALS point cloud registration in forest environments based on tree locations," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 4015–4035, 2024.

[17] N. Hao, F. He, C. Tian, and Y. Hou, "On the consistency of multi-robot cooperative localization: A transformation-based approach," *IEEE Robot. Autom. Lett.*, vol. 10, no. 1, pp. 280–287, Jan. 2025.

[18] T. Bailey, E. M. Nebot, J. K. Rosenblatt, and H. F. Durrant-Whyte, "Data association for mobile robot navigation: A graph theoretic approach," in *Proc. Millennium Conf. IEEE Int. Conf. Robot. Automation. Symposia*, vol. 3, Apr. 2000, pp. 2512–2517.

[19] J. Shi, H. Yang, and L. Carlone, "ROBIN: A graph-theoretic approach to reject outliers in robust estimation using invariants," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 13820–13827.

[20] P. C. Lusk and J. P. How, "CLIPPER: Robust data association without an initial guess," *IEEE Robot. Autom. Lett.*, vol. 9, no. 4, pp. 3092–3099, Apr. 2024.

[21] M. B. Peterson, P. C. Lusk, A. Avila, and J. P. How, "TCAFF: Temporal consistency for robot frame alignment," 2024, *arXiv:2405.05210*.

[22] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 19, no. 1, pp. 5–18, Jan. 2004.

[23] X. Zhang, J. Ma, J. Guo, W. Hu, Z. Qi, F. Hui, J. Yang, and Y. Zhang, "HyperGCT: A dynamic hyper-GNN-learned geometric constraint for 3D registration," 2025, *arXiv:2503.02195*.

[24] X. Zhou, Z. Ma, C. Wang, M. Chu, W. Zhao, and H. Zhang, "A factor graph optimization SLAM mapping method for autonomous vehicles considering dynamic target motion," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 11, pp. 20836–20849, Nov. 2025.

[25] H. Suljagic, E. Bayraktar, and N. Celebi, "Similarity based person re-identification for multi-object tracking using deep Siamese network," *Neural Comput. Appl.*, vol. 34, no. 20, pp. 18171–18182, Oct. 2022.

[26] E. Bayraktar, Y. Wang, and A. DelBue, "Fast re-OBJ: Real-time object re-identification in rigid scenes," *Mach. Vis. Appl.*, vol. 33, no. 6, p. 97, Nov. 2022.

[27] E. Bayraktar, "ReTrackVLM: Transformer-enhanced multi-object tracking with cross-modal embeddings and zero-shot re-identification integration," *Appl. Sci.*, vol. 15, no. 4, p. 1907, Feb. 2025.
[28] N. G. De Bruijn, "A combinatorial problem," *Proc. Sect. Sci. Koninklijke Nederlandse Akademie van Wetenschappen te Amsterdam*, vol. 49, no. 7, pp. 758–764, 1946.
[29] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, May 2008.
[30] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, 1955. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109
[31] J. Baek, J. Lee, H. Shim, S. Im, and Y. Han, "Target tracking initiation for multi-static multi-frequency PCL system," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 10558–10568, Oct. 2020.
[32] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.
[33] S. Y. Chen, "Kalman filter for robot vision: A survey," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4409–4420, Nov. 2012.
[34] C. V. Nguyen, S. Izadi, and D. Lovell, "Modeling Kinect sensor noise for improved 3D reconstruction and tracking," in *Proc. 2nd Int. Conf. 3D Imag., Model., Process., Visualizat. Transmiss.*, Oct. 2012, pp. 524–530.
[35] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "Approximate inference in state-space models with heavy-tailed noise," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5024–5037, Oct. 2012.

JAEUK BAEK received the B.S. degree in electrical engineering from Hanyang University, Seoul, South Korea, in 2015, and the Ph.D. degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2020. He was a Postdoc Research Fellow in electrical engineering with KAIST. He is currently a Senior Researcher with the Electronics and Telecommunications Research Institute (ETRI), Daejeon. His research interests include target tracking and signal processing, intelligent robot control systems, artificial intelligence, SLAM, robot software frameworks, and cooperative robot control.

SEOKHO JEONG received the B.S. degree in electrical engineering from Kyungpook National University, Daegu, South Korea, in 2025. He is currently pursuing the M.S. degree in electrical engineering with Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. His research interests include target tracking, robot control systems, artificial intelligence, and anomaly detection.

DONGGYU CHOI received the B.S. degree in computer engineering, the M.S. degree in software convergence, and the Ph.D. degree in computer engineering from Dong-Eui University, Busan, South Korea, in 2018, 2020, and 2023, respectively. He is currently a Postdoctoral Research Fellow with the Defense and Safety Convergence Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. His main research interests include deep learning programming and health care, intelligent robot control systems, artificial intelligence, robot software frameworks, and cooperative robot control.

CHANGEUN LEE received the B.S. and M.S. degrees in electronics engineering from Hanyang University, Seoul, South Korea, in 1996 and 1998, respectively, and the Ph.D. degree in information and communication engineering from Chungnam National University, Daejeon, South Korea, in 2017. From 1998 to 2000, he was a Researcher with LG Industry Systems, Seoul, where he worked on intelligent building automation systems. Since 2001, he has been with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea, where he has conducted research in the fields of intelligent robot control systems and home network systems. His research interests include artificial intelligence, SLAM, robot software frameworks, and cooperative robot control.