**Occupancy Map Abstraction for Higher Level Mission Planning of Autonomous Robotic Exploration in Hazardous Nuclear Environments**

David Batty1, Lupo Manes1, Andrew West2, Maulik Patel1, Ipek Caliskanelli3, Paolo Paoletti1

1 @LERT, University of Liverpool, , Brownlow Hill L69 3GH, UK  
2 RAIN, University of Manchester, UK  
3 RACE, UKAEA, Culham Science Park, UK  
Email: [sgdbatty@liverpool.ac.uk](mailto:sgdbatty@liverpool.ac.uk)

**Abstract**In the nuclear industry, the need for improved reliability in current and future technology hinders the deployment of autonomous robotic systems. Existing solutions to the radiation-aware path planning problem require significant computational power. The research aims to develop a method of reliably mapping a large environment and abstracting the map into a sparse node graph. The proposed data form allows for efficient storage whilst maintaining important map features and coverage. The method utilises an expanding node algorithm to convert standard occupancy maps to a sparse node graph representation. The algorithm effectiveness has been tested on simulated maps and real-world maps to test the compression factor for various scenarios. The algorithm is expanded to function on a semi-unknown map abstracting during exploration.

**Keywords:** Occupancy Map, Map Abstraction, Mobile Robot, Autonomous Navigation, Radiation Hazard Avoidance

1 Introduction

As the nuclear industry in the United Kingdom moves forward, increasing focus is being placed on the decommissioning of legacy and recently decommissioned nuclear facilities and assets. The Nuclear Decommissioning Authority (NDA) are at the forefront of leading the implementation of protocols and policies in the clean-up effort. Recently the NDA released a set of “Grand Challenges” [1] to accelerate the technical innovation in nuclear decommissioning. One of these challenges involves creating and developing technologies and methods to move humans away from harm. This challenge is critical for reducing the risks posed to the workforce by many factors involved in hands-on tasks.

Whilst the challenge of removing humans from harm can be achieved through various technological advances such as the use of VR or augmented reality allowing for remote operation, the key area of focus for this research is the utilisation of robotic systems to reduce the need for humans to enter hazardous locations to perform routine tasks. These tasks range from routine inspection of operational facilities to the exploration and survey of legacy decommissioned facilities. Currently, these types of inspections are carried out by highly trained operators wearing radiological protective suits and breathing apparatus that, due to contamination, are typically deemed single-use and treated as contaminated waste after each operation. This method of working for routine tasks is costly and exposes the operators to physical and radiological danger. However, with technological advances, these tasks are becoming suitable for a robotic system. This would allow the operators to teleoperate and, ideally, supervise missions from the safety of a remote location. Additionally, this change in work methods would increase operator safety and efficiency while reducing the costs of these tasks [2].

One of the main problems facing the deployment of robotic systems is the safety concerns with deploying a device into an active environment and ensuring the system's reliable operation when exposed to harsh environments. Three main factors pose a problem to a robotic system. The first two, temperature and humidity, are standard for various environments, and off-the-shelf components are typically relatively resilient to high temperature and humidity levels. Nuclear industry environments are unique, except for aerospace applications, as they feature the potential for increased radiation levels that a robotic system would be exposed to during regular operation. Typical off-the-shelf components are susceptible to damage and failure from relatively low radiation dosage levels. [3] This damage can cause various effects, from memory corruption to system failures. This potential for system failure has led the nuclear industry to prefer systems to be deployed in a tethered configuration when deployed. Whilst guaranteed high-speed network communications between the user and the robot has the advantage, the tether dramatically restricts the range of exploration into complex facilities with a substantial risk of snagging and tangling during missions. The logical solution would be to operate the robotic system wirelessly. However, this poses a significant challenge that the following research aims to solve. This challenge is that an untethered system would be significantly more challenging to recover in case of a system failure or loss of signal with the user. In this event, the robot would be unable to return to the base. It could become an obstacle to future robotic missions and pose issues to other operations. The disabled robot would likely need to be recovered by deploying human operators into the environment. At that point, the purpose of utilising robots is negated because human operators are still required to enter the environment [4]–[6].

