Current Journal of Engineering and Science Research
Volume 1, Issue 1, pp 18-26, November 2024
ISSN (online): 3049-1738
DOI: https://doi.org/10.47001/CJESR/2024.101003
Available Online at https://www.cjesr.com

Intercommunication in Multi Robotic Cleaning System with Efficient Route Planning Through Map Decomposition

Benny George David

Department of Computer Science & Design, Viswajyothi College of Engineering and Technology, Ernakulum, Kerala, India

Abstract: Due to their effectiveness in performing cleaning activities in different contexts, cleaning robots have recently gained popularity. In this study, we propose a unique method based on map decomposition to guide and coordinate multiple cleaning robots to clean an entire area. The proposed method consists of three elements: finding the shortest and most efficient path, decomposing the map, and identifying obstacles. The best route each robot should take to cover the designated area is selected by the component that finds the shortest and most efficient path. Depending on the map decomposition component, each robot only covers the partial map assigned by the monitoring robot. The obstacle detection component enables robots to detect and avoid obstacles on their path. Simulated experiments were used to test the proposed method. The results show that it successfully controls and coordinates multiple cleaning robots to cover a large area based on map decomposition. Compared with traditional cleaning techniques, the strategy significantly increases cleaning efficiency and can be extended to clean large environments. The proposed method can be used in various areas, including airports, libraries and office complexes.

Keywords: Obstacle detection, map decomposition, cleaning robots, shortest path, cleaning systems, multi-robot, route planning.

I. INTRODUCTION

Cleaning robots have become more common in recent years due to their ability to perform cleaning tasks in different environments. A major challenge is to ensure that cleaning robots cover the entire area sufficiently and efficiently. Overlapping or uncleaned areas are among the challenges in coordinating and controlling multiple robots. In this research paper, we propose a method for controlling and coordinating multiple cleaning robots based on map decomposition to solve these problems.

Our system consists of three essential parts: obstacle detection, map decomposition, and finding the shortest and most efficient route. The first part of the system involves creating algorithms to determine the shortest and most efficient route each robot should take to navigate its assigned submap. The second step, called map decomposition, involves dividing the map into smaller submaps. This way, each robot can effectively cover its own area thanks to inter-robot intercommunication. Reducing crosstalk increases the performance of cleaning robots by reducing the cleaning time. Obstacle detection is about developing algorithms to find and avoid obstacles on robot paths. The proposed solution seeks to optimize the cleaning process by effectively dividing the area and coordinating the movements of multiple robots. This approach will allow cleaning robots to clean the map effectively and efficiently and with less human intervention.

II. BACKGROUND STUDY

Global and local route planning, which is divided into online and offline route planning depending on the environment, are the two categories of route planning techniques discussed in this section. Many algorithms have been developed for real-time route planning systems, including traditional and heuristic approaches. This research presents a novel algorithm based on modified graph search algorithms to select the most efficient and shortest path for cleaning robots in a known environment with static obstacles.

The Department of Electrical and Computer Engineering, Universidad Norte Sur presented an automated delivery robot with an obstacle detection and avoidance system that uses the Dijkstra algorithm to calculate the shortest route. This research used the modified Dijkstra algorithm and vector mapping to determine the ctual path is measured by the distance between rows and columns.

The same way as the Dijkstra algorithm, but focuses its search on states with the highest probability of success,

Current Journal of Engineering and Science Research
Volume 1, Issue 1, pp 18-26, November 2024
ISSN (online): 3049-1738
DOI: https://doi.org/10.47001/CJESR/2024.101003
Available Online at https://www.cjesr.com

potentially saving significant computation time. As part of their research Maren Bennewitz from an undisclosed institution proposed a fast pathfinding algorithm called Minimal Construct. The algorithm finds the shortest path by traversing visibility graphs. The algorithm is much faster than more advanced visibility graph algorithms because it only calculates the necessary part of the visibility graph around obstacles that block the path. To find the shortest path for a robot moving in a corrupted environment, Mohammed Aldarwbi and Uthman Baroudi from the Department of Computer Science at the University of New Brunswick in New Brunswick, Canada, invented an approach called FreeD*. This method combines the advantages of the D*, Dijkstra, and APF (Artificial Potential Field) algorithms. If there are unknown obstacles between the sub-paths, the technique integrates them into a single diagonal path and uses Dijkstra to optimize the generated path. APF is then used to avoid unidentified obstacles. [2] The results show that FreeD* can find the shortest path while avoiding unexpected obstacles, and the newly added systems lead to a reduction in the false positive rate from 1,400 to 501, or about 65%.

