Indoor Obstacle Detection and Fire Risk Evaluation Robot

By

Akaash Varatharajan, Febin Fredi, Gokul Srinivasan, Devesh Datwani

A Capstone Project Report

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Robotics Engineering

Abstract

In a study on home structural fires by the National Fire Protection Association, by categorizing the death and injury by activity when injured, it is observed that those who were fatally injured were most likely to have been trying to escape, resulting in around 38% of the deaths. This includes navigating through obstacles and getting out of entrapment. While the main cause of death is usually smoke inhalation, exit blockage is a very dangerous underlying condition that often leads to excessive smoke inhalation. In another survey it is found that out of all the firegrounds fatalities of firefighters, one-and two-family houses constitute the maximum fatality of 39 percent, making homes– the place people feel safest from fire, also where they are at significant risk. Thus, the team has made its goal to make homes slightly safer by developing an indoor autonomous mobile robot that moves around freely and detects obstacles blocking the exit, if any. Furthermore, the robot would notify the user about the obstacle and give out a score on how dangerous that obstacle could be in case of an emergency. The robot was able to successfully detect obstacles in the closest path to the exit and also give out suitable risk scores, in a simulation of house floorplans.

Table of Contents

List of Figures

List of Table

Authorship

Akaash Varatharajan started work with the navigation stack. Contributed to reaching out and following up with stakeholders Documented the minutes of meetings. Worked on navigation and exploration. Contributed extensively to the development of the gradient scoring algorithm. He contributed to map generation and change detection outputs. Contributed to Gazebo world generation for different obstacle positions. Contributed to quantifying the results. Contributed to authoring the Informed RRT* Algorithm section and the inference to the Test Results in the report.

Febin Fredi contributed to developing navigation and exploration. Implemented Informed RRT* algorithm. He contributed to map generation. Contributed extensively to the development of the gradient scoring algorithm. Contributed to Gazebo world generation for different obstacle positions. Contributed to optimizing the code and making it efficient. Contributed to authoring the Navigation and Exploration, Gradient Map, and Gradient Score sections of the report. Provided insights into the inference of the Test Cases.

Gokul Srinivasan contributed to reaching out and following up with stakeholders. Documented the minutes of meetings. Carried out a literature survey on existing works related to the project. Authored Abstract, Acknowledgements, Introduction, Related Work, Background, Future Work, Bibliography, and Appendix sections of the report. Contributed to authoring the Informed RRT* Algorithm section and the inference to the Test Results in the report. Executed test scripts to generate outputs of the map and Informed RRT* algorithm. Generated outputs artifacts for the presentation. Contributed to debugging gradient scoring codebase. Contributed to Gazebo world generation for different obstacle positions.

Devesh Datwani contributed ideas related to gradient scoring. Contributed to stakeholder meetings by presenting intuitive questions. Worked on implementing the Change Detection. Authored Change Detection and State Machine sections of the report.

Acknowledgments

We would like to thank everyone who helped our project by providing comments, time, and support. Our project would not have been able to accomplish what it has without your support. We would like to especially thank our advisor, Professor Loris Fichera for his continued patience, support, and guidance.

We would like to thank Professors James Urban, Milosh Puchovsky, Pedro Reszeka, Carlo Pinciroli, and William Michaelson for providing insights into the types of fires, the different fuels of fires, and the standard operating procedures of firefighters. We would also like to thank them for picking apart our ideas, finding the weakest link, and the inputs to improve them.

We would like to thank Diane Poirier, Administrative Assistant of the WPI (Worcester Polytechnic Institute) Fire Protection Engineering Department for helping us set up meetings with the professors. We would also like to thank her for connecting us with the Worcester Fire Department.

Finally, we would like to thank Brian McGurl, District Chief of Worcester Fire Department for speaking with us about how robots could assist the firefighters and giving insights on how to improve our ideas to better suit their needs. Their input was invaluable to ensure our project was grounded in real strategies.

Introduction

The National Fire Protection Association (NFPA) defines fire as a rapid oxidation process, which is an exothermic chemical reaction, resulting in the evolution of light and heat in varying intensities [1]. The base substance can vary from wood, plastic, or even metals. According to a recent report by the National Fire Protection Association, fire departments across the country responded to an estimated 1,353,500 fires in the US in 2021. These fires caused an estimated 3,800 civilian deaths;14,700 civilian injuries; and $15.9 billion in direct property damage. Out of which, 338,000 home structure fires in 2021 (27 percent) caused 2,840 civilian fire deaths (75 percent); 11,100 civilian injuries (76 percent), and $8.4 billion in direct property damage (38 percent). The major contributor to this is the 256,500 fires in one- or two-family home structures, which caused 2,440 civilian deaths, 8000 civilian fire injuries, and $6.9 billion in property damage [2].

In a recent study on home structural fires by the NFPA, by categorizing the death and injury by activity when injured, it is observed that those who were fatally injured were most likely to have been trying to escape, resulting in around 38% of the deaths This is followed by those who are sleeping amounting to 30 percent of the death. In contrast, the non-fatally injured were much more likely to have been trying to fight the fires themselves [3]. In another report by the NFPA, out of 28 fireground fatalities, 16 were in structure fires, including 11 in fires in one-and two-family homes (39 percent), two in apartment buildings (7 percent), and one each in an assisted living facility (4 percent), a shed (4 percent), and a chicken house (4 percent) [4]. Thus making home – the place people feel safest from fire – is where they are at significant risk.

The recent innovation and technological influence of robots have created opportunities to make many of the tasks that endanger people who risk their lives, easier and safer. Robotics applications are becoming common in everyday lives as well as in highly hazardous environments, reducing potential risks to humans. With these challenges in mind, the goal of our project is to develop an autonomous mobile robot, that ensures safe passage to emergency exits by identifying the obstacles obstructing the paths and notifying the users to remove them.

Related Work

The several research works and consumer products that have inspired this project are discussed in this section.

Research has shown that robots can play a significant role during emergencies and help evacuees to exits. Sakour and Hu reviewed current state-of-the-art robotic technologies deployed in a simulation of crowd evacuations. The paper also demonstrates how autonomous robots could be effectively deployed in disaster evacuations and rescue missions [5]. Wagner discusses the challenges of developing robot evacuation systems and had proposed five principles as an ethical underpinning for the ethical developments of robots, to guide the shifting of robots from novelty to them commonplace actors in everyday life [6]. M. Nayyar Et al from The Pennsylvania State University investigated the effects of the explanation provided by the robot on a person's decision to follow the robot’s guidance during a simulated emergency. They found that explanations increase compliance, but also increase evacuation time if the explanations are not concise [7]. Nayyar and Wagner presented a method that allowed the robot to use and create plans for obstacle detection and path clearing. The experiments presented in their paper demonstrated that their approach is generalizable to multiple environments. The results showed that the robot’s efforts can remove blockages and improve average evacuation time [8].

E. Ferranti and N. Trigoni discussed the problem of discovering evacuation paths in emergency areas, whilst exploring them and studied the interplay between exploration algorithms and two ERD algorithms namely Agent2Tag-ERD and Tag2Tag-ERD, where the former relies on agent-to-tag communication and the later on inter-tag communication [9]. K.S. Kiangala and Z. Wang designed an experimental safety response mechanism for an autonomous mobile robot running in a small manufacturing environment. Using the Q Learning Reinforcement Learning algorithm, the mechanism allowed the robot to generate a trajectory free of obstacles from its location at evacuation time to the closest emergency exit [10]. Compared to conventional evacuation, robot-assisted evacuation has many advantages when an emergency happens, such as approaching the evacuees quickly, guiding them efficiently to the shortest and safest route, and accessing the dynamic change in an environment in real-time such as pedestrians' density and flow, as stated by Tang et al. [11]

H.W. Lee Et.al proposed a method to accurately detect entrapped victims in an indoor disaster, especially those isolated due to fire smoke. The results show that the HHD method proposed in this paper produces better results than when YOLO or RetinaNet is used alone [12]. C. Pang Et.al presented a new approach to the detection of roads and obstacles using a single downward-looking 2D LiDAR, which was very effective and economic to be used in unmanned ground vehicles in urban environments. While the proposed method had some limitations in distinguishing dynamic obstacles, there was a stable performance in detecting roads and static obstacles [13]. J. Han Et.al proposed a road boundary and obstacle detection method using a downward-looking LiDAR sensor. Their proposed method extracts line segments from the raw data of the sensor in polar coordinates and then classifies the line segments into road and obstacle segments. The paper also uses the estimated roll and pitch angles of the sensor relative to the scanning surface [14]. B. Chen Et.al proposed an approach to detecting dynamic obstacles using 2D laser radar in real-time. The approach considers both temporal and spatial factors to detect the state of obstacles, resulting in improved reliability of the map. The paper identifies static obstacles by comparing the state of the same grid cell in three consecutive grid maps and the eight-neighbor cells. This new grid map with the static obstacles is then compared to the current grid map to identify the dynamic obstacles. The paper effectively demonstrated the proposed approach using the MORCS-1 mobile robot [15].

S.M.H. Rostami Et.al proposed a modified artificial potential field method employed for path planning and collision avoidance for autonomous mobile robots in environments with fixed obstacles. The proposed method solved the local minimum problem, where the robot gets trapped at the local minimum and couldn't achieve the target [16]. One of the most common commercial indoor mobile robots is the Roomba, iRobot's robotic vacuum cleaner. iRobot has sold around 35 million of these robotic vacuum cleaners across the world making it a household name, so ubiquitous that it is shorthand for every other robot vacuum available [17].

Background

This chapter introduces the stakeholders of this project and explains the background information necessary for understanding

Stakeholders

The first step in developing any robotic solutions was to identify and isolate the problems that required robots, while not interfering with already existing procedures. To better understand the gravity of firefighting and other components involved with it, the team set up in-person meetings with the faculties of the WPI Robotics Engineering department that had previously worked with robots in firefighting. The main discussion with these professors was based on the challenges faced by them in their previous endeavors in this field.

Furthermore, the team interviewed the faculties of the WPI Fire Protection Engineering Department. The discussions ranged from the standard operating procedures of firefighting, the types of fuels, personal experiences of firefighting, and brainstorming of possible uses of robots in a fire.

The team also interviewed Chief Brian McGurl, District Chief of Worcester Fire Department, and discussed how robots could assist the firefighters and insights on how to improve the ideas to better suit their needs. The firefighting department also provided a list of situations where they felt robots would be able to aid them. It was clear from this meeting that, for a robot to be an aid on the fireground, it needs to perform on par with firefighters. It was also evident that with the extensive training that firefighters go through, they could perform standard operating procedures subconsciously without much effort, thus reducing the need for a robot to directly assist them on the fireground. After this meeting, the team decided to shift the focus from directly assisting at the fire-gound, to preventing fires and protecting lives in the first place.

Based on the reports from the NPFA, the team decided to make it a goal to make the one-and two-family house safer. The team interviewed householders, and discussed their needs and wants in case of fires. The brief of all the discussions can be found in Appendix A.

The stakeholders identified for this project are the Homeowners and the Worcester Firefighting Department. The stakeholders were surveyed and interviewed to learn more about how the standard operating procedures, the techniques used during firefighting, the causes of fire in residential complexes, and the reasons for casualties in these fires. Furthermore WPI Fire Protection Engineering department. was identified as a stakeholder and interviewed to better understand how technology can be used to aid firefighters. These stakeholders are described below and summarized in Table 1.

