PeraSwarm: Simultaneous Localization and Mapping in Mixed Reality Environment

Group 19

Supervisors:
Prof. Roshan Ragel
Dr. Isuru Nawinne
Mr. Nuwan Jaliyagoda

Team Members:
E/18/077 Nipun Dharmarathne
E/18/150 Yojith Jayarathna
E/18/227 Dinuka Mudalige

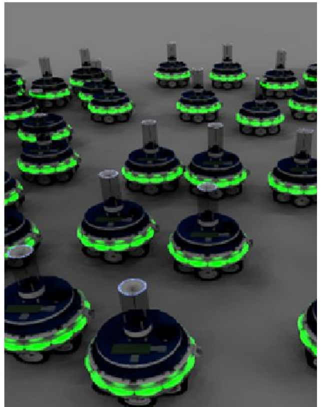
1. Introduction

a. Swarm Robotics
b. Mapping and Localization with Swarm Robots

Swarm Robotics

- Multiple robots collaboratively work together to accomplish tasks in a,
  - Distributed
  - Decentralized manner

- Inspired by the collective behaviour observed in natural swarms such as,
  - Ants
  - Bees

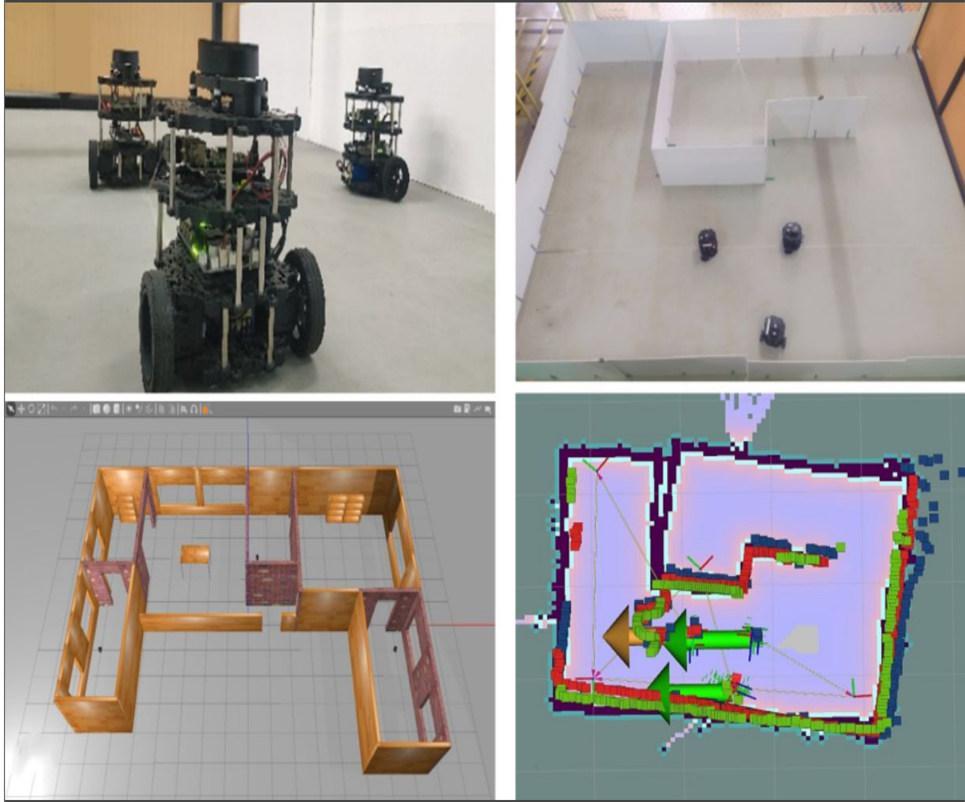


- Features:
  - Decentralized control
  - Self-organizing behavior
  - Robustness
  - Scalability
  - Efficiency, adaptability, and collaboration

- Applications:
  - Exploration
  - Surveillance
  - Search and rescue

Mapping and Localization with Swarm Robots

- Involves a team of robots working together to explore and map an unknown environment
- Advantages:
  - Increased efficiency
  - Rapid coverage
  - Distributed exploration



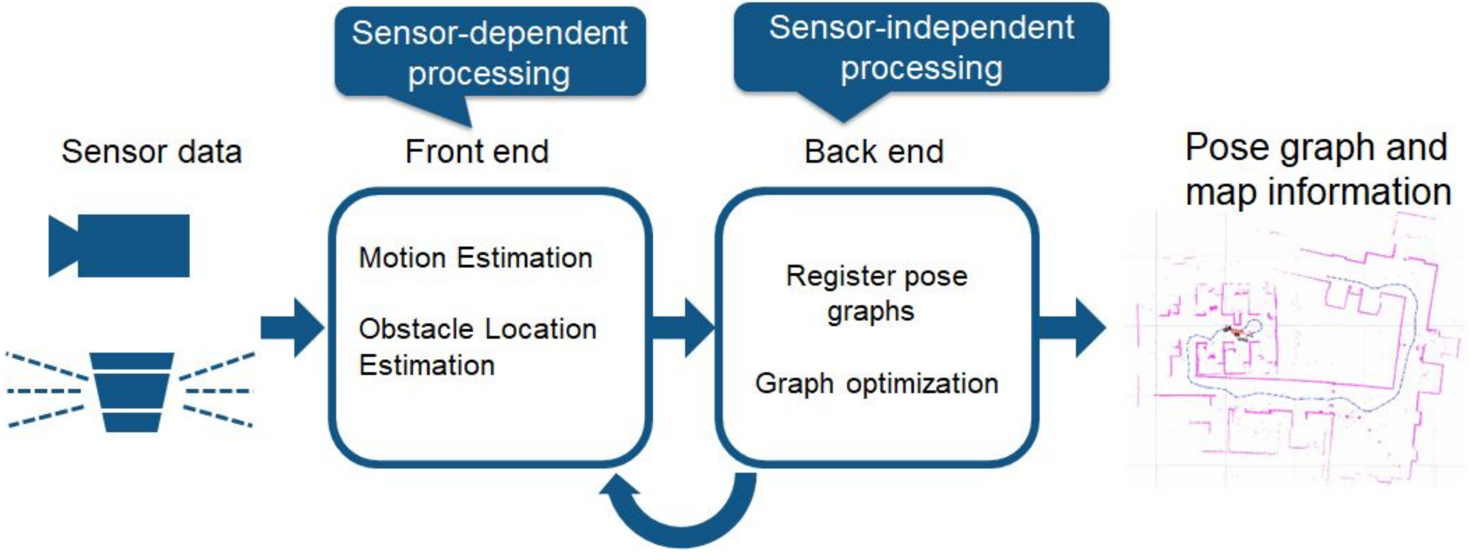
2. Methodology, Analysis & Conclusions

a. Literature Review
b. Research Gaps
c. Proposed Solution

Literature Review

Simultaneous Localization and Mapping (SLAM)