The following research focuses on further the capabilities of semi-autonomous ground vehicles. Currently, the industry's concern about the reliability of the technologies is holding back potential deployments of robotic systems into real-world scenarios. One of the main methods of improving the trust of the systems is to develop a method of allowing complex path planning and mapping to be carried out onboard the robot without relying on a centralised server. Existing literature solutions require a high-power computer to achieve the computation needed to process occupancy maps with multiple layers of information, for example, the work by A West et al. Additionally, the current literature explored in the next section only carried out the higher-level path and mission planning on relatively small and straightforward environments. This research aims to design a method of abstracting the current practice of storing mapping data into a sparse node graph representation. This method of modelling the environment will allow for significantly larger areas with even more data layers to be efficiently stored and computed using relatively low-power computing units. With robots looking to be deployed into mainstream facilities, the ability for the robot to achieve autonomy using low-power computers will significantly reduce the costs involved, further improving the argument for deploying systems into nuclear environments. This paper investigates the current state of research in this field (section 2), outlines how the algorithm functions (section 3), test the effectiveness of the algorithm in simulation and real-world testing (section 4), explores application when the environment is unknown (section 5) and concludes with potential applications of the work (section 6).

**2 Related Work**

This research is primarily inspired to be a precursor step to research carried out into developing a state-based risk framework for navigation in a hazardous environment featuring radiation and temperature level that may cause harm to the robot exploring them [7]. This concept utilises an assessment of the current health of the robot combined with an estimation of the environmental exposure expected along a selection of potential paths to determine which path will either cause the lowest harm to the robot or, if the damage is inevitable, which path will cause the least cumulative damage to the robot. The path planning for this method is standard occupancy map-based path planning. Additionally, research was conducted into mapping radiation in a similar environment and exploring how to locate possible radiation sources and subsequently avoid them during navigation to reduce the system's exposure [8][9]. Both methods are at the forefront of developing autonomy using ground rover navigation in a hazardous environment. However, they both feature the drawback of utilising occupancy maps for path planning. Given that multiple additional layers are added to the cost maps, both implementations' computation requirements are significant and require a high-power computer. These methods are also shown to work on relatively small single-room maps; however, for these systems to reach maturity, they will need to be capable of handling maps significantly beyond the reach of a tethered system.

The above systems represent the state of the art for avoiding radiation based on a mapped radiation environment. Several projects have been developed to achieve the radiation mapping and localisation of the source of the radiation. project CARMA, now succeeded by CARMA II, developed at the University of Manchester, is designed to be capable of searching for radiation sources autonomously. The system is based on the Clearpath Husky platform and is designed to patrol around a known area taking periodic readings to detect radiological leaks or contamination [9]. The system could avoid these radiation sources once detected.

Finally, several systems have successfully reached a high technological readiness level to be tested in active environments. The most notable ground-based systems are, firstly, Mirrax[10]. This system has been designed to be deployed through port holes at Sellafield and reconfigured inside a more traditional ground-based rover. Mirrax features a sensor set capable of mapping the environment to which it is deployed. However, this robot current is limited by regulations to be teleoperated and tethered, significantly reducing the range of exploration. Another similar system is project Lyra, formally Vega [11], [12]. Like Mirrax, this system is currently deployed in both active and non-active testing areas. However, it is also limited by requiring teleoperation and tethering. Both systems, whilst having the potential to be deployed with semi-autonomous capabilities, require more proven reliability of these technologies to be approved for active testing of the systems.

Self-driving cars are the other main area of research on the autonomous navigation problem. However, self-driving cars like the one created by Intel research for the urban challenge [13] feature powerful onboard custom computing hardware and network connections to compute complex environments. This technology isn’t feasible regarding practicality and expense when considering potentially disposable nuclear exploration robots, which, due to the nature of the environment, are typically low-cost disposable systems so advances in expensive technologies such as autonomous cars aren’t applicable.

**3.1 Methodology**

From the related work, the main limiting factor of the methods proposed is the significant amount of computational power required to work with even relatively small maps. The following procedure aims to abstract the occupancy map into a sparse nodal representation of the environment. This primarily aims to compress the data form that the map represents. The key features of the map must be maintained to allow any path planning carried out on the compressed version to have access to all the necessary features. The method creates a series of nodes by converting open spaces into nodes. The more open the area, the more spread out the nodes are placed.

On the contrary, features are preserved by more nodes being placed in confined and complex spaces. This method dramatically reduces the number of data points required to represent an area. This method is carried out recursively, allowing the algorithm to operate on maps of varying sizes and configurations without issue. The following pseudocode and figures show the algorithm functionality.