Householder

The term householder refers to a person who occupies a house or tenement alone or as the head of a household. They are the primary users of the system.

Fire Department

The term fire Department refers to an organization that provides fire prevention and fire suppression services. While they are not the direct users of the system, the usage of this system will aid their operation as saving lives is their topmost priority.

Academia

The term refers to the academic world, particularly the WPI Robotics department and the WPI Fire Protection Engineering Department They are not the direct users of the system, but have the expertise to provide feedback and assistance to the development of the system.

To gauge the need for robotics in firefighting, the team first identified the stakeholders

The concise report of all the stakeholder discussions can be found in Appendix A.

Stakeholder	**Involvement**	**Extent of Involvement**	**Rationale**
Householders (SH01)	Direct, Engaged, Informed, Positive	Feedback on the current system	Customer of system
Fire Department (SH02)	Indirect, Consulted, Positive	Feedback on the current system	The system aids some of their tasks

Academia (SH03)	Indirect, Neutral	Feedback on the current system and prototype testing	Will interact with the system as needed

Table 1: Stakeholders Identification and Analysis

After collecting the data from stakeholders, the information was compiled into a table of user needs, which was of great help to understand the essential needs that the system should satisfy. The needs of the stakeholders are summarized in Table 2 below. It was decided to focus the project on addressing the needs of the householders, as addressing the needs of the firefighters or the academics would require a much larger scale, cost, complexity, and experience.

Stakeholder	**Need**
Householders	• The system should know the safest exit during a fire and guides the users to it by taking the safest path. • The system should give regular updates on any obstacles in the path to the exits. • The system should detect spot fires that happen inside the perimeter of an estate and eliminate them.
Fire Department	• A system to move and rescue as many people as possible from a residential fire. • The system should be able to relay information including, but not limited to, video monitoring of fire before reaching the situation to the commanding officer.

	• A system to extinguish fires in high-rise buildings, where it is hard to reach with ladders and stairs. • A system that could predict the movements of a wildfire. • A system that can control wildfire spread by extinguishing firebrands. • A system that could aid in the identification of signs of life in buildings in fire. • A system that allows visualization of images in smoke-filled rooms, improving the user's vision to see through smoke. • A system that could drain and clean the hose after fighting the fire, and loading it back into the truck.
Academia	• A system that could collect data for situational awareness. • A system that can identify signs of life in a building.

Table 2: Stakeholders Need Elicitation

System Requirements

The scope of the project was narrowed down to cover the subset of needs that the team found to be most important to the householders and achievable with available resources within the given timeline. From the information collected from the stakeholders, discussed previously, it was evident that an unassuming object could turn into a very obstructive obstacle easily, in an emergency. With this in mind, the team decided to focus on removing these unassuming obstacles before they could pose a serious risk. The team also felt that a system that detects an obstacle and initiates the user to remove that object would best meet the stakeholder's need under the time and resource constraints of the project. The list of achievable needs and the system requirements can be found in Table 3 below.

<table>
<tr><th>Stakeholder</th><th>Needs</th><th>System Requirements</th></tr>
<tr><td rowspan="4">Householders</td><td rowspan="3">The system should give regular updates on any obstacles in the path to the exits</td><td>The system shall scan the floor space at regular intervals.</td></tr>
<tr><td>The system shall update the user on the obstacles.</td></tr>
<tr><td>The system shall give a score based on how risky the object is in an emergency.</td></tr>
<tr><td>The system should know the safest exit during a fire and guides the users to it by taking the safest path.</td><td>The system shall provide the user with the shortest path to the exit from various points in space, in a non-emergency situation.</td></tr>
</table>

Table 3: Stakeholder's Needs and System Requirements

System Architecture

Navigation and Exploration

First, the turtlebot3 robot is spawned into an unknown world in Gazebo by running the 'explore.launch' launch file. As soon as the robot is spawned, lidar initializes and robot odometry starts getting published on the topic /odom by the gazebo. The LiDAR sensor scans a radius of three meters around the robot as its center and starts updating the point cloud with new scan data by publishing on the topic /scan. The robot then starts its mapping and published the map data on /map topic. Now, we run a python script named explore.py which is used to publish goal points by publishing the goal pose to the /move_base/goal. Gazebo is subscribed to the /move_base/goal topic, thus receiving the goal pose from the topic, whenever the script publishes the goal pose. Once we have the goal location, a global planner generates a global path plan to the goal, and the local planner navigates locally around the obstacles.

To move the robot along the path toward the goal, we need the linear velocity and angular velocity parameters. These parameters are published on the /cmd_vel topic, which Gazebo subscribes to thus getting a trajectory to the goal point consisting of velocity parameters. These velocity parameters are used by Gazebo to move the robot in a sim environment. After the robot reaches the goal location, the explore.py script publishes another goal location and keeps on publishing until the list of goal points has been located by the robot. While traversing these goal points the laser scan is running as well as the odometry data is published, thus performing the 2D slam along the way and thus generating a 2D map of the layout. If any new obstacle is introduced in the environment, the obstacle is detected while mapping and then added to the 2D map being generated. Once, exploration is done, we run the map_server node and use the map_saver function to save the occupancy grid data from the /map topic and, the global cost map data from the /move_base/global_costmap topic.

2D SLAM: The method we use here for 2D mapping – using lidar scan data and odometry data - is known as GMapping, which is one of the 2D SLAM methods available, others being Hector Slam, Cartographer, RTAB-Map, etc. The advantage of Gmapping is that, for memory used and the computation complexity, Gmapping is much more accurate than others. The node turtlebot3_slam_gmapping performs the Slam Gmapping and published the occupancy grid map data to /map topic.

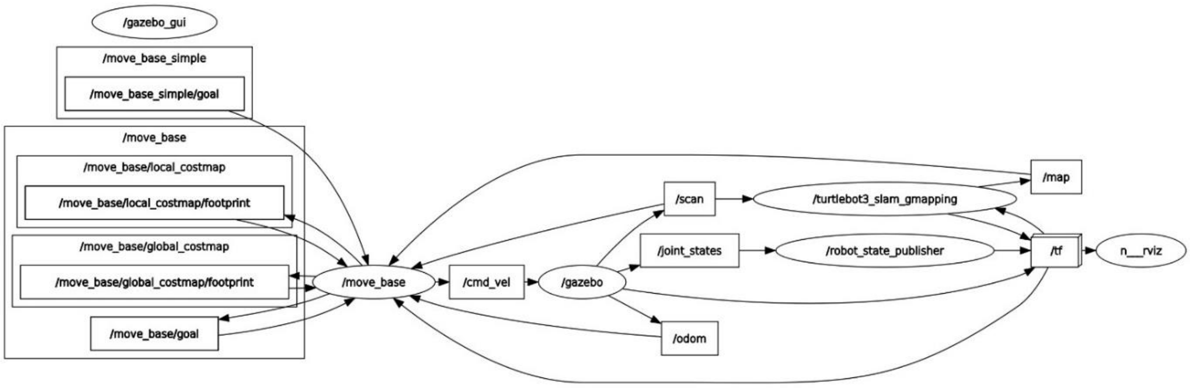


Figure 1: Ellipsoidal Informed Subset

Planners: Once the goal pose is received by Gazebo, the robot starts moving towards the goal location by following a global path to the goal location generated using a global planner using Dijkstra's algorithm. A local planner using the Dynamic Window Approach algorithm is used to avoid the obstacles on the path locally while navigating through the global path generated.

Maps: As shown in Figure 1, /move_base topic also has /move_base/local_costmap and /move_base/global_costmap. A costmap is a map with a cost assigned to every cell in a map, where the maximum cost is for the cells containing the obstacle and the zero for an empty cell. The costmap is generated by generating a cost around the walls to a distance up to the width of the robot so that, when the robot approaches the wall, it has enough space to move near the walls as the robot traverses the path with its center over the navigation path. A costmap is also important for the local planner to adjust the path to avoid obstacles, thus providing a safe path while traversing the map using the path generated by the global planner. Another map topic is /map which has the occupancy grid data, which defined whether a cell is occupied or not. While the robot explores the map, it assigns each cell as -1 or unknown/ not explored or a value ranging from 0-100 is the probability of the cell being occupied. The occupancy grid is used for fire safety score calculation, which is explained further below.

Map visualization using Rviz: To visualize the map generated, we are using rviz which subscribes to the topics and uses the data published on these topics to generate visual data. It uses data published on /map topic to show the occupancy grid, data from /move_base/global_costmap to show the global costmap, /move_base/local_costmap to show the local costmap, /scan to show the laser scan of the surroundings and, some other topics to show the global path generated and local planner generated path along with robot transforms and other data if necessary.

Informed RRT* Algorithm

Informed RRT* is a simple upgrade to the RRT* algorithm with the main difference being the informed direct sampling. The informed RRT* algorithm behaves as the RRT* path planning algorithm until the initial solution is found. Once the initial solution is found, it samples from the subset of states to improve the solution as defined by a heuristic. The subset of states requires no additional fine-tuning and it balances exploration and exploitation.

The informed RRT* is simplified with three simple steps after obtaining the initial solution. It is illustrated in the figure below.

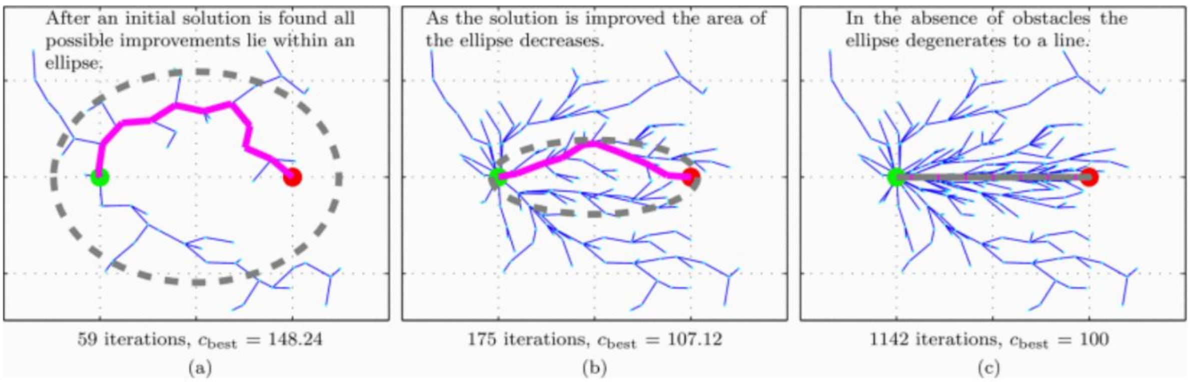
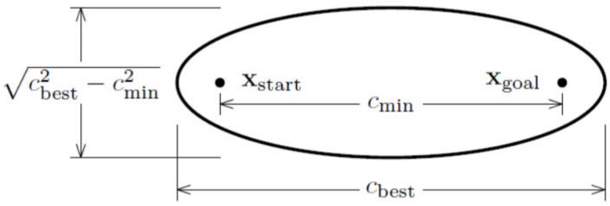


Figure 2: Ellipsoidal Informed Subset

The heuristic sampling domain $X_{\hat{f}}$ to minimize the path length is an ellipse with the Xstart and Xgoal states. The shape of the ellipse depends on the initial and final goal states as focal points. The minimum cost is calculated between the two as c_{min} and the cost of the best solution until then as c_{best}. The ellipse eccentricity is given by c_{min}/c_{best}



$$X_{\hat{f}} = \{\mathbf{x} \in X \mid ||\mathbf{x}_{\text{start}} - \mathbf{x}||_2 + ||\mathbf{x} - \mathbf{x}_{\text{goal}}||_2 \leq c_{\text{best}}\}$$

Figure 3: Ellipsoidal Informed Subset

For minimizing path length in R n, Euclidean distance is the heuristic for both terms about the start and goal positions (even with motion constraints). The subset of states would be improved for the current solution, $X_{fb} \supseteq X_f$, which can then be expressed in closed form as the cost of the current solution, c_{best}, as

$$X_{fb} = \ x \in X \quad ||x_{start} - x||_2 + ||x - x_{goal}||_2 \leq c_{best}\} \ ,$$

which is the general equation of an n-dimensional prolate hyper spheroid (i.e., a special hyper ellipsoid). The focal points are denoted as x_{start} and x_{goal}, the transverse diameter is denoted as c_{best}, and the conjugate diameters are the square root of ($c^2_{\ best} - c^2_{min}$).