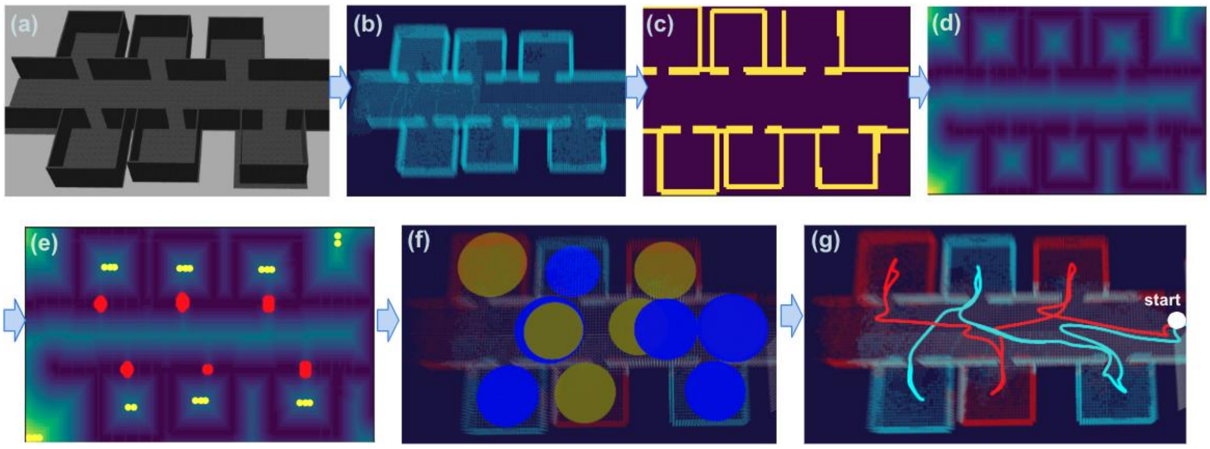

Figure 1: Overview of our exploration pipeline

In addition, duplicate paths in embedded decomposed maps prevent robots from cleaning efficiently because the full map is not known [6]. Effective distribution of cleaning among multiple cleaning robots requires assigning tasks to multiple robots, resilience is one of the most serious problems in centralized techniques. Real-time obstacle detection for an intelligent cleaning robot using a 3D depth sensor by S. Y. Kim et al. (2016) presents a real-time obstacle detection system for a cleaning robot using a 3D depth sensor. The system employs an RGBD camera to capture depth information of the environment and uses this data to detect obstacles in real-time. The proposed method is tested in various environments and is shown to be effective in detecting obstacles with high accuracy. [10]

Autonomous robot navigation and obstacle detection using ultrasonic sensor and Raspberry Pi by H. M. Sabra et al. (2017) describes a low-cost autonomous cleaning robot system that uses an ultrasonic sensor and Raspberry Pi for obstacle detection and navigation. The system uses the ultrasonic sensor to detect obstacles and then calculates the distance and direction of the obstacles. The Raspberry Pi then uses this information to control the motors of the robot and navigate it to its destination. [11] Obstacle detection and avoidance for a cleaning robot using a combination of LIDAR and an RGBD camera by S.

H. Park et al. (2018) propose an obstacle detection and avoidance system for a cleaning robot using a combination of a LIDAR and an RGBD camera. The system uses LIDAR for mapping and detecting obstacles in the environment, and the RGBD camera for obstacle recognition and avoidance. The proposed system is tested in various scenarios and has been shown to be effective in detecting and avoiding obstacles in real-time. [12]

Overall, these research papers demonstrate various approaches to obstacle detection and avoidance for cleaning robots using different sensors and algorithms. The proposed methods aim to improve the accuracy and robustness of obstacle detection and avoidance systems for cleaning robots, ultimately enhancing their safety and effectiveness.

Current Journal of Engineering and Science Research
Volume 1, Issue 1, pp 18-26, November 2024
ISSN (online): 3049-1738
DOI: https://doi.org/10.47001/CJESR/2024.101003
Available Online at https://www.cjesr.com

III. PROPOSED SYSTEM

a) Tools and technologies:

In this study, the Webots simulation tool was utilized to create a virtual environment with diverse obstacles and multiple robots. Webots, an open-source platform, facilitated the implementation of C-based robot controllers. The tool allowed for the creation of realistic cleaning scenarios. Multiple robots were deployed, and their coordination and intercommunication were supported by Webots. This combination provided a flexible and accurate simulation environment to evaluate the proposed method. By using Webots and C programming, it could optimize the performance and coordination of cleaning robots, enhancing efficiency in large-area cleaning tasks.

b) Find the shortest and most efficient path:

1) Path planning

- Route optimization: Several path planning methods will be applied, and data on each path's energy usage will be gathered. The algorithm that generates the most direct and economical path will be chosen.
- Changing Velocity and Acceleration: Using the chosen path planning method, the robot's velocity and acceleration will be changed to reduce energy consumption.

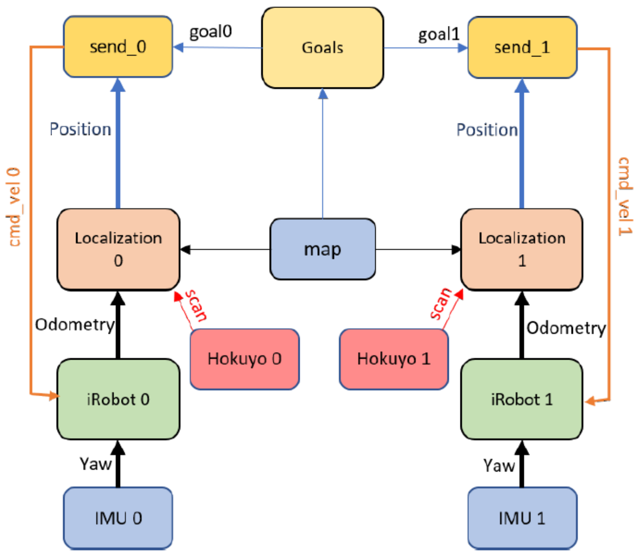

Figure 2: The flowchart of our multi-robotic system for cleaning

2) Energy consumption simulation

Simulate Energy Consumption: The distance traveled by the robot will be calculated, and the robot's velocity and acceleration will be adjusted to minimize energy consumption. The simulation will be run multiple times to collect data for different paths.

