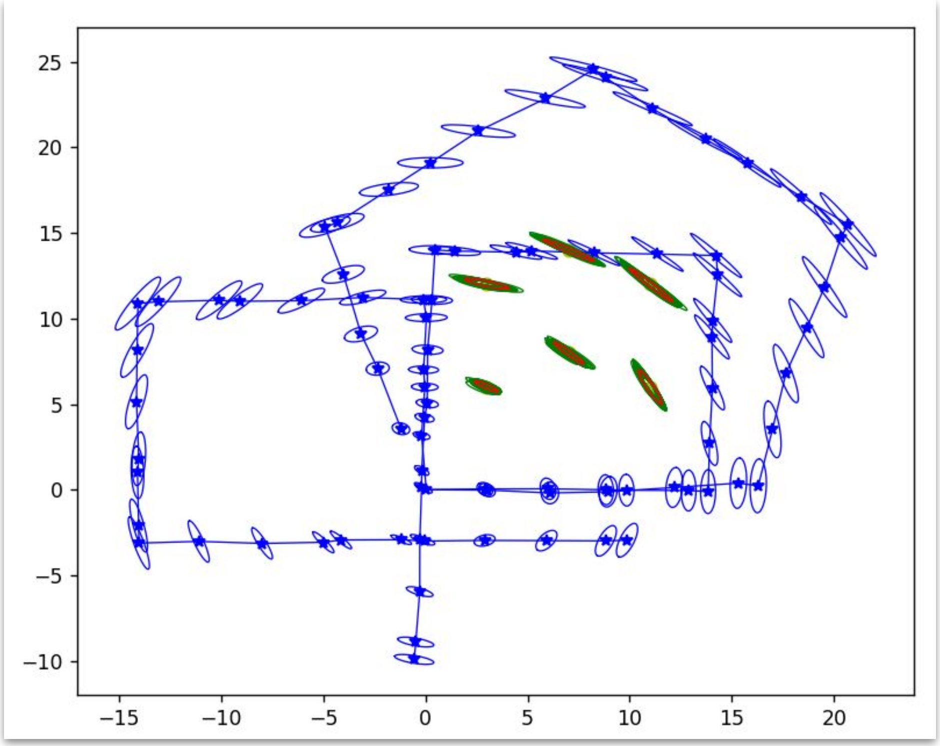
Multi-Agent SLAM Using Extended Kalman Filter

Course Name: 16-833 Robot Localization and Mapping

Team Members: Ryan Wu, Kevin Tang, Nithya Sampath, Tianqi Yu



Agenda

01 **Introduction & Motivation**

02 **Related Work**

03 **Algorithm**

04 **Evaluation**

05 **NVIDIA Isaac Visualization**

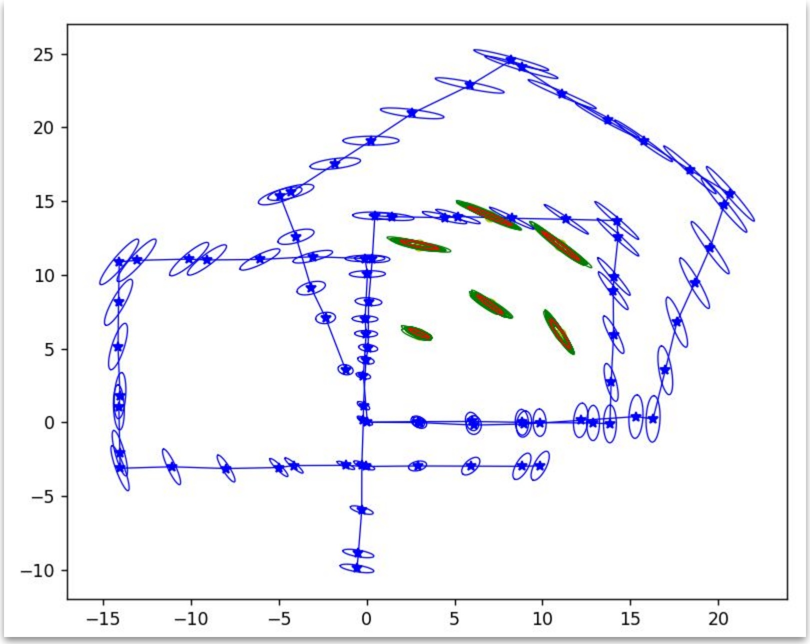
06 **Lesson Learned**

Project Introduction & Motivation

The team took inspiration from HW2 and would like to extend the EKF's capability to handle multiple agents and also provide a better visualization tool.

