Multi-Person Tracking in Cluttered Environments Using Hybrid Data Association

Nurettin Özçelik[1], Volkan Sezer[2]

[1]Department of Mechatronics Engineering
Smart and Autonomous Systems Laboratory (SASlab), Istanbul Technical University, Istanbul, Türkiye
ozcelikn22@itu.edu.tr
[2]Department of Control and Automation Engineering
Smart and Autonomous Systems Laboratory (SASlab), Istanbul Technical University, Istanbul, Türkiye
sezerv@itu.edu.tr

Abstract

Multi-target tracking remains a fundamental problem in robotics, where data association plays a critical role. Algorithms such as the Global Nearest Neighbor (GNN) and the Joint Probabilistic Data Association (JPDA) offer trade-offs between speed and robustness. GNN is computationally efficient but degrades in cluttered scenes, whereas JPDA enhances robustness through probabilistic reasoning at the expense of higher computation. This paper proposes a hybrid data association strategy that combines GNN and JPDA in a rule-based framework by adaptively switching between them according to scene ambiguity. The method is evaluated on simulated pedestrian-tracking scenarios in ROS and Gazebo with clutter and sensor models with noise. Performance is assessed using T-GOSPA metric and runtime analysis. Results show that the hybrid approach achieves accuracy comparable to JPDA while reducing the runtime gap between JPDA and GNN by about 18.5%, enabling real-time deployment on robotic systems with limited computational resources.

1. Introduction

Tasks like people guidance in public spaces [1] require robots to continuously perceive human activity and adapt accordingly. The robot must maintain up-to-date estimates of the positions of people over time and predict future motion to plan safe paths. Multi-target tracking refers to the task of estimating both the number of objects present in the environment and maintaining a separate track for each one. A track typically includes a unique identifier and an estimate of the object's dynamic state, such as its position, velocity, and possibly orientation, over time.

Tracking involves several fundamental challenges that have been the focus of ongoing research efforts within the field. One of them is data association, which is matching incoming sensor detections to the correct tracked people. This process is complicated by several factors that can introduce uncertainty into sensor measurements. Weak sensor signals, background noise, or closely spaced targets can lead to ambiguous measurements, which makes it difficult to select the correct measurement for updating a track. Wrong matches may cause identity swaps or duplicate tracks. Accumulated errors rapidly degrade tracker reliability and accuracy.

To address these challenges, various data association strategies can be employed. One of the earliest solutions was the Global Nearest Neighbor (GNN) algorithm, introduced in the early 1970s [2]. This method associates each measurement with the nearest predicted target state based on distance metrics. While it is effective in simple cases, GNN is highly sensitive to clutter and becomes unreliable when multiple objects are close together. In response to this limitation, Bar-Shalom proposed the Joint Probabilistic Data Association (JPDA) filter [3]. Instead of choosing a single measurement, JPDA jointly evaluates all feasible measurement-to-track assignments while accounting for mutual exclusivity. It uses a probabilistic weighting scheme to update the target state. This allowed the tracker to remain robust even when false alarms or missed detections occurred.

Several studies in the literature compare the performance of data association algorithms and propose hybrid approaches that combine their strengths to improve tracking accuracy and robustness. In [4], a simulation with four non-intersecting targets in a static area was repeated under varying clutter intensities $\lambda = \{0, 1, 2, 5, 10\}$. The results demonstrated that clutter increased the position error in JPDA, even for well-separated targets. In contrast, GNN provided more accurate position estimates up to a certain clutter level. However, JPDA showed a lower cardinality error, which is the discrepancy between the number of estimated tracks and the actual number of targets. These findings highlight that even in scenarios where targets are well-separated, JPDA may underperform compared to GNN in terms of positional accuracy. Therefore, identifying such situations and opting for GNN instead of JPDA can lead to improved tracking performance.

In [5], JPDA and MHT (Multiple Hypothesis Tracking) from MATLAB's Sensor Fusion and Tracking Toolbox were compared using the GOSPA metric in scenarios where targets moved from distant positions to closely spaced parallel paths. As targets moved closer, JPDA's performance degraded while MHT maintained better accuracy. However, MHT was significantly slower. By analyzing GOSPA values, a threshold was identified to determine when JPDA performs similarly to MHT. Based on this, a strategy was proposed to switch between algorithms depending on proximity for optimizing tracking performance while balancing computational cost.