3) Analyze Data:

The energy consumption data will be analyzed to identify the shortest and most energy-efficient path for the cleaning robot.

4) Limitation and ethical consideration

The assumption that the simulation environment adequately represents the real world and the limits of the Webots simulation engine in effectively reproducing realworld physics are two limitations of this study. Ethics: As this study exclusively uses simulation-based research, no ethical issues are pertinent to it. Finding the quickest and most energy-efficient route for a cleaning robot is one major objective of this research. We suggest using the A* algorithm and Dijkstra's algorithm, two path-planning techniques, to achieve this. The most energy-efficient method may be found by utilizing either one of the algorithms alone or in combination.

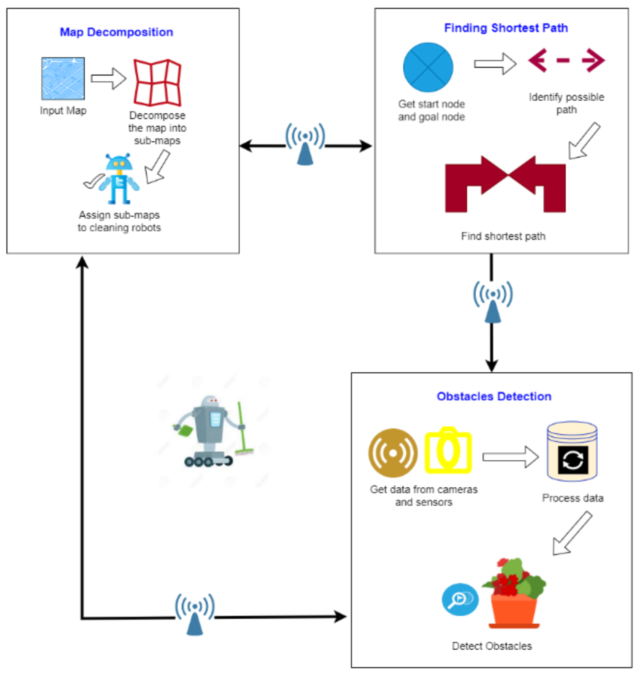

Figure 3: Finding the shortest and most efficient route

Once a node is selected, the algorithm checks if it is the last node. If so, the route has been found and the algorithm exists.

Current Journal of Engineering and Science Research
Volume 1, Issue 1, pp 18-26, November 2024
ISSN (online): 3049-1738
DOI: https://doi.org/10.47001/CJESR/2024.101003
Available Online at https://www.cjesr.com

Otherwise, the algorithm generates all neighbor nodes of the current node and calculates their preliminary g and f values.

c) Map decomposition and sub-map assigning:

In the beginning, a group of cleaning robots is divided into two groups called supervisor robot which is responsible for decomposing the map into sub-maps and sub-map cleaning robots. Utilizing a method that involves map decomposition, the objective of the supervisor robot is to explore the whole of the map. This robot's goal is to create a series of more detailed sub-maps by partitioning them into smaller regions.

After the supervisor robot has finished the process of exploration and decomposition, it will begin to build submaps based on the divisions that have been found. After that, the slave-cleaning robots are given responsibility for these sub-maps. Each robot that is used to clean sub-maps has a particular sub-map that is responsible for properly cleaning.

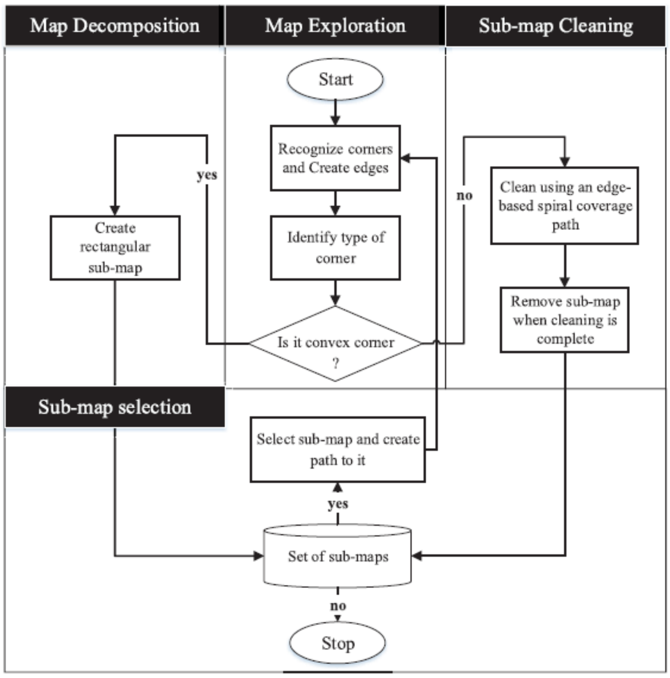

Figure 4: Summary of proposed CPP method

The cleaning chores for each sub-map are carried out by the robots that are allocated to that sub-map. They make use of a wide variety of cleaning methods and approaches to clean the areas that are specifically delegated to them in an efficient and effective manner.