Our proposed method:

- Generate synthetic datasets of the robot's measurements and control data.
- Solved the multi-agent problems using our multi-EKF solver.
- Compare the estimation results between single agent and multi-agent.
- Re-create the landmarks and robot motion in 3D using NVIDIA Isaac Sim.



Related Work

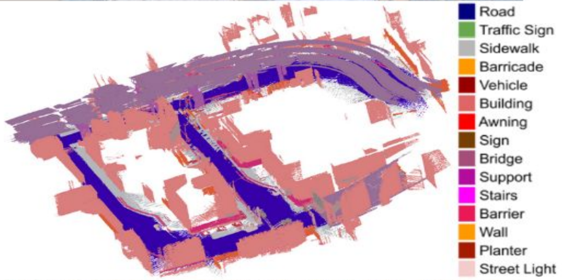
Kimera-Multi: Robust, Distributed, Dense Metric-Semantic SLAM for Multi-Robot Systems

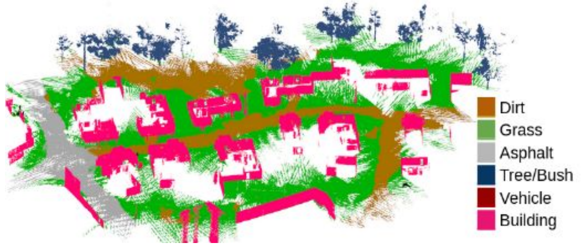
Kimera-Multi

- A fully distributed, fully modular multi-robot system for robust and dense metric-semantic SLAM.
- A framework capable of identifying and rejecting incorrect inter/intra-robot loop closures.

Advantages to other multi-agent frameworks

- A more robust and accurate trajectory estimation.
- Generates 3D meshes with improved metric-semantic accuracy.
- Significant robot communication reductions and increased efficiency.
- Can achieve estimation errors comparable to centralized SLAM systems.





Tian, Y., Chang, Y., Arias, F. H., Nieto-Granda, C., How, J. P., & Carlone, L. (2021). *Kimera-Multi: Robust, Distributed, Dense Metric-Semantic SLAM for Multi-Robot Systems* (arXiv:2106.14386). arXiv. http://arxiv.org/abs/2106.14386

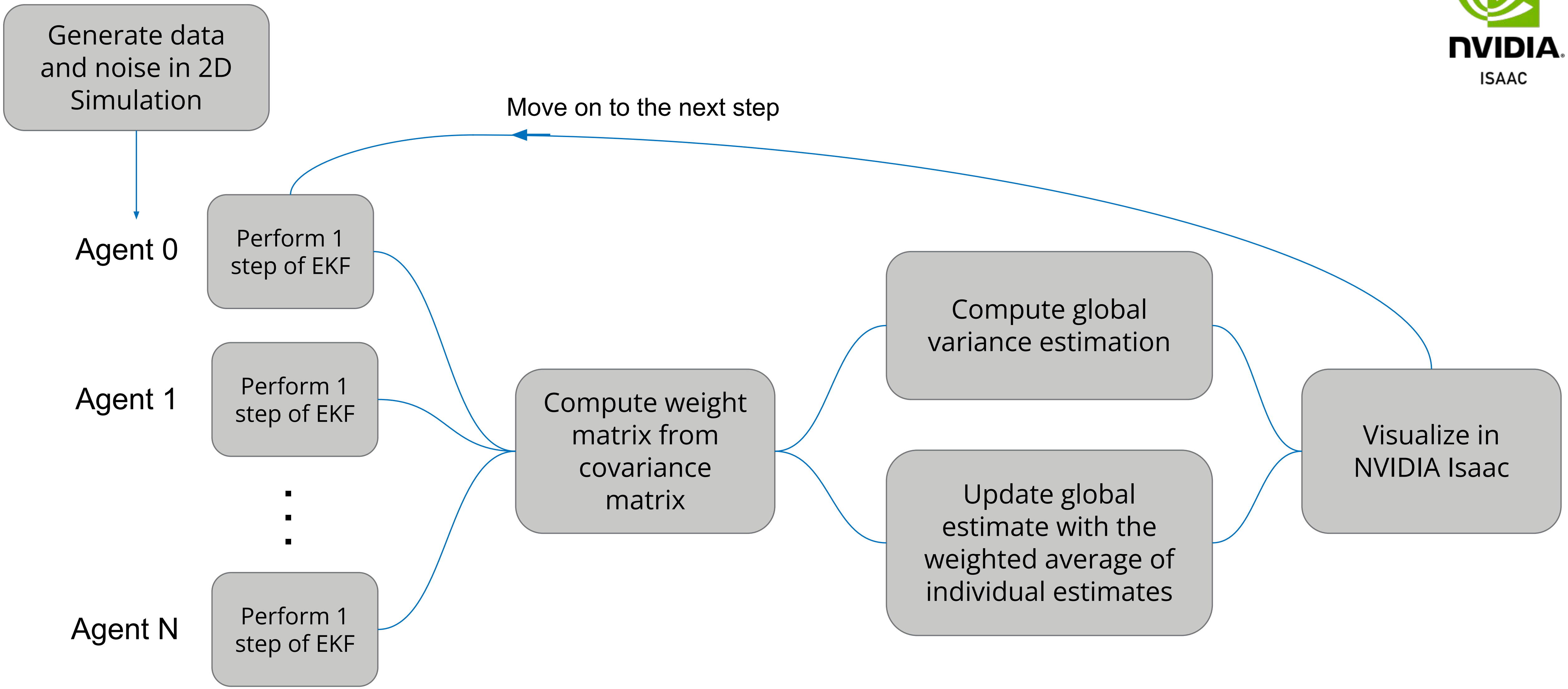
Multi-Robot SLAM via Information Fusion (Extended Kalman Filter)

