2026 NDIA MICHIGAN CHAPTER
GROUND VEHICLE SYSTEMS ENGINEERING
AND TECHNOLOGY SYMPOSIUM
***AUTONOMY AND ROBOTICS* TECHNICAL SESSION**
AUGUST 11–13, 2026 - NOVI, MICHIGAN

Cooperative Object Tracking for Improved World Modeling in Multi-Agent Systems

William Kennedy[1], Nicholas Alton[1]

[1]Southwest Research Institute, San Antonio, TX

ABSTRACT

This paper presents a cooperative object tracking framework that enables robots to share object observations, improving world model accuracy and consistency in environments with occlusions. An open-source, MHT-based object tracking package was applied in conjunction with an inter-robot communication system developed to fuse object observations across multiple robots. Experimental results demonstrate greater than 1.4-fold improvements in tracking performance in cluttered, occluded environments compared to individual tracking.

Citation: W. Kennedy, N. Alton "Cooperative Object Tracking for Improved World Modeling in Multi-Agent Systems," In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium* (GVSETS), NDIA, Novi, MI, Aug. 11–13, 2026.

1. INTRODUCTION

As robots and autonomy become more prevalent across many industries, there will be increased opportunities for autonomous assets to cooperate to achieve goals. Autonomous convoys, robotic formations, and "swarm" behaviors are all examples of collaboration among robots. For groups of robots to perform these types of coordinated tasks, they must maintain accurate knowledge of the surrounding environment, including the locations of collaborating robots and any dynamic objects such as nearby people or vehicles. In multi-robot systems, collaborating assets may have information that would be useful to the rest of the group. Objects that may be out of sensing range, blocked from view, or misclassified by one robot's perception system could be accurately detected by a cooperating asset. Sharing and fusing this data can help mitigate the shortcomings of individual perception sensors and lead to more accurate and robust world modeling, especially in dynamic, cluttered, and low-visibility environments.

A cooperative tracking system could be leveraged to support human-machine formations, allowing robots to maintain better awareness of the location and movement of soldiers and other vehicles in cluttered or occluded environments. Similarly, autonomous vehicles in a convoy could share obstacle information, enabling followers to anticipate hazards before they enter sensor range. Autonomous material handlers could also use such systems to

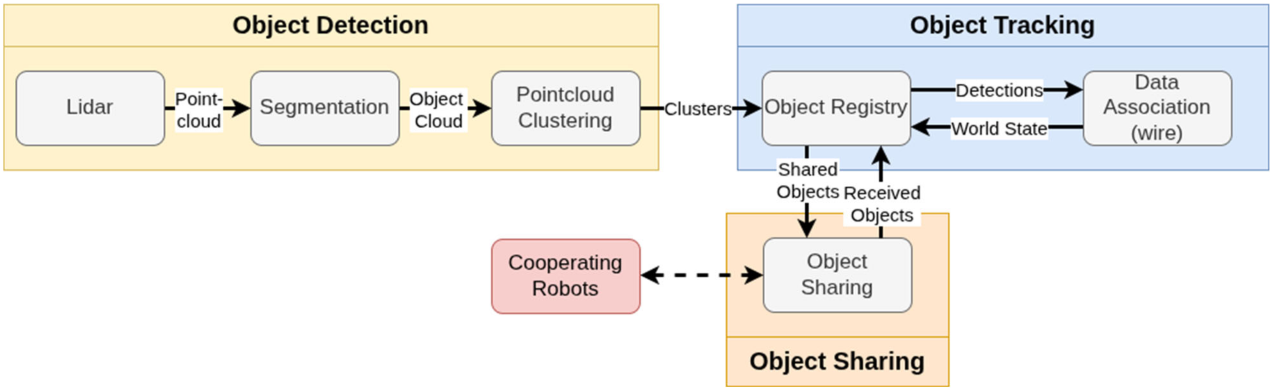

Figure 1: Cooperative tracking system architecture

improve situational awareness while navigating dynamic supply depots or warehouses, operating safely near people and other vehicles.