The supervisor robot and the sub-map cleaning robots work together to finish cleaning the entire area. They involve a cleaning distribution approach to guarantee that all areas of the map are thoroughly cleaned, and the cleaning procedure does not include any instances of overlap or omission. The consequence of the robots working together and coordinating their efforts results in the thorough and methodical cleaning of the whole area that has been allocated.

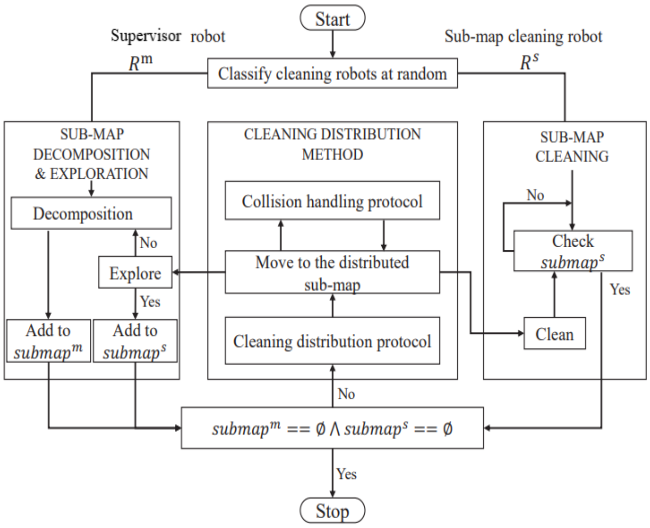

Figure 5: Flow chart for map decomposition

d) Obstacles Identification when changing light intensity:

Obstacle detection is a critical function for any cleaning robot. In Webots simulation, there are several ways to detect obstacles, depending on the specific sensors that are available on the robot.

In a Webots simulation, a cleaning robot can utilize a combination of cameras, GPS sensors, LiDAR, and distance sensors to effectively detect and overcome obstacles. Cameras provide visual input, allowing the robot to perceive its environment. The robot's image processing algorithms analyze the camera feed to identify objects or obstacles in its path. By detecting changes in pixel values or using object recognition techniques, the robot can determine the presence and location of obstacles. Cameras can also assist in identifying specific features or landmarks, aiding in navigation.

GPS sensors provide global positioning information, allowing the robot to determine its location within the simulated environment accurately. Although not directly used for obstacle detection, GPS sensors are valuable for mapping the

Current Journal of Engineering and Science Research
Volume 1, Issue 1, pp 18-26, November 2024
ISSN (online): 3049-1738
DOI: https://doi.org/10.47001/CJESR/2024.101003
Available Online at https://www.cjesr.com

environment and creating a reference for navigation. By combining GPS data with other sensor inputs, the robot enhances obstacle detection and path planning.

LiDAR sensors provide accurate distance measurements, allowing the cleaning robot to detect objects even in low light conditions. By analyzing the point cloud data generated by LiDAR, the robot can identify and avoid obstacles effectively.

Distance sensors, such as ultrasonic or infrared sensors, detect objects near the robot. These sensors are particularly useful for detecting objects close to the ground or at a short distance from the robot. They provide valuable information for obstacle avoidance and navigation in tight spaces.

By combining the information from these sensors, the cleaning robot creates a comprehensive understanding of its environment. Control algorithms process the sensor data to make informed decisions on obstacle detection, path planning, and avoidance strategies. This enables the robot to navigate autonomously and clean the environment effectively while avoiding obstacles and potential collisions.

In a Webots simulation environment, a cleaning robot can employ LiDAR sensors, infrared sensors, and cameras to detect obstacles effectively, even in low light conditions. LiDAR sensors emit laser beams and measure their return time to identify objects in the environment, independent of ambient lighting. This technology is particularly useful for obstacle detection in low-light environments. By analyzing the point cloud data generated by LiDAR, the cleaning robot can accurately determine the distance and location of obstacles, enabling successful navigation.

Infrared sensors, on the other hand, can detect objects based on emitted or reflected infrared radiation. They operate independently of visible light and are well-suited for obstacle detection in low-light conditions. The cleaning robot can utilize infrared sensors to provide proximity information, allowing it to detect obstacles in its immediate surroundings.

By analyzing the received infrared signals, the robot can identify objects or obstacles and take appropriate actions to avoid them. Cameras play a vital role in providing visual input to the cleaning robot, enabling it to perceive the environment.

