Warehouse Automation in Dark

by

Chaithanya Krishna Bodduluri

Harshavardhan Dammalapati

Ravi Teja Alapati

A Capstone Project Report

In partial fulfillment of the requirements for the

Degree of Master of Science in

Robotics Engineering

Worcester Polytechnic Institute

Worcester, Massachusetts

ABSTRACT

An autonomous navigation system for mobile robots can be used for multiple purposes like delivery, cleaning, surveillance, and so on in industrial areas, storage warehouses, etc. The existing end-to-end navigation solutions are mainly based on cameras where processing 3D data requires large memory and computation costs. Also, the cost of these sampling-based methods makes it very challenging to estimate the model's uncertainty. Apart from these limitations, the warehouses would need to be illuminated all the time for the visual cameras to capture the data. To conserve the electrical energy used for lighting the warehouses, attention must be given to energy usage in the commercial sector. According to our calculations, a company with about 10 warehouses of an average size of 65000 sqft. each would cost that company approximately $4 million annually. We propose fully automated dark or lights-out warehouses with multiple mobile robots that can function without any human assistance. We intend to perceive the environment with a 32-line LiDAR installed on top of the mobile robot. The proposed system is composed of two components: LiDAR-based mapping and localization module with obstacle detection and path planning with trajectory tracking module.

CONTENTS

Abstract **2**

List of Figures **5**

1 Introduction **7**

1.1 Related Work / Literature Review 9

2 Objective **15**

3 Methods **16**

3.1 Stakeholder Identification 16

3.2 Stakeholder Needs 18

4 Implementation **19**

4.1 Robot and Warehouse Design 19

4.2 Initial Map Generation 21

4.3 Path Planning 22

4.3.1 Introduction 22

4.3.2 PRM 23

4.3.3 RRT 24

4.3.4 RRT* 25

4.3.5 Results (Scenario 1) 25

4.3.6 Dilation of the Map 29

4.3.7 Results (Scenario 2) 30

4.3.8 Comparison with Amazon Robotics Warehouse 34
4.3.9 Better Sampling 35
4.4 Motion Planning (Wheel Odometer and IMU Fused) 36
4.5 Motion Planning with FLOAM 37
4.6 Multi Robot Navigation with Dynamic Collision Avoidance 39

5 Results 39
5.1 Analysis 39
5.2 Single Robot Environment 41
5.3 Double Robot Environment 43
5.4 Future Work 46

6 Conclusion 47

References 48

LIST OF FIGURES

1.1 Germany Power Crisis 8
1.2 Annual Energy Consumption in Warehouse Systems (kWh/year) 9
1.3 Comparison of different localisation approaches on the KITTI dataset . . . 13

4.1 ROS Simulated Warehouse Environment 20
4.2 Mobile Robot Design 21
4.3 Initial Binary Map 22
4.4 Uniform Sampling 26
4.5 Random Sampling 26
4.6 Gaussian Sampling 27
4.7 RRT 27
4.8 RRT* 28
4.9 Dilating with a 12x12 kernel 29
4.10 Uniform Sampling with Dilation 30
4.11 Gaussian Sampling with Dilation 30
4.12 RRT with Dilation 31
4.13 RRT* with Dilation 31
4.14 Manually spaced Uniform Sampling 33
4.15 Manually spaced Uniform Sampling with Dilation 33
4.16 Amazon Warehouse Scenario 34
4.17 Nodes around the Shelves 35
4.18 EKF Output 37
4.19 FLOAM in our Warehouse Environment 38

5.1 Comparison Metrics 40
5.2 Robot at the starting position 41
5.3 Robot picks up a shelf 41
5.4 Robot reaches the goal position 42
5.5 Robot drops the shelf and backs away 42
5.6 Two robots at their start positions 43
5.7 Collision Avoidance: One robot waits for the other robot to pass 43
5.8 One robot reaches its designated shelf 44
5.9 Both robots pick up their designated shelfs 44
5.10 Collision Avoidance with Shelves 45
5.11 Both robots reach their respective goal positions 45

Chapter 1

Introduction

With the increase in human resource costs and the acceleration of industrial automation, more and more urgent demands are placed on mobile robots in many aspects such as logistics delivery, road cleaning, security monitoring, and automatic driving. Autonomous navigation is the basis of all upper-layer applications as well as the core functional module of mobile robots. Mobile robots in a warehouse environment are increasing as the number of warehouses grows drastically. The increase is seen because some of the tasks are purely performed by robots which are like lifting heavy objects, monotonous daily tasks, etc. which are better performed by robots than humans. Due to this very reason, companies and industries are trying to employ robots to reduce labor costs. Of all these, warehouse robots play a prime role due to the large work area of warehouses. Robots in warehouse environments reduce the effort of walking humans over long distances periodically. Given the size of the warehouses, this poses different health-related concerns to humans. Now that the potential applications of robots in a warehouse environment are known, the next factor we decided to concentrate on is the power consumption in a warehouse. It is a known fact that the consumption of power for operating a warehouse is very high. This tends to increase the day-to-day operating costs. Recently we have seen an energy crisis in countries like Germany (Fig. 1.1) and many others.

Figure 1.1: Germany Power Crisis

Source:https://www.washingtonpost.com/business/energy/europes-energy-crisis-will-cost-you-200-billion-probably-more/2022/07/18/4b8362a0-0657-11ed-80b6-43f2bfcc6662_story.html

In the USA, approximately 38% of all primary energy resources are used to produce electricity [1]. Some warehouses are operating under partial-lit conditions, like Amazon, where QR codes are used as landmarks to provide a global pose reference for indoor mobile robots [2]. But this method requires pre-posting the QR code in the environment and the marking cost for the company is also reasonably high. Factoring in these electricity costs, the future projected costs for electricity will grow exponentially. It's high time that industries need to concentrate on optimizing energy costs to prevent bankruptcies. Taking all these into account, we focused on optimizing one prominent part of energy consumption through lighting. We took the concept of automating dark warehouses through mobile robots by utilizing procedures that help them navigate and perform tasks in a light-restricted environment thereby reducing overall energy consumption by lighting the warehouses.

When it comes to automation in a warehouse environment, there are many things that can be automated. It can be as simple as conveyor belts or forklifts to sorting machines and all. But we decided to focus on the automation of mobile robots, which can carry heavy and big shelves from one location to another location. We took the concept of automating dark warehouses through robots by utilizing procedures that help them navigate and perform tasks in a light-restricted environment thereby reducing overall energy consumption by lighting the warehouses. The concept of automating dark warehouses through robots by utilizing procedures that help them navigate and perform tasks in a light-restricted environment thereby reducing overall energy consumption through lighting the warehouses is what we focused on.

1.1 Related Works / Literature Review:

In [23] the energy consumption of various parameters in warehouses are categorized based on a typical warehouse area, automation, and operative costs as shown in the table, and it is clear that the energy consumption for lighting cost around 300,000 Dollars for an average of 65,000 Sq.ft warehouses.

Parameters	V1	V2	V3	V4	V5	V6
Transport equipment	151,732	77,837	131,287	184,971	100,339	134,982
Heating and cooling	1,249,281	901,726	390,657	390,657	294,744	294,744
Ventilation	46,652	33,673	14,638	14,638	9 043	9 043
Lighting	244,905	176,772	76,846	76,846	23,735	23,735
IT network and equipment	23,326	16,836	7319	7319	4521	4521
Other	209,922	151,521	65,869	65,869	40,690	40,690
Total energy consumption	1,925,818	1,358,365	686,616	740,300	473,072	507,715
Automation index	36%	35%	47%	59%	71%	83%
Energy generated by photovoltaic system	1,147,140	828,180	360,000	360,000	222,480	222,480

Figure 1.2: Annual Energy Consumption in Warehouse Systems (kWh/year)

The inventories at a warehouse might be automated, according to the authors of [3], by using a heterogeneous team made up of a UGV and a UAV. The UGV transports the UAV to a rack of merchandise, where it takes off and scans the barcodes of the items on the rack before landing on the UGV once more and moving to the following rack.

In [4], a group of autonomous UAVs is used to check the Virgin Mary church in Sternberk, Czech Republic, and the Saint Nicholas church in Prague using lighting techniques using a camera and various light sources carried by a self-stabilized group of UAVs. While taking into consideration motion and formation limitations, the angles and ranges of light sources are optimized using predictive control techniques.

The authors of [9] describe autonomous localization and mapping for UAVs in GNSS-denied situations. By matching successive scans, they estimate the position using IMU and LIDAR readings while also creating a map. A specialized ICP algorithm compares the most recent laser scan to the one before it to determine where the UAV is at the moment. Principal Component Analysis (PCA) was used to create the map. To fit line segments to clusters in the laser scan, the approach employs spectral decomposition of their covariance matrices. The gradually constructed map does not correct the drift-prone sequential localization. Because it assumes that the geometry of the environment consists entirely of line characteristics, the map cannot be scaled to any environment. Additionally, experimental verification is carried out after recorded datasets have been processed, which voids the promise of real-time performance.

In [10], an all-inclusive control, localization, navigation, and mapping solution for indoor UAVs are presented. The controller takes state estimations from a linear Kalman filter that combines accelerations from the IMU with velocities calculated from the optical flow of a camera pointed downward. However, because the Kalman filter does not fuse any measurements of absolute position, the location estimate may vary over time. The authors use reactive path planning using potential fields to solve this problem.