The pseudocode of the algorithm using direct-informed sampling, Informed RRT*, is presented in Figures. 4 and 5. It is identical to RRT* as presented in [18], with the addition of lines 3, 6, 7, 30, and 31. This algorithm builds an optimal path to the planning problem by building a tree in state space incrementally. The tree is denoted with T = (V, E), consisting of a set of vertices, $V \subseteq X_{free}$, and edges, $E \subseteq X_{free} \times X_{free}$. New vertices are added by growing the graph in free space towards randomly selected states. The graph is rewired with each new vertex such that the cost of the nearby vertices is minimized.

The algorithm differs from RRT* where once a solution is found, it improves the solution by focusing on the search part of the planning. This is done by sampling the ellipsoid heuristic directly. In the pseudocode, at line 30, after the solutions are found, they are added to the list of possible solutions in line 31.

It does this through direct sampling of the ellipsoidal heuristic. As solutions are found (line 30), Informed RRT* adds them to a list of possible solutions (line 31). It uses the minimum of this list (line 6) to calculate and sample X_f directly (line 7). As is conventional, we take the minimum of an empty set to be infinity [19].

Algorithm 1: Informed RRT*($\mathbf{x}_{start}$, $\mathbf{x}_{goal}$)

```
1  V ← {x_start};
2  E ← ∅;
3  X_soln ← ∅;
4  T = (V, E);
5  for iteration = 1 ... N do
6      c_best ← min_{x_soln ∈ X_soln} {Cost (x_soln)};
7      x_rand ← Sample (x_start, x_goal, c_best);
8      x_nearest ← Nearest (T, x_rand);
9      x_new ← Steer (x_nearest, x_rand);
10     if CollisionFree (x_nearest, x_new) then
11         V ← ∪ {x_new};
12         X_near ← Near (T, x_new, r_RRT*);
13         x_min ← x_nearest;
14         c_min ← Cost (x_min) + c · Line (x_nearest, x_new);
15         for ∀x_near ∈ X_near do
16             c_new ← Cost (x_near) + c · Line (x_near, x_new);
17             if c_new < c_min then
18                 if CollisionFree (x_near, x_new) then
19                     x_min ← x_near;
20                     c_min ← c_new;
21         E ← E ∪ {(x_min, x_new)};
22         for ∀x_near ∈ X_near do
23             c_near ← Cost (x_near);
24             c_new ← Cost (x_new) + c · Line (x_new, x_near);
25             if c_new < c_near then
26                 if CollisionFree (x_new, x_near) then
27                     x_parent ← Parent (x_near);
28                     E ← E \ {(x_parent, x_near)};
29                     E ← E ∪ {(x_new, x_near)};
30         if InGoalRegion (x_new) then
31             X_soln ← X_soln ∪ {x_new};
32 return T;
```

Figure 4: Informed RRT* Algorithm [19]

Algorithm 2: $\texttt{Sample}\left(\mathbf{x}_{\mathrm{start}}, \mathbf{x}_{\mathrm{goal}}, c_{\max}\right)$

if $c_{\max} < \infty$ **then**
- $c_{\min} \leftarrow \left\|\mathbf{x}_{\mathrm{goal}} - \mathbf{x}_{\mathrm{start}}\right\|_2$;
- $\mathbf{x}_{\mathrm{centre}} \leftarrow \left(\mathbf{x}_{\mathrm{start}} + \mathbf{x}_{\mathrm{goal}}\right)/2$;
- $\mathbf{C} \leftarrow \texttt{RotationToWorldFrame}\left(\mathbf{x}_{\mathrm{start}}, \mathbf{x}_{\mathrm{goal}}\right)$;
- $r_1 \leftarrow c_{\max}/2$;
- $\{r_i\}_{i=2,\ldots,n} \leftarrow \left(\sqrt{c_{\max}^2 - c_{\min}^2}\right)/2$;
- $\mathbf{L} \leftarrow \mathrm{diag}\{r_1, r_2, \ldots, r_n\}$;
- $\mathbf{x}_{\mathrm{ball}} \leftarrow \texttt{SampleUnitNBall}$;
- $\mathbf{x}_{\mathrm{rand}} \leftarrow \left(\mathbf{C}\mathbf{L}\mathbf{x}_{\mathrm{ball}} + \mathbf{x}_{\mathrm{centre}}\right) \cap X$;

else
- $\mathbf{x}_{\mathrm{rand}} \sim \mathcal{U}(X)$;

return $\mathbf{x}_{\mathrm{rand}}$;

Figure 5: Sample function of Informed RRT* Algorithm [19]

Change Detection

Once a two-dimension map is generated. Our system locates possible obstacles and their locations relative to the map. This is done by a change detection algorithm. The change detection algorithm is predominantly used for detecting changes in satellite imagery of a geographical location. It is called unsupervised change detection using PCA and K-Means clustering.

Our system carries this out by comparing maps of ground truth value to the maps generated in real-time. First, the ground truth map and real-time map are subtracted. The difference image is partitioned into hxh non-overlapping blocks. A set of orthonormal eigenvectors is extracted with Principal Component Analysis. Each pixel in the difference image is represented with an s dimensional vector where s <= h. This feature set is clustered using the K-Means algorithm. We use a cluster size of two to effectively cluster each pixel into changed or not changed. We then display the resulting new image with binary values where 1 represents change and 0 represents unchanged. In our experiments, we found this to be an effective way to detect and locate a change/obstacle in the floor plan [20].

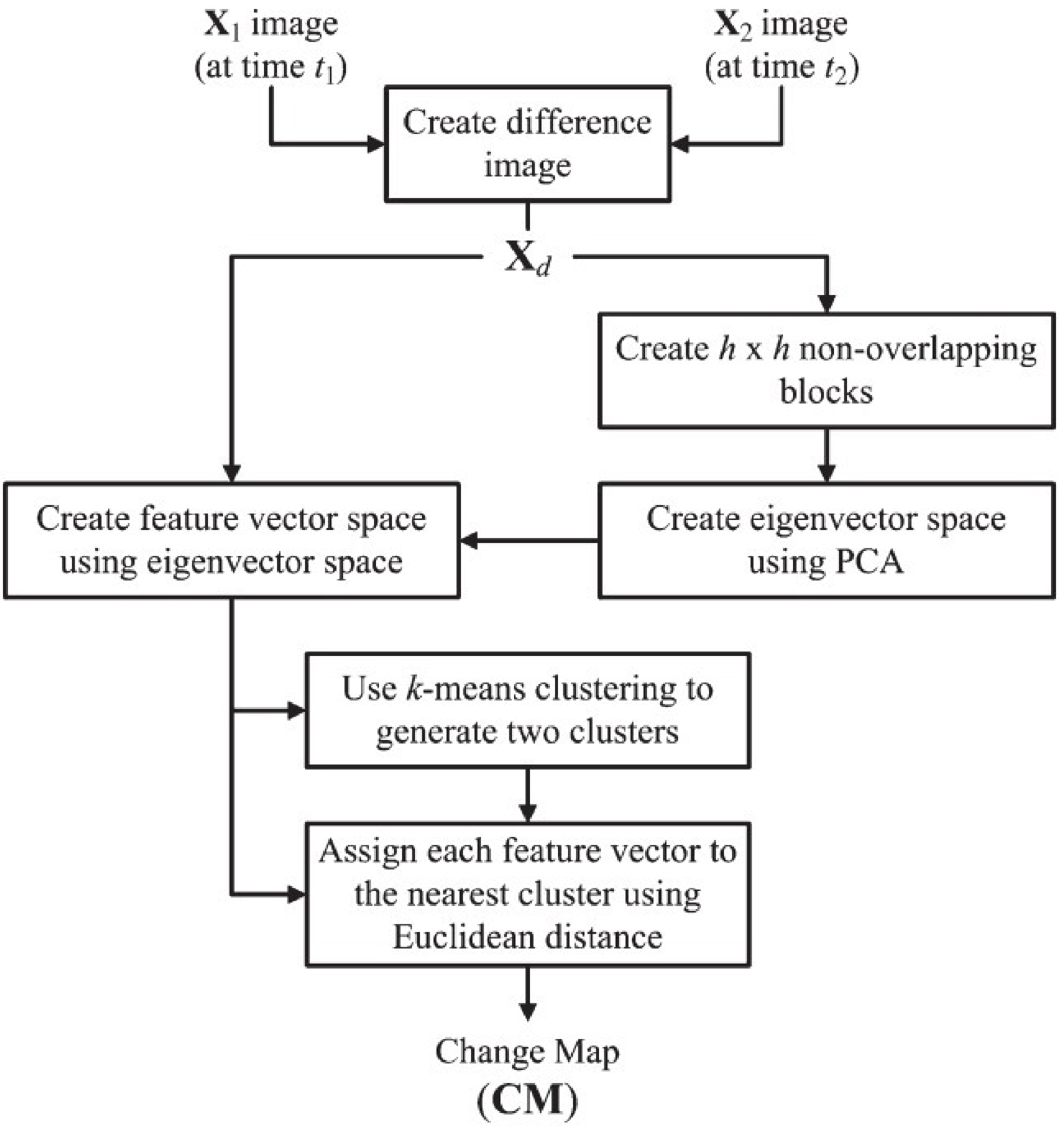


Figure 6: Change Detection

State Machine

State machines have become increasingly popular in robotics and web development. A finite state machine is a framework that defines a robot state with exactly one state out of the finite possible states. Each state executes a subroutine it is responsible for. Depending on the outcome of the routine, the state returns a value that switches the state from the current to another. The states and their trigger responses are pre-defined. We define our system with four states. Please note that this is created for prototyping and presenting proof of concept thus the complexity of the finite state machine is limited. The four states are

- Rest
- Map
- Score
- Notify

Since there is no set safety standard for our approach, the ideal time to rest is arbitrary and user-defined. In the rest state, the robot does nothing and sits idly at its home position. The resting time can be anywhere from 2 hours to 10 hours.

