Multi-Robot Frontier Exploration with Information-Theoretic Task Allocation

Team: Mikul Saravanan (ms6858), Samay Lakhani (svl2111), Jeffrey Kravitz (jk5024)

1. Problem Statement and Objectives

Modern multi-robot exploration frequently wastes time and energy due to redundant coverage and poor coordination under partial observability and bandwidth limits. Standard frontier-based policies, e.g., “nearest-frontier per robot”, lead to assignment collisions and oscillations; empirical overlap rates >30% are standard in cluttered maps, causing excess path length and delayed coverage, and result in overlap rates exceeding 30% in complex environments, wasting computational resources and battery life.

This project develops an improved multi-robot exploration system incorporating information-theoretic metrics into task allocation decisions to achieve efficient coverage with reduced redundancy.

Specific Measurable Objectives:

1. **Reduce redundant exploration by 20%** compared to nearest-frontier baseline using information gain estimation
2. **Achieve 95% environment coverage** in equal or less time through dynamic task reallocation
3. **Maintain 80% map merging success rate** across three test environments with limited initial overlap

2. Technical Approach and Methodology

Core Approach

The system builds on ROS packages (slam_toolbox, m-explore) with three key innovations:

Information-Theoretic Frontier Scoring: Evaluates frontiers based on expected information gain rather than distance alone:

```
None
Score = w1*InformationGain + w2*(1/Distance) +
w3*UniquenessFactor
```

Information gain considers frontier size, expected map expansion, and overlap with other robots' areas.

Enhanced Task Allocation: Modified Hungarian algorithm using information-theoretic cost matrix with allocation history to prevent oscillation and redundant exploration penalties.

Build and solve a Hungarian assignment over active frontiers. Sparsify by keeping top-K frontiers per robot to bound runtime; history-aware tie-breaks retain the previous target unless a new one improves cost by >x. Re-solve at 0.5–1 Hz and on IG change events.

Robust Map Merging: Accept the merge if the confidence (inlier count, final ICP RMSE, and overlap ratio) exceeds the thresholds; otherwise, defer and keep the submaps separate. Provide a fallback to known-initial-pose when available

System Architecture

Distributed architecture with each robot running independent SLAM and frontier detection. Central coordination node handles task allocation and map merging through bandwidth-aware ROS topics.

ROS 2 (Humble), slam_toolbox, nav2, m-explore, Python/C++ nodes for scoring/assignment and merge. Use compressed map deltas for bandwidth (e.g., image_transport/ros2 compressed) and tile updates.

Timeline

Weeks 1-3: Setup Docker/ROS environment, implement baseline single-robot exploration
Weeks 4-6 (Milestone 1): Add multi-robot coordination, demonstrate two-robot exploration
Weeks 7-9: Integrate information gain, confidence-based merging, failure recovery
Weeks 10-11 (Milestone 2): Parameter tuning, multi-environment testing
Weeks 12-13: Final evaluation and documentation

3. Expected Outcomes and Evaluation

Deliverables

1. **ROS Package**: Complete multi-robot exploration implementation with frontier detection, task allocation, and map merging modules compatible with TurtleBot3 simulations
2. **Evaluation Results**: Performance metrics and trajectories from experiments across three test environments
3. **Documentation**: Setup instructions and reproducibility guidelines

Evaluation Metrics

Exploration Performance: Coverage percentage over time, total path length, redundant exploration percentage, time to reach 50%, 90%, 95% coverage

Map Quality: Absolute Trajectory Error (ATE), map consistency vs. ground truth, successful merge percentage

System Performance: Message exchange rate, CPU/memory usage, scalability (2-4 robots)

Baselines: (B1) per-robot nearest frontier; (B2) global greedy nearest. Our method: IG-Hungarian (with/without history and penalties for ablation). Trials: 10 seeds/env/config; report mean±std. Ablations: −Uniq, −RedunPenalty, −OscPenalty, −IG (distance-only).

Testing will use three Gazebo environments: office (cluttered rooms), warehouse (open with obstacles), and maze-like environment, each presenting different coordination challenges.

Risk Mitigation

Primary Risk: Map merging failure with insufficient overlap
  Mitigation: Fallback to known initial poses mode

Secondary Risk: Docker performance on Mac
  Backup: Columbia Linux clusters for intensive testing

Bandwidth spikes. *Mitigation:* map tiling + periodic deltas; cap allocator frequency; topic compression.

The modular design allows graceful feature reduction if scope exceeds timeline while maintaining core functionality.

4. Innovation and Course Relevance

This project applies course concepts including motion planning (frontier navigation), SLAM (mapping), multi-robot coordination, and optimization (Hungarian algorithm, ICP). ThThe novelty lies in a unified, information-theoretic frontier scoring with history-aware global assignment that explicitly penalizes redundancy and oscillation under bandwidth constraints, yielding practical gains (≥20% redundancy reduction) in realistic settings. While incremental, the expected 20% efficiency improvement has meaningful implications for applications like search-and-rescue and warehouse automation.

References

[1] Yamauchi, B. "Frontier-based Exploration Using Multiple Robots." International Conference on Autonomous Agents, 1998.

[2] Horner, J. "Map-merging for Multi-robot System." Charles University Prague, 2016.

[3] Burgard, W., et al. "Coordinated Multi-Robot Exploration." IEEE Trans. Robotics, 2005.

[4] Valencia, R. and Andrade-Cetto, J. "Mapping, Planning and Exploration with Pose SLAM." Springer, 2018.