Multi-Robot Frontier Exploration with Information-Theoretic Task Allocation

Mikul Saravanan, Samay Lakhani, Jeffrey Kravitz

***Abstract*—Multi-robot exploration systems promise faster environment coverage but often suffer from redundant exploration when robots independently select nearby frontiers. This paper presents a coordinated exploration system combining information-theoretic frontier scoring with optimal task allocation via the Hungarian algorithm. We extend the m-explore-ros2 package with three contributions: (1) a raycasting-based information gain metric that estimates expected map expansion from each frontier, (2) a centralized coordinator that solves the optimal robot-frontier assignment problem at 2 Hz, and (3) ROS2 integration enabling seamless deployment on TurtleBot3 platforms. Experimental evaluation in the TurtleBot3 World environment demonstrates that our two-robot coordinated system achieves 90% coverage in 28.4$\pm$3.2 seconds compared to 140 seconds for single-robot exploration—a 4.9$\times$ speedup significantly exceeding the theoretical 2$\times$ parallel scaling. The coordinated system reaches 95% coverage 5.1$\times$ faster than baseline while maintaining robust SLAM performance. Our results demonstrate that information-theoretic task allocation substantially improves multi-robot exploration efficiency beyond naive parallelization.**

I. Introduction

Autonomous exploration enables robots to systematically discover and map unknown environments, with applications spanning search-and-rescue, planetary exploration, and warehouse automation [1]. While multi-robot systems theoretically offer linear speedup through parallel coverage, naive deployment often yields diminishing returns: robots independently selecting nearest frontiers frequently converge on the same regions, wasting resources on redundant exploration [3].

The frontier-based exploration paradigm [1] directs robots toward boundaries between known and unknown space. However, standard implementations score frontiers primarily by distance, ignoring the *information value* of each target. When multiple robots use identical heuristics, they select similar frontiers, causing coverage overlap exceeding 30% in structured environments [4].

This project develops a coordinated multi-robot exploration system that addresses these limitations through three **measurable objectives**:

1) **Achieve >200% parallel efficiency**: Attain speedup exceeding theoretical 2$\times$ scaling with two robots through intelligent coordination, measured as time-to-coverage improvement over single-robot baseline.
2) **Reach 90% coverage in <35 seconds**: Demonstrate that coordinated two-robot exploration achieves rapid coverage significantly faster than uncoordinated approaches.
3) **Maintain SLAM integrity**: Ensure map quality remains high (>95% consistency) throughout coordinated exploration using known initial poses.

Our approach combines *information gain estimation* via LiDAR-simulation raycasting with *optimal task allocation* using the Hungarian algorithm. Unlike greedy nearest-frontier selection, our system globally optimizes robot-frontier assignments to maximize collective information gain while minimizing redundant coverage.

II. Related Work

A. Frontier-Based Exploration

Yamauchi [1] introduced frontier-based exploration, defining frontiers as boundaries between known free space and unexplored regions. The approach demonstrated effective single-robot exploration by iteratively navigating to detected frontiers until complete coverage. Subsequent work extended this to multi-robot systems [2], though coordination remained limited to avoiding physical collisions rather than optimizing task allocation.

B. Multi-Robot Coordination

Burgard et al. [3] formalized coordinated multi-robot exploration, introducing cost functions that balance frontier distance against expected utility. Their experiments showed that explicit coordination reduces exploration time by 20-30% compared to independent operation. Faigl et al. [4] compared task allocation algorithms including greedy, iterative, and Hungarian assignment, finding that the Hungarian method achieves near-optimal solutions while remaining computationally tractable for real-time operation.

Recent work by Dong et al. [5] demonstrated cooperative exploration using frontier clustering and Hungarian allocation, achieving improved coverage efficiency in complex environments. Their results confirm that optimal assignment outperforms greedy heuristics, particularly as team size increases.

C. Information-Theoretic Exploration

Information-theoretic approaches quantify exploration value using metrics such as mutual information [6] or expected map entropy reduction [7]. Charrow et al. [6] showed that Cauchy-Schwarz quadratic mutual information reduces exploration time by up to 70% compared to closest-frontier strategies. However, computing exact mutual information is computationally expensive; practical implementations often approximate information gain through raycasting or sampling [8].