SLAM is a computational technique used by robots and autonomous systems to construct maps of unknown environments while simultaneously tracking their own position

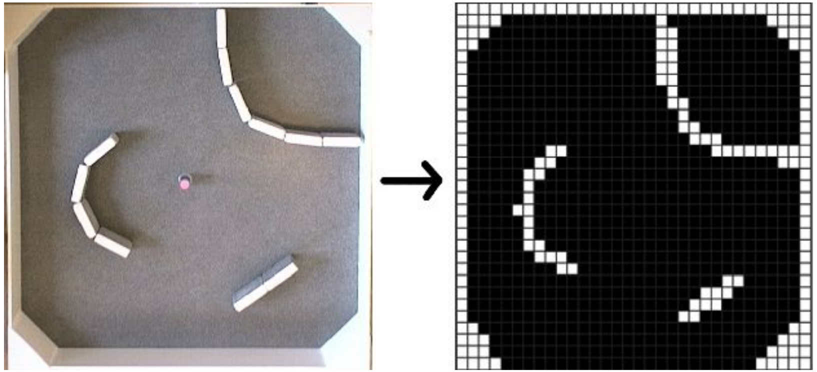


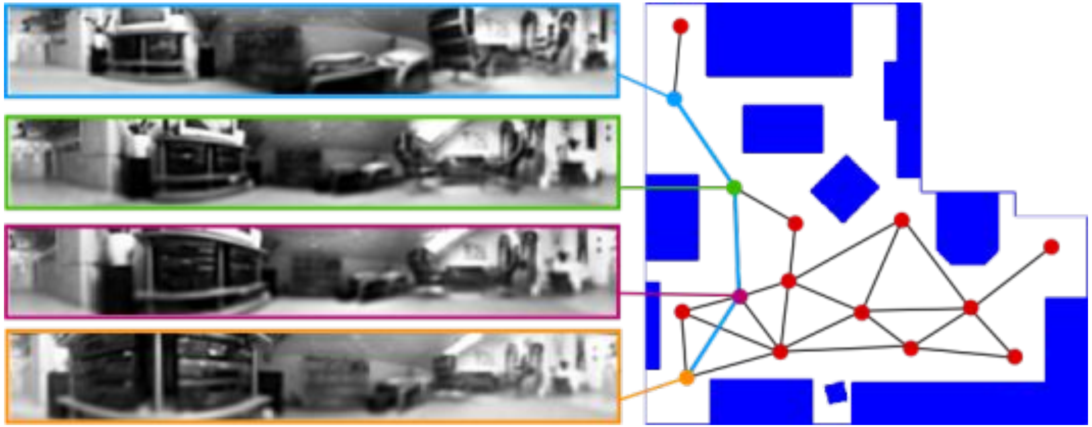
https://www.mathworks.com/discovery/slam.html

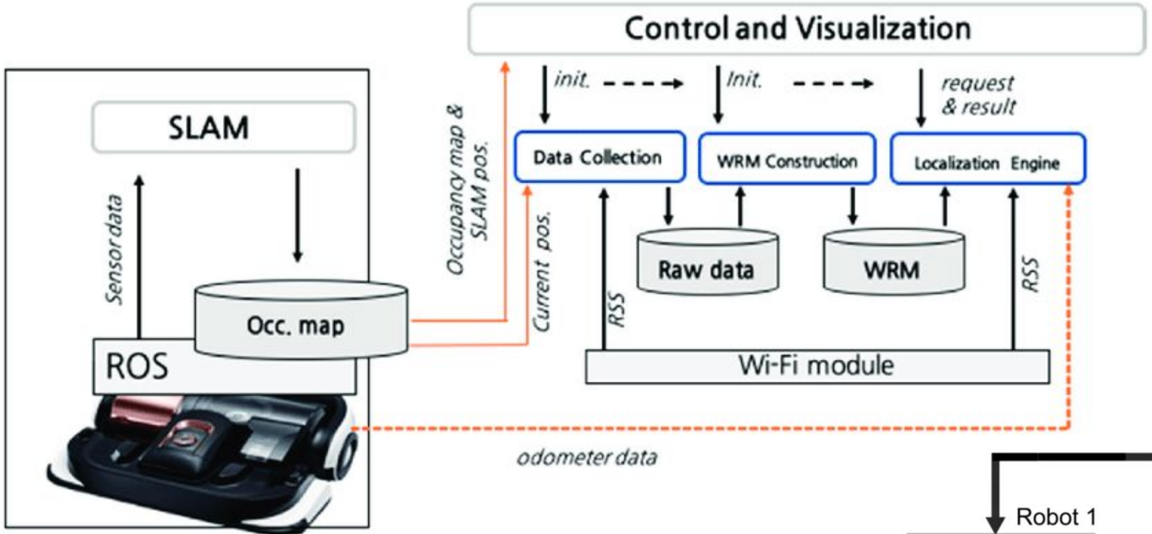
Swarm SLAM: Challenges and Perspectives:

- Challenges:
  - How swarm explore the environment?
    - Communication? Sensing?
  - How robots share the gathered information?
  - How these information used to produce an overall map?

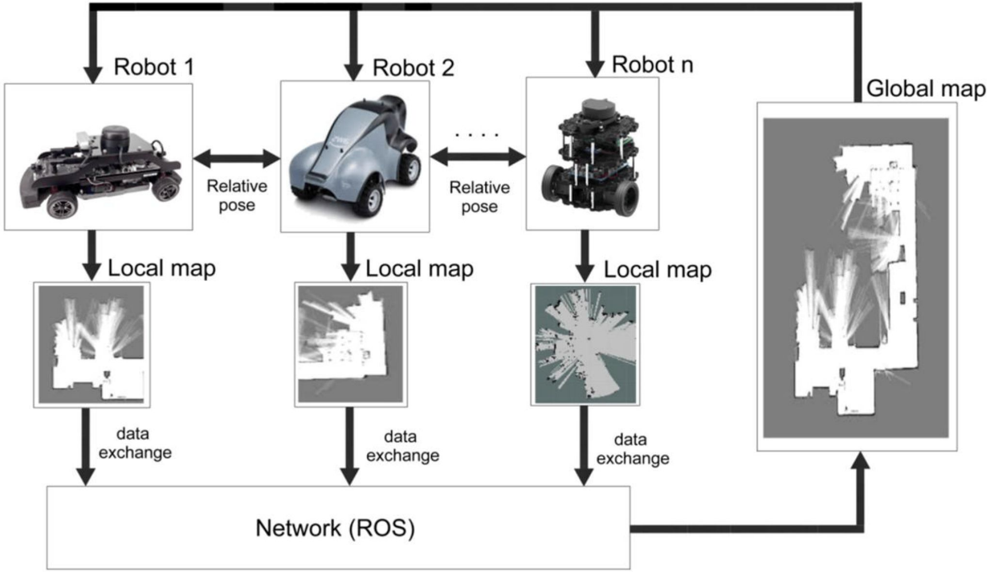
- Current Approaches
  - Centralized vs Decentralized Methods
  - Wireless Communication
  - Data compression and processing
    - LIDARs, Cameras, Sonars
  - Consensus algorithms (Odometry)
  - Map generation
    - Occupancy grid mapping
      - Divides into cells
    - Topological mapping
      - Nodes - Landmarks
      - Edges - Paths