Section II presents the multi-target tracking algorithms used in this study and the proposed hybrid data association method. Section III describes the system architecture, simulation setup, and scenario generation framework. Section IV reports the experimental results and performance analysis. Lastly, Section V concludes the paper and outlines potential directions for future work.

2. Multi-Target Tracking Algorithms

In this study, GNN and JPDA algorithms are used in combination with a Linear Kalman Filter (LKF). LKF is a widely adopted state estimator in tracking applications due to its efficiency and simplicity. At each cycle, it predicts the next state of a target based on its previous state using a motion model and then corrects this prediction using the latest sensor measurements.

At a detection frequency of approximately 10 Hertz, which corresponds to a measurement interval of $\Delta t = 0.1$ s, human motion can generally be assumed to be locally linear over short durations. This makes the Nearly Constant Velocity (NCV) motion model a suitable choice. The state vector in the NCV model is defined as $\boldsymbol{x} = [x, y, v_x, v_y]^T$ representing the 2D position and velocity of a tracked object. For the NCV, F and Q are given in (1) and (2), respectively. A process noise level of $q_L = 0.1$ is used.

$$F = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$Q = \begin{bmatrix} \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & 0 & \Delta t & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 & \Delta t \end{bmatrix} q_L \tag{2}$$

Since the sensors in this study provide measurements only of the position in x and y directions, the measurement model is defined to extract these components from the full state vector.

2.1. GNN

The Global Nearest Neighbor (GNN) algorithm is one of the most commonly used methods for data association in multi-target tracking, particularly due to its implementation simplicity and computational feasibility. In GNN, each measurement is associated with at most one track, and vice versa. At each cycle, a cost matrix is constructed, where each element represents the cost of matching a specific track with a particular measurement. This cost is usually calculated based on the Mahalanobis distance. For track-measurement pairs that fall outside the gating threshold, their cost is typically assigned a very large number, such as 999. Solving this matching problem requires finding the optimal one-to-one assignment that minimizes the total cost across all valid pairings. This is a classical instance of the linear assignment problem. In the context of GNN, it is solved efficiently using the JVC algorithm [6]. Once the optimal assignment is determined, each track is updated using the corresponding assigned measurement.

GNN is prone to misassignments, especially when targets are close, sensor noise is high, or false positives are present. Relying on hard assignments and ignoring the full association probability space can lead to incorrect updates.

2.2. JPDA

The Joint Probabilistic Data Association (JPDA) algorithm offers a more robust solution in cluttered environments by considering all feasible measurement-to-track associations with probabilistic weighting. The first step in JPDA is constructing a validation matrix. It is a binary matrix that identifies which measurements fall within the gating region of each target. Using the validation matrix, JPDA enumerates all valid association hypotheses that meet two constraints: each measurement links to at most one target or clutter, and each target to at most one measurement. Each hypothesis represents a possible joint association event, and its probability is evaluated using Bayes' rule, where the likelihoods of the hypotheses incorporate a Poisson model for false alarms.

For each measurement-target pair, the marginal probability is obtained by summing the likelihoods of all joint hypotheses in which that specific association occurs. The marginal association probability β_{jt} that measurement j is associated with target t is used as a weighting factor in the state update step.

JPDA has some limitations. One well-known issue is track coalescence, where the estimated tracks of closely spaced targets tend to merge, leading to identity loss. JPDA also assumes a fixed number of targets, which reduces its effectiveness without a reliable track management mechanism in dynamic scenes where targets may appear or disappear. Furthermore, as the number of targets or clutter increases, the number of possible associations grows rapidly. This makes the approach computationally expensive.

To address track coalescence, several extensions of JPDA have been proposed. One of them is Scaled JPDA [7], which applies a scaling factor to the association probabilities. This increases the weight of highly likely associations, making the algorithm behave more decisively. Adjusting this factor controls the softness of JPDA's probabilistic assignments. In our work, we used [7] to mitigate merging of closely spaced, parallel-moving pedestrians. A scaling factor of 2 is used as it is found in the original study to provide a good compromise between accuracy and stability for many scenarios.

2.3. Proposed Hybrid Method

In dynamic environments such as human–robot interaction scenes, where target density, sensor quality, and occlusions vary over time, a fixed association method can be inefficient. Using JPDA when GNN would suffice wastes resources, while sticking with GNN when ambiguities are high compromises accuracy. To address this, we propose a hybrid approach that leverages the strengths of both algorithms. We keep both GNN and JPDA readily available, adaptively switching to the more computationally intensive JPDA only when its enhanced reasoning is genuinely required. Our objective is to achieve JPDA-level robustness while maintaining computational cost close to that of GNN.