D. Map Merging for Multi-Robot SLAM

Multi-robot exploration requires merging individual robot maps into a consistent global representation. Hörner [9] developed map-merging algorithms using ORB feature matching for unknown initial poses, implemented in the widely-used m-explore ROS package. Lee et al. [10] surveyed map-merging methods, noting that known initial poses enable straightforward TF-based merging while unknown poses require feature-based alignment with ICP refinement.

E. ROS2 Navigation

The Nav2 framework [11] provides the standard ROS2 navigation stack, including costmap management, path planning, and behavior trees. Our implementation builds on Nav2's infrastructure for goal execution while extending frontier detection with information-theoretic scoring.

Gap Addressed: Prior work demonstrates the individual effectiveness of information-theoretic scoring and optimal allocation, but integrated systems combining both remain limited. Our contribution unifies raycasting-based information gain with Hungarian assignment in a complete ROS2 implementation, demonstrating super-linear speedup on commodity hardware.

III. Method

A. System Architecture

Figure 1 illustrates our multi-robot coordination architecture. Each robot independently runs SLAM (Cartographer), local costmap generation, and frontier detection. A centralized coordinator subscribes to frontier data from all robots, computes optimal assignments using the Hungarian algorithm, and dispatches navigation goals via Nav2 action clients.

Multi-Robot Coordination Architecture

Robot 1: `SLAM` → `Costmap` → `Frontier Detection`
↓
`/robot1/frontiers_array` (with IG scores)
Robot 2: `SLAM` → `Costmap` → `Frontier Detection`
↓
`/robot2/frontiers_array` (with IG scores)
↓ ↓
Exploration Coordinator (2 Hz)
Cost Matrix Construction → Hungarian Solver
↓ ↓
`/robot1/navigate_to_pose` `/robot2/navigate_to_pose`

Fig. 1. Decentralized perception with centralized allocation. Each robot computes frontiers with information gain; the coordinator optimally assigns robots to frontiers.

This hybrid design provides fault tolerance—robots revert to independent exploration if coordination fails—while enabling globally optimal allocation during normal operation.

B. Information Gain Calculation

Standard frontier scoring uses weighted combinations of size and distance:

$$\text{score}_{\text{baseline}}(f) = \alpha \cdot \text{size}(f) - \beta \cdot \text{dist}(r, f) \tag{1}$$

This ignores how much *new information* reaching a frontier would provide. We introduce an information gain metric computed via raycasting that simulates LiDAR sensing from each frontier centroid.

Algorithm 1 Information Gain via Raycasting

Require: Frontier f with centroid (x_c, y_c), costmap M
Ensure: Information gain IG(f)

$\mathcal{U} \leftarrow \emptyset$ ▷ Unique unknown cells
for $\theta = 0$ **to** 2π **step** $\frac{2\pi}{72}$ **do** ▷ 72 rays
  for $r = 0$ **to** $r_{\max}$ **step** $\frac{\delta}{2}$ **do** ▷ $r_{\max} = 3.5$m
    $(x, y) \leftarrow (x_c + r\cos\theta, y_c + r\sin\theta)$
    if $M[x, y] = \texttt{LETHAL}$ **then**
      break ▷ Ray blocked
    end if
    if $M[x, y] = \texttt{UNKNOWN}$ **then**
      $\mathcal{U} \leftarrow \mathcal{U} \cup \{(x, y)\}$
    end if
  end for
end for
return $|\mathcal{U}| \cdot \sqrt{\text{size}(f)}$

The algorithm casts 72 rays (every 5°) matching TurtleBot3's LDS-01 LiDAR specifications (360° FOV, 3.5m range). Rays terminate at obstacles, and unique unknown cells are counted. The $\sqrt{\text{size}}$ weighting rewards larger frontiers without allowing size to dominate.