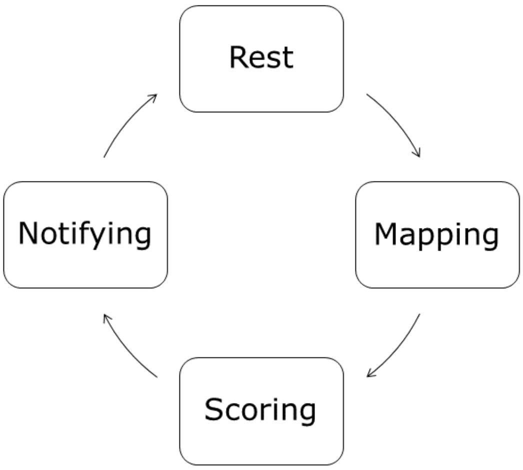


Figure 7: State Machine routine

Map: In the mapping state, the robot autonomously navigates to the waypoints, or explores the building floor and returns to its home position. At the current complexity, there are no fail-safes for this routine. Hence the robot will have to ideally map the home and navigate itself back to the homing position.

Score: During the scoring state, the system compares the current map with the ground truth map to detect changes and hence possible obstacles. Further, the informed RRT star algorithm scoring is factored in with the final score, and then the state switches to either Rest or Notify.

Notify: If this state is triggered by the map state, the end user is notified of the possible obstacle through a push notification. After this state, the state is transitioned to rest again.

Gradient Map

The gradient map contains the gradients around specified risk points, which are coordinates on space, where an obstacle will be a critical danger to the safety of the user. The gradient region is defined around each risk point with the risk points having the maximum cost and decreasing outwards from the center, thus creating a gradient region around each risk point. The gradient region is created using a kernel of K x K size created using Gaussian distribution. The value of the gradient map array ranges between one and the maximum cost.

Gradient Score

Change Map

The change map compares the changes between the two maps and highlights the region where changes are observed. The change map is a binary map with obstacle pixels being black and every other pixel being white. Finally, the binary change map is converted to a binary array where obstacle pixels are assigned a value of one and every other pixel a value of zero.

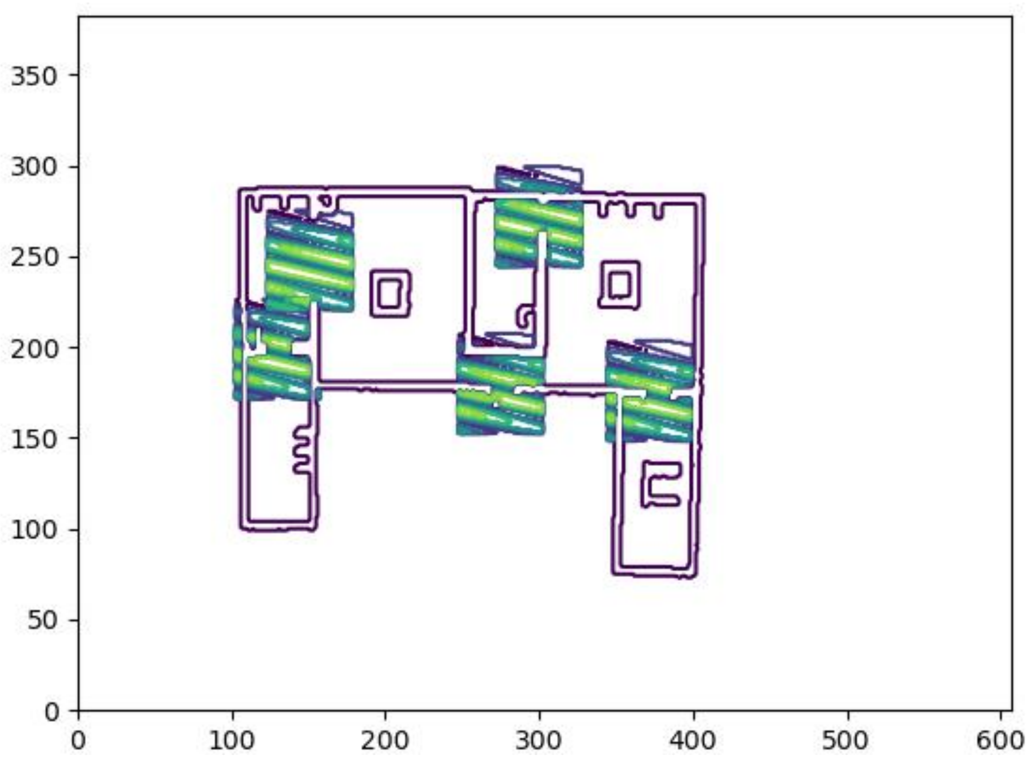


Figure 8: Gradients implemented at the door location of the house

Gradient Score Calculation

To calculate the gradient score, an element-wise multiplication between the Gradient Map array and the Change Map array is performed. This results in an array of values ranging from zero to the max cost. Then all the values in the array are added, giving us a gradient score for the whole map. For calculating the gradient score for every room, we find the gradient score for a specific region of the gradient map array by using the room coordinates.

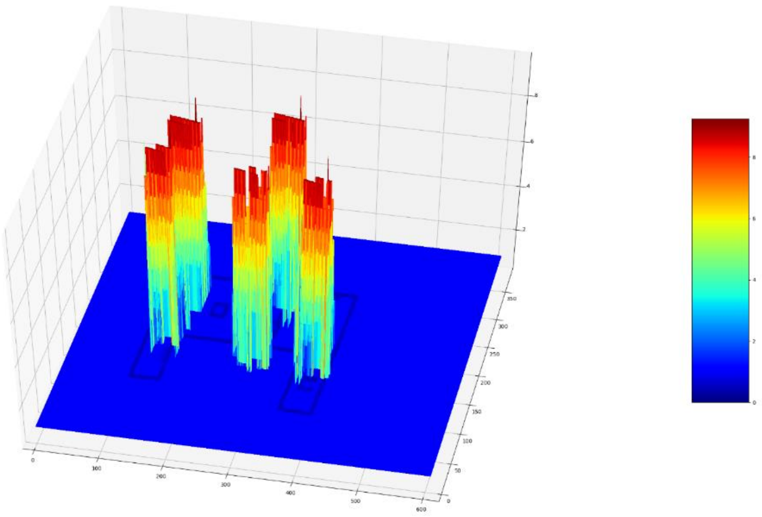


Figure 9: Gaussian gradient visualization in 3D

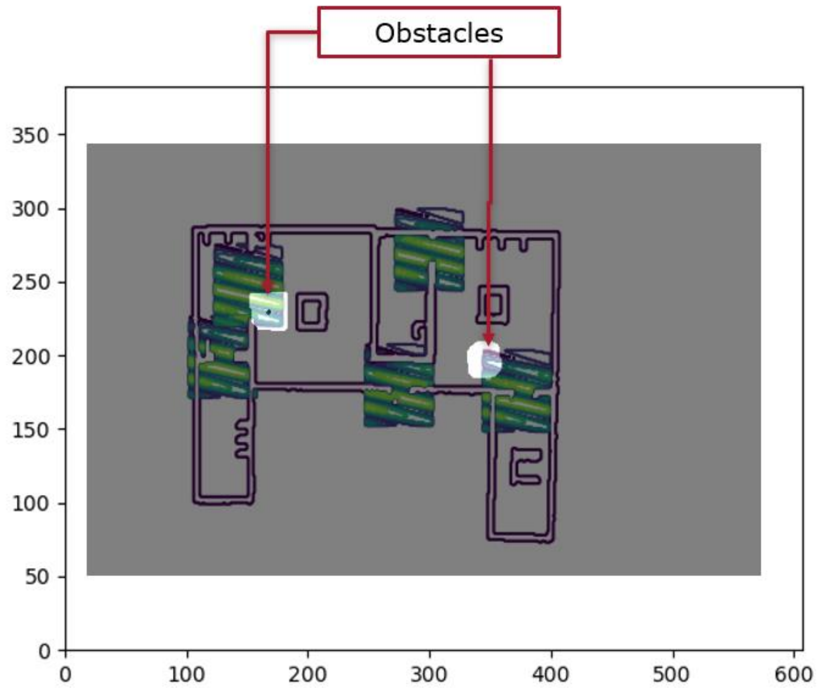


Figure 10: Combining Gradient Map and the Change Map

Results

Test Setup

Turtlebot3 was chosen as the robot for simulation. The Noetic version of the Robotic Operating System was used as the communication medium between the team, the turtlebot, and the Simulation environment.Gazebo11 was used as the simulation medium for generating the environment and simulating the navigation and map generation. The turtlebot3 packages such as Turtlebot3 Simulation, Turtlebot3 Navigation, and Turtlebot3 GMapping were used to implement navigation and map generation. The codebase was developed using Python3.8.

A 3D house layout was built as a world file and imported into the Gazebo simulator. The turtlebot was spawned into this world and navigation is based on user-defined points. While navigating to these points, the bot simultaneously maps the entire area and stores the map. This process is then performed again after introducing new objects into the world and the navigation is performed based on the same user-defined points used previously. Thus resulting in maps with and without obstacles.

The navigation was performed on different object orientations and their maps are obtained. Since the project aims at using robotic vacuum cleaners as a base, immovable obstacles like pieces of furniture, closets, etc are considered part of the ground truth.

Test Results

Map Generation

The navigation was performed on environments, with user-defined points, that the bot had to reach. The bot maps the area with its LiDAR sensor while navigating to these points. During this process, the bot creates an occupancy grid of the data. At the end of this process, the occupancy map is converted into a binary image. This process is repeated after introducing new objects in the environment. The maps obtained from these processes are shown below in Figure 10.

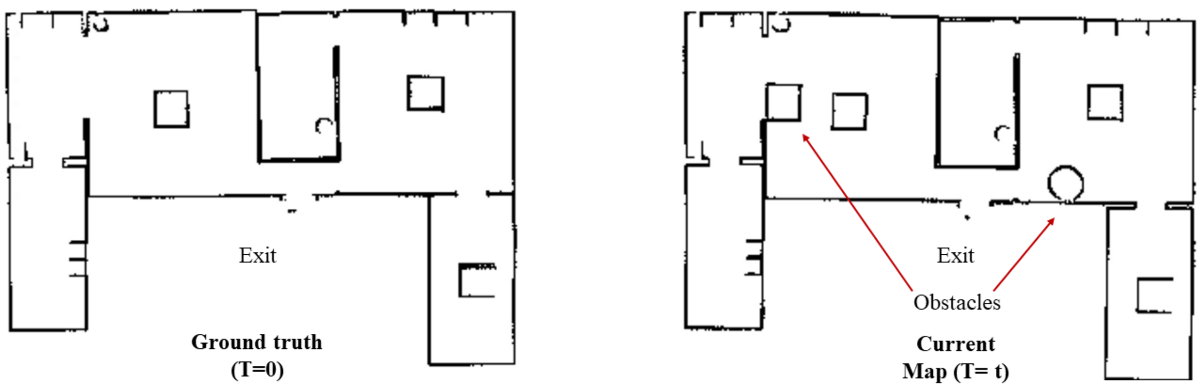
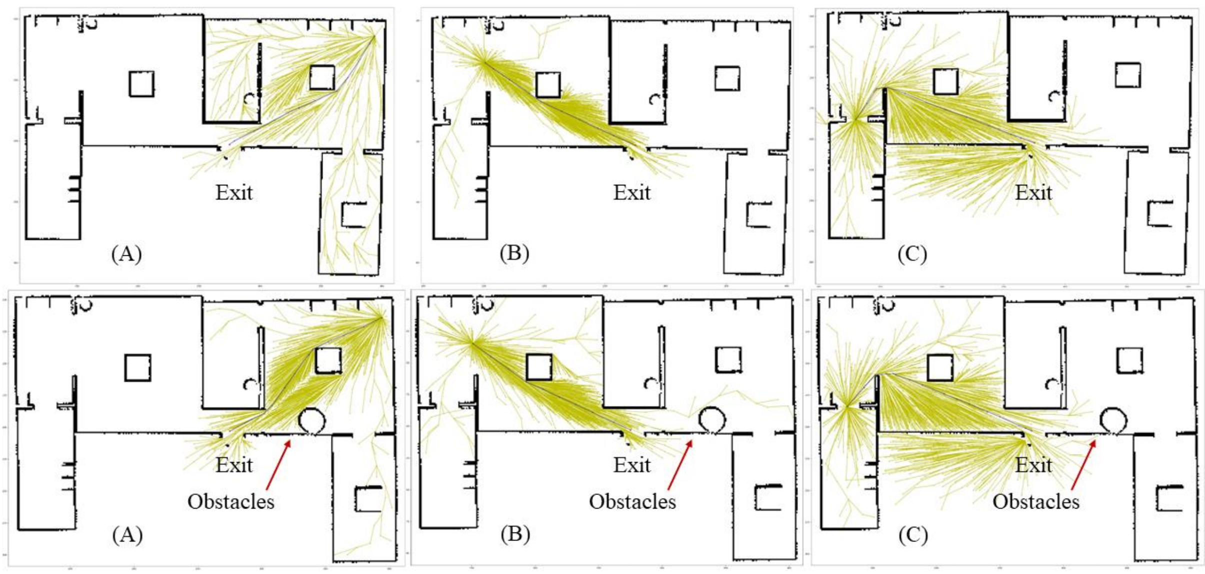