**3.2 Pseudocode for Map Abstraction**

***Function:***

*Read in Occupancy Map from map topic*

*Read in Current/Starting Position from SLAM node topic*

*Select the location for the initial node*

*Create a list of possible nodes*

*Create initial node*

*Append the initial node to the list of possible nodes*

*WHILE (list of possible nodes isn’t empty)*

*Expand out from the current node until a wall is found (Free space radius)*

*Identify neighbours of the current node (All those lying at free space radius +1)*

*IF (neighbour doesn’t lie within the radius of any nodes)*

*Calculate the free space radius of the neighbour*

*Append neighbour to the possible node list*

*Add the current node to the final node list*

*Remove the current node from the potential list*

*Sort potential node list (largest to smallest free space radius)*

*Select the next current node based on the largest free space radius*

*Remove any nodes from the possible list within the radius of the new current node*

*FOR each final node*

*Identify neighbour nodes IF Distance between nodes < threshold value*

*Line of sight between the two nodes*

*Output the list of computed nodes on the desired topic*

**3.3 Data Storage**

For the above algorithm, the recursive nature of the algorithm means that each cell's free space radius could be computed many times. To avoid wasted computation, a global array of free cell space radii is filled as each cell is calculated and checked each time a cell is called on for consideration. A cell also tracks whether it is within the radius of a chosen cell again to avoid this being repeatedly computed.

**3.4 Graphical Outline of Algorithm**

Shape

Description automatically generated with medium confidence Text

Description automatically generated Square

Description automatically generated with medium confidence Shape, square

Description automatically generated

Shape, square

Description automatically generated Shape, square

Description automatically generated Diagram

Description automatically generated

Figure 1 – Series of diagrams showing the application of the algorithm on an example occupancy-style map.

4.1 Implementation

Following the trend of robotics research, the system was developed using the ROS framework. The robotics community is progressively adopting the ROS framework for the speed at which it can be used to implement complex systems and the vast libraries allowing for hardware integration without needing to rewrite code.[14], For both methods, the primary computation was conducted on a setup ROS Noetic running on Ubuntu 20.04 on a computer with an Intel Core i7 10-core CPU and an Nvidia RTX 3070Ti laptop. Further testing is also carried out with the same setup running on an Aaeon Core Up Squared Pro single-board computer for operation onboard the physical rover. The simulation and real-world testing will utilise Hector Mapping as the mapping node for these experiments. Hector mapping is a node that uses the laser depth point cloud produced from LiDAR, either simulation or real-world sensor, to compute a 2D occupancy map of the environment [15]. For this testing, the cell size of the mapping was left at the default of 5 cm to allow for a relatively high resolution to be mapped. Hector mapping can also determine the pose of the sensor in the environment using the additional Hector SLAM (simultaneous location and mapping) package included in the library. Both sources of information are published as ROS topics which can be received by the abstraction node to be computed and run onboard the computer. The package has been designed to allow ROS1 and ROS2 compatibility to utilise ever-advancing libraries.



Figure 2 - Outline the high-level data flow from raw point cloud data to sparse node abstraction.

4.2 Simulation Environment

The effectiveness of the algorithm proposed in this paper was assessed through a series of simulations. These simulations test the algorithm in a controlled and repeatable environment. This allows for the initial development of the algorithm due to the input data being clean and easily obtainable. The simulation environment of choice is Gazebo, an industry standard Physics simulation package that allows the seamless integration of digital robots to be simulated using the same ROS packages as their real-world counterparts. Gazebo is advantageous, allowing a wide range of environments to be efficiently tested. This range allows for specific behaviours of the algorithm to be tested in isolation and combination to ensure that the system responds as expected.

To test the algorithm effectively, generating a series of maps of varying sizes and complexities is required to analyse the effectiveness of the algorithm's compression. Using Gazebo, it is simple to construct a series of maps to emulate typical scenarios that the system would find during operation. For this testing, three simulation maps were created. Firstly, a simple hexagonal arena ensured the algorithm was functioning correctly. Secondly, a maze-type structure was built. This maze featured diverging and converging paths and dead ends to ensure the algorithm could handle these features. Finally, a multi-room building was constructed to compare real-world testing later in the report. To simulate a real-world environment, further hazards such as tables, chairs and barrels were added to the world to allow the map to feature some nonstandard shapes.

Each of the tests was carried out in a standard manner. The robot was loaded into the world and teleoperated by the user around the map, as shown in figure 4a, until a complete map was captured using the Hector mapping package. Teleoperation was achieved using the Turtlebot3 teleop node. This stage was completed by simply running the abstraction node, which runs once when it sees a suitable map published into the map topic. Figure 2b shows an example of world abstraction; each sphere in the RViz window represents an assigned node. For each test, the initial node was set to be the starting spawn location of the robot. Due to the complete nature of the map, the algorithm should only be run once to abstract the whole environment.

