MAES, a Realistic Simulator for Multi Agent Exploration and Coverage

Malte Z. Andreasen, Philip I. Holler, Magnus K. Jensen, Michele Albano

Department of Computer Science, Aalborg University
Selma Lagerløfs Vej 300, 9220 Aalborg, Denmark
malte@mza.dk, philipholler94@gmail.com, magnjensen@gmail.com, mialb@cs.aau.dk

Abstract

This demo concerns the open source physics-driven simulation tool Multi Agent Exploration Simulator (MAES), which can be used for developing and evaluating exploration and coverage algorithms in an unknown continuous 2D environment. MAES supports development efforts for new algorithms by means of debugging tools for developed algorithms. It provides a fully user-controlled camera over the environment under exploration, which can be attached to a single robot to follow its operations. MAES represents graphically the SLAM process and the areas explored by one or all robots, to visualize exploration, coverage, heatmap, slam etc, both for a specific robot or for the entire swarm. Finally, MAES provides both a simple interface for developing algorithms in C# as well as a Robot Operating System 2 (ROS2) interface, the latter allowing to integrate with existing robot controllers. MAES aims to bridge the gap between unrealistic, simple simulations, usually executed on grid environments, and heavy, time consuming, but realistic simulations, such as ARGoS or Gazebo. The accompanying video is available at https://youtu.be/0RzPN0oW7v8. MAES' source code can be found at https://github.com/MalteZA/MAES.

Introduction

Many algorithms target the problem of collective terrain exploration and coverage using swarm robotics, since it has both theoretical interest, and real life applications (Schranz et al. 2020). In most models, each agent has a field of view (FOV), which is a circular area that the agent is able to get measurements from. **Exploration** describes the amount of accumulated open area an agent's FOV has covered during the process of moving through an environment. Exploration is important when the agent aims to generate a navigable map of the environment, or in search & rescue scenarios where it is vital to quickly explore large areas. An agent's physical body has an area that covers some movement space, and **Coverage** describes the amount of accumulated area that an agent's body-area has physically covered. Coverage scenarios comprise robotic lawn mowers and cleaning robots such as autonomous vacuum cleaners, where the robots need to efficiently and precisely cover areas.

This work focuses on Online Exploration and Coverage, i.e. the terrain is unknown before the algorithms is executed.

Even though many great algorithms have been proposed for these problems, it is hard to compare them since they model their agents in very different manners, and most of them operate in unrealistic environments where the agents (i) maneuver by moving from cell to cell in a grid-like structure; (ii) communication between robots has unlimited range and is not blocked by walls; (iii) robots know always each other's location, and are provided with real-time distributed Simultaneous Localization And Mapping (SLAM). On the opposite hand, other simulators such as the multi-physics robot simulator Autonomous Robots Go Swarming (ARGoS) (Pitonakova et al. 2018) provide realistic simulation environments for robots. However, they are complex to program, and have bad computational performance since they simulate each sensor or actuator present on the robot.

This demo presents the tool Multi Agent Exploration Simulator (MAES), which provides a trade-off between realism and simplicity. MAES represents the movement space as a continuous 2D plane, and maps the different approaches on a common ground in terms of hardware capabilities of the robots. A preliminary version of MAES, used to showcase a general simulation framework, was presented in (Andreasen. et al. 2022). This demo focuses on the MAES tool itself, and it presents its capabilities and its recent developments.

The MAES Tool

MAES is a deterministic 2D discrete time-step physics-based simulation, visualized in 3D. Timing of communication between ROS nodes is not deterministic, and MAES' behavior also is not deterministic when used with the ROS interface. The simulator uses the Unity Engine (Unity Technologies 2021), for visualization and physics simulation. Physics are simulated at a rate of 100 ticks per simulated second. MAES contains a map generator than can automatically generate two types of maps: building maps featuring hallways and rooms, and cave maps full of irregular shapes, or the map generator can generate maps from a .pgm file.

An agent is able to rotate in place and moving straight ahead. Movement is simulated through the Unity 2D Physics Engine by applying force at the position of each wheel. The simulation accounts for inertia, drag, and collisions with obstacles and other agents. The agent is able to sense other agents and detect collisions with walls and other agents. The agent senses its surroundings by means of simulated LIDAR

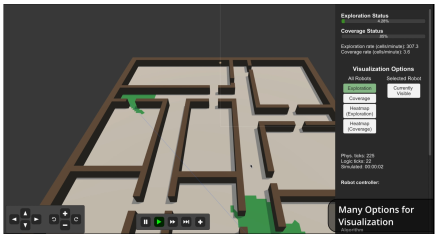

Figure 1: Camera showing the whole area.