1) Obstacles Detection in Low light Environment Simulation:

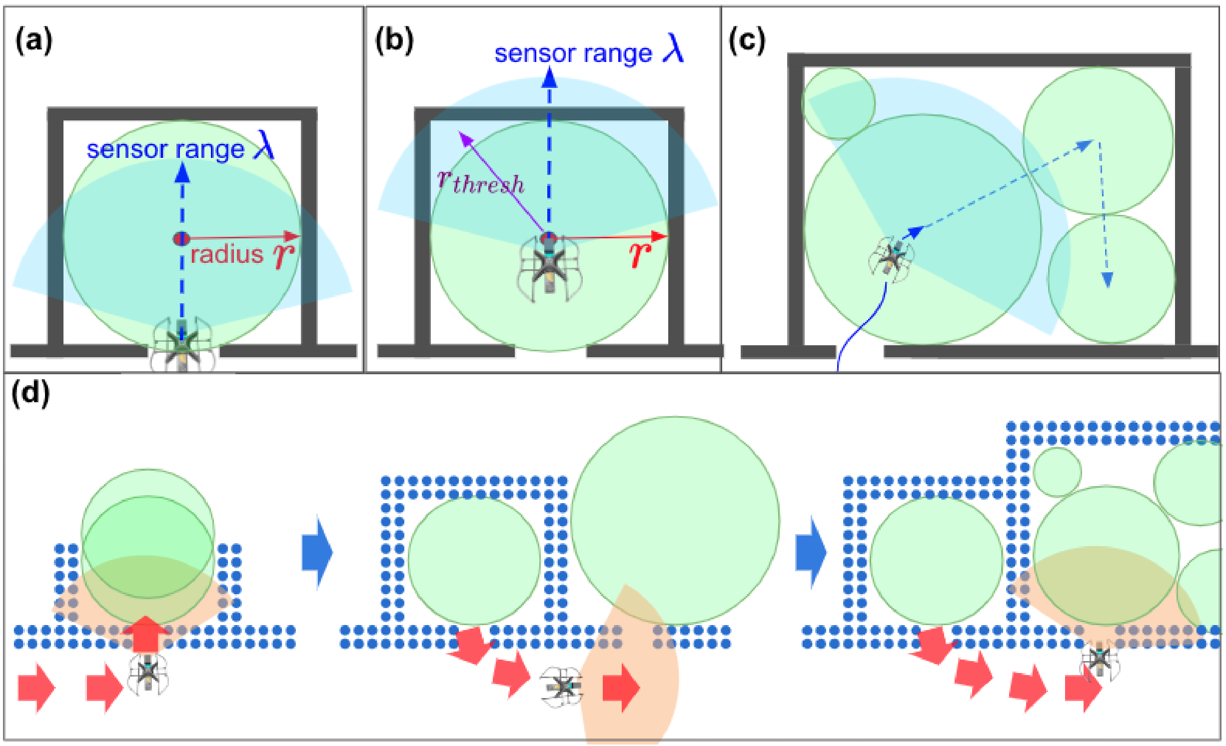

Figure 6: Extracting Geometric Cues of Doors and Rooms

CJESR JOURNAL

Current Journal of Engineering and Science Research
Volume 1, Issue 1, pp 18-26, November 2024
ISSN (online): 3049-1738
DOI: https://doi.org/10.47001/CJESR/2024.101003
Available Online at https://www.cjesr.com

Although cameras may be affected by low light conditions, there are techniques to enhance their performance. The robot can utilize cameras with low-light capabilities or employ image processing algorithms designed specifically for lowlight environments. By analyzing the camera feed, the robot's algorithms can identify changes in pixel values or use object recognition techniques to detect obstacles.

Cameras also offer contextual information, such as identifying landmarks or specific features in the environment, which aids in navigation and obstacle avoidance. By integrating the data from LiDAR sensors, infrared sensors, and cameras, the cleaning robot can create a comprehensive understanding of its environment, even in low light conditions. The robot's control algorithms process this sensor data to make informed decisions on obstacle detection, path planning, and avoidance strategies. For instance, if an obstacle is detected by LiDAR or infrared sensors, the robot can utilize the camera to further analyze the obstacle and gather more detailed information for navigation. By combining the information from these sensors, the cleaning robot can autonomously navigate, avoid obstacles, and effectively clean the environment, even when faced with low-light conditions.

2) Visualization:

To visualize obstacle detection in low light conditions using RViz in Webots and visualize the light condition, you can follow a series of steps. First, set up your Webots simulation environment to replicate low light conditions by adjusting the lighting parameters. This can involve reducing the intensity of light sources or adding filters to simulate low-light environments.

Next, enable the necessary sensor data output in your Webots simulation. Configure sensors like LiDAR, infrared sensors, and cameras to output relevant data such as point cloud data, distance measurements, and image frames. This data will be used for visualization in RViz. To establish communication between Webots and RViz, set up the ROS (Robot Operating System) interface in Webots. This allows for the transfer of sensor data from Webots to RViz for visualization purposes.

Launch RViz and configure the necessary settings to visualize the obstacle detection and light condition. This typically involves adding appropriate RViz plugins and configuring them accordingly. To visualize obstacle detection, import and visualize the sensor data in RViz. Use plugins like "PointCloud2" for LiDAR data, adjusting parameters such as point size, color scheme, and intensity values. For infrared sensor data, use plugins like "MarkerArray" to visualize detected obstacles as markers, or use the "Image" plugin to display distance maps. Camera data can be visualized using the "Image" plugin to display the captured images or video stream.

To visualize the light condition, you can add additional visual elements or overlays in RViz. For example, create a custom marker or object in RViz representing the lighting conditions, such as a sphere indicating the light source's position and intensity. Adjust the marker's color or opacity to accurately represent the low light conditions.

With the sensor data and light condition visualizations set up in RViz, you can observe and analyze the obstacle detection and the simulated low light conditions. This allows you to evaluate the performance of the cleaning robot's obstacle detection algorithms and assess the impact of low light on its capabilities.