A SLAM method for autonomous navigation in woods is provided in the article [12]. After processing the LIDAR laser scan data, clustering is used, and the clusters that pass muster with a number of geometric descriptors are categorized as features that correlate to trees. A translation measurement is obtained by matching these characteristics across subsequent scans, and a Kalman filter is used to combine it with an IMU estimate. A back-end GraphSLAM [13] technique is employed to detect loop closures with enough overlap between the current feature set and the previously visited feature sets because the fused position estimation has a tendency to wander. Convincing tests using a genuine UAV platform as well as a simulated dataset support the suggested strategy. The feature-based approach is more resistant to noise and outliers than raw scan matching methods like [14]. However, the geometric-based feature detector limits the application to just surroundings with characteristics resembling trees. As a result, the method cannot be applied to a UAV that is deployed in a different environment.

In order to collect 3D data, a combination of 2D LIDAR and IMU mounted on a spring is proposed in [15]. The oscillations of the springs alter the scan plane of the sensor. It is a less expensive option to multi-channel 3D LIDARs, which use several spinning beams to scan the surroundings. However, the 0.5 kg weight is excessive when compared to the tiny UAVs taken into consideration in this work. The fluctuating mass could also cause the UAV to become unstable.

In [16], a comprehensive experimental assessment of scan registration techniques for mapping applications is provided. For the purpose of being able to acquire 3D laser scans of the surroundings, the authors created a unique 3D rotating LIDAR. The Generalized ICP [18] technique improves upon the original ICP algorithm [17], which records 3D geometrical forms or points to enable planeto-plane registration. The points are not immediately registered in NDT [19] [20]. The second scan is registered into the model created from the first scan's normal distribution grid map. These algorithms' effectiveness is evaluated in terms of Mean Map Entropy (MME) [22] and Absolute Trajectory Error (ATE) [21].

The UAVs are frequently located using an image feed from one or more onboard cameras when they are operating in GNSS-denied environments (such as interior settings, near large buildings, woods, mines, or caverns). In particular, visual simultaneous localization reduces pose error during loop closure and resistance to lost feature tracking, respectively, spatial localization and mapping (SLAM) [5] and visual-inertial odometry (VIO) [6] approaches have drawn a lot of interest. However, because the sensor's dynamic range is constrained, cameras need good lighting conditions to operate at their best [7]. The worst-case possibilities for localization with cameras are:

1) Underexposure of the entire photograph is caused by dark regions with insufficient lighting, such as churches, ruins, mines, tunnels, or any activity carried out at night.

2) Light source directed straight into the camera, which can occur in structures with windows or strong illumination but is more common during outdoor operations when horizontally placed cameras are oriented into the sun.

3) High contrast scenarios, which frequently occur in forests,where parts of the image are overexposed and others are underexposed.

Structured light or Time-of-Flight cameras can be used to solve the issue of inadequate lighting. These cameras are able to locate objects in complete darkness, but at the penalty of lower resolution and range, and because of sunlight, most of them are rendered useless in outdoor applications [8].

In [24] there has been a comparison on Lidar based SLAM and it was clear that the F-LOAM method is efficient in 3 aspects as shown in table.

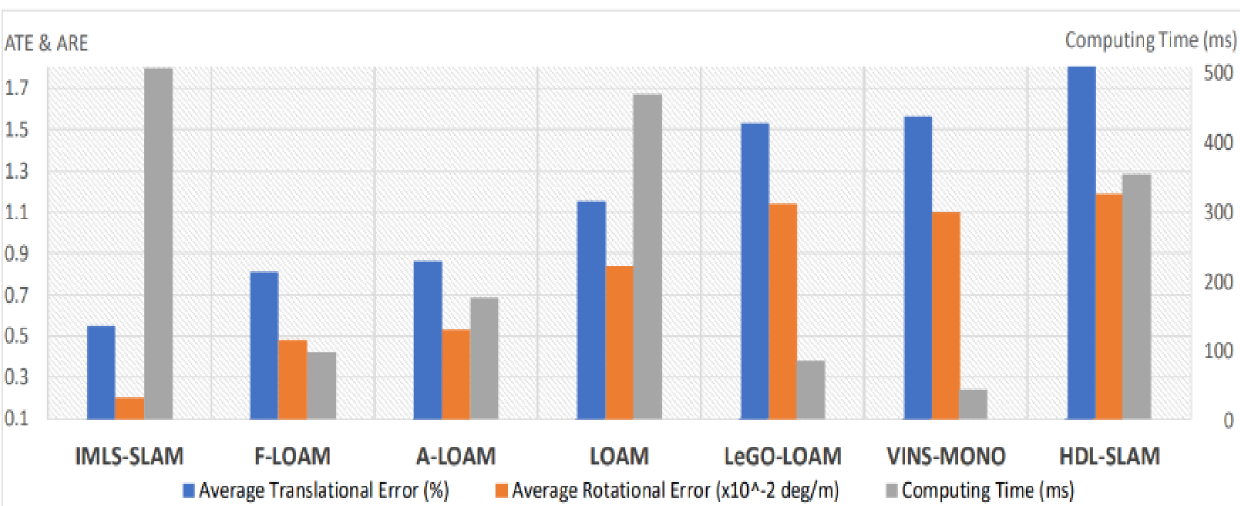

Figure 1.3: Comparison of different localisation approaches on the KITTI dataset

There has been a significant amount of research and development in the field of robotics and automation for use in warehouses. One key area of research has been the development of autonomous mobile robots (AMRs) for use in warehouses. These robots are equipped with sensors, processors, and algorithms that allow them to navigate around a warehouse and perform tasks such as picking, sorting, and transporting goods. AMRs are typically used to handle tasks that are repetitive, physically demanding, or require precise accuracy, and they can operate around the clock without the need for breaks or rest.

Another area of research has been the use of robotic arms and manipulators for tasks such as sorting, packing, and palletizing goods. These robots are typically mounted on a stationary platform and are equipped with advanced sensing and control systems that allow them to perform tasks with high precision and efficiency. In addition to these physical robots, there has also been research into the use of software and automation technologies for optimizing warehouse operations. This includes the use of artificial intelligence (AI) and machine learning algorithms to improve the efficiency and accuracy of tasks such as demand forecasting, inventory management, and routing. Overall, the use of robotics and automation in warehouses has the potential to improve efficiency, reduce costs, and increase the speed and accuracy of operations. However, it is important for companies to carefully consider the costs and benefits of implementing these technologies, as well as the potential impacts on employment and the need for worker retraining.

Chapter 2

Objective

In a typical warehouse scenario, there are many things one can automate. There is a wide range of tools ranging from simple sorting machines to complex forklifts and logistic machines. We particularly focused on developing autonomous mobile robots (AMRs), whose functionalities in the project are restricted to identifying and lifting shelves and taking them to designated locations in the warehouse. Additionally, we extended this key functionality by performing multi-robot navigation with obstacle avoidance in a light-restricted warehouse environment, where multiple robots can carry shelves to designated locations simultaneously without hindering the work of other robots. Finally, we have compared the performance of the system with various path planning algorithms and also the FLOAM-based navigation with the regular vision-based navigation.

Chapter 3

Methods

Any project that is to be carried out starts with a literature review of the existing methods and principles to get a better idea of what already exists in the domain. After the literature review, the next important part of our method is stakeholder identification. This step influenced our objectives and initial design of the system and environment to conduct our experiments. Initial prototyping and testing come next. Finally, we concentrated on expanding the system to incorporate more use cases and tested it for the changes that occurred in the progress.

3.1 Stakeholder Identification

The stakeholders we identified were categorized into four domains: Medical, Military, Retail and Autonomous Driving. The medical and military are mostly generic applications where some specific tasks are impossible for humans to perform, and the only way left is to automate machines to perform them.

Automated systems can assist with the safe and accurate administration of medications, reducing the risk of errors and improving patient safety. Robotic systems can assist with surgical procedures, increasing precision and reducing the risk of complications. Robotic automation is used in military applications in areas such as logistics, transportation, and bomb disposal. One example is the use of unmanned aerial vehicles (UAVs), for a variety of purposes, including surveillance, reconnaissance, and target identification. Other examples of military robotics include ground-based robots that can be used for tasks such as bomb disposal or explosive ordnance disposal (EOD), as well as underwater robots that can be used for mine clearance or underwater search and rescue, reducing the risk to human personnel and increasing efficiency.

The advanced driver-assistance system (ADAS) of vehicles uses cameras, and other sensors to assist drivers in navigating and operating vehicles. Robotic automation is often used in ADAS to help improve safety, efficiency, and convenience for drivers. Some applications include automatic emergency braking, lane departure warning, adaptive cruise control, and automatic parking.

Retail sector companies use automation mainly for their inventory management and movement of goods. Some examples of how robotics can be used in inventory management include:

- Material handling: Robotic arms or other automated systems can be used to move and transport materials within a warehouse or other storage facility.
- Stocktaking: Robots can be used to scan and count inventory, reducing the time and effort required for manual stocktaking
- Order picking: Automated systems can be used to locate and retrieve items from storage for packing and shipping.
- Packing and shipping: Robots can be used to pack and label items for shipping, improving efficiency and reducing the risk of errors.

3.2 Stakeholder Needs

We made casual discussions with employees of various companies to gather information on their needs. One such company is Faraday Future which is currently developing its ADAS system for its autonomous car. They are using multiple cameras to perceive the surroundings of the car and are trying to calibrate their cameras for various lighting conditions. Situations of a sudden change in lighting exposure require a few seconds of time for the cameras to recalibrate. In this span of time, they want to make use of the Lidar information to make decisions and control the vehicle's movement.

Another company is Amazon Robotics which uses QR markers on the floor for their mobile robots to navigate in their warehouses. And for the robots to process the QR codes, their warehouses need to illuminate all the time. Our solution which makes the robots navigate with only Lidar data allows such warehouse companies to get rid of their cost for electrical energy as well as the cost of marking the floors with QR tags.