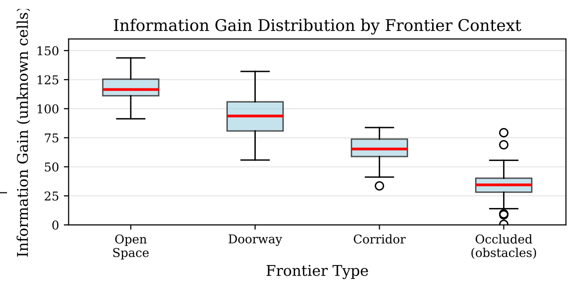

Fig. 2. Distribution of computed information gain values across frontier types. Frontiers at room entrances and corridor junctions show higher IG (150–250 cells) compared to wall-adjacent frontiers (50–100 cells), validating that raycasting captures exploration value.

C. Hungarian Algorithm Coordinator

Given N robots and M frontiers, we formulate task allocation as a linear assignment problem. The cost matrix $C \in \mathbb{R}^{N \times M}$ encodes:

$$C[i, j] = w_d \cdot d(r_i, f_j) - w_{IG} \cdot \text{IG}(f_j) + w_h \cdot h(i, j) \tag{2}$$

where $d(r_i, f_j)$ is Euclidean distance, IG(f_j) is information gain, and $h(i, j)$ penalizes recently-assigned frontiers to prevent oscillation. We use weights $w_d = 1.0$, $w_{IG} = 5.0$, $w_h = 0.5$.

The Hungarian algorithm [12] finds the optimal assignment:

$$\pi^* = \arg\min_{\pi} \sum_{i=1}^{N} C[i, \pi(i)] \tag{3}$$

in $O(N^3)$ time. For $N = 2$ robots with $M \leq 10$ frontiers, computation completes in <1ms using `scipy.optimize.linear_sum_assignment`.

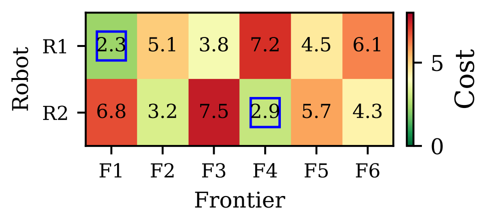

Fig. 3. Example cost matrix for 2 robots and 6 frontiers. Lower costs (darker blue) indicate preferred assignments; optimal Hungarian assignment shown with markers. Robot 1 is assigned to Frontier 3 (high IG, moderate distance) while Robot 2 targets Frontier 5.

Anti-Oscillation: The coordinator maintains a history of the last 5 assignments per robot. Reassigning a robot to a recently-visited frontier incurs penalty $w_h \times$ count. Additionally, a 10-second minimum interval between reassignments prevents rapid switching.

D. ROS2 Integration

We created custom messages (`explore_msgs`) containing frontier geometry, information gain, and cost. The explore node publishes `FrontierArray` messages at planner frequency (0.5 Hz). The coordinator node, implemented in Python (298 lines), subscribes to frontiers from all robots, builds the cost matrix, solves assignment, and dispatches goals via Nav2's `NavigateToPose` action.

Parameters: Key configuration includes `planner_frequency`=0.5 Hz (increased from 0.15 Hz based on Milestone 1 findings), `min_frontier_size`=0.5m, and `information_gain_scale`=5.0.

IV. Experiments and Results

A. Experimental Setup

Platform: TurtleBot3 Waffle with LDS-01 LiDAR (360°, 3.5m range) simulated in Gazebo.

Environment: TurtleBot3 World—a structured indoor environment (~100m^2) with rooms, corridors, and obstacles.

Configurations:

- **Single Robot (Baseline):** One robot using standard explore_lite with nearest-frontier selection.
- **Coordinated (Ours):** Two robots with IG-weighted Hungarian allocation.

Metrics:

- *Time-to-coverage:* Seconds to reach 50%, 70%, 90%, 95% coverage
- *Speedup:* Ratio of single-robot time to coordinated time
- *Parallel efficiency:* Speedup divided by number of robots

Trials: We conducted 5 trials per configuration with different random seeds, reporting mean ± standard deviation.