There are several current approaches to cooperative object tracking and cooperative perception in general. Many share raw sensor data or full point clouds or use complex machine learning-based algorithms for data fusion. These approaches, while effective, may not scale well for all types of systems and would not be feasible for applications where communication bandwidth is limited or communications are intermittent. Instead, we implemented an MHT-based algorithm to track and to manage detected objects. Rather than share unprocessed sensor data, we share processed, object-level data between assets. This reduces the necessary bandwidth and communication frequency, resulting in a scalable framework. In this work, we demonstrate our object tracking system using lidar point cloud clustering algorithms for object detection, however, the system is designed to ingest detection data from all types of perception sensors.

2. BACKGROUND

Collaborative object tracking and the broader concept of cooperative perception have been researched for a variety of applications, including CAVs, UASs, and AMRs. Much of the recent research in the CAV field focuses on tracking other vehicles for the purpose of path planning and collision avoidance [1, 2, 3, 4, 5]. Cooperative perception has also been investigated for off-road autonomy [6] and SAR [7], but studies in off-road or unstructured environments are not as prevalent. In the UAS field, cooperative perception and other collaborative behaviors have been explored for “swarm” applications [8].

Regardless of the specific application, key concerns for cooperative perception systems include scalability and communication constraints. This paper presents a generalized cooperative object tracking framework that can be applied to a variety of autonomous systems with different sensor modalities, focusing on scalability and low communication overhead.

3. SYSTEM OVERVIEW

The goal of the system is to facilitate sharing object data within a group of cooperating robots. It was developed using a ROS-based architecture. The system layout is presented in Figure 1, and the main components are described in more detail in the following sections.

3.1. Object Detection

The first step of the cooperative tracking pipeline is object detection. This detection can be accomplished using several types of sensors, the most common of which are lidar and cameras. In this work, we used a lidar sensor coupled with a Euclidean point cloud clustering algorithm for object detection; however, the important part is that the output of the detection subsystem is compatible with the `wire` package used for data association [9]. The lidar-based detection system could be replaced with an alternative detection method, such as camera-based detection, provided its output remains compatible with the `wire` package.

Within the object detection component, raw point clouds are first segmented into "ground" and "object" clouds. The "object" point cloud is then clustered into individual objects based on the spatial proximity of the points relative to one another. These objects are sent on to the object tracking subsystem for processing. A visualization of a point cloud clustered into individual objects is shown in Figure 2.

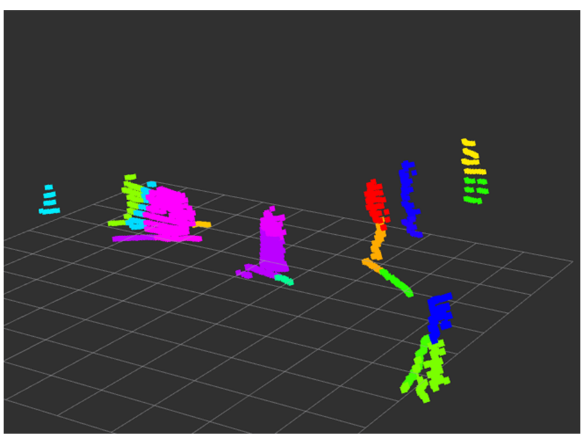

Figure 2: Clustered object point cloud

Once the clustered object point cloud is received from the object detection subsystem, it is filtered and distilled into a format compatible with `wire`. Because the focus of this research was the tracking of dynamic objects, static objects were filtered out of the clustered point clouds. A side effect of the current filtering approach is that the system stops tracking dynamic objects when they become stationary and only resumes tracking them once they move again. Incorporating camera data or other semantic information into the tracking system could enable it to determine which objects to track based on their type rather than solely on their motion. Detection messages corresponding to each clustered object are then sent to the `wire` package describing the objects' positions as well as any relevant semantic information.