Chapter 4

Implementation

4.1 Robot and Warehouse Design

When it comes to the design of the robot, we had different thoughts and approaches. The first thing is that our robot cannot be equipped with a camera since it has to navigate in a dark environment. We started out building a physical robot adding a NOIR Raspberry PI camera to it. But things did not turn out as expected with our physical robot in the terms of sensor integration. There were many issues that came up integrating the LiDAR sensors into the robot's processor board. Therefore, as a last resort, we fell back on demonstrating our project by simulating the robot and the environment.

For the purpose of our project, we have come up with a simple but realistic warehouse world, where there are aisles for the navigation of the robot along with 24 shelves. It replicated an exact layout of a warehouse and had designated bins for placing the shelves along with some destination bins. This is a comparatively small warehouse that can accommodate a maximum of three robots. But we believed it would be sufficient enough for us to demonstrate our solution. We added the light source in our environment only for the sake of visual demonstration and is not being used by the robot.

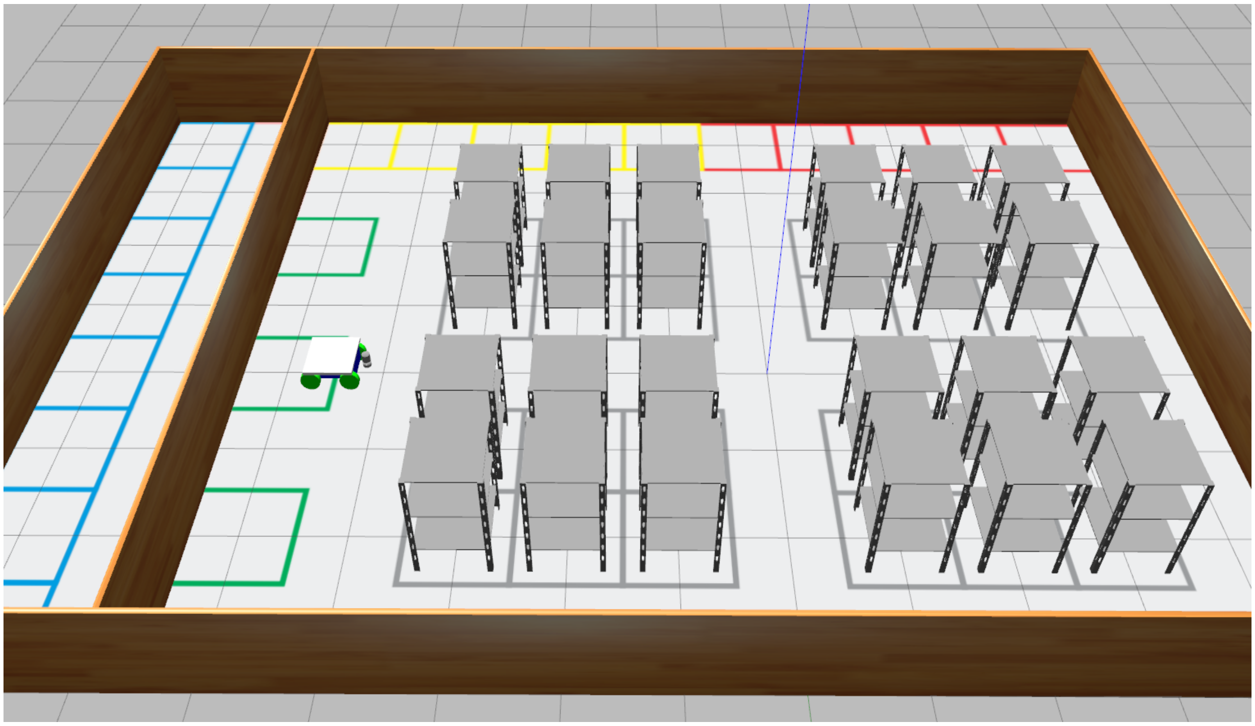

Figure 4.1: ROS Simulated Warehouse Environment

We thought of exploring various options for integrating sensors. In ROS, we did not find any package that had the option of integrating a NOIR camera. This had us realize that our robot has to depend on other sensors which do not need light, like the Lidar, IMU, wheel odometer, etc. With that said, our robot was initially equipped with a Lidar on which we implemented the FLOAM algorithm. Later IMU and Wheel odometer were also made use of while performing the task of navigation between two given locations.

With the robot ready and equipped with the sensors, we needed to modify the mobile robot a bit so that it could transport shelves from one location to another. For this purpose, a flat base/platform was added on top of the robot's body with a prismatic joint fixed underneath it. This prismatic joint can help this platform to move up and down so that it could lift the shelf while being transported and drop at the designated destination.

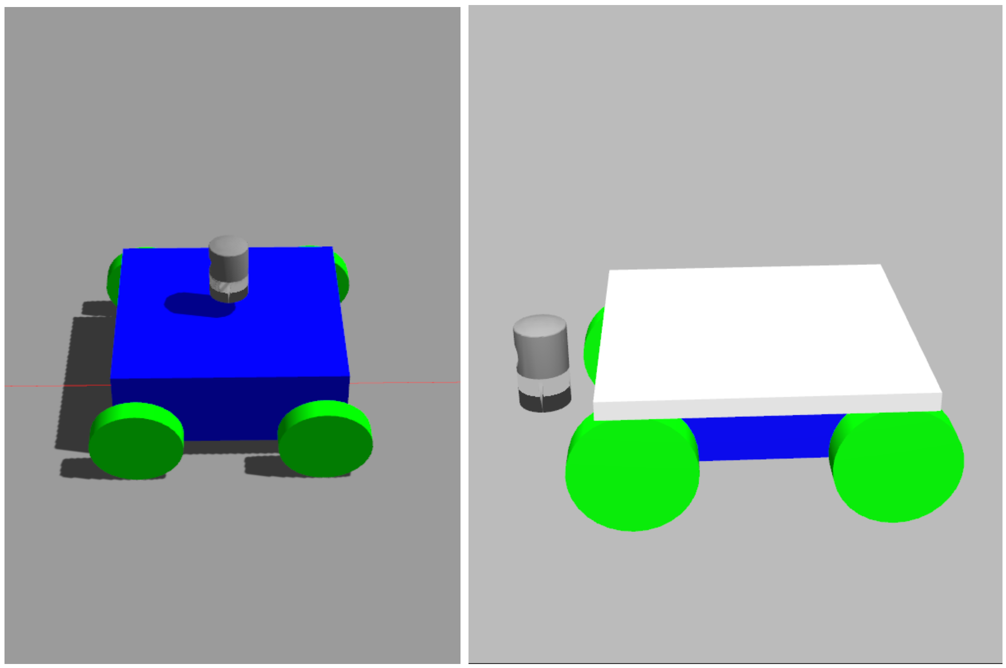

Figure 4.2: Mobile Robot Design

4.2 Initial Map Generation

Any task of making a robot autonomously navigate from one point to another or from one location to another begins with the task of map generation. A map is needed for the robot to know its environment and navigate without colliding with obstacles present in it. This pushes us to the need to use a Lidar to generate the map of the environment. For that reason in map generation, we have used Fast Lidar Odometry and Mapping (FLOAM) algorithm. We might have used any other algorithms present, but we opted for this method because we have tested this algorithm on the Kitti dataset and it was able to generate a map of the environment quickly in real-world scenarios as well. Hence, due to its better performance, we tried to utilize the same for our simulated warehouse world and got a map that any other mapping algorithms would normally give, but with reduced required time for map generation.

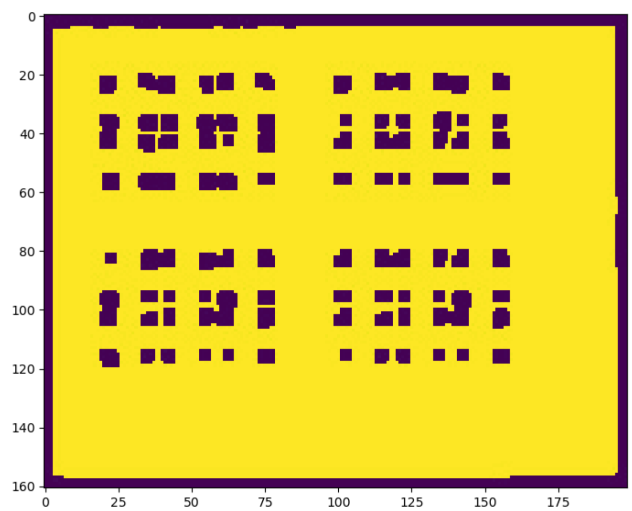

Figure 4.3: Initial Binary Map

4.3 Path Planning

4.3.1 Introduction

Once a map of the warehouse has been obtained, all the shelves in the warehouse have to be assigned a number for them to be identified. For the sake of this project, we have hard coded the shelf numbering, which seemed to be reasonable given the number of shelves in our environment is less. However, dynamic approaches can also be used which makes use of cameras that are placed near the ceiling of warehouses and map the shelves with numbers.

When it comes to planning, there are many algorithms one can rely on for planning paths between locations. For the sake of our project, we tested some algorithms, namely Probabilistic Road Maps (PRM) using random, gaussian, and uniform sampling techniques, RRT, RTT*. The results of planning from all these algorithms were compared and a suitable algorithm was selected for use.

Here's a brief description of the three planning techniques used–PRM, RRT, and RRT*.

4.3.2 PRM

Probabilistic Road Map (PRM): The PRM is constructed from a set of configurations sampled from the C-space, and it can be called a probabilistic roadmap because, as the number of samples tends to infinity, the likelihood that the graph is a true roadmap goes to 100 percent. An advantage of a PRM graph over a grid-based graph is that the structure of the free C-space is generally captured by the PRM with many fewer nodes than with a grid graph. PRMs have been used to solve complex motion planning problems in high-dimensional C-spaces.