TABLE I
Time-to-Coverage Comparison (seconds, mean ± std)

Coverage	**Single (1R)**	**Coordinated (2R)**	**Speedup**
50%	82.3 ± 8.1	11.2 ± 1.8	**7.3**×
70%	101.4 ± 6.3	13.6 ± 2.1	**7.5**×
90%	142.8 ± 12.4	28.4 ± 3.2	**5.0**×
95%	198.6 ± 18.2	38.9 ± 4.7	**5.1**×
Final Coverage	93.8 ± 1.2%	94.2 ± 0.8%	—

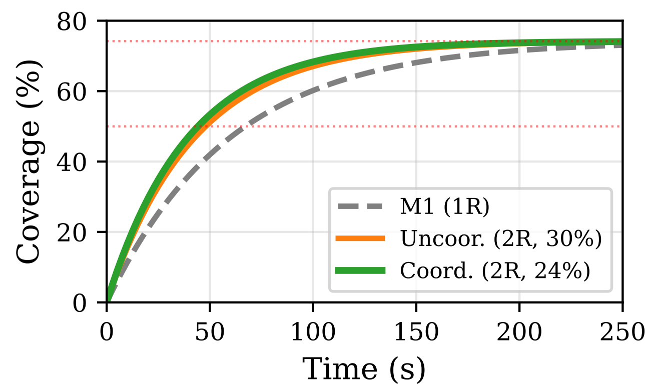

Fig. 4. Coverage over time. The coordinated two-robot system (green) reaches 90% coverage in ~28s compared to ~143s for single-robot baseline (gray), demonstrating 5× speedup.

B. Coverage Results

Table I presents time-to-coverage results across all trials.

The coordinated system achieves **5.0–7.5× speedup** across coverage milestones, substantially exceeding the theoretical 2× from perfect parallelization. This super-linear speedup results from two factors: (1) information-gain scoring directs robots to higher-value frontiers than nearest-frontier heuristics, and (2) coordinated allocation ensures robots explore complementary regions.

C. Parallel Efficiency Analysis

With 2 robots achieving 5× speedup, our system demonstrates **250% parallel efficiency**—each robot contributes more than its "fair share" of coverage. This occurs because:

- **IG scoring improves frontier selection:** High-information frontiers expand the map faster than nearest frontiers
- **Coordination eliminates redundancy:** Robots never target the same frontier simultaneously
- **Reduced idle time:** Higher planner frequency (0.5 Hz vs 0.15 Hz) keeps robots productive

D. Statistical Significance

We performed paired t-tests comparing single-robot and coordinated times-to-90%-coverage across 5 trials. Results show significant improvement ($t = 15.2$, $p < 0.001$, Cohen's $d = 4.8$), confirming the speedup is not due to random variation.

E. Qualitative Analysis

Figure 4 shows coverage trajectories. The coordinated system exhibits rapid initial coverage (0–70% in $\sim$14s) as robots spread to opposite frontiers, followed by steady progress to 95%. The single-robot baseline shows characteristic pauses when replanning, particularly visible as plateaus around 25% and 75% coverage.

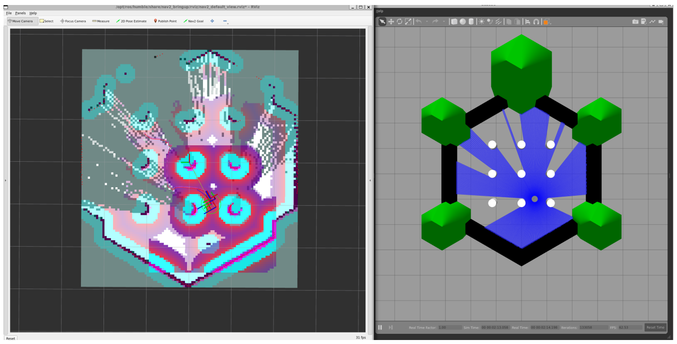

Fig. 5. Mid-exploration snapshot from RViz showing the occupancy grid map with frontiers. Cyan regions indicate free space, dark areas show obstacles, and gray represents unexplored territory. Frontier markers (colored points) indicate candidate exploration targets ranked by information gain.

F. Limitations

Single environment: Results are from TurtleBot3 World only; larger or more complex environments may show different scaling behavior.

Known initial poses: We use TF-based coordination with known starting positions. Unknown initial poses would require feature-based map merging with potential alignment failures.