Information Fusion Method

- Our method is adapted from Sasaoka, et. al's information fusion technique.
- Main Steps:
  - 1. The robots compute (and maintain) state estimates $\hat{\boldsymbol{x}}^i_{t|t}$ and covariance matrices $P^i_{t|t}$ based on their individual measurements.
  - 2. Collect all of the state estimates (after each robot has performed its update step) to compute cross-covariance matrices $P^{ij}_{t|t}$.
  - 3. Find optimal weight matrices $A = [A_1 \; A_2 \; \cdots \; A_M]$ which will be multiplied by the post-update state estimates to obtain a final global state estimate.
- Although our implemented method is still a weighted average method, it deviates from the paper's method slightly in that we (1) do not attempt to track the positions of other agents, and (2) use scalar weight values computed based on the covariance.

Sasaoka, T., Kimoto, I., Kishimoto, Y., Takaba, K., & Nakashima, H. (2016). *Multi-robot slam via information fusion extended kalman filters*. IFAC-PapersOnLine, 49(22), 303–308. https://doi.org/10.1016/j.ifacol.2016.10.414

Algorithm

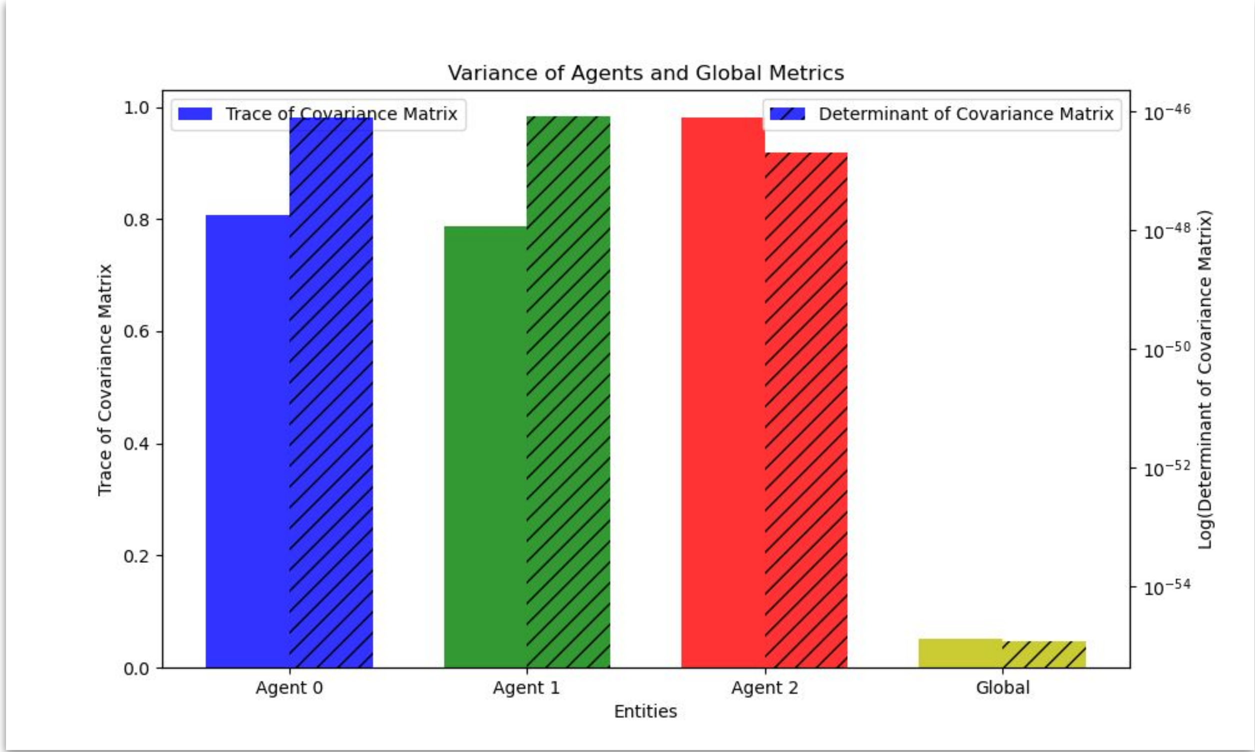

Generate data and noise in 2D Simulation
Move on to the next step
Agent 0
Perform 1 step of EKF
Agent 1
Perform 1 step of EKF
Agent N
Perform 1 step of EKF
Compute weight matrix from covariance matrix
Compute global variance estimation
Update global estimate with the weighted average of individual estimates
Visualize in NVIDIA Isaac

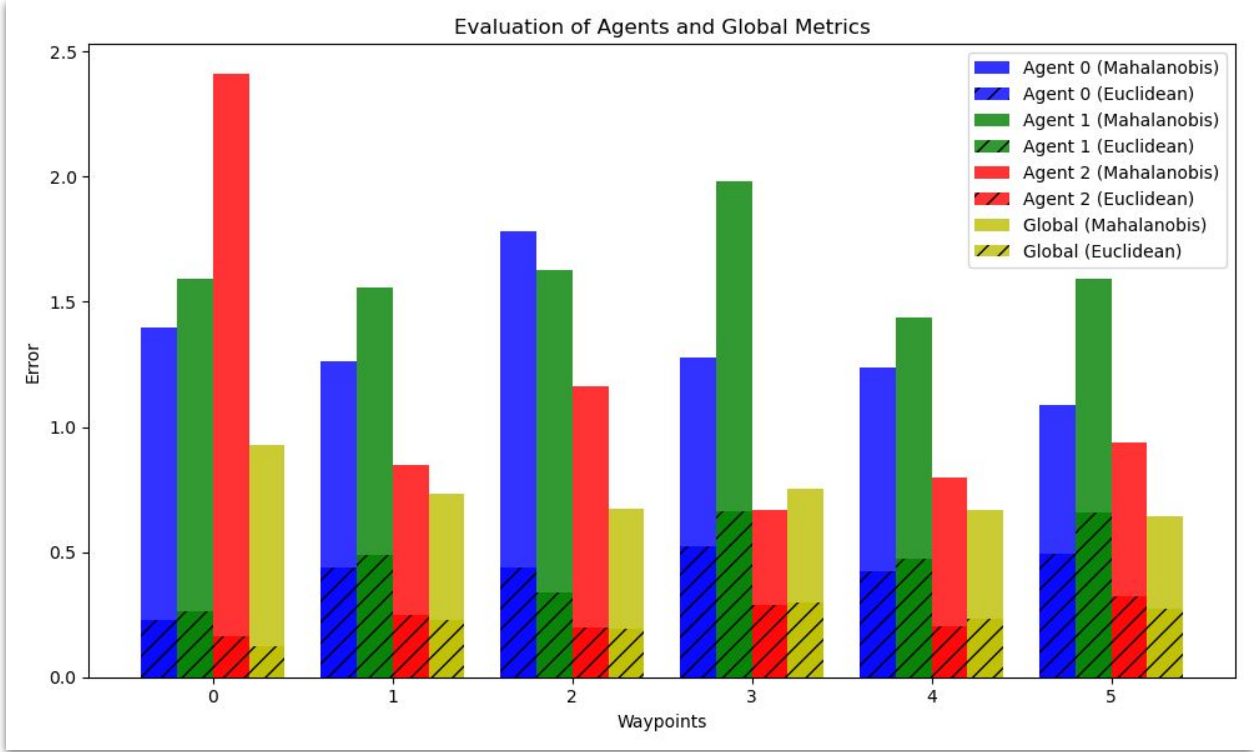
Evaluation

Evaluation: constant measurement noise

Metrics:

- Compare the trace of the covariance matrix.
- Compare the determinant of the covariance matrix.
- Compare the Euclidean error and Mahalanobis error.