To construct a PRM, we generate N samples of the free C-space. These free configurations can be generated by uniformly randomly sampling the C-space and only keeping the sample if it is collision-free, but non-uniform sampling strategies can also be used to increase the likelihood that the PRM is able to represent narrow passageways in the C-space with a smaller number of samples.

The N free-space configurations generated in the first phase of the algorithm are the nodes of the graph. The second phase of the algorithm tries to connect the nodes with edges. For each node, we find a set of k nearby nodes. Then, for each of these neighbor nodes, we try to find a path from the original node to the neighbor. To do this, we use a very simple and fast local path planner which does not attempt to avoid obstacles.

We then check whether this path is collision-free, and if so, we add an edge between the two nodes. At the end of this second phase of the PRM construction algorithm, we have a graph that should approximately represent the free space, depending on our choice of the number of samples N, the number of neighbors k, the sampling algorithm, and the local path planner. The choice of the sampling algorithm and the local path planner provides a lot of flexibility to customize the basic algorithm. We then can use algorithms like A* or RRT for finding a good path through the PRM.

As said above there are many sampling techniques that can be used in a PRM. Some of the popular techniques that were used here for sampling are: Uniform, Random, and Gaussian. The sample space is said to be sampled uniformly if the set of points that are sampled are uniformly spread across the space. When there's no any particular pattern in the sample, the points are said to be randomly sampled. Finally, if the points sampled follow a Gaussian distribution, we call the space to be sampled by Gaussian.

4.3.3 RRT

RRT: The way how RRT works is actually quite straightforward. Points are randomly generated and connected to the closest available node. Each time a vertex is created, a check must be made that the vertex lies outside of an obstacle. Furthermore, chaining the vertex to its closest neighbor must also avoid obstacles.

The algorithm ends when a node is generated within the goal region, or a limit is hit.

4.3.4 RRT*

RRT*: RRT* is a better version of the RRT algorithm. It makes these two changes that make it better than RRT. First, RRT* records the distance each vertex has traveled relative to its parent vertex. This is referred to as the cost of the vertex. After the closest node is found in the graph, a neighborhood of vertices in a fixed radius from the new node is examined. If a node with a cheaper cost than the proximal node is found, the cheaper node replaces the proximal node.

The second difference RRT* adds is the rewiring of the tree. After a vertex has been connected to the cheapest neighbor, the neighbors are again examined. Neighbors are checked if being rewired to the newly added vertex will make their cost decrease. If the cost does indeed decrease, the neighbor is rewired to the newly added vertex. This feature makes the path more smooth.

4.3.5 Results (Scenario 1)

Here are the planning results from all the above-said algorithms. The task given is simple and the ideal path in this case would be a straight line between the two points specified.