Graphical user interface

Description automatically generated Graphical user interface

Description automatically generated

Figure 4—a) Simulated environment in Gazebo b) Abstraction of the Turtlebot3 House World as viewed in RViz

**4.3 Simulation Results and Analysis**

Following the successful abstraction of the map, it is possible to determine how compressed the abstraction of the occupancy map has become whilst ensuring that the map's features have been maintained. Initially, this is done by comparing the total number of unoccupied cells in the occupancy map against the total number of nodes in the abstracted map. These values are absolute and were run post-computation. Additionally, testing was conducted to determine the proportion of occupancy map cells considered when an A\* path-planning algorithm [16] is performed on the occupancy map. If a cell is checked for its heuristic value, it has been considered. This value is the average taken from several pairs of points positioned randomly around the map. This value indicates the performance increase achieved through the abstraction, as when considering path planning, not all cells of an occupancy map will be considered. This comparison is more indicative than measuring compute time of the algorithms as factors such as language and library inefficiencies and data source loading delays cannot be ruled out and tested comparably. The following figure shows this analysis applied to four artificially created maps like a real-world environment.

Figure 5 - Comparison between the number of Occupancy Map free cells vs Abstract Map Nodes

It can easily be seen that the nodal abstraction provides a data form that is consistently several orders of magnitude more compressed than the raw occupancy map data whilst visually maintaining the essential features of the map whilst maintaining coverage of the map with 99.9% of the unoccupied cells falling withing the free space radius of one of the nodes meaning that every cell is still reachable by the new abstract network. Simulation testing has shown that the proposed algorithm functions in the intended manner under various conditions. For example, when looking at the house model, the occupancy map created consisted of 72300 unoccupied cells. A\* running at the extremities of the map required the evaluation of 53103 cells. Following abstraction, this map is reduced to 102 nodes that have 100% coverage of cells within the building. Simulation has also been done to ensure that given a random list of nodes the robot is able to traverse between all of the nodes.

**4.4 Real-World System Testing**

Following the successful simulation, the next logical progression is to test the effectiveness of the algorithm compression on real-world maps. These maps differ from the simulations due to the level of noise and unpredictability compared to the pure nature of the maps produced in the simulation. The same mapping and abstraction nodes from the simulation testing were used, with the main difference being that the data was collected using a real LiDAR, RPLiDAR LiDAR S2 from SLAMtec. The LiDAR was mounted onto a ground rover, Miti Rover, by Rover Robotics.



Figure 6 - Miti Robot by Rover Robotics featuring custom payload with Intel Real sense Camera and SLAMtec S2 LiDAR

The robot's speed was limited to 0.5ms-1 to avoid issues with Hector mapping losing synchronisation. The abstraction was carried out after mapping was conducted on the Intel/NVidia-based laptop and successfully onboard the robot on the Aaeon Core Squared Pro board to show that the abstraction can be handled on the system without impacting the performance of the robot’s operation. Figure 4 shows the testing setup and the successful abstraction in a sizeable looping environment. Due to the real-world nature of the data collected, some minor clean-up was conducted to remove some minor particulate noise caused by reflections from the environment. This clean-up was later solved by using a higher-quality sensor. The next section features a similar numerical analysis conducted during the simulation testing.

A picture containing text, several

Description automatically generated

Figure 7 - Real-world Mapping and Abstraction Results

**4.4 Real-World System Testing**

As with the simulation testing, even with noisy non-standard maps, the compression from abstraction is equally as effective, with a significant reduction in the number of data points whilst maintaining map features and coverage. Figure 8, which outlines a series of three real-world maps, each shows a significant reduction, with the whole lab loop showing a reduction of 99.94% from the full occupancy map size to the abstracted map. This is very useful when calculating multiple cost map layers and higher-level path planning in terms of computational power required. Each map shows a similar order of magnitude reduction in the compression of the occupancy map representation.

Figure 8 - Free space cells vs considered cells vs nodes in abstract map comparison

**6 Conclusion**

The results shown in this paper has successfully demonstrated that the proposed algorithm is an effective method for abstracting occupancy map data into a significantly compressed representation to achieve several order-of-magnitude reductions in the quantity of data required to represent an environment. The sparse node graphs still provide a representation that maintains the features of the environment. The aim of creating a means of applying the risk-based framework proposed in related literature to a significantly larger environment in a substantially more computationally efficient manner can be achieved with the application of the above work. Instead of applying to an occupancy map, when applied to a nodal map will allow significantly higher-level decisions in more complex environments to be computed much more efficiently. The abstraction approach would allow for this to not only be attempted but also for the concept to be applied either for use in multi-mission applications with multiple map compilation to be attempted as well as multi-robot map consolidation across a fleet of robots.