By following these steps, you can effectively utilize RViz in Webots to visualize obstacle detection in low-light conditions and visualize the simulated lighting conditions. This comprehensive visualization provides valuable insights into the cleaning robot's perception and performance under challenging lighting situations.

3) Limitations:

Simulations may not fully capture the computational constraints and limitations of real-world systems. Detection algorithms and processing times in the simulation may differ from real-time requirements. Evaluating the performance of obstacle detection algorithms in terms of computational efficiency and responsiveness may require additional considerations.

4) Algorithm for obstacle detection:

Canny Edge Detection is a popular algorithm for detecting image edges. Although it is not typically used to directly detect obstacles in a cleaning environment, it can be used as part of a broader obstacle detection process. A hysteresis threshold is then edges above a certain intensity threshold and remove edges below a lower threshold. In the context of cleaning robots, Canny edge detection can be used as part of a broader obstacle detection process that includes thresholding and segmentation algorithms. For example, Canny's edge detection algorithm can be applied to an image on the robot the resulting edges can be used to identify

Current Journal of Engineering and Science Research
Volume 1, Issue 1, pp 18-26, November 2024
ISSN (online): 3049-1738
DOI: https://doi.org/10.47001/CJESR/2024.101003
Available Online at https://www.cjesr.com

areas of interest in the image that could correspond to potential obstacles in the cleaning environment.

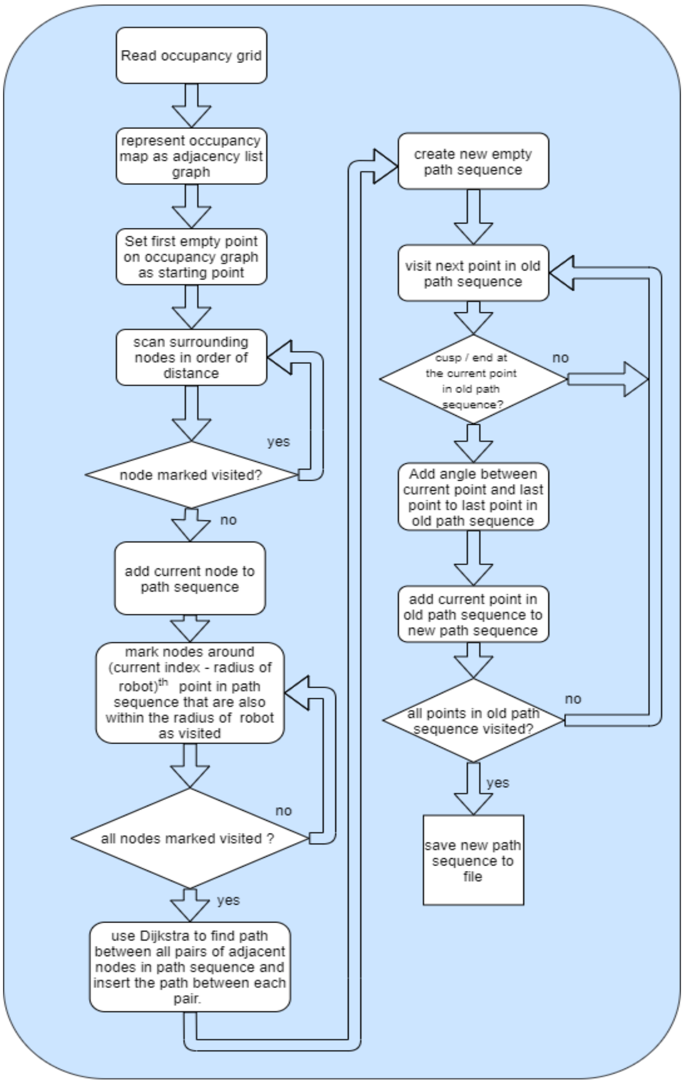

Figure 7: Flow chart for map decomposition

To refine obstacle detection, additional processing steps can be applied to the identified regions of interest. For example, Object detection algorithms can also be used to classify regions of interest and determine if they are obstacles or other objects in the environment.

Below are the general steps to implement the refined edge detection algorithm for obstacle detection in a cleaning environment:

- Image will be captured from the camera.
- Have to convert the images to grayscale.
- Apply Gaussian blur image conditions to reduce noise.
- Calculate the gradient of the image using the Sobel operator or other edge detection techniques.
- Apply hysteresis thresholding using the Canny algorithm.
- Identify regions of interest in the image based on detected edges.
- Apply morphological operations (erosion, dilation) to remove small objects or fill gaps.
- Use object detection algorithms to classify regions of interest as obstacles or other objects.
- Update the robot's obstacle detection system with new obstacle locations.
- Repeat for subsequent images captured by the robot.

IV. RESULTS AND DISCUSSION

Using a computer simulator and Webot, the proposed strategy was put into practice to control and coordinate multiple cleaning robots based on map decomposition. Using URDF, the physical properties of a cleaning robot as well as its model were described. The environment was designed to look like a normal house, with furniture, walls, and other obstacles, using Webot objects.