3.2. Data Association

The next step in the cooperative object tracking pipeline is the data association step. Here, we must determine whether each detection represents a new object, clutter, or a previously tracked object. Data association strategies range from relatively simple global nearest neighbor algorithms to complex, machine learning-based approaches. We implemented an open-source, ROS-based MHT package called `wire` to handle data association. MHT-based systems track multiple possible hypotheses for object positions and trajectories, continuously updating them over time and selecting the hypothesis with the highest probability to represent the current state of the world. These types of algorithms are well-suited to dynamic environments and can maintain tracks in the face of clutter, occlusions, and intersecting trajectories.

We made several modifications to the `wire` package to improve its suitability for the cooperative tracking system. Originally, `wire` assumed that tracked objects never disappeared, which led to objects persisting after they had left the vehicle's sensor range, resulting in old objects accumulating around the edge of the world model. To address this issue, we updated the `wire` code to prune old objects that had not been observed for a specified period, allowing the system to stay focused on active and relevant objects. We also customized `wire` to output velocity estimates for tracked objects, which was not included in the original implementation. This modification was important to enable the prediction of object trajectories, which is necessary for the filtering logic implemented in the object sharing node.

3.3. Object Sharing

Once objects are tracked in a robot's world model, it must identify cooperating robots and share relevant object data with each of them. To enable this exchange, we developed an inter-robot communication node. The node utilizes a C++-based serialization library called Cista++ [10], allowing the communication framework to remain ROS-agnostic. Communication messages are sent using UDP over a radio network. The communication system supports the sharing of both "object evidence" messages, containing information about objects in the sharing robot's registry, and "vehicle information" messages, which contain information about robot positions, velocities, unique IDs, and IP addresses. The vehicle information messages are broadcast at set intervals and allow cooperating robots to discover one another dynamically. Once a new cooperating robot is discovered, it is added to a registry of cooperating robots. The position, velocity, and predicted trajectory of each cooperating robot are tracked by all robots in the network.

We implemented filtering logic to determine which objects to share with cooperating robots, minimizing unnecessary transmissions. The filtering process is based on the current position and predicted trajectory of each object relative to the position and trajectory of each cooperating robot. Objects in the registry are propagated forward in time using their most recent estimated position and velocity, and the same propagation is applied to each cooperating robot based on their last reported pose and velocity. Objects with current or predicted positions within a defined proximity threshold of a cooperating robot are shared, while those outside this threshold are excluded.

The proximity threshold and trajectory prediction horizon are tunable parameters, providing the ability to adjust filtering for different environments or use cases. Relevant objects selected for sharing are transformed into a global coordinate frame and transmitted over the radio network to the appropriate robots. This filtering approach reduces the communication burden, especially in highly cluttered environments, ensuring that each robot only shares objects relevant to its peers.

Once the shared object data is received from cooperating robots, it is deserialized, its timestamp is checked to verify that it is recent enough to use, and it is translated into a format compatible with the `wire` package. Next, it is forwarded to the data association subsystem (`wire`), where it is treated like detections from the robot's own sensors and fused with the current world model.

4. EXPERIMENTAL TESTING

To validate the cooperative tracking system, we deployed it on two small mobile robots, nicknamed SPAR and Rover. Several tests were conducted to assess the system's ability to track multiple dynamic objects in complex and cluttered environments.

4.1. Test Setup

The test platforms, pictured in Figure 3, were each equipped with a lidar sensor for perception and radios for inter-robot communications. Each robot has a sensor range of approximately 25 meters, and testing was conducted with the robots within 15 to 30 meters of each other.

Figure 3: Platforms used for experimental testing. Rover (left) and SPAR (right)

To mimic a cluttered environment, jersey barriers and traffic cones were set up in various configurations between the robots to occlude parts of each robot's field of view. An example of one of these configurations is shown in Figure 4. Data was collected with various types and numbers of dynamic obstacles moving throughout the test area.