As detailed in Fig. 1, whenever new detections arrive, the tracker runs the same quick loop. It performs gating, builds a validation matrix, and passes it to the switch module. The module evaluates three conditions: ambiguous matches, overlapping gates, and close tracks. If any test says the scene is uncertain or ambiguous, the tracker switches from GNN to JPDA. If the scene becomes clear again, it switches back to GNN. A cool-down timer $n_{cooldown}$ prevents rapid oscillation by enforcing a minimum run time of $n_{cooldown}$ cycles after each switch. The chosen method then assigns detections to tracks and updates each track's state.

The first test is done by looking at the validation matrix to see whether the current scene is clear or ambiguous in terms of data association. If a single measurement takes place in more than one row cell, or a single track has more than one "1" in its column, the scene already contains conflicting pairings. GNN would be forced to make a hard choice, while JPDA can keep both options

Algorithm 1: Selection logic of data association algorithm

Data: validation matrix V, Current track set T
Cycle counter s, last-switch cycle s_{last}
Params: distance threshold, n_{cooldown}
Result: selected data association algorithm alg_{DA}
if V=None or len(T) < 2 or ($s - s_{last}$) ≤ n_{cooldown} **then**
return previous alg_{DA}
endif
if V is ambiguous **then**
$alg_{\text{DA,temp}}$= JPDA
elseif any two tracks in T have overlapping gates **then**
$alg_{\text{DA,temp}}$= JPDA
elseif min(intertrack distance) < distance threshold **then**
$alg_{\text{DA,temp}}$= JPDA
else
$alg_{\text{DA,temp}}$= GNN
endif
if previous $alg_{\text{DA}} \neq alg_{\text{DA,temp}}$
alg_{DA}= $alg_{\text{DA,temp}}$
s_{last}= s
endif
return alg_{DA}

Fig. 1: Selection logic of data association algorithm

alive. In the gate-overlap test, the goal is to anticipate situations where two tracks might soon interfere with each other due to proximity and uncertainty. Instead of explicitly checking for an intersection between their elliptical gates, the system uses a simple test based on Mahalanobis distance. If the predicted position of one track is within a gate of the other, their gates are considered potentially overlapping. This means the two tracks might start competing for future measurements, which can lead to association conflicts. As soon as one overlapping pair is found, the switch module flips to JPDA for the coming frame. The last test is the intertrack Euclidean distance, which checks whether any two predicted centers of track are closer than a threshold. A threshold of 0.75 m is used. If the minimum intertrack distance is below the threshold, JPDA is used to avoid the bad performance of GNN with closely spaced targets.

The switching logic follows a fixed priority order rather than evaluating all tests equally. Rules are checked from the simplest to the most computationally demanding, allowing fast decisions in clear cases. Once a rule selects JPDA or GNN, the remaining tests are skipped. The validation-matrix ambiguity test has the highest priority, as it directly reflects the current association state. Gate-overlap and inter-track distance tests share equal priority since they capture the same proximity effects from different perspectives, with the former sensitive to covariance growth and the latter to spatial closeness.

2.4. Track Management

To avoid creating tracks from clutter or noisy detections, a two-stage confirmation method was used during track initialization. In the first stage, unmatched measurements in the previous cycle become candidates. Each candidate is then checked against all active tracks that haven't missed more than five consecutive updates. Candidates within 0.25 m of any active track are discarded to avoid duplicates. For the current unmatched measurements, a second check is made by comparing them with candidates. If a measurement is found within 0.15 m of a candidate, the candidate's internal confirmation level is increased, and the nearby measurement shares this level. Once a candidate reaches level 2, a tentative track is created. This way, only measurements that appear consistently across two frames are promoted to tracks. After that, candidates with level 0 are deleted.

Tracks missing more than 30 consecutive updates are considered lost and are deleted. If a track's estimated position falls into an area where fewer than 90% of cells within a 0.15 m radius are free in the occupancy grid, it is assumed to have drifted into an obstacle and is discarded [8]. In addition, tracks that have been matched successfully at least 10 times are also monitored for consistency. If the sum of their positional variances grows beyond 0.2 m², the track is killed to avoid accumulating uncertainty or drift. Tentative tracks missing over 20 consecutive updates are likewise discarded, while those with more than 10 matches are promoted to confirmed status.