**References**

[1] “The NDA’s ‘Grand Challenges’ for technical innovation - Cleaning up our nuclear past: faster, safer and sooner.” https://nda.blog.gov.uk/the-ndas-grand-challenges-for-technical-innovation/ (accessed Jan. 13, 2023).

[2] R. Smith, E. Cucco, and C. Fairbairn, “Robotic development for the nuclear environment: Challenges and strategy,” *Robotics*, vol. 9, no. 4, pp. 1–16, Dec. 2020, doi: 10.3390/robotics9040094.

[3] K. Nagatani *et al.*, “Gamma-ray irradiation test of Electric components of rescue mobile robot Quince-Toward emergency response to nuclear accident at Fukushima Daiichi Nuclear Power Station on March 2011,” 2011.

[4] T. Robinson, “Challenges for Robotics in Nuclear Decommissioning,” *Engineering & Technology Reference*, 2016, doi: 10.1049/etr.2016.0097.

[5] R. Smith, E. Cucco, and C. Fairbairn, “Robotic Development for the Nuclear Environment: Challenges and Strategy,” *Robotics*, vol. 9, no. 4, p. 94, Nov. 2020, doi: 10.3390/robotics9040094.

[6] J. de Geeter, M. Decréton, and E. Colon, “Challenges of telerobotics in a nuclear environment,” *Rob Auton Syst*, vol. 28, no. 1, 1999, doi: 10.1016/S0921-8890(99)00025-1.

[7] J. David, T. Bridgwater, A. West, B. Lennox, and M. Giuliani, “Internal State-Based Risk Assessment for Robots in Hazardous Environment,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2022, vol. 13546 LNAI, pp. 137–152. doi: 10.1007/978-3-031-15908-4\_12.

[8] A. West, T. Wright, I. Tsitsimpelis, K. Groves, M. J. Joyce, and B. Lennox, “Real-Time Avoidance of Ionising Radiation Using Layered Costmaps for Mobile Robots,” *Front Robot AI*, vol. 9, Mar. 2022, doi: 10.3389/frobt.2022.862067.

[9] K. Groves, E. Hernandez, A. West, T. Wright, and B. Lennox, “Robotic Exploration of an Unknown Nuclear Environment Using Radiation Informed Autonomous Navigation,” *Robotics*, vol. 10, no. 2, p. 78, May 2021, doi: 10.3390/robotics10020078.

[10] W. Cheah, T. B. Garcia-Nathan, K. Groves, S. Watson, and B. Lennox, “Path Planning for a Reconfigurable Robot in Extreme Environments,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2021, vol. 2021-May, pp. 10250–10256. doi: 10.1109/ICRA48506.2021.9561834.

[11] B. Bird *et al.*, “A robot to monitor nuclear facilities: Using autonomous radiation-monitoring assistance to reduce risk and cost,” *IEEE Robot Autom Mag*, vol. 26, no. 1, pp. 35–43, Mar. 2019, doi: 10.1109/MRA.2018.2879755.

[12] B. Bird *et al.*, “Vega—A small, low cost, ground robot for nuclear decommissioning,” *J Field Robot*, 2021, doi: 10.1002/rob.22048.

[13] D. Ferguson and M. Likhachev, “Efficiently Using Cost Maps For Planning Complex Maneuvers,” 2008. [Online]. Available: https://repository.upenn.edu/grasp\_papers/20

[14] M. Quigley *et al.*, “ROS: an open-source Robot Operating System.” [Online]. Available: http://stair.stanford.edu

[15] C. Kolhatkar and K. Wagle, “Review of slam algorithms for indoor mobile robot with lidar and rgb-d camera technology,” *Lecture Notes in Electrical Engineering*, vol. 661, pp. 397–409, 2021, doi: 10.1007/978-981-15-4692-1\_30/TABLES/2.

[16] N. Sariff and N. Buniyamin, “An overview of autonomous mobile robot path planning algorithms,” in *SCOReD 2006 - Proceedings of 2006 4th Student Conference on Research and Development “Towards Enhancing Research Excellence in the Region,”* 2006, pp. 183–188. doi: 10.1109/SCORED.2006.4339335.