The environment was successfully divided into smaller sub-areas, which were then assigned to multiple cleaning robots by the system. To avoid collisions and overlapping coverage areas, robots interact with each other. The robots' trajectories contained obstacles that were identified and effectively avoided by the obstacle detection algorithms.

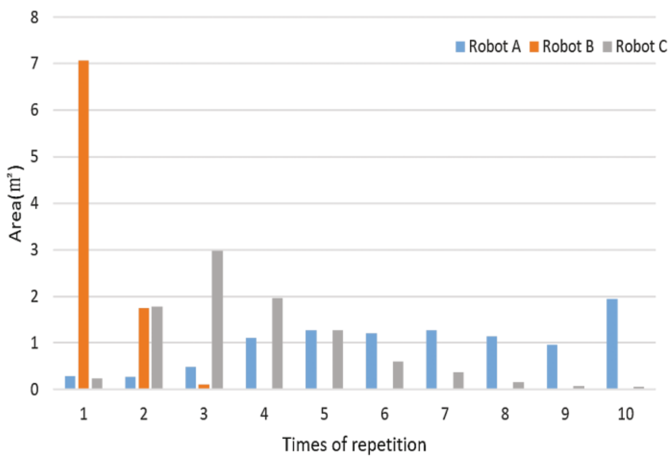

Figure 8: Robot repetition time

Each robot was able to navigate its designated area using the shortest and most efficient path according to the path

Current Journal of Engineering and Science Research
Volume 1, Issue 1, pp 18-26, November 2024
ISSN (online): 3049-1738
DOI: https://doi.org/10.47001/CJESR/2024.101003
Available Online at https://www.cjesr.com

planning algorithms. The proposed approach based on map decomposition to control and coordinate multiple cleaning robots of cleaning operations. The system cleans a larger area in less time, more efficiently and with less human interaction by dividing the environment into smaller submaps and assigning them to multiple cleaning robots. The effectiveness and efficiency of the system also benefits from route planning and obstacle detection algorithms. The proposed approach has several limitations. The system assumes that the environment is static and does not take into account changes that may occur during the cleaning process. The system also assumes that the cleaning robots have the same physical properties and capabilities and are identical. This is not necessarily the case in real-world situations where different types of cleaning robots could be used. The system may still need to be evaluated and validated in real-world conditions, even if it has already been tested with the Webots simulator.

V. CONCLUSION

In summary, our proposed approach to control and coordinate multiple cleaning robots significantly increases the efficiency and effectiveness of cleaning operations by decomposing maps, identifying obstacles, and finding the shortest and most efficient route. Our method can minimize the need for human interaction while expanding the coverage area of robots by optimizing the cleaning process. To achieve optimal efficiency and reduce the risk of collision, additional work can be done to further optimize route planning, obstacle detection, and intercom algorithms. Research could also focus on developing smarter cleaning robots that can adapt to different environments. The usefulness and feasibility of the proposed approach in many sectors can be demonstrated through further testing and deployment in real-world conditions.

REFERENCES

[1] K. M. Wurm, C. Stachniss, and W. Burgard, “Coordinated Multi-Robot Exploration using a Segmentation of the Environment,” in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp. 1160–1165.

[2] E. Brunskill, T. Kollar, and N. Roy, “Topological Mapping using Spectral Clustering and Classification,” in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 3491–3496.

[3] A.Zelinsky, R. A. Jarvis, J. C. Byrne, and S. Yuta, ``Planning paths of complete coverage of an unstructured environment by a mobile robot," in Proc. 6th Int. Conf. Adv. Robot., vol. 13. Tokyo, Japan, Nov. 1993, pp. 533538.

[4] M. Y. Jeon and H. S. Kim, "Obstacle detection and collision avoidance system for a cleaning robot using a laser scanner," 2013 IEEE International Conference on Mechatronics and Automation, Takamatsu, Japan, 2013, pp. 1155-1160, doi: 10.1109/ICMA.2013.6617986.

[5] H. Choset, ``Coverage for roboticsA survey of recent results," Annal. Mathemat. Intell., vol. 31, nos. 14, pp. 113126, Oct. 2001.

[6] J. S. Oh, Y. H. Choi, J. B. Park, and Y. F. Zheng, ``Complete coverage navigation of cleaning robots using triangular-cell-based map," IEEE Trans. Ind. Electron., vol. 51, no. 3, pp. 718726, Jun. 2004.

[7] Arulananth T S, Baskar M, P. Hari babu , R. Divya Sree , R.Sanjana, S.Bhavana: IR Sensor Based Obstacle Detection and Avoiding Robot - Palarch’s Journal Of Archaeology Of Egypt/Egyptology 17(9). ISSN 1567-214x.

[8] H. M. Sabra, A. Y. Abulila and R. S. Ganesan, "Autonomous robot navigation and obstacle detection using ultrasonic sensor and Raspberry Pi," 2017 International Conference on Smart Technologies for Smart Nation (SmartTechCon), Bangalore, India, 2017, pp. 575-578, doi: 10.1109/SmartTechCon.2017.8358443.

[9] S. H. Park, D. H. Kim and D. H. Cho, "Obstacle detection and avoidance for a cleaning robot using a combination of a LIDAR and an RGBD camera," 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, 2018, pp. 2020-2025, doi: 10.1109/ICRA.2018.8460862.