3. System Architecture and Simulation Setup

3.1. Simulation Environment

The tracking simulations were carried out in ROS and Gazebo with Python on a simulated Ackermann robot, derived from the MIT Racecar platform [9]. The robot has a ZED stereo camera and RPLIDAR A2M6. Its SolidWorks CAD model was exported to URDF as shown in Fig. 2. Prior to experiments, a 2D occupancy grid was built via *gmapping* [10] and used for localization by *als_ros* [11] package.

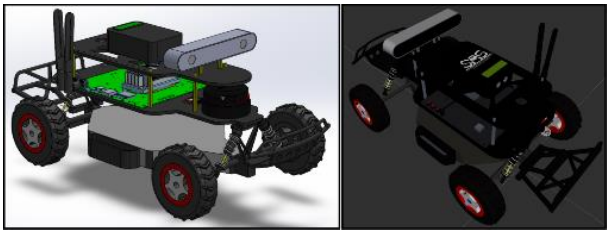

Fig. 2: Solidworks (left) and URDF (right) model of robot

3.2. Tracking Framework

The tracking framework implemented in this study builds upon the foundation of the SPENCER project [8], a well-regarded effort in the domain of multi-target tracking. Several key components from SPENCER were integrated into the system, including custom ROS message definitions and a measurement fusion module. For human detection, the *upper_body_detector* and *leg_tracker* packages were adopted. The *upper_body_detector* [12] is a depth-based template matching approach that slides a learned upper-body template over the depth image. The *leg_tracker* package [13] is a module for pedestrian tracking using 2D LiDAR data. It detects individual leg clusters and estimates human positions by associating them through a GNN. RViz was used to see real-time estimated tracks as shown in Fig. 3. Blue spheres represent generated false alarms, while green spheres at the foot level indicate detections produced by the fused output of two detectors.

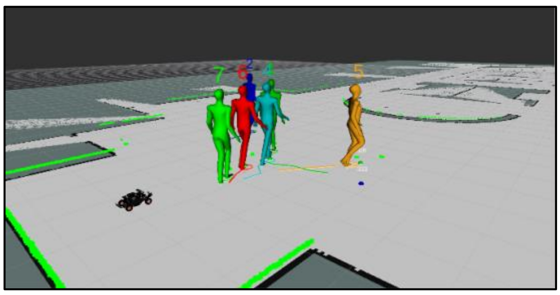

Fig. 3: Visualization of tracking results in RViz

The measurement covariance matrix R, defined as a 2x2 diagonal matrix with diagonal elements equal to 0.005, was uniformly applied to both detector outputs and generated clutter. False alarms are generated by modelling clutter as a homogeneous Poisson point process with rate λ, which is the average number of false alarms per second. Each clutter point is placed independently according to a uniform spatial density over the field of view of the depth camera. In this study, $\lambda = 10$ is used.

To simulate realistic sensor imperfections, zero-mean Gaussian noise with the same covariance matrix R mentioned earlier was added to the measurements, despite the fact that human detections derived from the depth camera and LiDAR data in the simulation environment exhibited negligible positional error.

In high-update-rate scenarios, generating hypotheses for every possible track–measurement pairing across the entire scene can overwhelm computational resources. To manage this complexity, a clustering strategy is applied in the JPDA implementation. Hypothesis enumeration is performed separately within these smaller clusters. This significantly reduces computational load and enables real-time tracking.

3.3. Scenario Generation

The generator converts high-level scene parameters such as geometry, traffic flow, and pedestrian kinematics into Gazebo *.world* files with time-synchronized, animated actors. Pedestrian motion is simulated using RVO2 (Reciprocal Velocity Obstacles 2) [14], a real-time multi-agent model that computes collision-free, goal-directed velocities until all agents reach their destinations. As shown in Fig. 4, the walkable area is modelled by five rectangles covering the main hall, its side corridors, and the entrance beside the robot. Pedestrians spawn in one of four regions (entrance left/right or side left/right) with entrances weighted at 60% and sides at 40%. Their positions are sampled uniformly within the chosen rectangle and oriented toward their destination. Destinations are drawn from a 4×4 origin–destination matrix that directs entrance spawns toward side exits and side spawns across to the opposite side. This structured flow produces frequent crossings, lane formation, and occlusions, creating challenging scenarios for data association.