Figure 4: Example test setup

4.2. Results

To evaluate the accuracy of the tracking system, data was collected of a pedestrian and a car moving through the test area. The tracking system output was compared to the GPS position of the car, while the pedestrian followed a pre-measured route to provide a known position reference. During these tests, the two robots remained stationary to minimize potential sources of error. The mean and maximum tracking errors for both the pedestrian and car are presented in Table 1. The larger mean errors observed when tracking the car were likely due to the lidar data only picking up one side of the car, causing the centroid of the detected object to align with the edge of the vehicle rather than the center. Integrating camera data could enable adjustments to the detection point based on the object's classification and estimated dimensions. The small differences in tracking error between the two robots were likely due to the geometry of the test setup and small differences in their individual lidar accuracy and calibration.

Table 1: Single object tracking accuracy

Robot	Tracking Error (m)			
	Vehicle		Pedestrian	
	Mean	Max	Mean	Max
SPAR	1.3	2.7	0.40	1.1
Rover	1.2	3.1	0.20	0.63

To determine the tracking performance in the presence of multiple dynamic objects, data was recorded with three pedestrians moving in various patterns throughout the test area. The objective was for each robot to maintain a unique and consistent track for each pedestrian throughout the duration of the test. Table 2 shows the percentage of the five-minute trial during which each robot achieved this goal. Both robots were able to track all three pedestrians for over 80% of the total time when object sharing was enabled. Due to the test setup and the paths taken by the pedestrians, tracking all three for 100% of the time was not possible, as there were portions of the test where one or multiple pedestrians were out of view of both robots. The SPAR platform demonstrated a 44% improvement over single-vehicle tracking, while the Rover showed a 57% increase. The difference in improvement between the two robots is likely due to the geometry of the test setup and the paths of the objects resulting in the objects being occluded from one robot more often than the other.

Table 2: Multi-object tracking results

Robot	% Time all objects tracked (no sharing)	% Time all objects tracked (sharing)
SPAR	37.6%	81.7%
Rover	30.7%	87.6%

During testing, the average bandwidth used to share object data was measured both for "high-clutter" environments (10–20 shared objects) and "low-clutter" environments (2–6 shared objects) for a group of two robots. Using the current communication process, where objects are filtered and sent selectively to each cooperating robot, bandwidth usage will scale quadratically with each additional robot, assuming each robot shares an equal number of objects. Figure 5 presents the estimated bandwidth usage in "high" and "low" clutter environments for groups of up to 10 robots, extrapolated from the measurements obtained for a group of two robots.

5. CONCLUSIONS

The results presented in this paper show the performance of a cooperative object tracking system leveraging an open source MHT-based object tracking package for data association. The system was deployed and tested using two robots in a cluttered environment with multiple occlusions. The testing provided valuable insight into the strengths and limitations of the system. The system was successful in increasing tracking duration for a group of dynamic objects in the presence of occlusions.

We observed limitations in the accuracy of shared object positions due to the inaccuracy of the robots' global position estimates. To address this issue, a cooperative localization algorithm could be implemented to align the robots' coordinate frames, or in some specific applications, RTK-corrected GPS could be incorporated into the localization solution. We also observed limitations in the handling of static objects, which could be addressed by incorporating semantic data from a camera to better classify types of objects. Despite the limitations, this system would be useful for robotic formations, autonomous convoys, collaborative scouting or sentry behaviors, or any application where robots must cooperate in a shared workspace near soldiers or other dynamic obstacles.

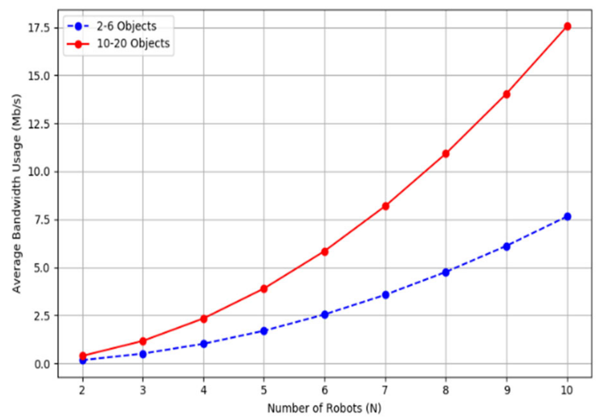

Figure 5: Measured and extrapolated communication bandwidth usage versus number of cooperating robots

6. FUTURE WORK

There are several additions that could be made to improve the accuracy and robustness of the object tracking system, along with opportunities for further testing.