Structure of the robot localization application in SLAM system (researchgate.net)



Feature-Based Occupancy Map-Merging for Collaborative SLAM (mdpi.com)

Algorithmic Aspects…

SwarMap: Occupancy Grid Mapping with a Robotic Swarm:

- The swarm is divided into two teams:
  - Landmark robots
  - Mapper robots
- A method is used that employs Bayesian filters
- These filters update a shared occupancy grid stored in the cloud
- Robots localize themselves based on odometry
- Robots use pair-to-pair neighborhood communication to exchange pose information
- Uncertainty is reduced through a Kalman filter

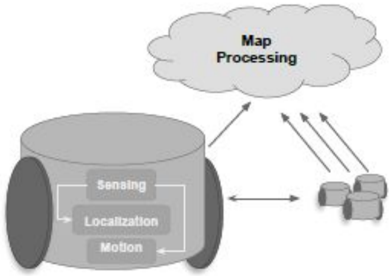
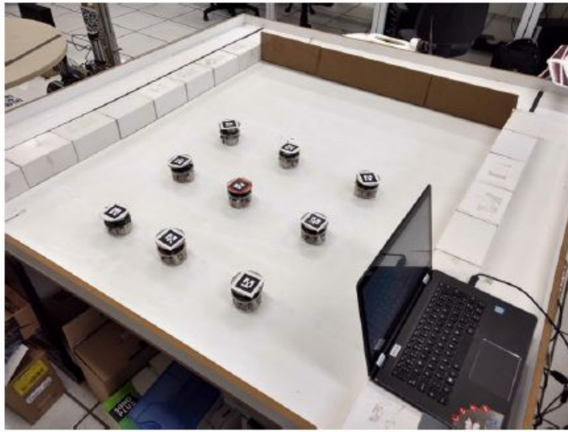


Fig. An overview of SwarMap: each robot computes its own pose estimation, exchanges this information with neighbors to improve its localization and cooperatively builds an occupancy grid map in the cloud.

Fig. Example of a real experiment setup using nine e-pucks in two scenarios. The leader robot $\mathcal{R}_l$ is identified by the red marker.



Cooperative Localization and Mapping with Robotic Swarms **[An extension of previous paper]**

- Explain more on some motion strategy used to guide the swarm
  - Covariance Intersection algorithm for decentralized localization

- Also discuss some steps performed by the robots to produce updated estimates
  - Absolute and Relative Pose Measurement Models

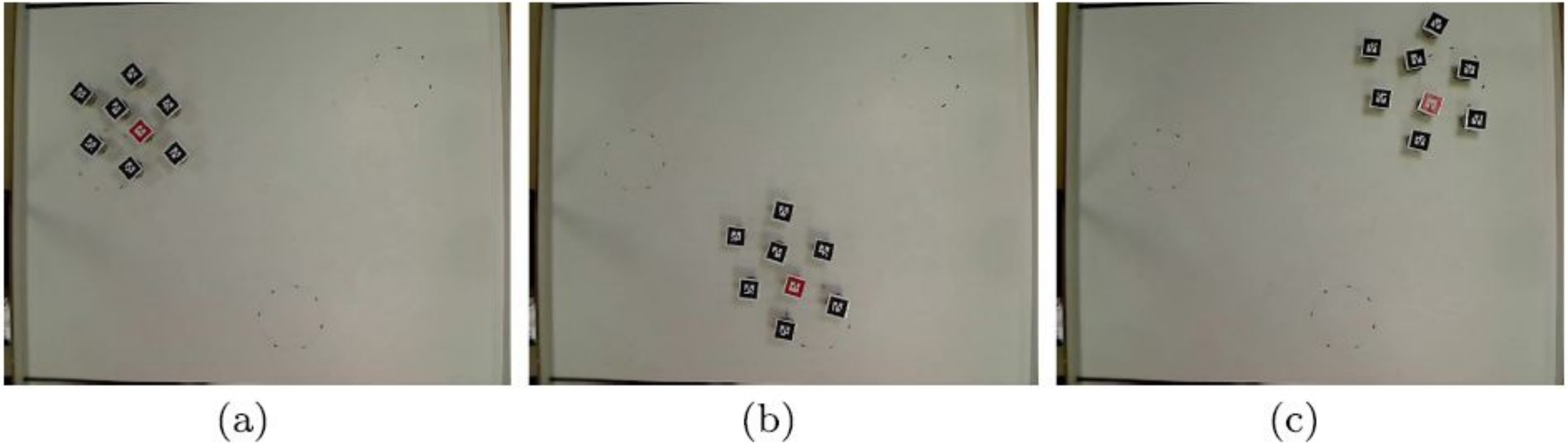
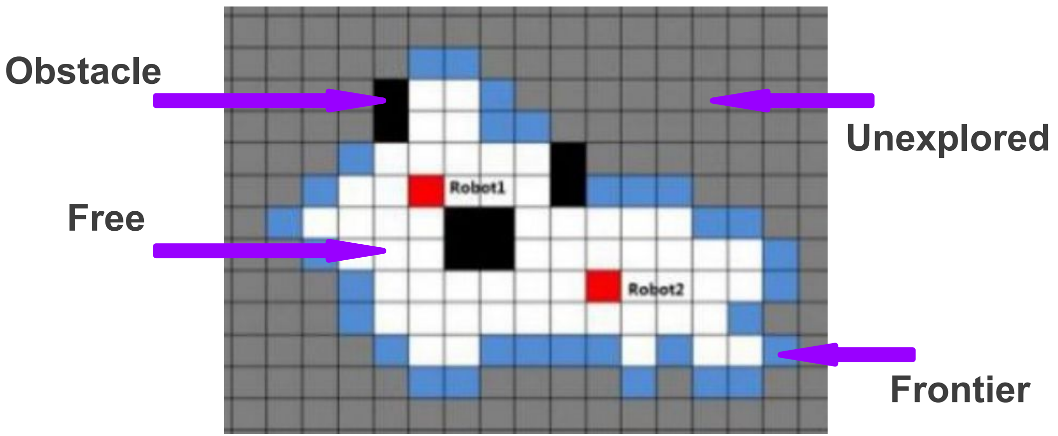
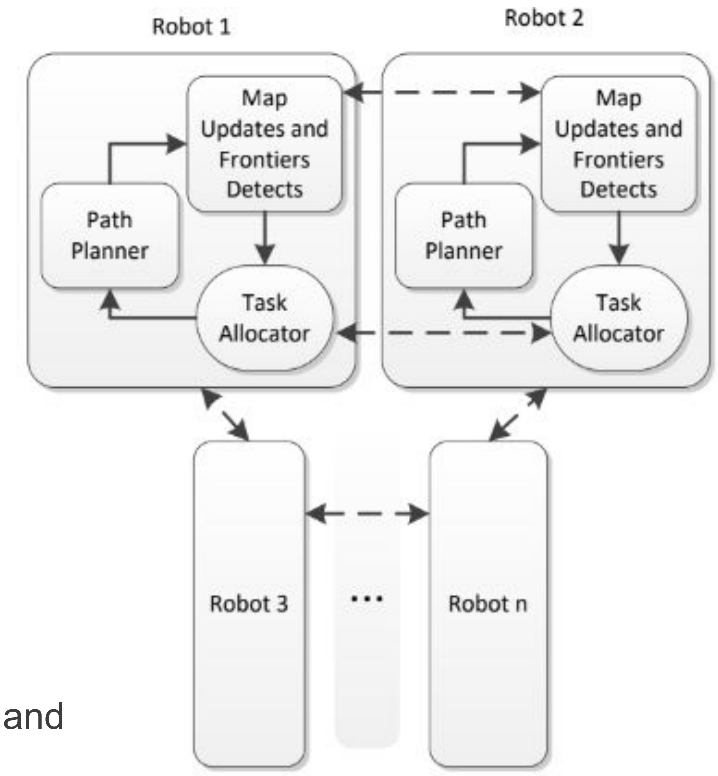