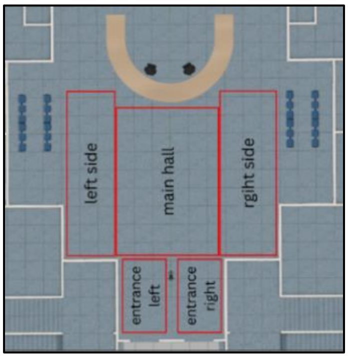

Fig. 4: Top-down view of the simulated environment

The number of pedestrians per scenario was sampled from a categorical distribution over 1–10, with most scenarios containing 3–5 pedestrians. Each pedestrian has a 25 % chance to join a group of up to three; grouped agents temporarily target their group's centroid. Walking speeds are sampled from a normal distribution ($\mu = 1.34$ m/s, $\sigma^2 = 0.26^2$) truncated to 0.4–2.1 m/s. Pedestrians have a 0.30 m radius and asymmetric dynamics of 0.8 m/s² acceleration and 1.5 m/s² deceleration to reflect natural stopping behavior. The simulation runs at 0.1 s steps with the RVO2 solver considering neighbors within 6 m. After the RVO2 simulation, trajectories are exported into a Gazebo *.world* file, where each pedestrian is defined as an *<actor>* element following its time-stamped path with a looping walk animation. Fig. 5 shows an example scenario rendered in Gazebo.

Fig. 5: Example scenario with multiple walking pedestrians in Gazebo

4. Simulation Results

The GNN, JPDA, and proposed hybrid models were evaluated using the Trajectory Generalized Optimal Sub-Pattern Assignment (T-GOSPA) [15] metric for tracking accuracy and their average frame-processing runtime for computational efficiency. This approach effectively penalizes localization errors for properly detected targets, missed detections, false detections, and track switches throughout the track lifecycle.

In this study, the metric is configured with an order of $p = 2$, a cut-off distance of $c = 0.5$ m, and a switching penalty of $\gamma = 0.5$. A cardinality error of one, whether it arises from a single missed target or a single false target, contributes $\sqrt{0.125} = 0.354$ m to the T-GOSPA score. It means it is treated exactly like a 0.354 m localisation error. A full switch error corresponds to a localisation error.

A total of 200 pedestrian scenarios were generated and simulated in Gazebo. Each run is accompanied by a corresponding ROS bag file recorded during the simulation with detectors. For each scenario, the ROS bag was replayed three times, once for each data association strategy. The average of individual cost components for each algorithm is reported in Table 1. All experiments were carried out on an Intel Core i5-8300H CPU@2.3 GHz with 16 GB RAM and an NVIDIA GeForce GTX 1050 GPU, running Ubuntu 18.04 under Python 3.8 and 2.7 with ROS Melodic.

Table 1. Cost statistics per algorithm

Cost Type (m^2/frame)	GNN	JPDA	Hybrid
Localization	0.0814±0.0506	0.0804±0.0502	0.0806±0.0502
Missed Target	0.3427±0.1864	0.3441±0.1862	0.3427±0.1870
False Track	0.0505±0.0423	0.0399±0.0333	0.0394±0.0340
ID Switch	0.0101±0.0104	0.0091±0.0095	0.0087±0.0086

Missed-target penalties constitute the majority of the total T-GOSPA cost (≈72.01%) across all three methods, while localization error accounts for about 16.96%. False-track and ID-switch terms together form only ≈11.03% of the total cost, but they are the most sensitive to the choice of data-association strategy. This distribution indicates that measurement recall (detection failures) is the principal performance bottleneck. Changing the association approach alone cannot eliminate that dominant error source. A likely contributing factor is the large allowable zone used when spawning pedestrians. Some agents begin or wander near the outer edges of the simulated hall, temporarily moving outside the effective field–of-view of the depth camera and LiDAR. Consequently, the detector never produces observations for those targets, which leads to an increase in the missed-target cost regardless of how cleverly detections are associated. Reducing the spawn area or widening sensor coverage could be prioritised in future work.

Compared with the deterministic GNN, probabilistic JPDA reduces the false-track cost by 20.99% and lowers ID-switches by 9.90%. These gains are consistent with JPDA's soft assignment. Ambiguous detections are fractionally attributed rather than forced into hard one-to-one matches. It suppresses spurious tracks and prevents identity flips. The hybrid approach fully retains all the benefits of JPDA. False-track cost of it is the smallest among the three and slightly better than JPDA. ID-switch cost drops further to 0.0087 m^2/frame, a modest 4.40% improvement over JPDA and 13.86% over GNN. The localization and missed-target costs stay nearly unchanged with respect to JPDA.