Figure 11: Floor maps

<u>Informed RRT* Implementation with a singular object</u>

In this case, a single obstacle was placed in the environment and the path differences between the ground truth and the new map are observed. Six points were chosen that are generally the opening of the room and Informed RRT* was performed to get the optimal path between these user-defined points as the start points and the exits are always considered as the endpoint. It was observed that there is a positive percentage increase in the path length whenever an obstacle is present and thus giving us an indication of an obstacle being present in that room.



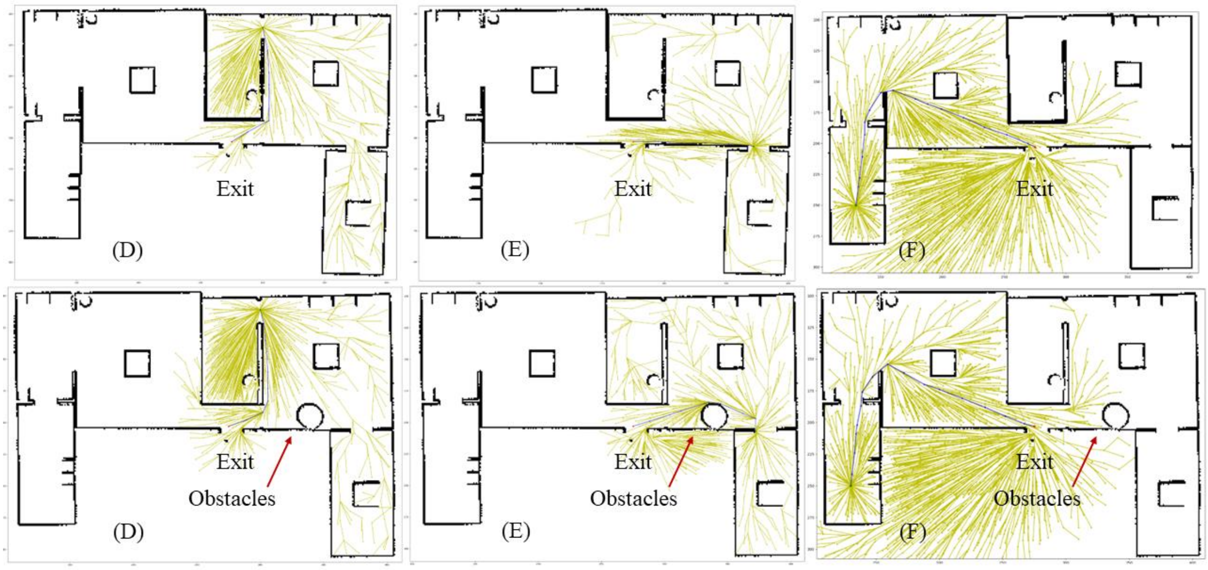


Figure 12: More Informed RRT* Implementation with singular obstacle

Scenarios	Path length in the ground truth	Path length in the current Map	Percentage change in path length
Scenario A	152	152	0.00%
Scenario B	141	141	0.00%
Scenario C	165	166	0.61%
Scenario D	108	111	2.78%
Scenario E	97	107	10.31%
Scenario F	228	228	0.00%

Table 4: Path Length changes with singular obstacle

Two unique cases were observed:

- Case A: In the top right room when path planning is performed, two possible optimal paths of the same path lengths are found, above or beyond the table (square in the middle of the room). Thus when an obstacle is introduced (bottom map), the path planning algorithm takes the other path (above the table), thus indicating no path change.

- Section D: In this case, the optimal path is not affected by the obstacle even though the obstacle is creating a bottleneck.

<u>Informed RRT* Implementation with Multiple Obstacles</u>

Now instead of one object, multiple objects are introduced into the environment with different orientations, these are classified into different cases and are discussed here.

CASE – 1

In this case, along with the object used previously, another object in front of the top left room is introduced.

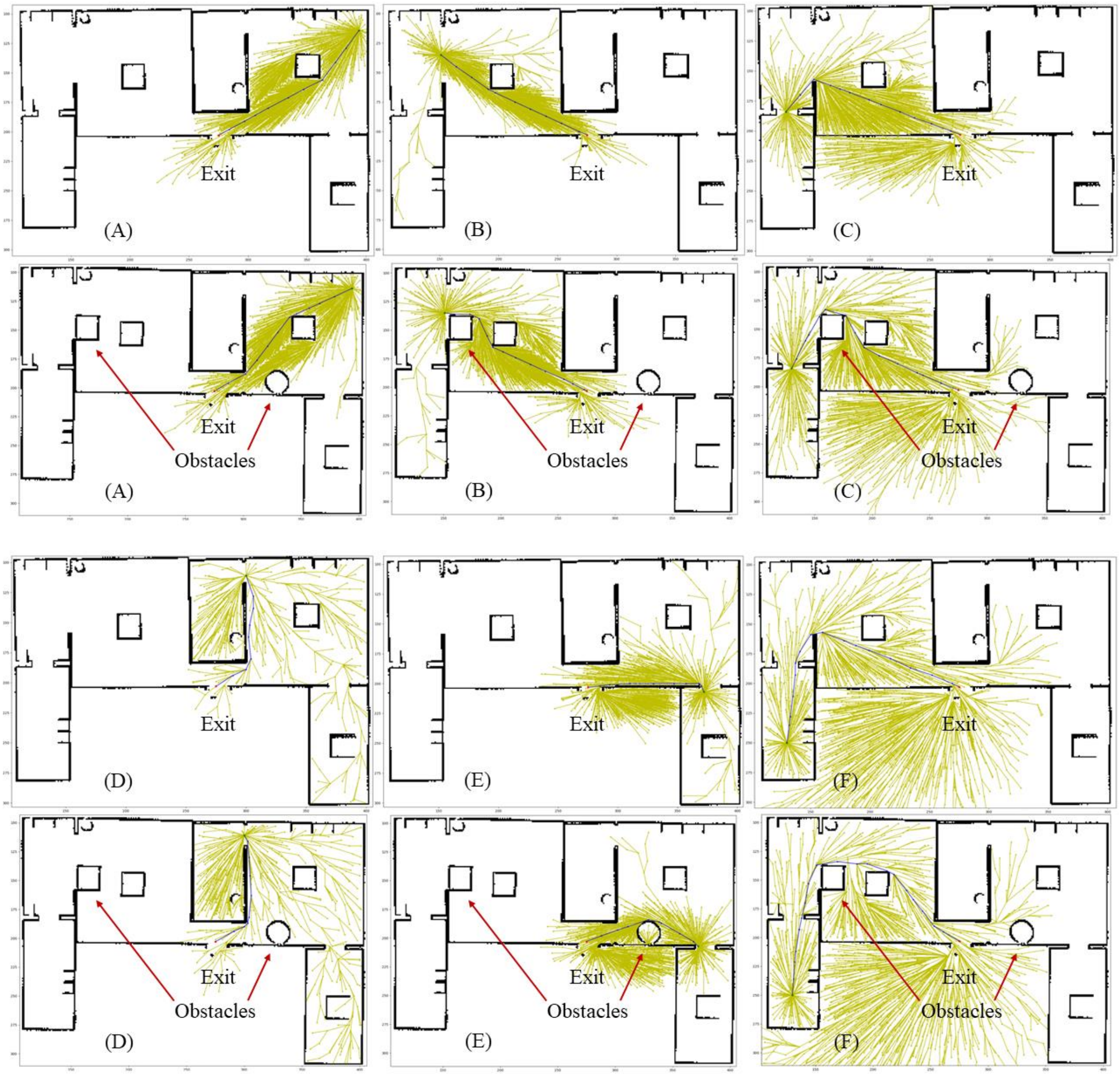


Figure 13: Informed RRT* Implementation with multiple obstacles: Case-1

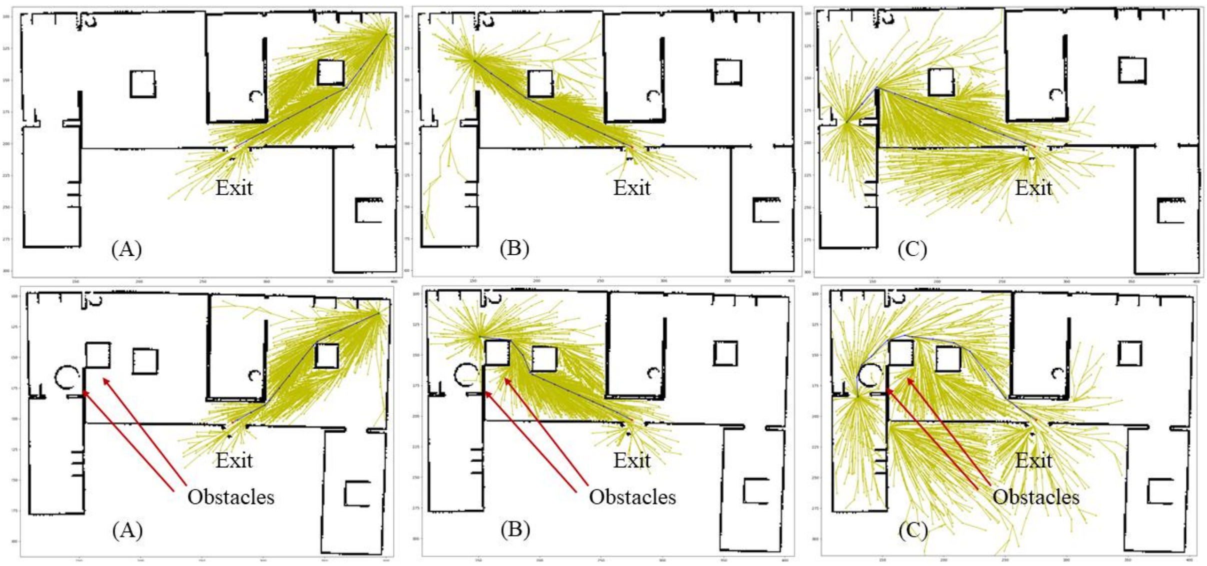
Room	Path length in the ground truth	Path length in the current Map	Percentage change in path length
Scenario A	152	152	0.00%
Scenario B	142	149	4.93%
Scenario C	165	200	21.21%
Scenario D	106	109	2.83%
Scenario E	97	107	10.31%
Scenario F	229	266	16.16%

Table 5: Path Length changes with multiple obstacles: Case-1

It is observed that informed RRT* can generate the shortest path to the exit. Even in the case of an obstacle, the algorithm can give us the shortest path. Furthermore, an increase in path length is observed when the obstacle is directly in the path of informed RRT*. Similar to the situation before, in cases A and D even though the obstacle is creating a bottleneck in the region, the path length remains unchanged.