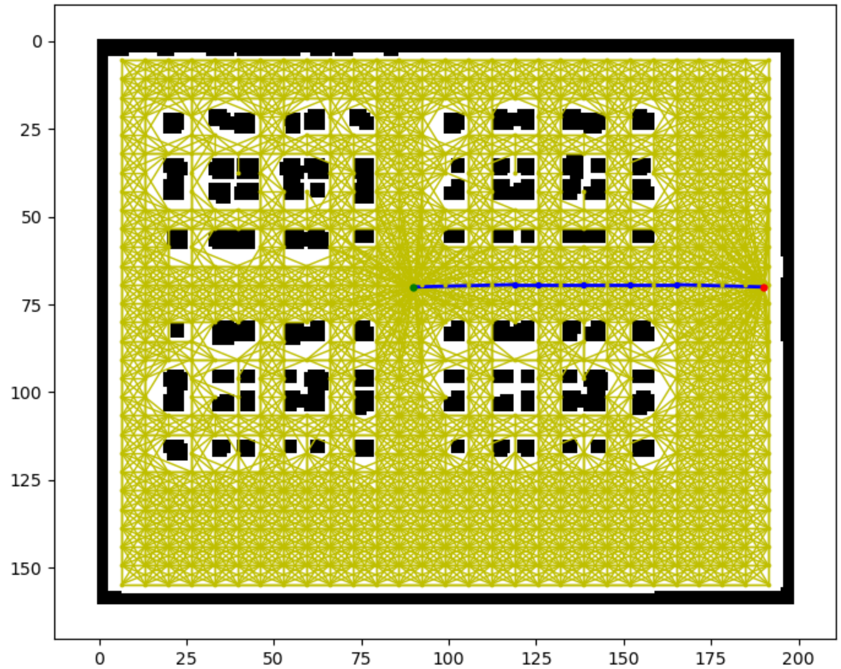

Figure 4.4: Uniform Sampling

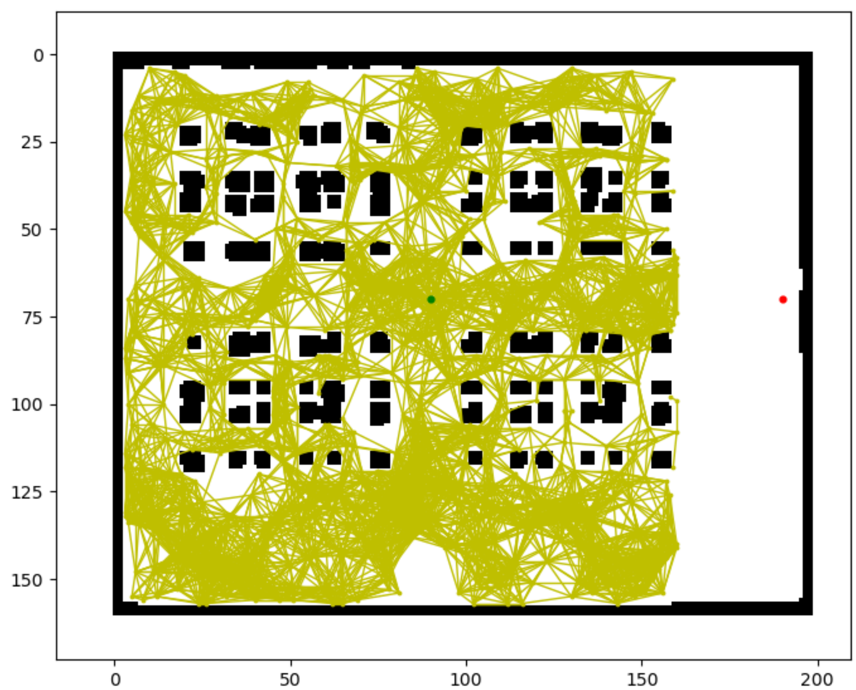

Figure 4.5: Random Sampling

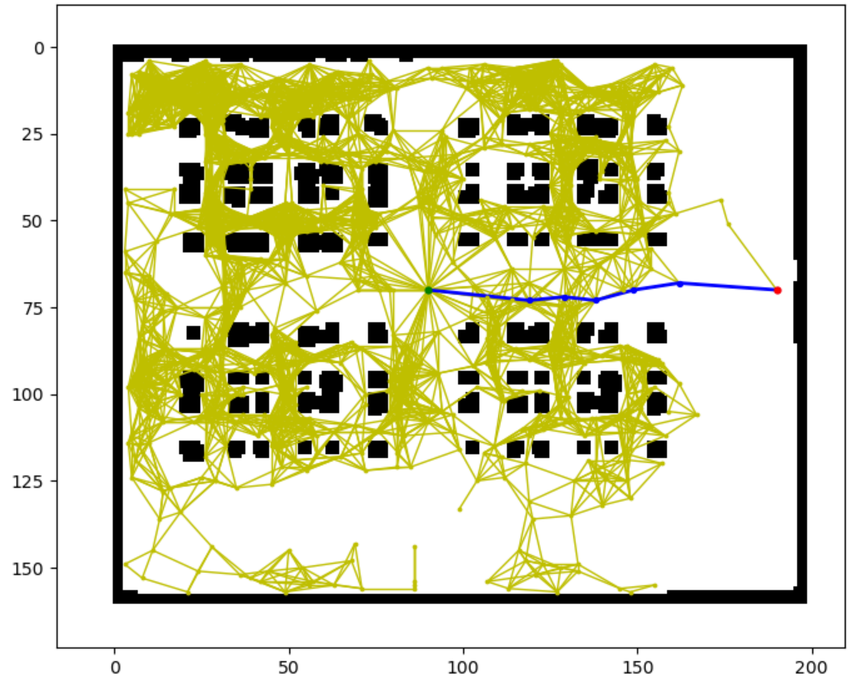

Figure 4.6: Gaussian Sampling

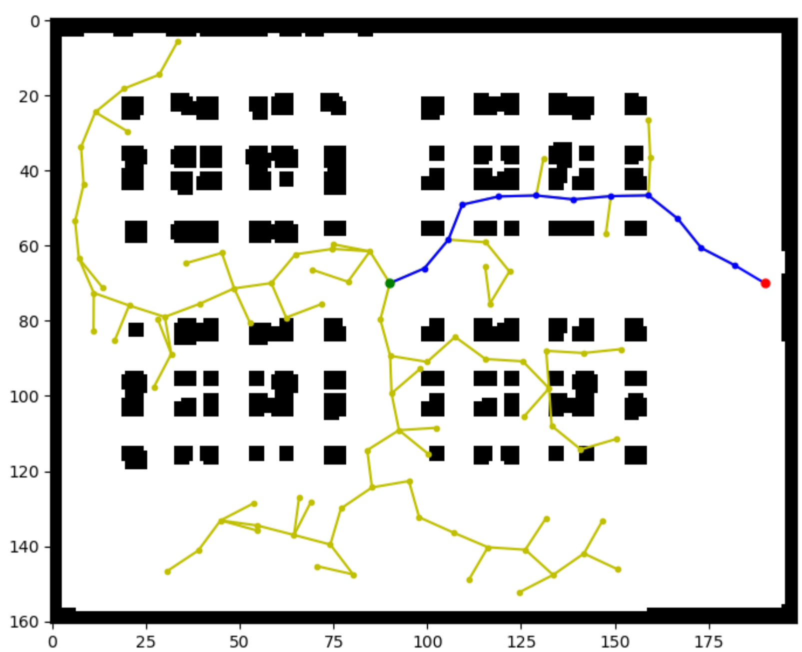

Figure 4.7: RRT

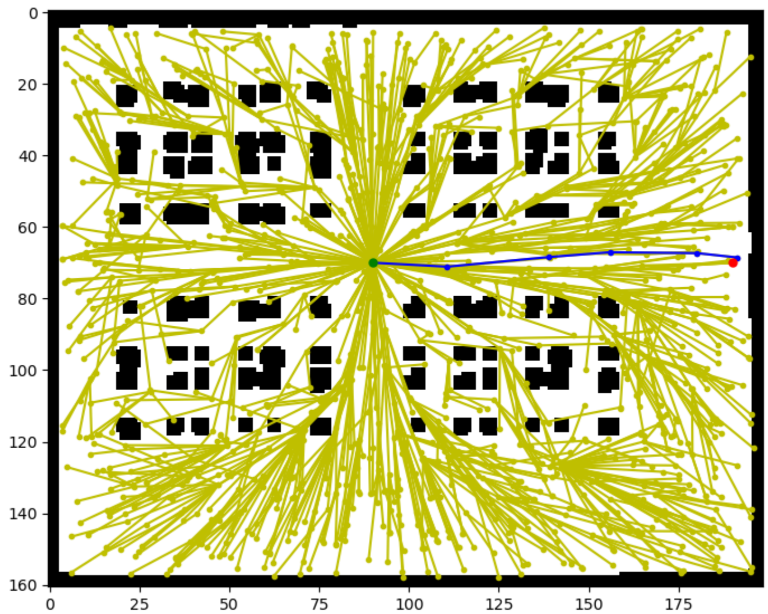

Figure 4.8: RRT*

We can visualize that PRM with Uniform sampling and RRT* gave us those ideal paths between the points. However, if we consider the case of PRM with Random sampling, there is no path generated even though it is as simple as a straight line. This puts forth the result that there is no guarantee for a path to be generated in the case of random sampling. There is however another problem that needs to be addressed. It is evident from the images above that these planning algorithms are giving out paths that go through the shelves. This is a problem because if navigation is done through spaces under the shelves, there's a high risk of collision of the robot with the shelf and since we did not model the physical/wheel constraints of the robot into the planning algorithm, problems can be encountered while the robot turns or tries to make a turn.

4.3.6 Dilation of Map

In order to solve this problem, we came up with a technique in Computer Vision called dilation. We dilated the black pixels of the initial map using a 12x12 kernel so that the gaps between/underneath the shelves are modeled as obstacles instead of free spaces and the paths that are generated are around these shelves in the aisles.

Figure 4.9: Dilating with a 12x12 kernel

Once the free spaces in the environment were modeled properly, we carried out our experimentation further and visualized some more results using these algorithms except random sampling, which turned out as follows:

4.3.7 Results (Scenario 2)

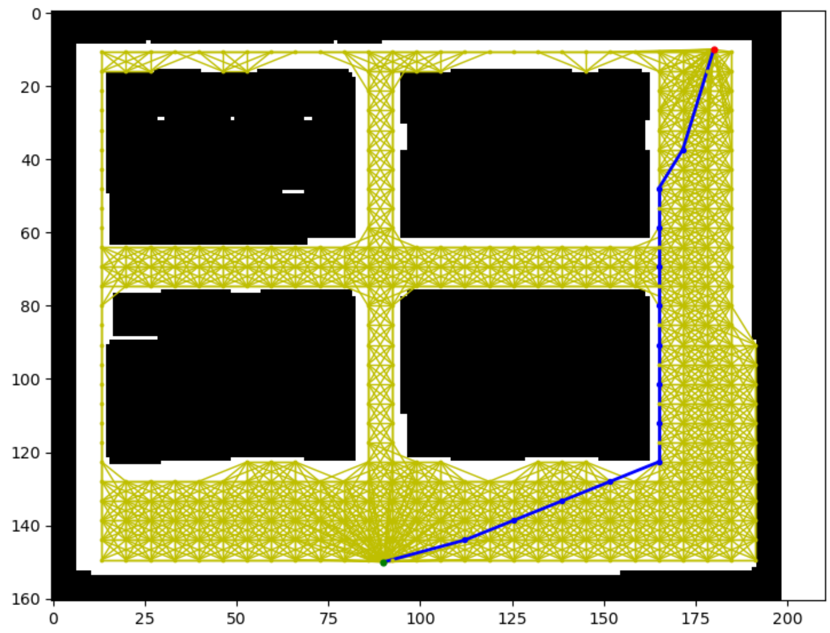

Figure 4.10: Uniform Sampling with Dilation

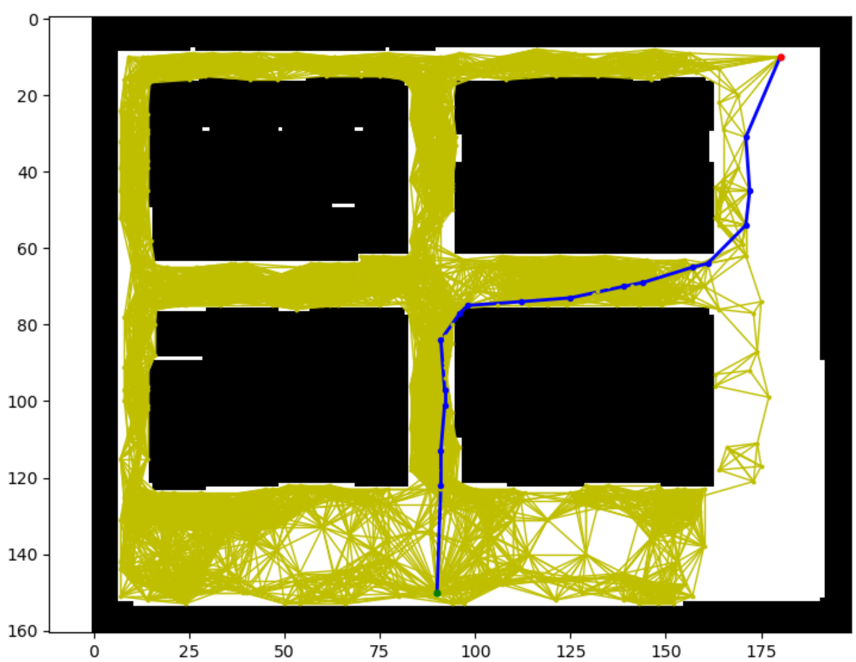

Figure 4.11: Gaussian Sampling with Dilation

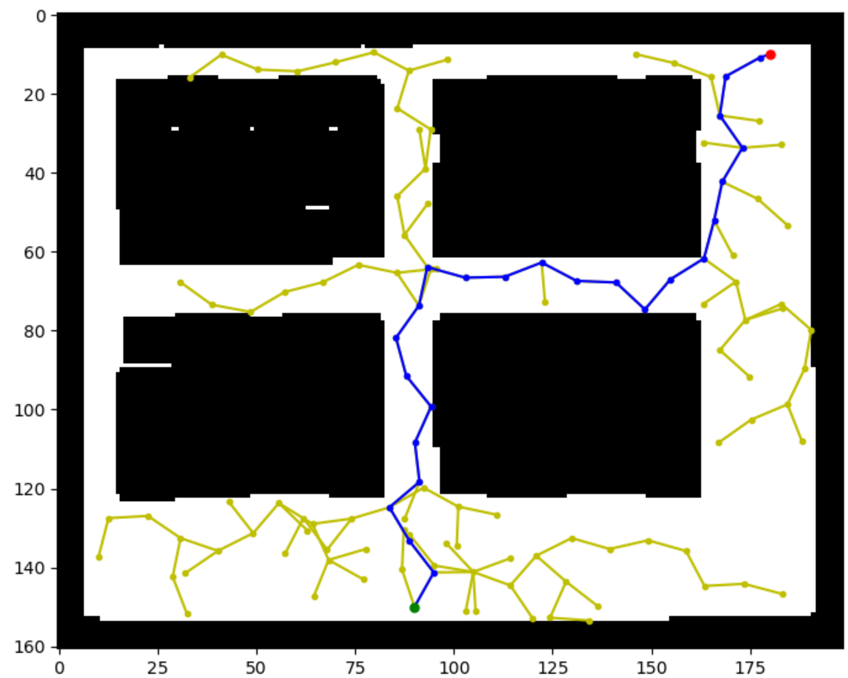

Figure 4.12: RRT with Dilation

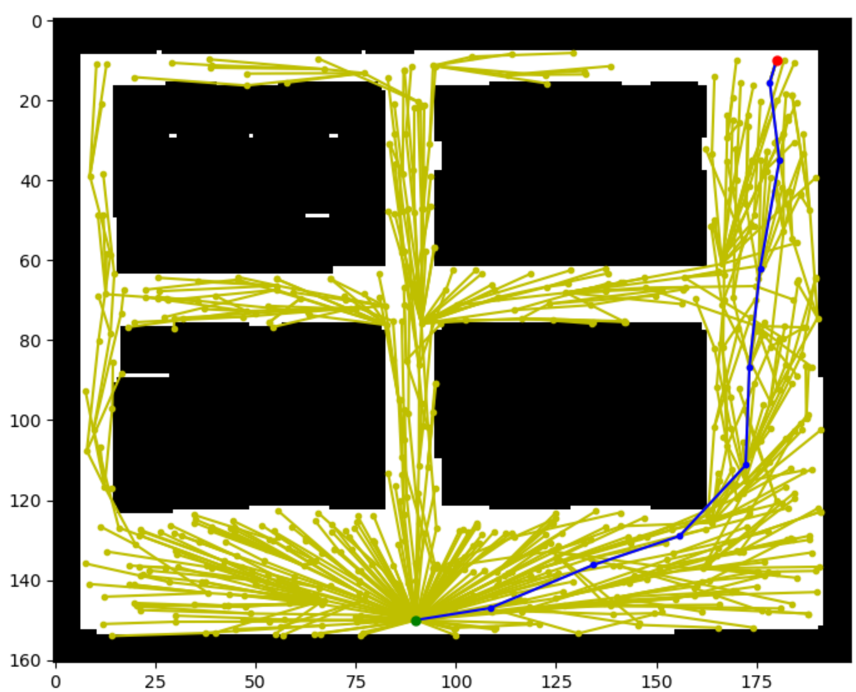

Figure 4.13: RRT* with Dilation

The problem with RRT is that the path generated by it is not smooth in any of the two above cases, which makes it difficult for the mobile robot to traverse such kind of paths. Also, the exploration done by RRT is very nominal in comparison with RRT*. RRT* is a very powerful extension of the RRT algorithm, which generates a lot better and smooth paths than RRT, with the right amount of exploration. But that kind of exploration takes a lot of time. Path generation with RRT* in our small warehouse environment takes about 1.2 seconds, but if this warehouse is scaled to be about 1 square mile to be in the area, the time taken by it will be more and requires online path planning or sophisticated onboard computation.

Finally, path planning boils down to the question of using PRM with either Gaussian or Uniform distribution, as these best are suited for us given the constraints. But we proceeded to use Uniform sampling because of the following advantages:

1) Be able to create a uniform virtual grid with it
2) Adjust the spacing of grid cells in the environment
3) Able to use just 50 sample points to generate a grid for the entire environment
4) Straight line paths

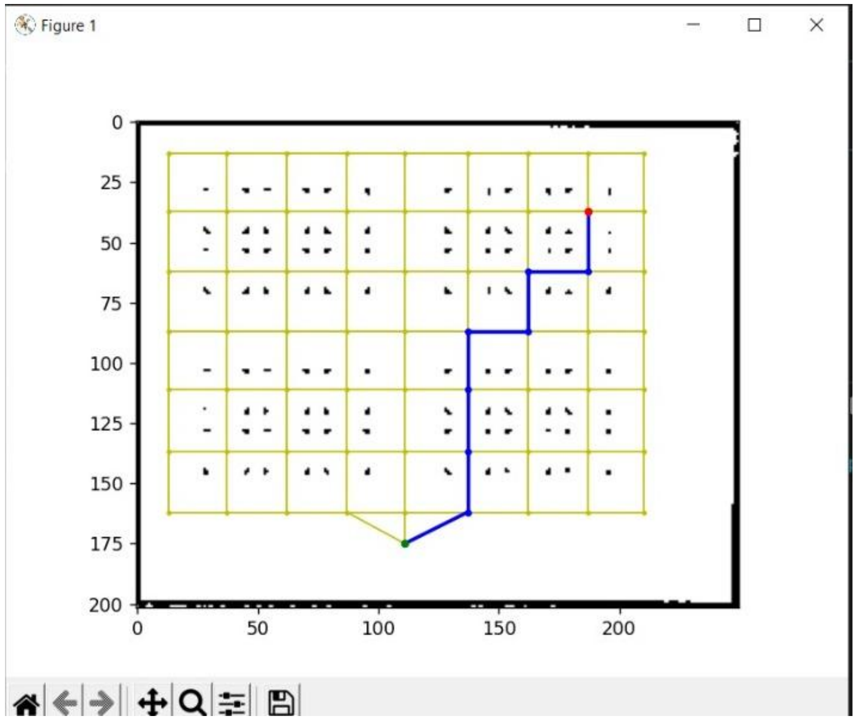

Figure 4.14: Manually spaced Uniform Sampling

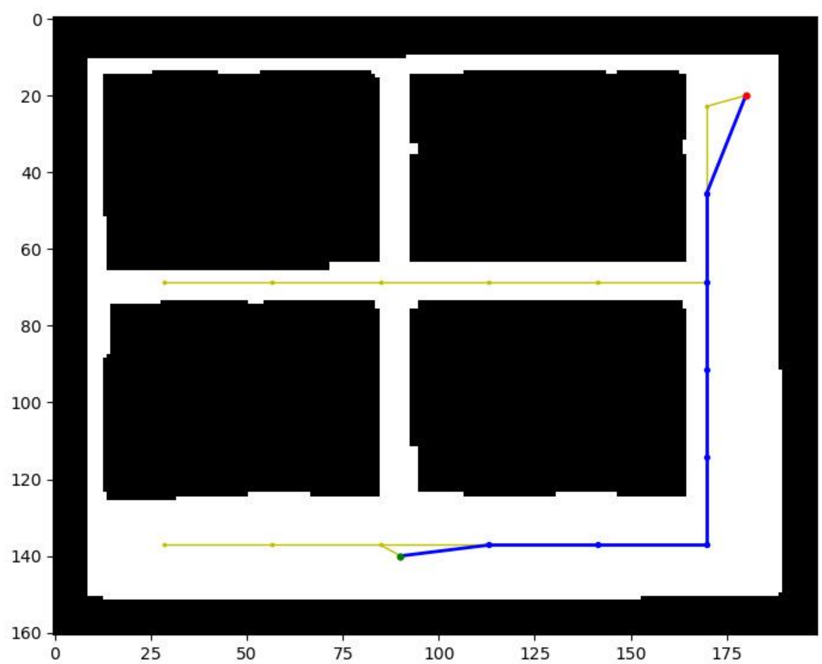

Figure 4.15: Manually spaced Uniform Sampling with Dilation

We further modified the uniform sampling algorithm to better check for collisions with the shelves. While generating paths, instead of directly generating intermediary paths between two sub-points, we added collision checking to check for collisions for up to 5 pixels on top of the prior generated path. If a collision occurs, then the path is regenerated to give a better collision-free path than the previous one. We also added the mapped shelves' locations as nodes to the map. This drastically reduced the risk of collision while navigating.

4.3.8 Comparison with Amazon Robotics Warehouse

The robots at the Amazon Robotics warehouse made use of visual markers on the floor for navigation between locations. The robots navigated in a grid-based static world where no humans would interfere with the robots or their paths. But our solution removed the use of such visual markers completely. With that being said, we also put forward a note of caution because our navigation is done in an idealistic world, where there are no errors and the navigation takes almost perfectly. But in order to account for real-world noise and errors, although the use of visual markers might not be removed completely, it can be greatly reduced and used for path corrections while navigation.

Figure 4.16: Amazon Warehouse Scenario

4.3.9 Better sampling

To further improve the time complexity and reduce the additional computation required we have sampled the nodes for uniform sampling in such a way that they are only placed in the aisles where the robot can navigate. Although this is a manual process, this reduces the number of sample points required for navigation from 50 points to about 20 points. 20 points for our whole environment is a great increase in performance. If this environment is scaled by say 20 times, or even 100 times, the sample points will not surpass 2000, which is the number of sample points we used for RRT* in our current environment.

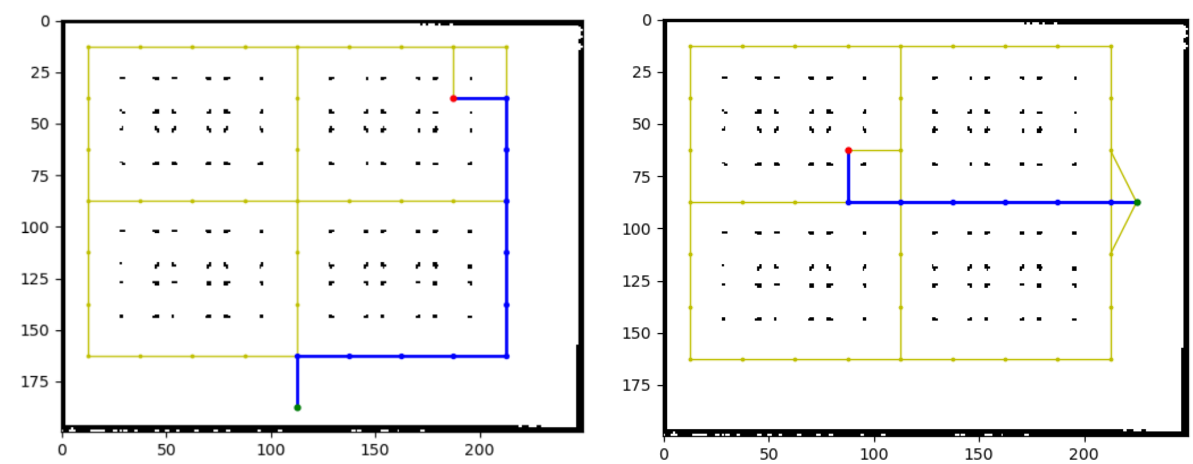

Figure 4.17: Nodes around the Shelves

Along with all these sampled nodes, which are placed in the aisles, the last node is added at the place of destination. This enables the robot to navigate from start to destination without any additional computation or changes.