Cooperative localization. The system currently relies on each robot's estimated pose in the global frame to transform shared objects into their local coordinate frames. This reliance on estimated global pose leads to errors due to offsets and inaccuracies in the robots' global localization solutions. A cooperative localization algorithm could address this issue, allowing the robots to register against shared objects or static landmarks to align their coordinate frames and improve overall tracking accuracy.

Additional sensor modalities. For this work, testing was limited to lidar-based object detection. Incorporating camera-based detection could add semantic information, enabling object classification and tailored motion models for different object classes. The addition of semantic data could improve both the accuracy and robustness of the system, especially in cluttered environments with many dynamic and static obstacles.

Larger groups. Testing in this work was limited to a group of two cooperating robots. Future evaluations should include larger groups to validate the system's performance at scale.

7. REFERENCES

[1] S. Fang and H. Li, "Multi-Vehicle Cooperative Simultaneous LiDAR SLAM and Object Tracking in Dynamic Environments," *IEEE Transactions on Intelligent Transportation Systems,* vol. 25, no. 9, pp. 11411-11421, 9 2024.

[2] D. Qiao and F. Zulkernine, "Adaptive Feature Fusion for Cooperative Perception using LiDAR Point Clouds," in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA, 2023.

[3] A. Rauch, S. Maier, F. Klanner and K. Dietmayer, "Inter-vehicle object association for cooperative perception systems," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013.

[4] A. Willecke, R. Stahl and L. C. Wolf, "Object Fusion for Collective Perception: Will it Scale?," in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, 2024.

[5] Q. Xie, X. Zhou, C. Wang, T. Qiu and W. Qu, "KeyCoop: Communication-Efficient Raw-Level Cooperative Perception for Connected Autonomous Vehicles via Keypoints Extraction," in *2024 21st Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2024.

[6] Y. Yue, C. Yang, J. Zhang, M. Wen, Z. Wu, H. Zhang and D. Wang, "Day and Night Collaborative Dynamic Mapping in Unstructured Environment Based on Multimodal Sensors," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[7] J. P. Queralta, J. Taipalmaa, B. Can Pullinen, V. K. Sarker, T. Nguyen Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju and T. Westerlund, "Collaborative Multi-Robot Search and Rescue: Planning, Coordination, Perception, and Active Vision," *IEEE Access,* vol. 8, pp. 191617-191643, 2020.

[8] Z. Du, C. Luo, G. Min, J. Wu, C. Luo, J. Pu and S. Li, "A Survey on Autonomous and Intelligent Swarms of Uncrewed Aerial Vehicles (UAVs)," *IEEE Transactions on Intelligent Transportation Systems,* vol. 26, no. 10, pp. 14477-14500, 2025.

[9] J. Elfring, S. Dries, M. J. G. Molengraft and M. Steinbuch, "Semantic world modeling using probabilistic multiple hypothesis anchoring," *Robotics and Autonomous Systems,* vol. 61, no. 2, pp. 95-105, 1 2 2013.

[10] F. Gundling, "Cista++," 2023. [Online]. Available: https://github.com/felixguendling/cista.

8. CONTACT INFORMATION

William Kennedy
Research Engineer
Southwest Research Institute
6220 Culebra Rd
San Antonio, TX
william.kennedy@swri.org
210-522-6241

9. ACRONYMS

MHT	Multiple Hypothesis Tracking
CAV	Connected and Automated Vehicles
UAS	Uncrewed Aerial System
AMR	Autonomous Mobile Robot
SAR	Search and Rescue
ROS	Robot Operating System
UDP	User Datagram Protocol
RTK	Real-Time Kinematic