CASE-2

In this case, two objects are placed in a very close vicinity to the entrance of the top left room.



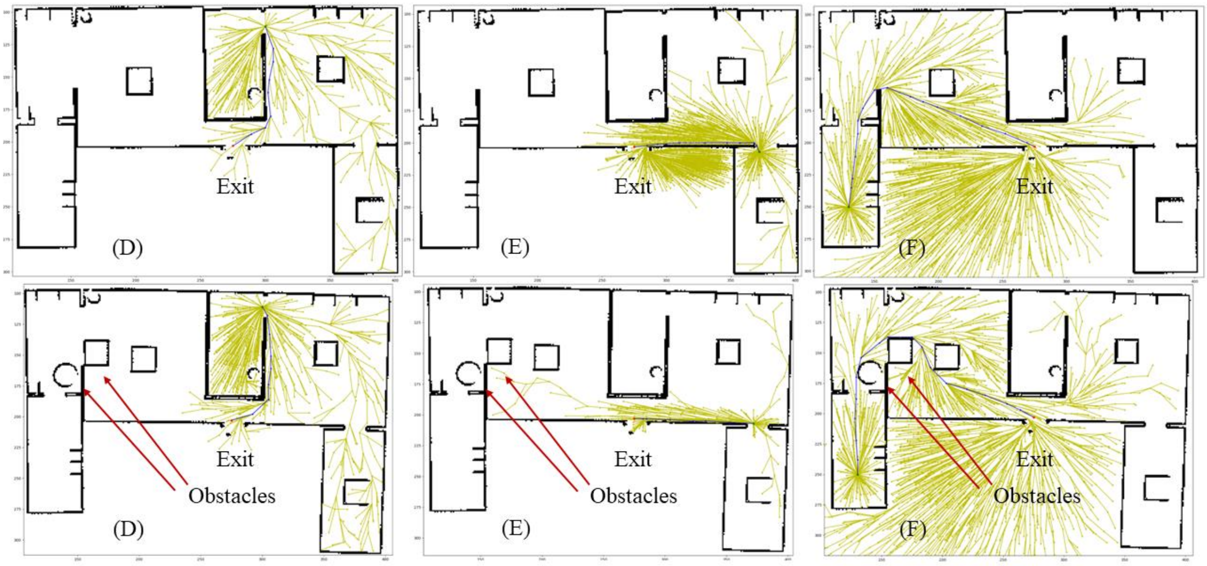


Figure 14: Informed RRT* Implementation with multiple obstacles: Case-2

Room	Path length in the ground truth	Path length in the current Map	Percentage change in path length
Scenario A	152	152	0.00%
Scenario B	142	149	4.93%
Scenario C	165	203	23.03%
Scenario D	106	106	0.00%
Scenario E	97	97	0.00%
Scenario F	229	272	19.30%

Table 6: Path Length changes with multiple obstacles: Case-2

It is observed that informed RRT* can generate the shortest path to the exit. Even in the case of an obstacle, the algorithm can give us the shortest path. Furthermore, an increase in path length is observed when the obstacle is directly in the path of informed RRT*.

CASE-3

This case is similar to case-1, but the object close to the bottom right room is moved much closer to the entrance of that room.

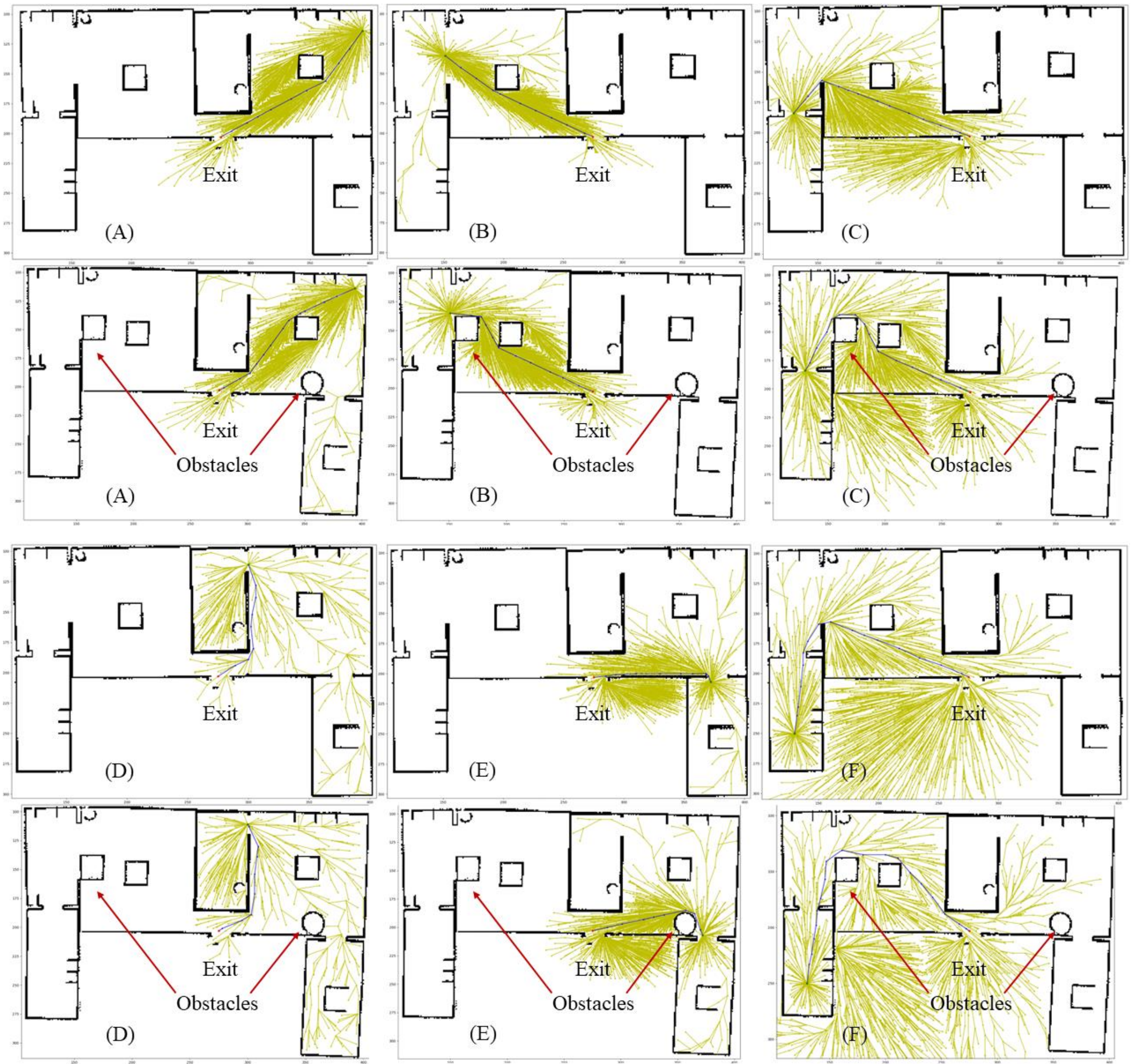


Figure 15: Informed RRT* Implementation with multiple obstacles: Case-3

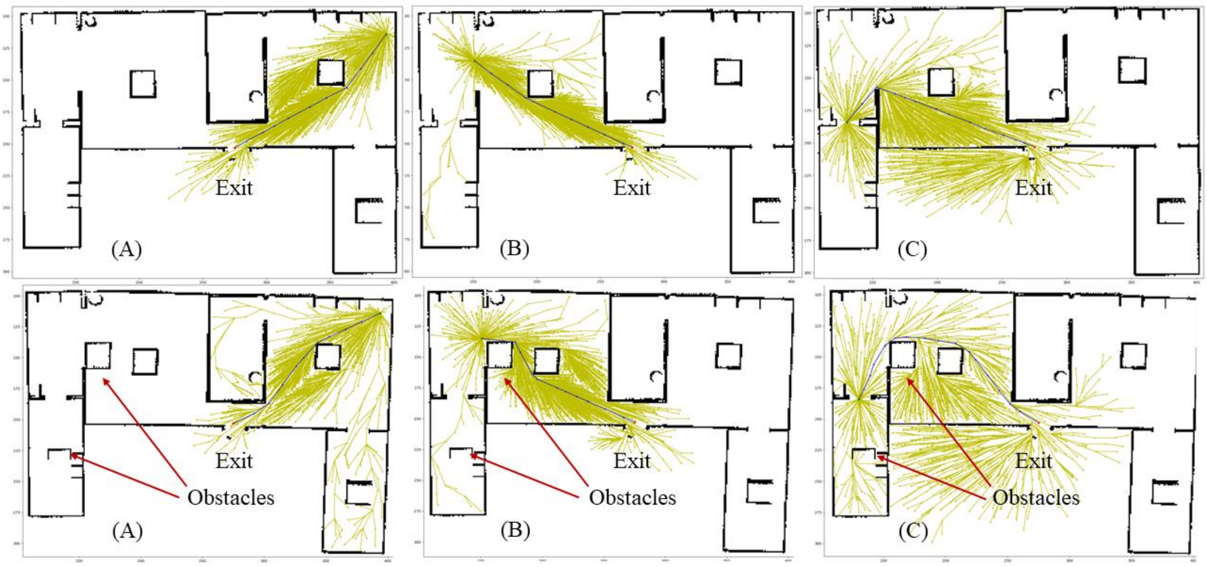
Room	Path length in the ground truth	Path length in the current Map	Percentage change in path length
Scenario A	152	152	0.00%
Scenario B	142	149	4.93%
Scenario C	165	203	23.03%
Scenario D	106	106	0.00%
Scenario E	97	107	10.31%
Scenario F	228	268	17.54%

Table 7: Path Length changes with multiple obstacles: Case-3

It is observed that informed RRT* can generate the shortest path to the exit. Even in the case of an obstacle, the algorithm can give us the shortest path. Furthermore, an increase in path length is observed when the obstacle is directly in the path of informed RRT*. But unlike in case-1, in cases A and D the obstacle is not creating a bottleneck in the region, thus a no path change is not an issue.

CASE-4:

In this case, one object is placed in the middle of the bottom left room and another object at the entrance of the top left the room.