Fig. Robots' motion based upon local information. The leader guides the group by using waypoints, which are depicted as dashed circles

An Unknown Environment Exploration Strategy for Swarm Robotics Based on Brain Storm Optimization Algorithm:

- Fully decentralized approach
- Frontier-based exploration
- An occupancy grid map is used with 4 states to represent,





- At each time step each robot will broadcast its own position and local map and integrate the information received from other robots to its local map

Collaborative mapping and navigation for a mobile robot swarm

- Each robot in the swarm was equipped with:
    - A limited field of view
    - A limited-range finder
    - A magnetometer to infer its orientation
- The robots were assigned fixed stationary targets
- They initially explored towards their target areas
- After exploration, they were guided towards their targets
- The robots exchanged their maps collaboratively during the exploration

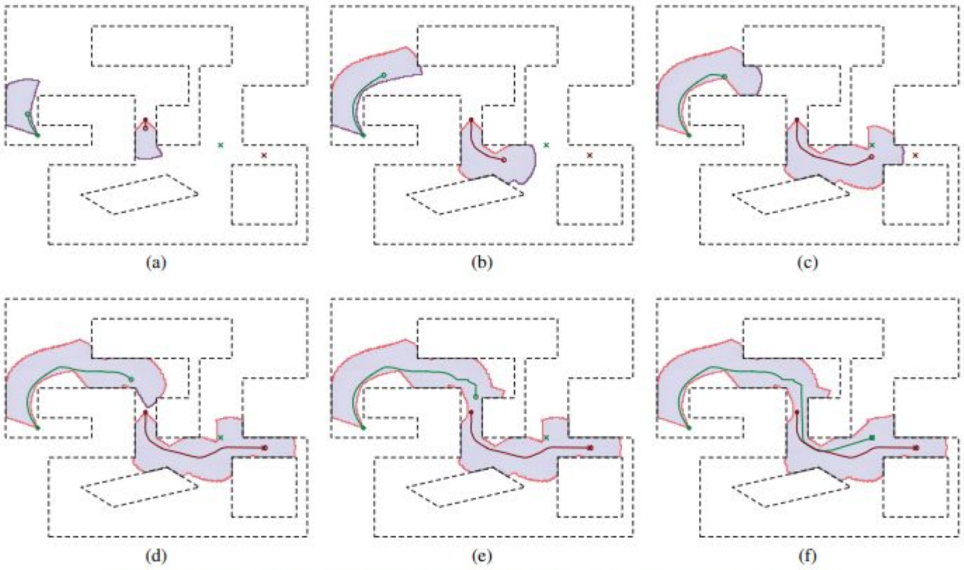
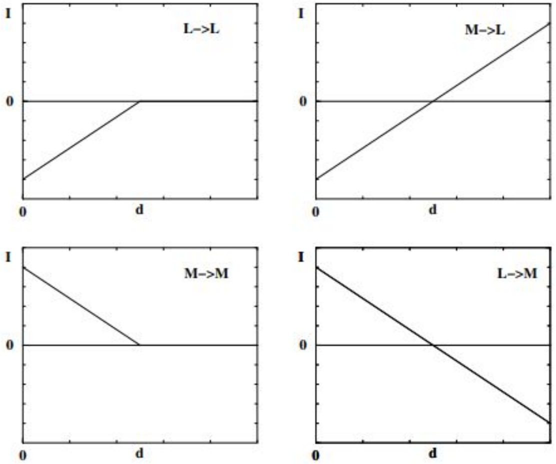


Fig. : Evolution of the robot navigation towards the target location with respect to the actual area

[Distributed Localization and Mapping with a Robotic Swarm](#):

- Research paper focuses on distributed localization and mapping with a robotic swarm
- Addresses fundamental questions in swarm robotics:
    - When and how to use swarms,
    - measuring and improving performance and
    - dealing with environmental complexity and sensor limitations
- Proposes three algorithms:
    - Collaborative localization
    - Dynamic task allocation
    - Collaborative mapping



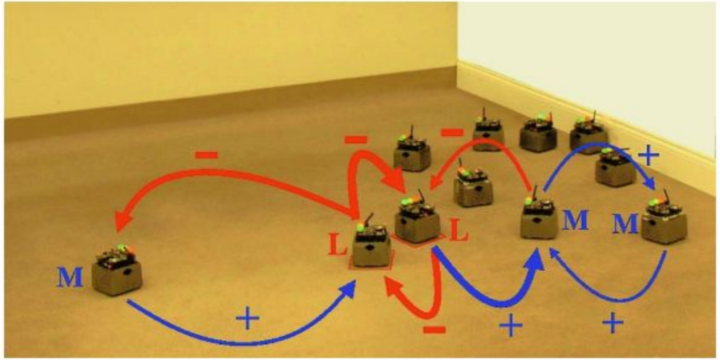


Fig. 2. Robot desire-to-move interactions shown on an actual swarm.