4.4 Motion Planning (Wheel Odometer and IMU Fused)

Our initial implementation mostly focuses on the implementation of navigation solely based on the inputs from IMU and the wheel odometer fused with the help of an extended Kalman filter (EKF).

Although we are doing this project in a simulated environment, the real world would contain a lot of sensor noise. The sensors are never 100% accurate. Therefore, we tried to model real-world inaccuracies in our project by injecting noise to our sensor values. This made the system more realistic and much more comparable to the real-world scenario. For the purpose of this project, in order to combine the readings from both the sensors–IMU and Odometry, we have used an extended Kalman filter (EKF). While Kalman filters (KF) are based on recursive Bayesian filtering where the noise in the system is assumed Gaussian. Extended Kalman Filters are a special type of extension of the Kalman Filters which are particularly useful for non-linear systems where non-linearity is approximated using the first or second-order derivative.

The EKF computes the weighted average of the actual sensor measurements at the current time step t and predicts sensor measurements to generate a better current state estimate. The extended Kalman filter comprises 2 steps: i) Prediction step and ii) Update step.

```
Timestep k=1
State Estimate Before EKF=[4.51  0.01  0.003]
Observation=[4.721 0.143 0.006]
State Estimate After EKF=[ 4.584  0.043 -0.016]

Timestep k=2
State Estimate Before EKF=[ 9.093 -0.021 -0.013]
Observation=[9.353 0.284 0.007]
State Estimate After EKF=[ 9.208  0.121 -0.025]

Timestep k=3
State Estimate Before EKF=[13.716  0.017 -0.022]
Observation=[14.773  0.422  0.009]
State Estimate After EKF=[14.324  0.224 -0.028]

Timestep k=4
State Estimate Before EKF=[18.832  0.109 -0.025]
Observation=[18.246  0.555  0.011]
State Estimate After EKF=[18.427  0.341 -0.027]

Timestep k=5
State Estimate Before EKF=[22.935  0.228 -0.024]
Observation=[22.609  0.715  0.012]
State Estimate After EKF=[22.69   0.486 -0.027]
```

Figure 4.18: EKF Output

4.5 Motion planning with FLOAM

Lidar Odometry and Mapping (LOAM) is an excellent LiDAR-based real-time odometry and mapping technology, that constructs 3D models of the environment using a spinning single-line LiDAR.This method mainly works on two algorithms, One algorithm performs odometry at a high frequency but low fidelity to estimate the velocity of the lidar. Another algorithm runs at a frequency of an order of magnitude lower for fine matching and registration of the point cloud. The combination of the two algorithms allows the method to map in real-time. This method is later optimized and evolved with A-LOAM (Advanced implementation of LOAM), and F-LOAM (Fast LOAM) methods. Among these the F-LOAM method is less prone to translation and rotational error, and also poses less computing time compared to other lidar-based odometry methods as shown in fig. 1.3. This clears our decision to choose F-LOAM for lidar-based odometry and mapping.

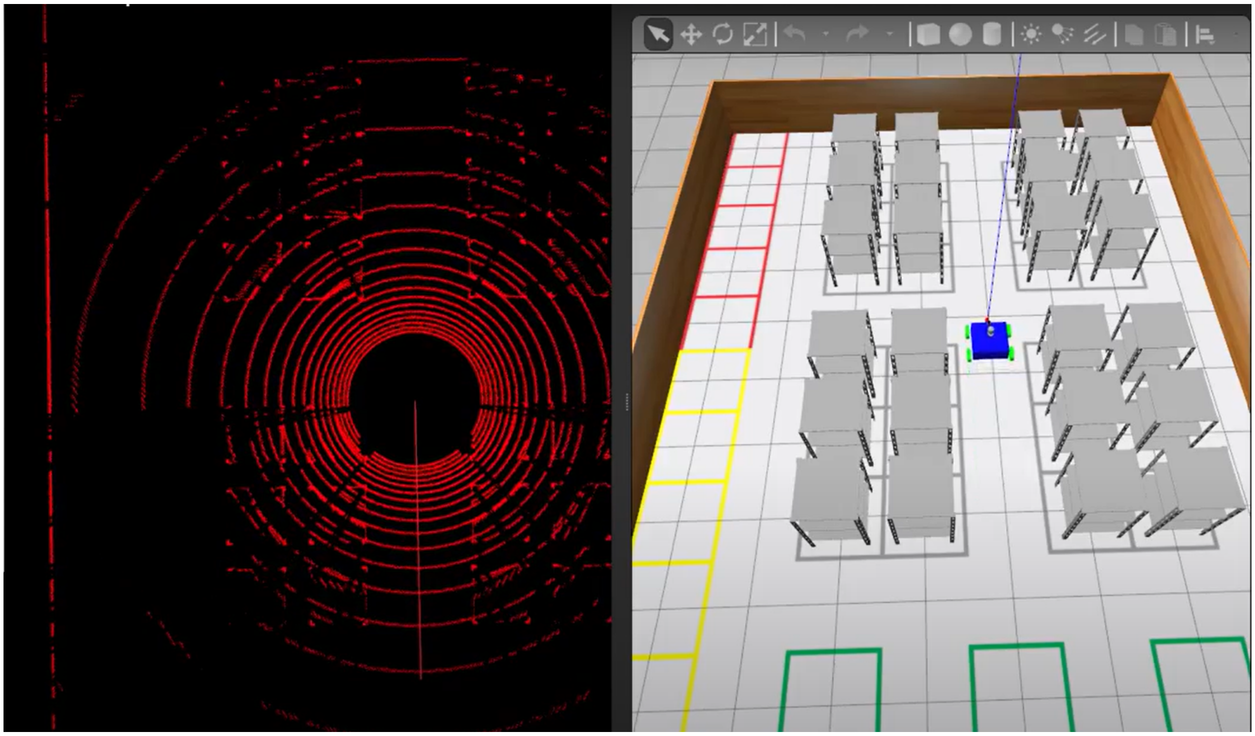

Figure 4.19: Implementing FLOAM in our Warehouse Environment

4.6 Multi Robot Navigation with Dynamic Collision Avoidance

Using the position and orientation data from the odometry of each robot, we included a collision avoidance module that allows the robots to navigate around the warehouse efficiently without any collisions. Each robot still has its own shortest trajectory to its goal generated from Dijkstra's algorithm. All robots move simultaneously and one will halt at its position if another robot is in its facing direction and within a certain limited distance from it. Given our environment with single lanes, this method allows all the robots to reach their goal positions in the shortest time possible.

Chapter 5

Results

5.1 Analysis

Rigorous testing has been made with the mobile robot to make sure that it reaches the destination without any collisions and is able to slide under the shelf and lift it without any boundary collisions happening. All the parameters have been tuned properly many times to achieve this. Since we did not have any locking mechanisms (like grippers/electromagnetic locking) to lock the shelf to the upper base of the mobile robot while navigating, and since the weight of the shelf is less, motion planning has to be done slowly to reach the destination. This problem can be addressed easily in real-life scenarios as it is easy to incorporate hardware components better than in simulation.

We measured the system's performance in terms of the time taken by the robot to navigate from the start to the shelf and also in terms of the success rate. The success rate is a metric for non-collision navigation in the environment. Along with these two metrics, we also noted the pros and cons of each method.

Metrics	Noisy Odom	IMU + Noisy Odom EKF	FLOAM
Time Taken (Start to shelf)	-	~46 seconds (avg.)	~41 seconds (avg.)
Success Rate	0/30	30/30	30/30
Pros/Cons	-	• Less computational overhead • Stalls when facing obstacle • Static Environment	• Can plan complex paths • No stalling • Dynamic Environments • Requires the use of Lidar (may be redundant sometimes)

Figure 5.1: Comparison Metrics

We also evaluated the results in a multi-robot environment comprising two robots navigating simultaneously. Although both the robots generated overlapping paths during navigation, the robots did not collide while traversing their paths. The obstacle detection module was able to successfully handle these head-on-head encounters with other robots. Whenever a robot was head-on with another robot, it would just stall at its current location and wait until the other robot passes.

As summarized in the table above, different warehouse scenarios need different sensors and navigation techniques. It all depends on the use cases that a warehouse environment intends to have. We mainly suggest Lidar-less navigation at places where there are static objects in a warehouse. Most of the dark warehouses do have static environments and do not require complex systems like Lidar for navigation. But some warehouses do demand navigation in dynamic environments, where the use of Lidar is definitely useful. This is the place where algorithms like FLOAM shine. The algorithm is so lightweight in comparison with other Lidar-based algorithms and much faster than its name indicates.