As the three filters have been evaluated in terms of their average per-iteration runtime, a brief overview of their computational performance is provided below. GNN is the fastest, requiring 9.01 ± 3.44 ms per iteration, which is about 13.02% faster than JPDA that operates at 10.37 ± 4.26 ms per iteration. The hybrid approach has a runtime of 10.12 ± 4.55 ms per iteration. It is slower than GNN by 12.20% while it is faster than JPDA by 2.41%.

GNN maximizes frame rate yet tolerates more assignment errors. JPDA is more reliable but 14.97% slower. The hybrid method reduces the JPDA-to-GNN runtime gap by 18.52% while maintaining JPDA-level accuracy. These results show that the hybrid method offers a good balance between speed and accuracy. It runs faster than JPDA while keeping the same level of tracking quality, making it well-suited for real-time use on robots with limited computing power.

5. Conclusion

This paper presented a hybrid data association framework for real-time multi-target tracking in cluttered environments. The approach integrates GNN and JPDA algorithms using a rule-based switching mechanism that selects the appropriate method based on scene ambiguity. A realistic simulation environment was developed in ROS, incorporating dynamic pedestrians, sensor noise, and clutter. Experimental results show that the hybrid method achieves JPDA-level accuracy while reducing the runtime gap between JPDA and GNN by approximately 18.52%. The proposed method offers a reliable and efficient solution for robust multi-person tracking in dynamic environments.

As future work, the tracking framework will be evaluated under varying clutter rates and sensor noise covariance levels to simulate different sensor conditions and assess robustness. A real-world implementation is planned using a real autonomous robot as a future study.

6. References

[1] U. Uğuzlar, E. Cansu, E. C. Contarlı, and V. Sezer, "Autonomous Human Following Robot Based on Follow the Gap Method," 2023 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). IEEE, pp. 139–144, Apr. 26, 2023.

[2] Y. Bar-Shalom and T. E. Fortmann, Tracking and Data Association, vol. 179, Mathematics in Science and Engineering: Academic Press, 1988.

[3] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," IEEE Journal of Oceanic Engineering, vol. 8, no. 3, pp. 173–184, 1983.

[4] J. Smith, F. Particke, M. Hiller, and J. Thielecke, "Systematic Analysis of the PMBM, PHD, JPDA and GNN Multi-Target Tracking Filters," 2019 22th International Conference on Information Fusion (FUSION). IEEE, pp. 2019

[5] B. L. Soboliev, Adaptive Multi-Target Tracking Through Optimal Data Association Technique Selection, Ph.D. dissertation, Dept. Elect. Eng., Univ. of Massachusetts Lowell, Lowell, MA, USA, May 2023.

[6] DF Crouse. On implementing 2D rectangular assignment algorithms. IEEE Transactions on Aerospace and Electronic Systems, 52(4):1679-1696, 2016.

[7] H. L. Kennedy, "Controlling track coalescence with scaled Joint Probabilistic Data Association," 2008 International Conference on Radar. IEEE, pp. 440–445, Sep. 2008.

[8] T. Linder, S. Breuers, B. Leibe, and K. O. Arras, "On multi-modal people tracking from mobile platforms in very crowded and dynamic environments," 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, May 2016.

[9] The MIT RACECAR, The MIT RACECAR project website, 2016. [Online]. Available: https://mit-racecar.github.io

[10] B. Gerkey, slam_gmapping: ROS wrapper for OpenSlam's GMapping, OpenSource Robotics Foundation, 2010. [Online]. Available: https://wiki.ros.org/gmapping

[11] N. Akai, "Reliable Monte Carlo Localization for Mobile Robots," Journal of Field Robotics, vol. 40, no. 3, pp. 595–613, 2023.

[12] Jafari O. Hosseini and Mitzel D. and Leibe B., "Real-Time RGB-D based People Detection and Tracking for Mobile Robots and Head-Worn Cameras," IEEE International Conference on Robotics and Automation (ICRA'14), 2014.

[13] A. Leigh, J. Pineau, N. Olmedo and H. Zhang, "Person Tracking and Following with 2D Laser Scanners," International Conference on Robotics and Automation (ICRA), 2015.

[14] J. van den Berg, Ming Lin, and D. Manocha, "Reciprocal Velocity Obstacles for real-time multi-agent navigation," 2008 IEEE International Conference on Robotics and Automation. IEEE, 2008.

[15] Á. F. García-Fernández, A. S. Rahmathullah and L. Svensson, "A Metric on the Space of Finite Sets of Trajectories for Evaluation of Multi-Target Tracking Algorithms," in IEEE Transactions on Signal Processing, vol. 68, pp. 3917-3928, 2020.