- Algorithms enable robots to estimate positions, decide movements, and map environments.

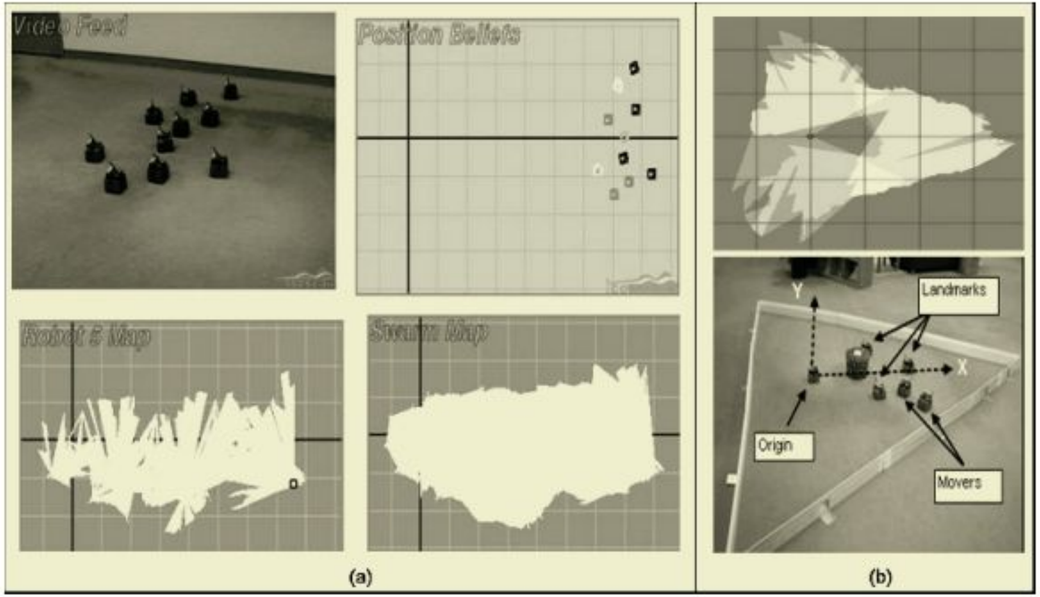
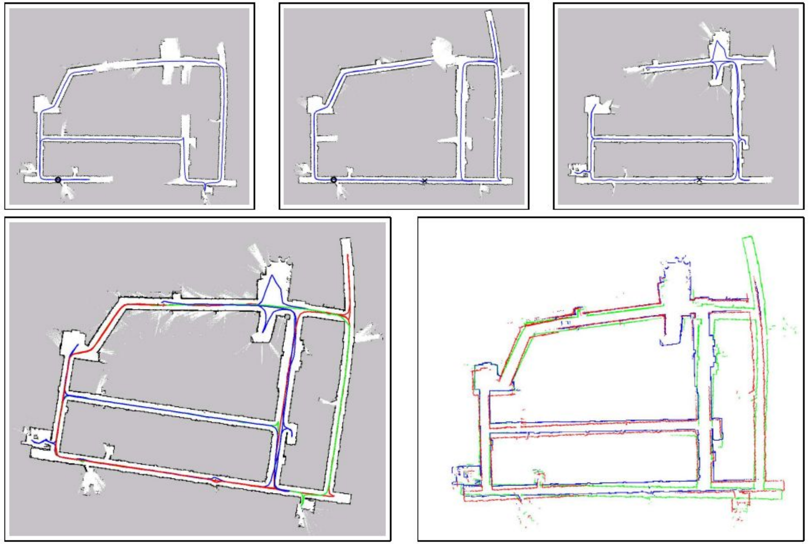


Fig. 3. (a) Localization and mapping along a straight hallway. (b) Mapping triangle shaped area with object in center.

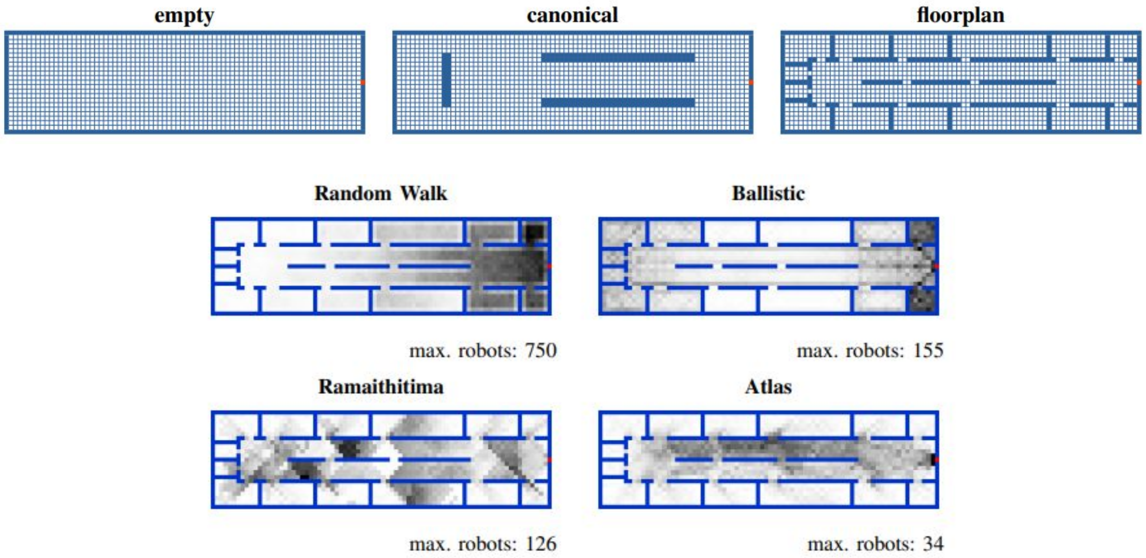
Distributed Multi-robot Exploration and Mapping:

- Addresses distributed multi-robot exploration and mapping challenges.
- Proposes decision-theoretic coordination for exploration and relative location verification.
- Introduces efficient algorithm using particle filters for relative location estimation.
- Utilizes predictive indoor environment modeling for enhanced accuracy.
- Implements constraint-based map merging for global map consistency.
- Demonstrates system effectiveness in real-world conditions and advantages over existing approaches.