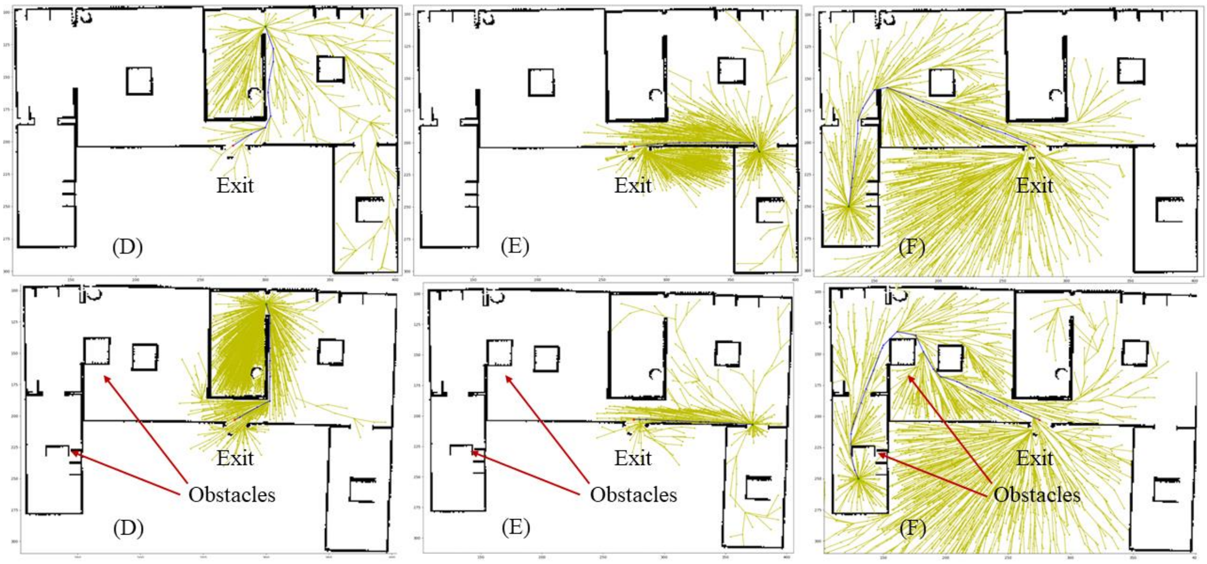


Figure 16: Informed RRT* Implementation with multiple obstacles: Case-4

Scenarios	Path length in the ground truth	Path length in the current Map	Percentage change in path length
Scenario A	152	152	0.00%
Scenario B	141	141	0.00%
Scenario C	165	166	0.61%
Scenario D	108	111	2.78%
Scenario E	97	107	10.31%
Scenario F	228	228	0.00%

Table 8: Path Length changes with multiple obstacles: Case-4

As observed previously, informed RRT* can generate the shortest path to the exit. Even in the case of an obstacle, the algorithm can give us the shortest path. Furthermore, an increase in path length is observed when the obstacle is directly in the path of informed RRT*.

Furthermore, accurate risk scoring cannot be calculated by using only the path lengths obtained by informed RRT*, and thus it needs to be used alongside another metric like gradient-based scoring.

Risk Scoring

When it comes to risk scoring, there are two major factors to be considered with the first one being the gradient score and the second being the path length score. Initially, a basic summation of the two factors was considered. So, to make the scoring robust, the following scenarios were considered.

High gradient score, Change in path length

Low gradient score, Change in path length

Scenario 1:

Figure 17: Risk scoring with a high gradient score

In the case of scenario 1, it is evident that the obstacles are located well within the gradient. So, the gradient score for the obstacles in room 1 is high. The path length is longer compared to the previous map. We can conclude that if there is an obstacle inside the gradient and the path length is long, the risk will be high

Scenario 2:

In the case of scenario 2, it is observed that the obstacle is present outside the gradient zone. It means that the gradient score of the obstacle will be zero. The issue is that the obstacle is obstructing the path to the exit. The path length is high but there is no value in the gradient cost.

Figure 18: Risk scoring with a low gradient score

From the above two scenarios, it is evident that neither the gradient costmap nor the change in path length affects the risk score individually. So, we come up with a formulation that balances the gradient cost and change in path length that results in the risk score. The formulation is shown below:

$$Risk = a * X' + (1 - a) * Y'$$

Where a is the empirically obtained weight

X' is the normalized gradient cost

Y' is the normalized change in path length

The value of X' is obtained as $X' = \frac{X}{X_{max}}$ where X_{max} is the maximum cost for each room and X is the currently obtained gradient score. Similarly, the value of Y' is obtained as $Y' = \frac{Y}{Y_{gnd}}$ where Y_{gnd} is the path length at ground truth and Y is the currently obtained path length

There is a special case wherein the value of Y' is 1 which implies that there is no change in path length. So, we consider the value of Y' in such a scenario as zero

We have three cases inferred for risk scoring as follows:

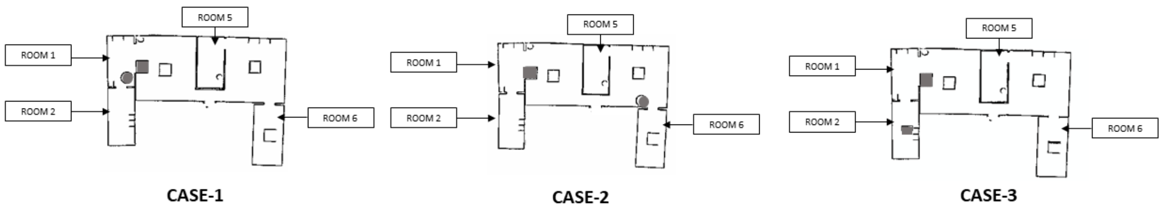


Figure 19: Risk scoring scenarios

Table 9 below depicts the risk scoring for Rooms 1,2,5,6 for the three different cases. It can be observed that in Room 1 of Case 1, the risk score percentage is at the highest as there are two obstacles in the path from room 1 which impends the faster escape scenario and it is followed by room 2 with a risk score of 80%. In case 2, we can see that the obstacles at the entrance of room 1 and 6 increases the risk score and for case 3 it is observed that there is a similar score as that of case 2 for room 1.

Risk Score	Case - 1	Case - 2	Case - 3
Room 1	85%	34%	35%
Room 2	80%	0%	13%
Room 5	2 %	1%	2%
Room 6	0 %	37%	3%

Table 9: Risk Scores Obtained for each room in each case.

The minor percentiles seen in the table are a result of noise in the change detection map generated as a result of inaccuracies in mapping in ROS. The noise in the change map is shown below in Figure 20.