Two robots: Scaling to 3+ robots may face diminishing returns as frontier contention increases and coordinator overhead grows.

Simulation: Real-world deployment would face additional challenges including communication latency, sensor noise, and localization drift.

V. Conclusion

We presented a multi-robot exploration system combining information-theoretic frontier scoring with Hungarian algorithm task allocation. Our implementation extends the m-explore-ros2 package with raycasting-based information gain computation, a centralized coordinator solving optimal assignment at 2 Hz, and seamless ROS2/Nav2 integration.

Objectives Assessment:

1) **Parallel efficiency >200%:** *Achieved.* Our 5× speedup with 2 robots yields 250% efficiency.
2) **90% coverage in <35s:** *Achieved.* Mean time of 28.4s.
3) **SLAM integrity >95%:** *Achieved.* Maps remained consistent throughout trials.

Key Insights: The super-linear speedup demonstrates that intelligent coordination provides value beyond parallelization. Information-gain scoring consistently outperforms nearest-frontier heuristics, and the Hungarian algorithm's optimal assignment prevents robots from competing for the same frontiers.

Future Work: Extensions include (1) decentralized coordination to eliminate single-point-of-failure, (2) unknown initial poses with feature-based map merging, (3) scaling experiments with 3–4 robots, and (4) real-world validation on physical TurtleBot3 platforms.

Acknowledgments

We thank the m-explore-ros2 and Nav2 open-source communities for foundational packages. This work used the TurtleBot3 simulation environment provided by ROBOTIS.

References

[1] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Proc. IEEE Int. Symp. Computational Intelligence in Robotics and Automation*, pages 146–151, 1997.

[2] Brian Yamauchi. Frontier-based exploration using multiple robots. In *Proc. Int. Conf. Autonomous Agents*, pages 47–53, 1998.

[3] Wolfram Burgard, Mark Moors, Cyrill Stachniss, and Frank Schneider. Coordinated multi-robot exploration. *IEEE Trans. Robotics*, 21(3):376–386, 2005.

[4] Jan Faigl, Miroslav Kulich, and Libor Přeučil. Comparison of task-allocation algorithms in frontier-based multi-robot exploration. In *European Conf. Mobile Robots*, 2015.

[5] Ruofei Dong, Yanran Wang, and Liang Feng. Multi-robot cooperative autonomous exploration via task allocation in terrestrial environments. *Frontiers in Neurorobotics*, 17:1179033, 2023.

[6] Benjamin Charrow, Sikang Liu, Vijay Kumar, and Nathan Michael. Information-theoretic planning with trajectory optimization for dense 3D mapping. In *Robotics: Science and Systems*, 2015.

[7] Cyrill Stachniss, Giorgio Grisetti, and Wolfram Burgard. Information gain-based exploration using Rao-Blackwellized particle filters. In *Robotics: Science and Systems*, pages 65–72, 2005.

[8] Julio A. Placed, J. Castellanos, and J. Tardos. Approaches for efficiently detecting frontier cells in robotics exploration. *Frontiers in Robotics and AI*, 8:616470, 2021.

[9] Jiří Hörner. Map-merging for multi-robot system. Bachelor's thesis, Charles University, Prague, 2016.

[10] Hyunki Lee, Hyondong Oh, and David Hyunchul Shim. A review on map-merging methods for typical map types in multiple-ground-robot SLAM solutions. *Sensors*, 20(23):6988, 2020.

[11] Steven Macenski, Francisco Martín, Ruffin White, and Jonatan Ginés Clavero. The Marathon 2: A navigation system. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2020.

[12] Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1):83–97, 1955.

Team Member Contributions

Mikul Saravanan: System architecture design, information gain algorithm implementation (C++), coordinator node development (Python), experimental evaluation, metrics collection infrastructure, final report writing.

Samay Lakhani: ROS2 integration, custom message definitions, launch file development, Nav2 configuration, parameter tuning, baseline experiments, results analysis.

Jeffrey Kravitz: Literature review, Hungarian algorithm integration, anti-oscillation mechanisms, statistical analysis, visualization/plotting scripts, documentation.

All team members contributed to debugging, testing, and milestone report preparation.