[10] I.A. Wagner, M. Lindenbaum, and A. M. Bruckstein, “Distributed covering by ant-robots using evaporating traces,” IEEE Transactions on Robotics and Automation, vol. 15, no. 5, pp. 918– 933, 1999.

[11] Z. L. Cao, Y. Huang, and E. L. Hall, ``Region lling operations with random obstacle avoidance for mobile robots," J. Robot. Syst., vol. 5, no. 2, pp. 87102, Apr. 1988.

[12] L. Paull, S. Saeedi, H. Li, and V. Myers, ``An information gain based adaptive path planning method for an autonomous underwater vehicle using sidescan sonar," in Proc. IEEE Conf. Autom. Sci. Eng., Toronto, ON, Canada, Aug. 2010, pp. 835840.

[13] S. Y. Kim, S. H. Park, J. Y. Lee and D. H. Cho, "Real-time obstacle detection for an intelligent cleaning robot

Current Journal of Engineering and Science Research
Volume 1, Issue 1, pp 18-26, November 2024
ISSN (online): 3049-1738
DOI: https://doi.org/10.47001/CJESR/2024.101003
Available Online at https://www.cjesr.com

using a 3D depth sensor," 2016 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, 2016, pp. 67-68, doi: 10.1109/ICCE.2016.7430464.

[14] Kang, Mun-Cheon & Kim, Kwang-shik & Noh, Dong-ki & Han, Jong- woo & Ko, Sung-Jea. (2014), "A Robust Obstacle Detection Method for Robotic Vacuum Cleaners", IEEE Transactions on Consumer Electronics. 60. 587-595. 10.1109/TCE.2014.7027291.

[15] T. Pilarski, M. Happold, H. Pangels, M. Ollis, K. Fitzpatrick, and A. Stentz, ``The demeter system for automated harvesting," Auton. Robots, vol. 13, no. 1, pp. 920, Jul. 2002.

[16] ElHalawany, Basem & Abdel-Kader, H & Tag Eldien, Adly & Ahmed, Alaa eldeen & Nossair, Z. (2012), "Vision-based obstacles detection for a mobile robot". MM-93.

[17] C. S. Kong, N. A. Peng, and I. Rekleitis, "Distributed coverage with multirobot system," in Proc. IEEE Int. Conf. Robotics and Automation (ICRA). IEEE, May 2006, pp. 2423–2429.

[18] I.Rekleitis, A. P. New, E. S. Rankin, and H. Choset, "Efficient boustrophedon multi-robot coverage: an algorithmic approach," Annals of Mathematics and Artificial Intelligence, vol. 52, no. 2-4, pp. 109–142, 2008.

[19] D. Zhu, C. Tian, X. Jiang, and C. Luo, "Multi-AUVs cooperative complete coverage path planning based on GBNN algorithm," in 29th Chinese Control and Decision Conference (CCDC). IEEE, May 2017.

[20] Y. Zhai, B. Ding, X. Liu, H. Jia, Y. Zhao, and J. Luo, "Decentralized Multi-Robot Collision Avoidance in Complex Scenarios with Selective Communication," IEEE Robotics and Automation Letters, vol. 6, no. 4, pp. 8379–8386, 2021.

[21] L. Wu et al., "Voronoi-based Space Partitioning for Coordinated Multi-Robot Exploration," Journal of Physical Agents, 2007.

[22] A.Solanas and M. Garcia, "Coordinated Multi-Robot Exploration through Unsupervised Clustering of Unknown Space," in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004.

[23] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning," IEEE Trans. Veh. Technol., vol. 69, no. 12, pp. 14 413–14 423, 2020.

[24] F. Gao, W. Wu, W. Gao, and S. Shen, "Flying on Point Clouds: Online Trajectory Generation and Autonomous Navigation for Quadrotors in Cluttered Environments," J. Field Robotics, vol. 36, no. 4, pp. 710–733, 2019.

[25] Y. Ren, F. Zhu, W. Liu, Z. Wang, Y. Lin, F. Gao, and F. Zhang, "Bubble Planner: Planning High-speed Smooth Quadrotor Trajectories using Receding Corridors," in IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2022, Kyoto, Japan, October 23-27, 2022. IEEE, 2022, pp. 6332–6339.

[26] T. Musil, M. Petrl´ık, and M. Saska, "Spheremap: Dynamic Multi-Layer Graph Structure for Rapid Safety-Aware UAV Planning," IEEE Robotics Autom. Lett., vol. 7, no. 4, pp. 11 007–11 014, 2022.

[27] S. Thrun, "Learning Metric-Topological Maps for Indoor Mobile Robot Navigation," Artificial Intelligence, vol. 99, no. 1, pp. 21–71, 1998.

[28] Y. Gabriely and E. Rimon, ``Spiral-STC: An on-line coverage algorithm of grid environments by a mobile robot," in IEEE Conf. Robot. Autom., vol. 1. Washington, DC, USA, May 2002, pp. 954960.

Citation of this Article:

Benny George David. (2024). Intercommunication in Multi Robotic Cleaning System with Efficient Route Planning Through Map Decomposition. *Current Journal of Engineering and Science Research.* 1(1), 18-26. Article DOI: https://doi.org/10.47001/CJESR/2024.101003

***** End of the Article *****