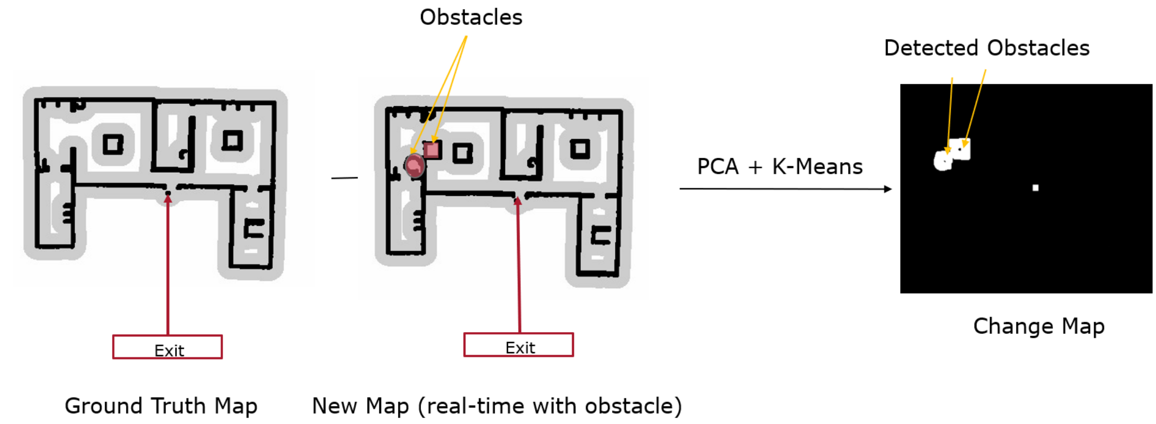


Figure 20: Change Map obtained in Case 1

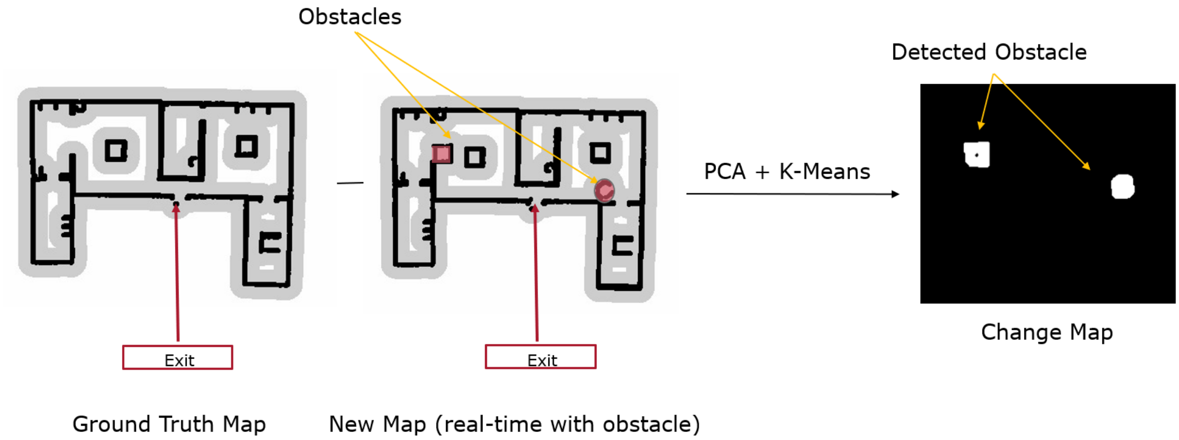


Figure 21: Change Map obtained in Case 2

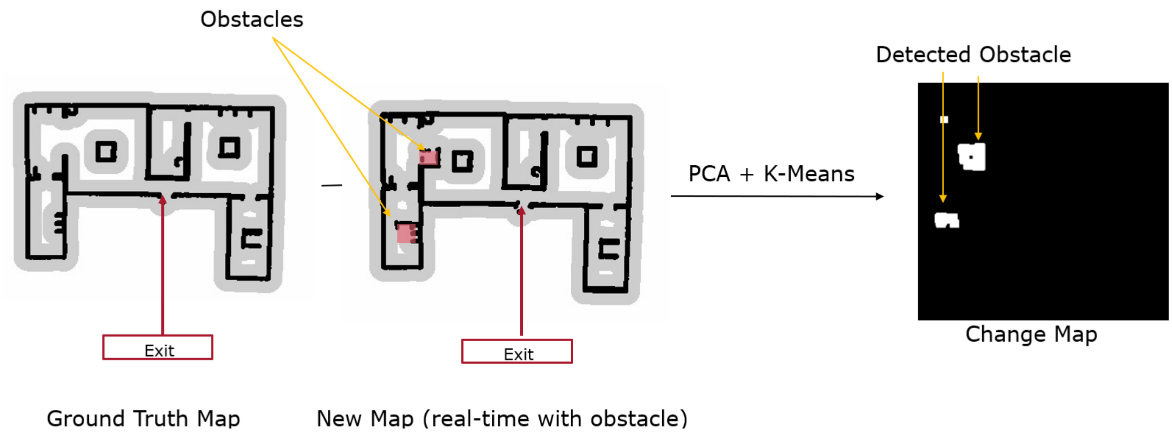


Figure 22: Change Map obtained in Case 3

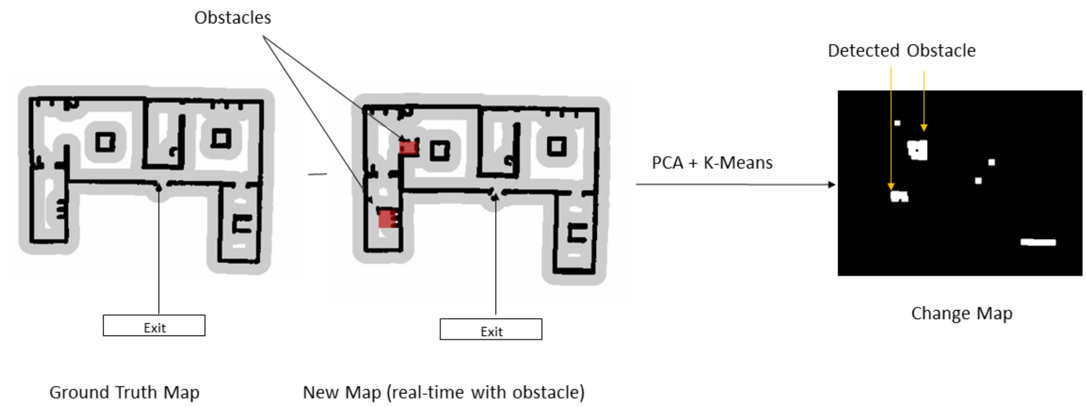


Figure 23: Noise in change detection map

In the case of using gradients on the house map, the gradients are positioned at the entrance door of each room as shown in Figure 8. But the case of using gradients at the door entrance is invalidated in the case shown in Figure 16. The case of using gradients for narrow corners has not been considered. Since gradient application at each room utilizes using a uniform-sized gradient across all rooms, specifying the same sized gradient for the corridor can result in exponentially high gradient scores which will further deprecate the risk scores obtained in rooms when it comes to normalization thereby affecting the overall scoring

Future Work

While the system demonstrated the ability to detect obstacles in the shortest path and score the object based on the risk it might pose, the system could use further development to make it more robust and reliable. The system should be tested in many more environments, increasing the data points collected and using them to improve path planning. This will also help in fine-tuning the risk score parameters to better reflect the actual risk an obstacle possesses. A future group could scale the system from 2-D mapping to 3-D mapping. This would open a wide variety of opportunities for the system to perform better. For example, the system could be programmed to identify the object and differentiate them i.e the system could differentiate between chairs and tables. This differentiation will eliminate the assumption – all obstacles are solid obstacles, that the current system works with, and can lead to better path planning. Currently, the risk-scoring system is rudimentary, and in the future, a better scoring system can be designed that uses the extra information that the testing and the new mapping yield, to reflect a much more realistic score.

Due to time constraints, the notification system was not fully implemented. With the current setup, the system does what it's supposed to do, but unfortunately, it doesn't communicate the same with the user. In the future, the push notification needs substantial work and must be developed such that it is highly reliable, as communication is a cornerstone of this project. The system currently lacks data on its effectiveness to the users. In the future, the system needs to be tested in the real world and user feedback must be collected to evaluate the system's effectiveness and further improve upon it.

Bibliography

[1] NFPA Glossary of Terms, National Fire Protection Association, 2021.

[2] S. Hall and B. Evarts, “Fire Loss in the United States During 2021,” National Fire Protection Association, September 2022.

[3] M. Ahrens and R. Maheshwari, “NFPA's "Home Structure Fires",” National Fire Protection Association, October 2021.

[4] R. F. Fahy and J. T. Petrillo, “Firefighter Fatalities in the US in 2021,” National Fire Protection Association, August 2022.

[5] I. Sakour and H. Hu, “Robot-Assisted Crowd Evacuation under Emergency Situations: A Survey,” *Robotics*, vol. 6, 2017.

[6] A. R. Wagner, “Principles of evacuation robots,” in *Living with Robots: Emerging Issues on the Psychological and Social Implications of Robotics*, Elsevier, 2019, pp. 153-164.

[7] M. Nayyar, Z. Zoloty, C. McFarland and A. R. Wagner, “Exploring the Effect of Explanations During Robot-Guided Emergency Evacuation,” in *Social Robotics*, Springer International Publishing, 2020, pp. 13-22.

[8] M. Nayyar and A. R. Wagner, “Aiding Emergency Evacuations Using Obstacle-Aware Path Clearing,” in *2021 IEEE International Conference on Advanced Robotics and Its Social Impacts (ARSO)*, 2021, pp. 7-14.

[9] E. Ferranti and N. Trigoni, “Robot-assisted discovery of evacuation routes in emergency scenarios,” in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 2824-2830.

[10] K. S. Kiangala and Z. Wang, "An Experimental Safety Response Mechanism for an Autonomous Moving Robot in a Smart Manufacturing Environment Using Q-Learning Algorithm and Speech Recognition," *Sensors,* vol. 22, 2022.

[11] B. Tang, C. Jiang, H. He and Y. Guo, "Human Mobility Modeling for Robot-Assisted Evacuation in Complex Indoor Environments," *IEEE Trans. Human-Machine Syst,* vol. 46, no. 5, pp. 694-707, 2016.

[12] H.-W. Lee, K.-O. Lee, J.-H. Bae, S.-Y. Kim and Y.-Y. Park, "Using Hybrid Algorithms of Human Detection Technique for Detecting Indoor Disaster Victims," *Computation,* vol. 10, no. 197, 2022.

[13] C. Pang, X. Zhong, H. Hu, J. Tian, X. Peng and J. Zeng, "Adaptive Obstacle Detection for Mobile Robots in Urban Environments Using Downward-Looking 2D LiDAR," *Sensors,* vol. 18, no. 6, 2018.

[14] J. Han, D. Kim, M. Lee and M. Sunwoo, "Enhanced Road Boundary and Obstacle Detection Using a Downward-Looking LIDAR Sensor," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 971-985, 2012.

[15] B. Chen, Z. Cai, Z. Xiao, J. Yu and L. Liu, "Real-time Detection of Dynamic Obstacle Using Laser Radar," in *2008 The 9th International Conference for Young Computer Scientists*, 2008, pp. 1728-1732.

[16] S. M. H. Rostami, A. K. Sangaiah, J. Wang and X. Liu, "Obstacle avoidance of mobile robots using modified artificial potential field algorithm," *EURASIP Journal on Wireless Communications and Networking,* vol. 70, 2019.

[17] R. Knox, "Amazon's Dangerous New Acquisition," The Atlantic, 21 August 2022. [Online]. Available: https://www.theatlantic.com/ideas/archive/2022/08/amazon-roomba-irobot-acquisition-monopoly/671145/.

[18] S. Karaman and E. Frazzoli, “Sampling-based Algorithms for Optimal Motion Planning,” *CoRR,* vol. abs/1105.1186, 2011.

[19] J. D. Gammell, S. S. Srinivasa and T. D. Barfoot, “Informed RRT: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 2997-3004.

[20] T. Celik, “Unsupervised Change Detection in Satellite Images Using Principal Component Analysis and k-Means Clustering,” *IEEE Geoscience and Remote Sensing Letters,* vol. 6, pp. 772-776, 2009.

Appendix

Appendix A: Stakeholder Discussions

Chief Brian McGurl	**Oct 26, 2022:**

1. Discussion of ideas given by firefighters, that might help them
    a. A robotic system that could help the officers in primary search surveillance of drug lab situations close to the fire ground.
    b. A drone system that can keep a fire in check by deploying extinguishing agents from the top.
    c. A robotic system that could drain supply line hoses, wash them and load them up the fire engine
    d. A system that could increase the visibility through smoke in the fireground.
2. Tour detailed explanation of firemen’s gear, trucks, and the training grounds.

Chief Brian McGurl	**Oct 19, 2022:**

1. Discussion of wildfire and structural fires.
2. Discussion of standard operating procedures.
3. Discussion on various techniques used by the firefighters for search and rescue.

Prof. James Urban	**Sept 29, 2022:**

1. Information about structural fires:
    a. How many people are inside?
    b. Give information about the environment inside.
    c. Can investigate petrochemical industries.
    d. Can look at fires in ships and vessels.
2. The use of drones in an active fire is highly frowned upon by firefighters.
3. The use of drones in a wildfire will cause issues with the FAA.

Prof. Milosh Puchovsky	**Sept 16, 2022:**

1. Information about structural fires:
    a. How many lives are inside?
    b. Give information about the environment inside.
    c. Can investigate petrochemical industries.

    d. Can look at fires in ships and vessels.

2. Conveying the information to the Incident Commander
    a. The fire dept only knows the location of the fire incident from the 911 call.
    b. Do not have much information about the fire situation on site
    c. Understand the situation by asking the locals around.
    d. Do not just directly go into a building on fire without knowing anything.
    e. Their priority is saving lives if they believe there is someone inside the building
3. Research more about firefighting tactics from NFPA and international firefighter safety website
4. The behavior of the fire is very dynamic in different situations. We need to know where the fire starts, what the fuel for the fire is, and how the ventilation of the building is.
5. The biggest concern – Who do they have to rescue? Is there anyone inside (Worcester 6 incident)
6. Identification of signs of life inside the building.

Prof. Pedro Reszeka	**Sept 8, 2022:**

1. Fire squad:
    a. Fire protection teams - 2 squads.
    b. One for rescue. They go in first.
    c. The other team is the fire brigade. They come in trucks and set up hoses
2. What kills firefighters?
    a. Smoke, heat
    b. Fire phenomenon types
        i. Flashover: Fire grows out of control suddenly. It is quite common.
        ii. Backdraft: Less common phenomenon.
3. Warehouse buildings go down quickly.
4. Saving people is the topmost priority.
5. From your experience, where can robots be used?
    a. “Search and rescue would be something. The search part should be quick. The robot should be deployed quickly.”
6. Firefighting is like a military operation. There is an incident commander and strategies.

7. Wildfires:
    a. For wildfires that go on for hours or days, robots can be deployed. (He does not have experience in it).
    b. Look for wildfire tactics online.
    c. Heat map of the area.
    d. Could help with predicting the direction of the fire.
    e. Intelligence for people on the ground. Situational awareness.
8. Structural fires:
    a. Initial stages of fire, there are two layers of air. Smoke goes down. Fresh air is up.
    b. There will be connection issues in the building. If buildings are made of reinforced concrete or steel, it hampers communications.
    c. High-rise building fires.
9. Tunnel fires:
    a. Fires and tunnels are complicated.
    b. Need to have a better idea of what is burning.
    c. Key features:
        i. One vital component when focusing on robots in firefighting is working in real-time.
        ii. Retrieving and feeding information.
        iii. Passing information to the incident commander so that he will investigate the bigger picture.
        iv. Situational awareness.

Professor William Michealson	**Sept 8, 2022:**

1. Describe project ideas, in what situations you think would be useful, and how they can aid firefighters.
2. Will need more help with larger and taller buildings compared to the smaller residential buildings due to my experience with the layouts.
3. The IC will have information about the fire site from various sources.
4. How can we add something to this aspect?
5. Saving lives is the topmost priority and then everything follows.
6. Anything that can save time is useful, even if it is just 5 seconds.

7. Firefighters must reach the fire site within a certain time, so what can we do to help them save time?
8. Robots should not interfere with their routine, instead, they should provide more time (easily deployable that is in seconds).
9. Research more about hazard situations for firefighters and about how robots are used in firefighting.
10. MQP project had many more neat ideas, but they could not implement them due to time constraints and proper testing was still lacking.