Atlas: Exploration and Mapping with a Sparse Swarm of Networked IoT Robots:

- Paper introduces Atlas, a mapping algorithm for sparse robot swarms.
- Atlas ensures full exploration even with a single robot.
- Open-source simulator compares Atlas with three algorithms:
    - Ramaithitima
    - Random walk
    - Ballistic random walk



- Atlas excels in exploration speed, map completeness, and quality across different scenarios.
- Advantages of Atlas:
    - Systematic exploration
    - Robustness to network failures
    - Scalability
- Atlas emerges as a promising solution for diverse exploration and mapping challenges.

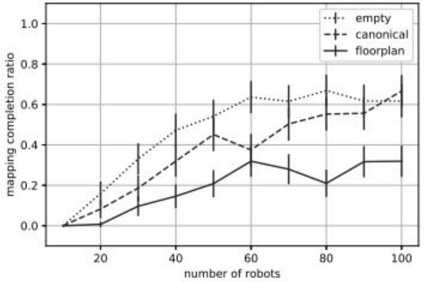


Fig. 3: Completion ratio of Ramaithitima: the portion of runs where the robots map out the entire area.

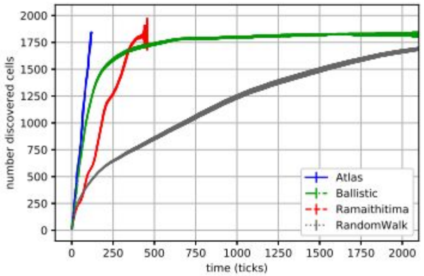


Fig. 4: Mapping profiles of the algorithms for the Floorplan scenario: the number of cells discovered over time.

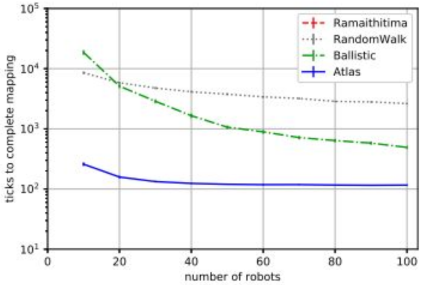


Fig. 5: Mapping speed for the floorplan scenario: time until the area is fully explored and mapped.

Research Gaps

- Complex sensors used in current systems

- So far, mapping with robots are mainly focused on single and centralized multi-robot communication systems.

- So far experiments have done using,
    - Simulated environment
    - Physical environment with less number of robots

- SLAM algorithms are often computationally demanding and as the number of robots increases, scalability becomes a significant challenge.

Proposed Solution

- Cost effective robots with simple sensors
- Decentralized system
- By using mixed reality system,
    - Can do experiments with both simulated and real environments
    - Can do experiments using many robots

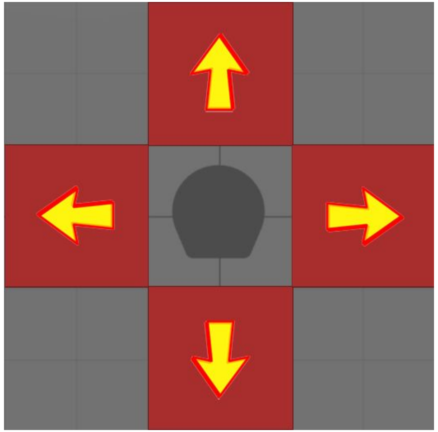
3. Current Progress

a. Our Approach
b. Tasks Carried Out with Demonstration

Our Approach

- Virtual robots in our simulated environment
- Occupancy grid representation
- 2D integer array was used to keep track of the map
  - 0 - Unexplored area
  - 1 - Free area
  - 2 - Obstacles
  - 3 - Robot’s position
- Random movements were used in robots
- Assumed robots can only move one step at a time to
  - North
  - South
  - East
  - West
- Tested mapping algorithms using our simulated environment

1	1	1	1	1	3	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	2	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	2	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	2	1	1	1	1	1	2	0
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1



Task Carried Out

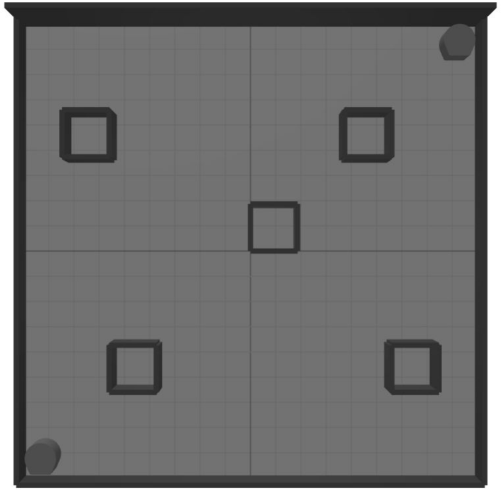
Mapping and Localization

With known,
- Initial positions
- Heading directions

With unknown,
- Initial positions
- Heading directions

Mapping with Known Initial Positions and Heading Directions

- Handled intercommunication between robots
  - Simple communication
  - For each step a robot broadcast its position and local map with others
- Implemented real time map merging in robots