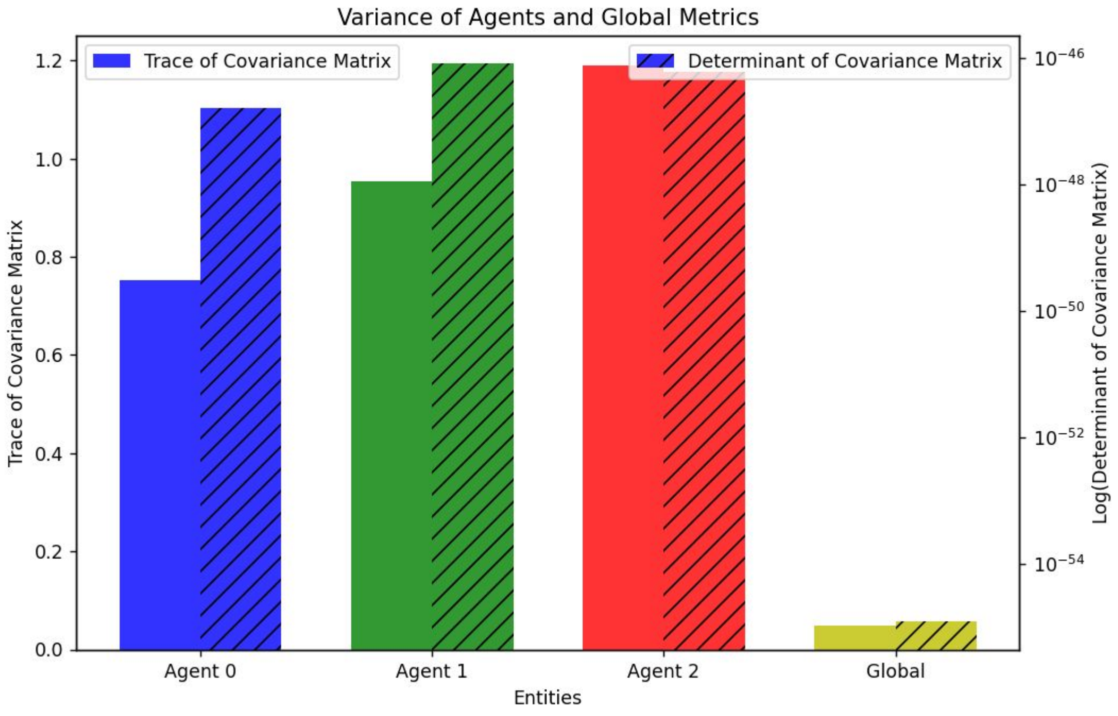


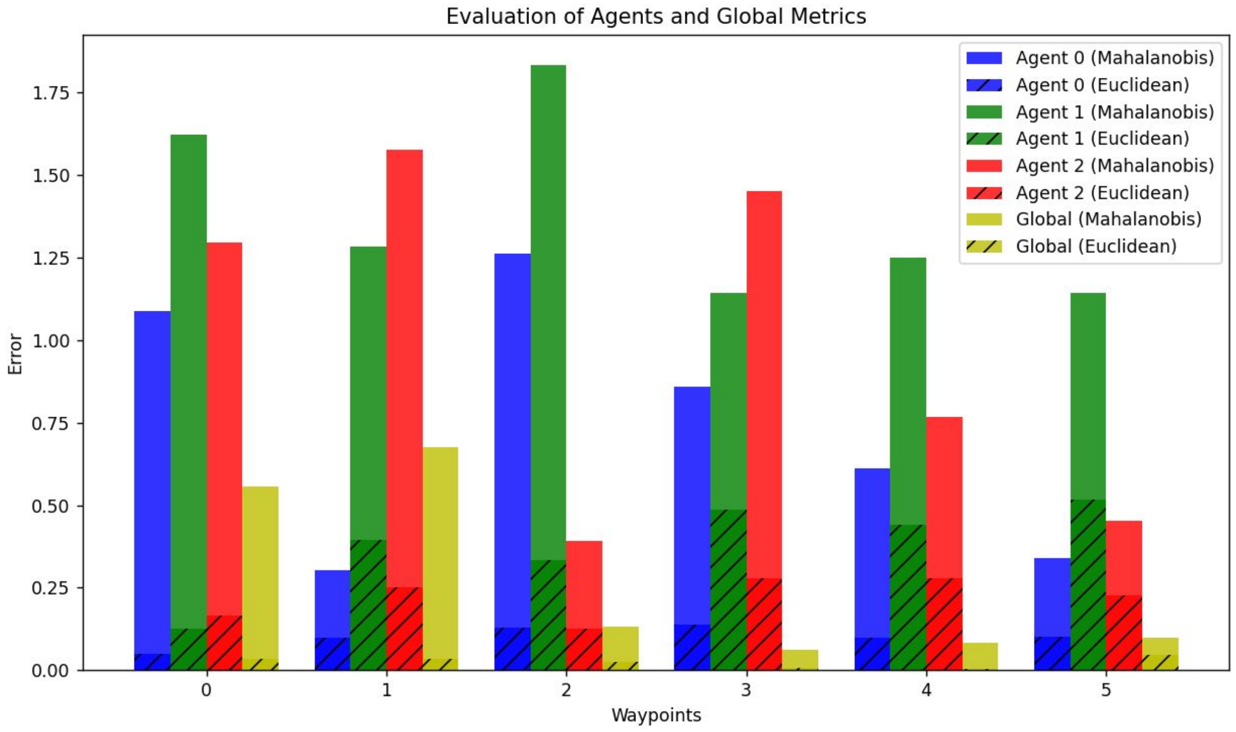


Evaluation: varying measurement noise

The same metrics used but with different measurement noise:

- Agent0: low noise.
- Agent1: standard noise.
- Agent2: high noise.





Evaluation: varying measurement noise

Comparing Multi-EKF estimated landmark positions to ground truth:

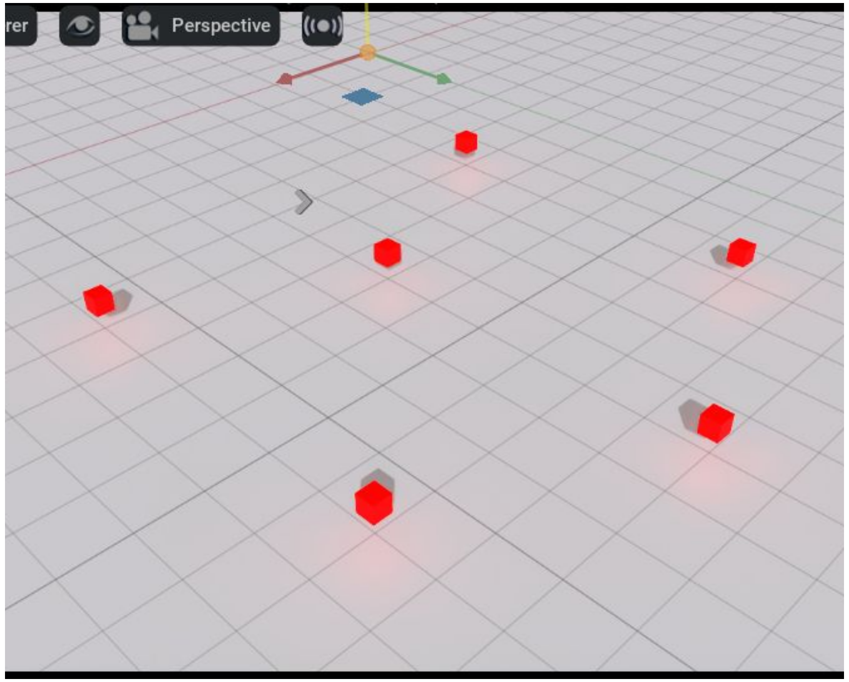
Landmark number	Landmark's ground truth x, y	Agent0's x, y (Single-agent EKF)	Agent1's x, y (Single-agent EKF)	Agent2's x, y (Single-agent EKF)	Landmark's x, y (Multi-agent EKF)
1	3, 6	2.988, 5.951	3.126,6.005	2.871, 6.104	2.988, 6.031
2	3, 12	2.906, 12.028	3.388, 11.931	2.779, 12.118	3.010, 12.034
3	7, 8	6.946, 8.116	7.215, 7.745	6.902, 8.078	7.015, 7.981
4	7, 14	6.864, 14.028	7.425, 13.765	6.772, 14.162	7.006, 14
5	11, 6	10.923, 6.061	11.204, 5.611	10.882, 6.254	10.996, 5.997
6	11, 12	10.919, 12.059	11.389, 11.659	10.830, 12.149	11.032, 11.969