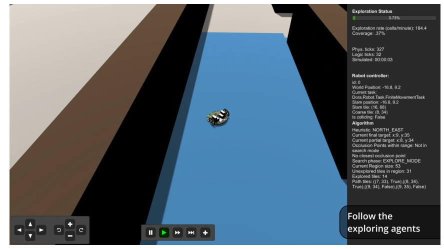

Figure 2: Camera following a specified agent.

scanner, implemented by means of ray tracing. Agents can communicate through broadcasting, where communication range and the capabilities to pass walls is defined via simulation parameters. Finally, agents can deposit information in environment tags and communicate indirectly with other agents. If enabled, the agent can provide an environment map generated via SLAM to the algorithm being simulated.

MAES provides two agent control interfaces that allow implementation of exploration algorithms. These interfaces provide access to movement controls, communication, object detection and SLAM. The first interface is a simple C# interface, that can be extended and the algorithm thus injected, while the second is a ROS2 (Erős et al. 2019) interface. The latter is aimed at integrating MAES with existing robot controllers, which usually use ROS2 interfaces to interact with real-world robots. Morever, MAES supports the definition of waypoints for robot's movements using a Nav2 interface and visualization of the robot's LIDAR via a Rviz2 (Kam et al. 2015) interface, both being mainstream technologies in the domain.

As MAES should function as a testbed for algorithms comparison, it includes a wide variety of debugging tools. A camera is placed over the area being explored (see Figure 1). Additionally, agents can be individually selected, which makes the camera follow the agent as well as reveal debugging information in a side bar regarding the selected agent (see Figure 2). Below the controller information, a section for algorithm specific debugging information is included. Furthermore, slam maps, communication, and environment tagging can be visualized. When using the ROS interface, a ROS menu is included where all topics and the data being sent through the topics can be visualized.

To showcase MAES as a tool to compare online exploration algorithms, our video features the concurrent simulation of three state-of-the-art algorithms, namely Spiraling and Selective Backtracking (Gautam et al. 2018), Local Voronoi Decomposition (Fu, Bandyopadhyay, and Ang 2009) and The Next Frontier (Colares and Chaimowicz 2016). The algorithms had to be adapted to be executed in MAES, since they had different requirements and modeled the environment in different manners. These adaptations are discussed in (Andreasen. et al. 2022).

The results of the simulations hint that Spiraling and Selective Backtracking outperforms the other algorithms in both building maps and cave maps. This performance comes at the cost of reduced realism, since this particular algorithm assumes perfect global communication and distributed SLAM capabilities.

Acknowledgments

This work was partly funded by the Villum Investigator Project "S4OS: Scalable analysis and Synthesis of Safe, Small, Secure and Optimal Strategies for Cyber-Physical Systems".

We thank Simon Bøgh and Casper Schou for their support when developing the ROS2 interfaces for MAES.

References

Andreasen., M.; Holler., P.; Jensen., M.; and Albano., M. 2022. Comparison of Online Exploration and Coverage Algorithms in Continuous Space. In *Proc. of the 14th International Conference on Agents and Artificial Intelligence - Volume 1: SDMIS,*, 527–537. INSTICC, SciTePress.

Colares, R. G.; and Chaimowicz, L. 2016. The next Frontier: Combining Information Gain and Distance Cost for Decentralized Multi-Robot Exploration. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, SAC '16, 268–274. New York, NY, USA: ACM.

Erős, E.; Dahl, M.; Bengtsson, K.; Hanna, A.; and Falkman, P. 2019. A ROS2 based communication architecture for control in collaborative and intelligent automation systems. *Procedia Manufacturing*, 38: 349–357.

Fu, J. G. M.; Bandyopadhyay, T.; and Ang, M. H. 2009. Local Voronoi Decomposition for multi-agent task allocation. In *2009 IEEE International Conference on Robotics and Automation*, 1935–1940.

Gautam, A.; Richhariya, A.; Shekhawat, V. S.; and Mohan, S. 2018. Experimental Evaluation of Multi-Robot Online Terrain Coverage Approach. In *2018 IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, 1183–1189.

Kam, H. R.; Lee, S.-H.; Park, T.; and Kim, C.-H. 2015. Rviz: a toolkit for real domain data visualization. *Telecommunication Systems*, 60(2): 337–345.

Pitonakova, L.; Giuliani, M.; Pipe, A.; and Winfield, A. 2018. Feature and performance comparison of the V-REP, Gazebo and ARGoS robot simulators. In *Annual Conference Towards Autonomous Robotic Systems*, 357–368. Springer.

Schranz, M.; Umlauft, M.; Sende, M.; and Elmenreich, W. 2020. Swarm robotic behaviors and current applications. *Frontiers in Robotics and AI*, 7: 36.

Unity Technologies. 2021. Unity Real-Time Development Platform | 3D, 2D VR & AR Engine. https://unity.com/.