Arena used for testing

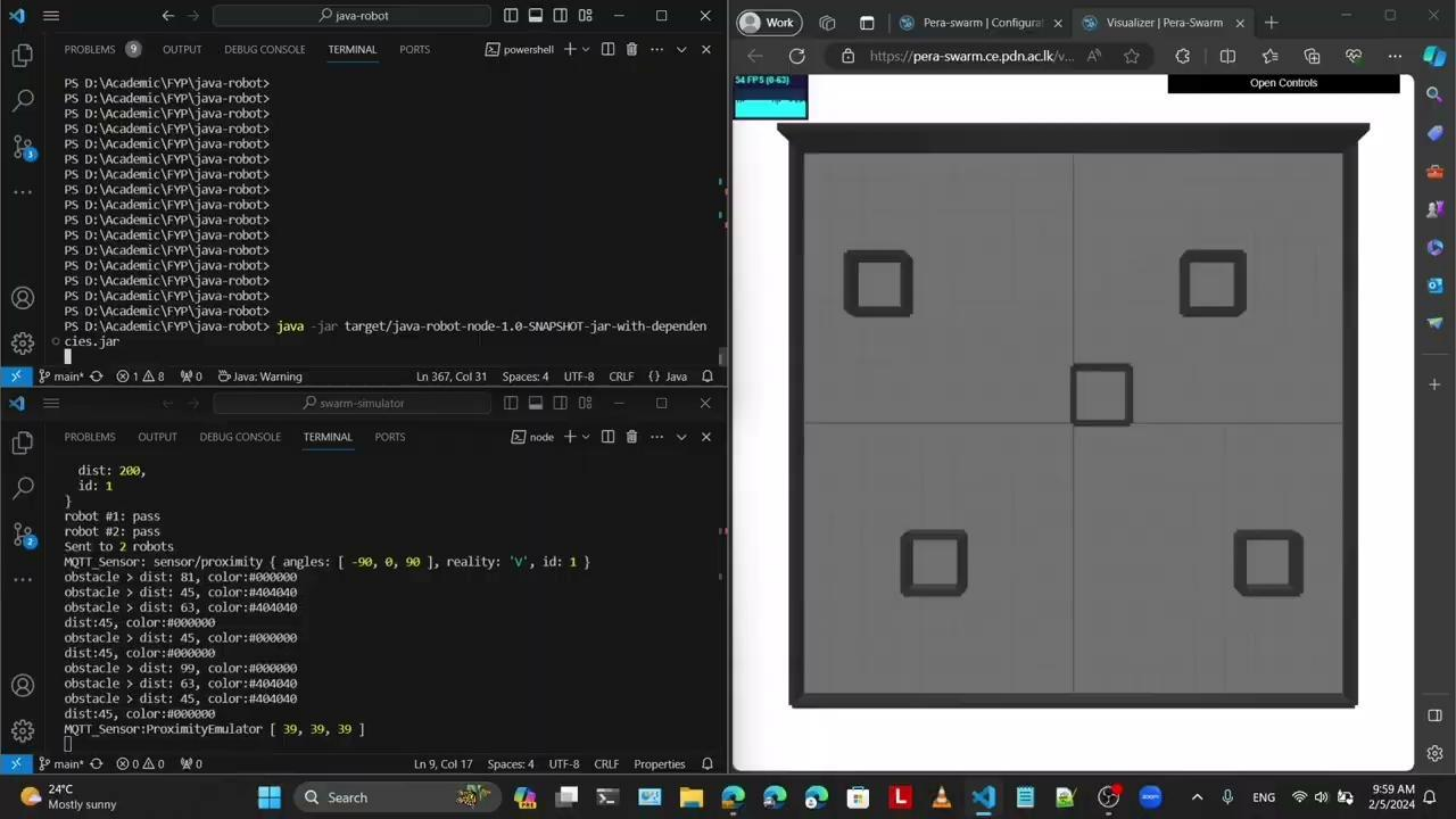
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	2	1	1	1	1	1	2	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	2	1	3	1	1
3	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	2	1	1	1	1	1	2	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

Map generated after few minutes

0 - Unexplored area   1 - Free area
2 - Obstacles   3 - Robot's position

Demonstration Video

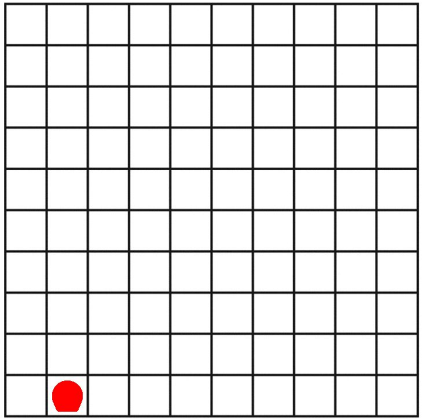
Mapping with Known Initial Positions and Heading Directions


java-robot
PROBLEMS 9
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
PS D:\Academic\FYP\java-robot>
PS D:\Academic\FYP\java-robot> java -jar target/java-robot-node-1.0-SNAPSHOT-jar-with-dependencies.jar
main*
Java: Warning
Ln 367, Col 31 Spaces: 4 UTF-8 CRLF {} Java
swarm-simulator
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
node
dist: 200,
id: 1
}
robot #1: pass
robot #2: pass
Sent to 2 robots
MQTT_Sensor: sensor/proximity { angles: [-90, 0, 90], reality: 'V', id: 1 }
obstacle > dist: 81, color:#000000
obstacle > dist: 45, color:#404040
obstacle > dist: 63, color:#404040
dist:45, color:#000000
obstacle > dist: 45, color:#000000
dist:45, color:#000000
obstacle > dist: 99, color:#000000
obstacle > dist: 63, color:#404040
obstacle > dist: 45, color:#404040
dist:45, color:#000000
MQTT_Sensor:ProximityEmulator [39, 39, 39]
main*
Ln 9, Col 17 Spaces: 4 UTF-8 CRLF Properties
Work
Pera-swarm | Configurat
Visualizer | Pera-Swarm
https://pera-swarm.ce.pdn.ac.lk/v...
Open Controls
24°C
Mostly sunny
Search
ENG
9:59 AM
2/5/2024

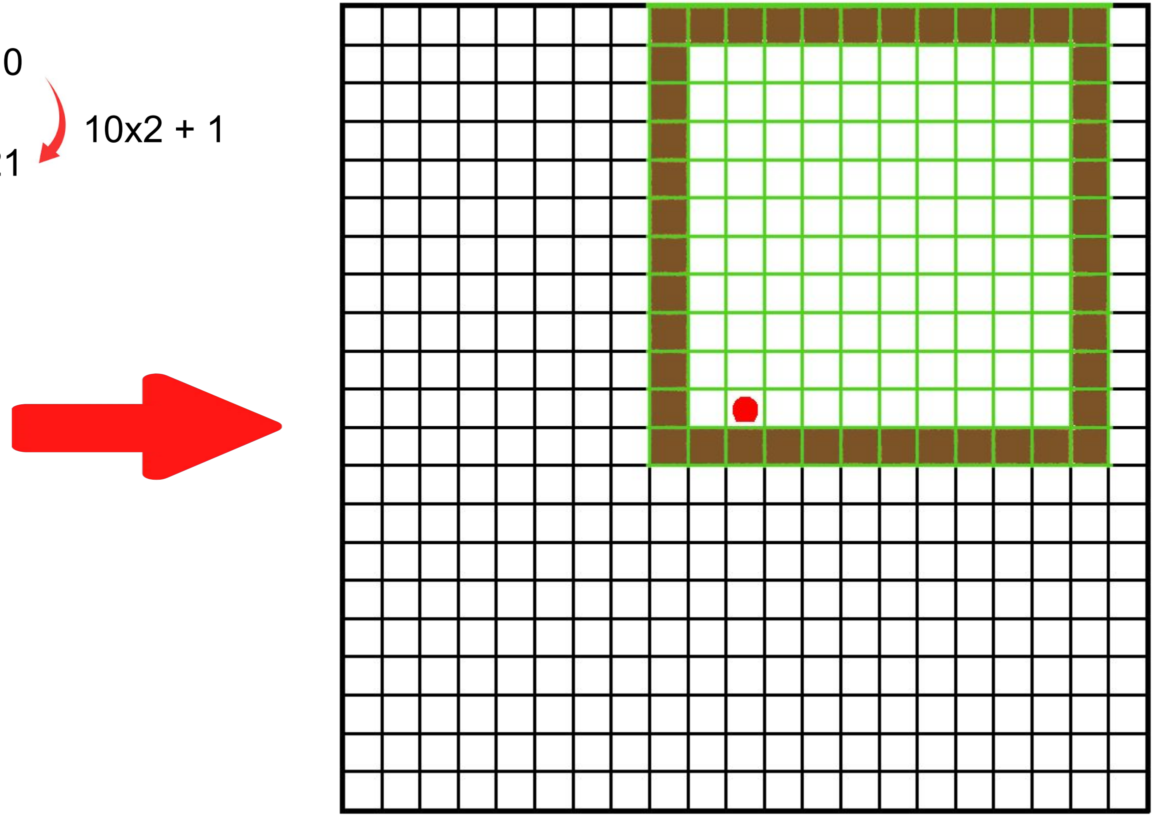
Mapping with Unknown Initial Positions and Heading Directions