NVIDIA Isaac Sim

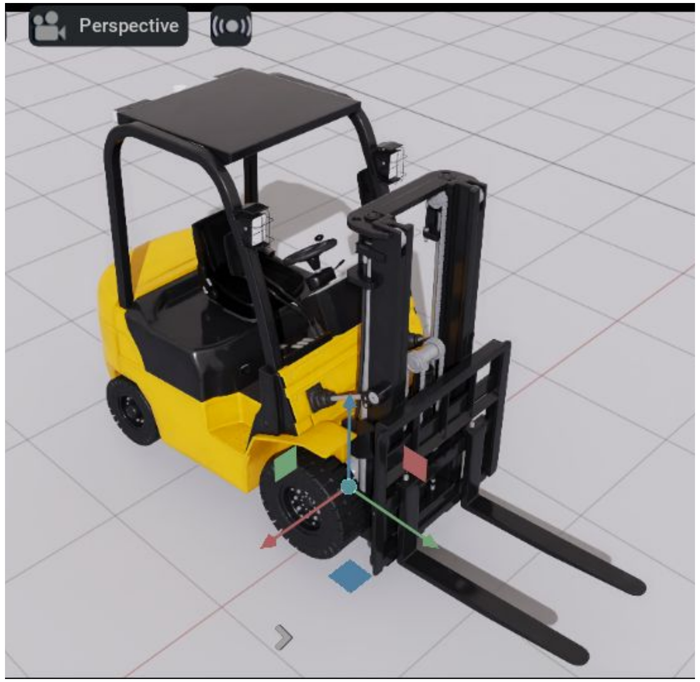
NVIDIA Isaac Sim Visualization (1/2)

The simulation environment setup:

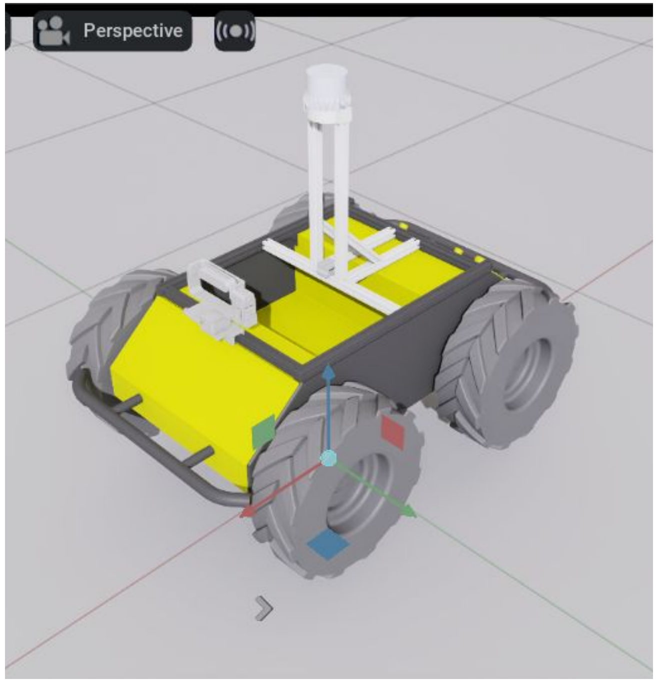
Ground truth landmarks



Agent: Forklift



Agent: Theia UGV



NVIDIA Isaac Sim Visualization (2/2)

The simulation animation for both agents running different control inputs:

Lesson Learned

The Good:

- Multi-agent gave us better landmark estimation compared to single-agent.
- Explored the state-of-the-art frameworks in the multi-agent SLAM space.
- Experienced with a powerful 3D visualization tool for robotic research.

The Not so Good:

- Data fusion and association problem for multi-agent system.
- Lack the practical experience in generating datasets for SLAM operations.
  - Creating 2D datasets from 3D is not trivial (quaternion math).
- Using naive global covariance generation method for result evaluation.

Thank you for your time!

Questions?