- Arena used : 10 x 10
- 2D array used : 21 x 21

10x2 + 1



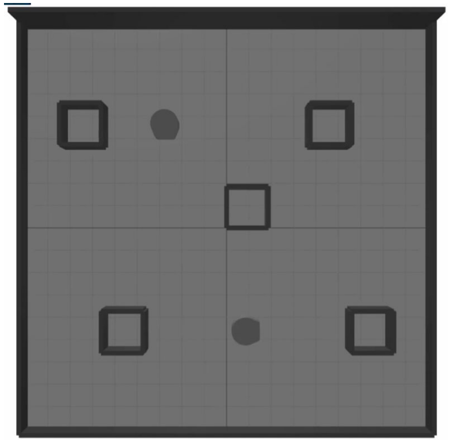
Arena



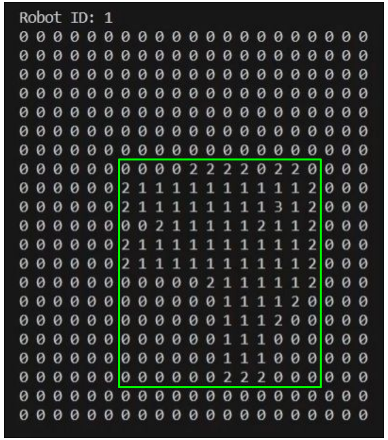
2D array

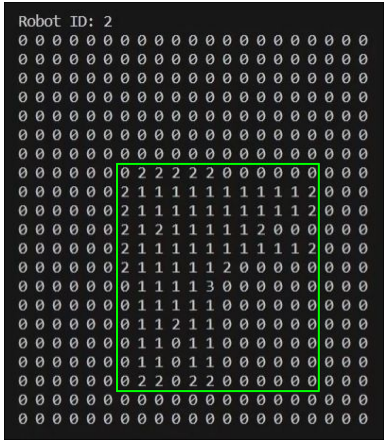
Mapping with Unknown Initial Positions and Heading Directions

Robot	Random Initial Position	Random Heading Direction
Robot 1	(2, -3)	South
Robot 2	(-2, 3)	North



Arena used for testing

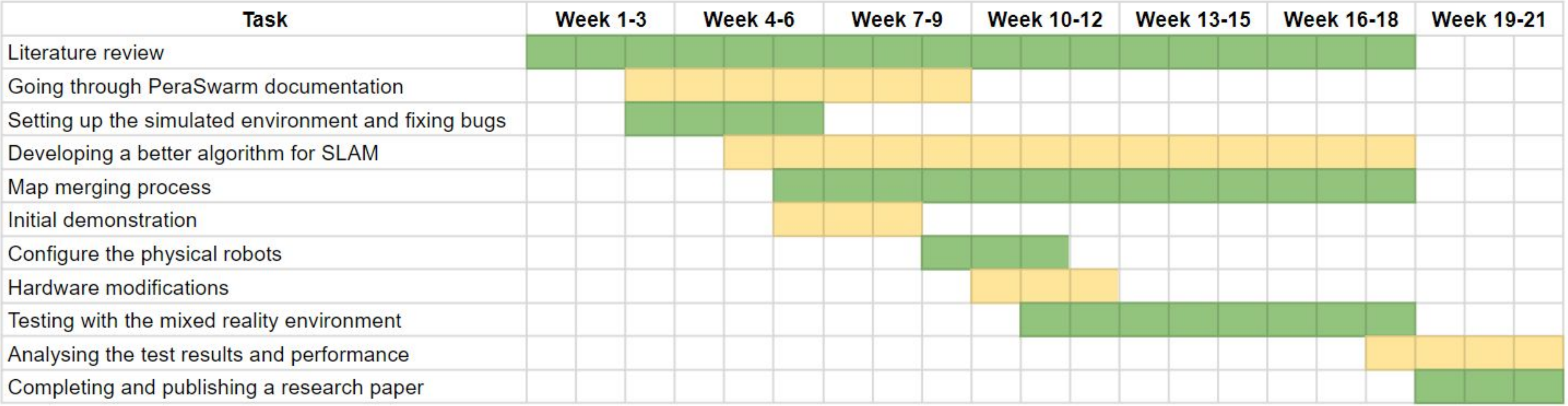




Maps generated after few minutes

0 - Unexplored area 1 - Free area
2 - Obstacles 3 - Robot's position

4. Work Plan


Task
Week 1-3
Week 4-6
Week 7-9
Week 10-12
Week 13-15
Week 16-18
Week 19-21
Literature review
Going through PeraSwarm documentation
Setting up the simulated environment and fixing bugs
Developing a better algorithm for SLAM
Map merging process
Initial demonstration
Configure the physical robots
Hardware modifications
Testing with the mixed reality environment
Analysing the test results and performance
Completing and publishing a research paper

5. Use of AI

consensus

ChatGPT

Copilot

Your everyday AI companion

SCISPACE

Gemini

Heuristica

AI-powered knowledge exploration

grammarly

Thank You!

Q&A