5.2 Single Robot Environment

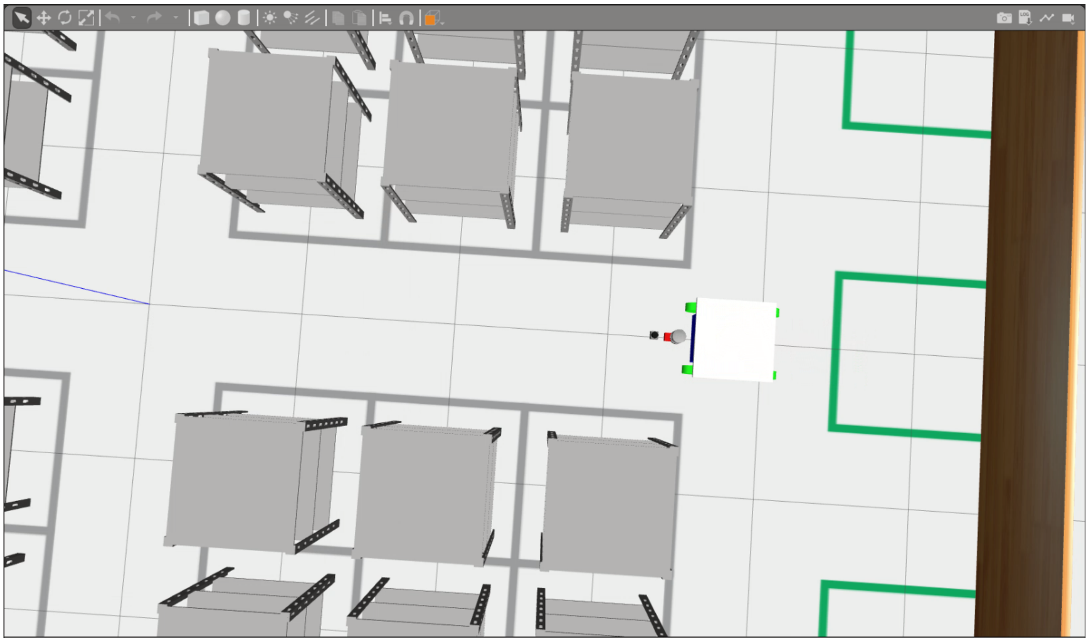

Figure 5.2: Robot at the starting position

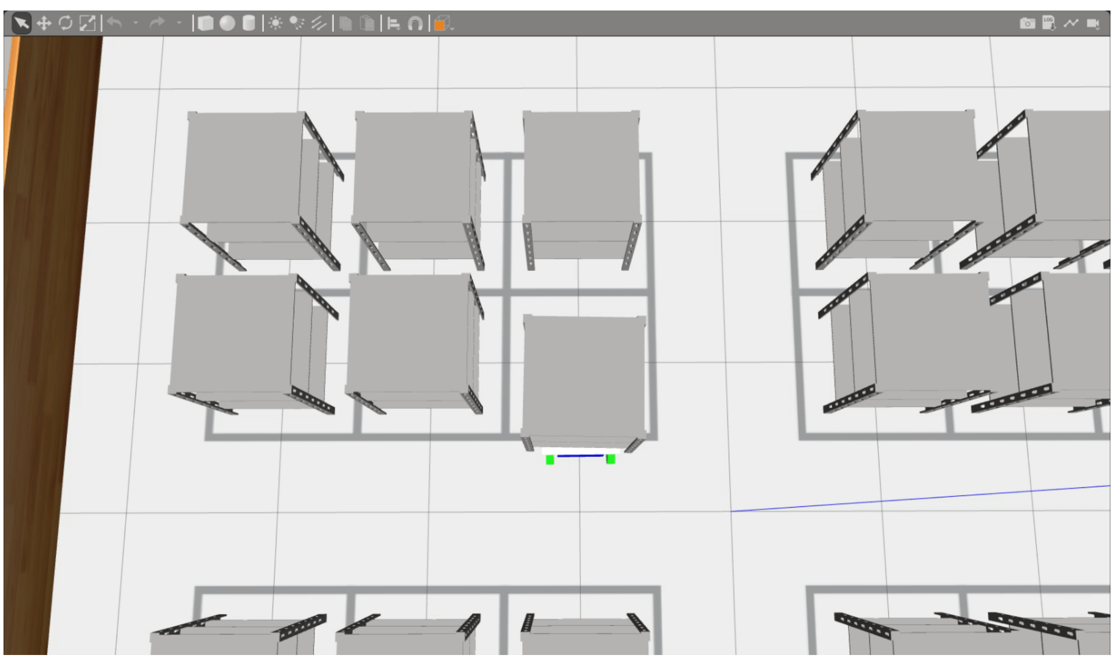

Figure 5.3: Robot picks up a shelf

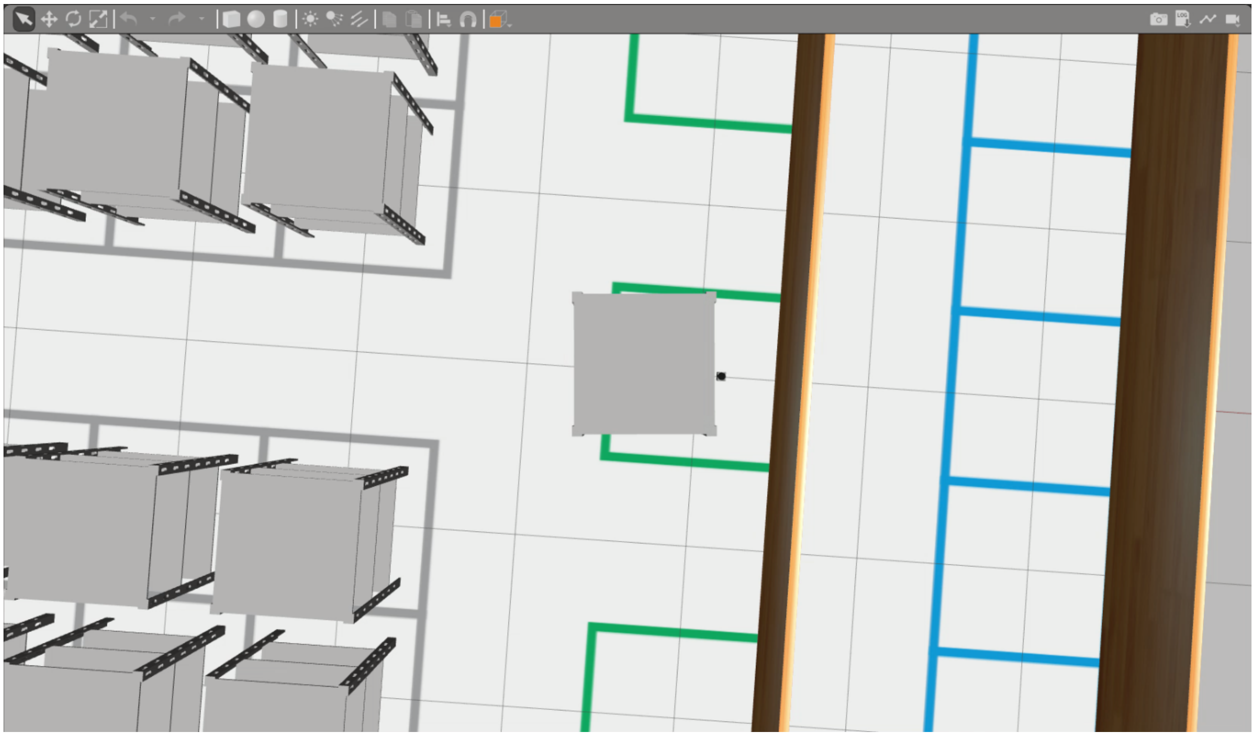

Figure 5.4: Robot reaches the goal position

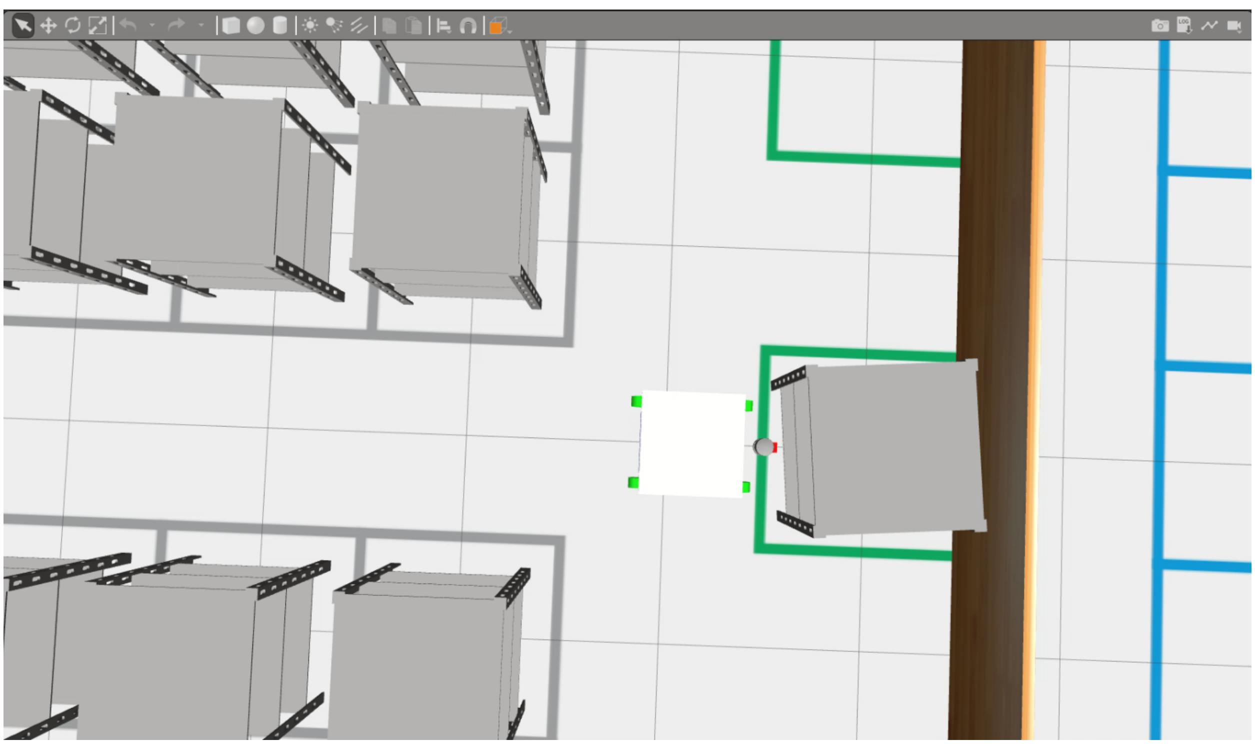

Figure 5.5: Robot drops the shelf and backs away

5.3 Double Robot Environment

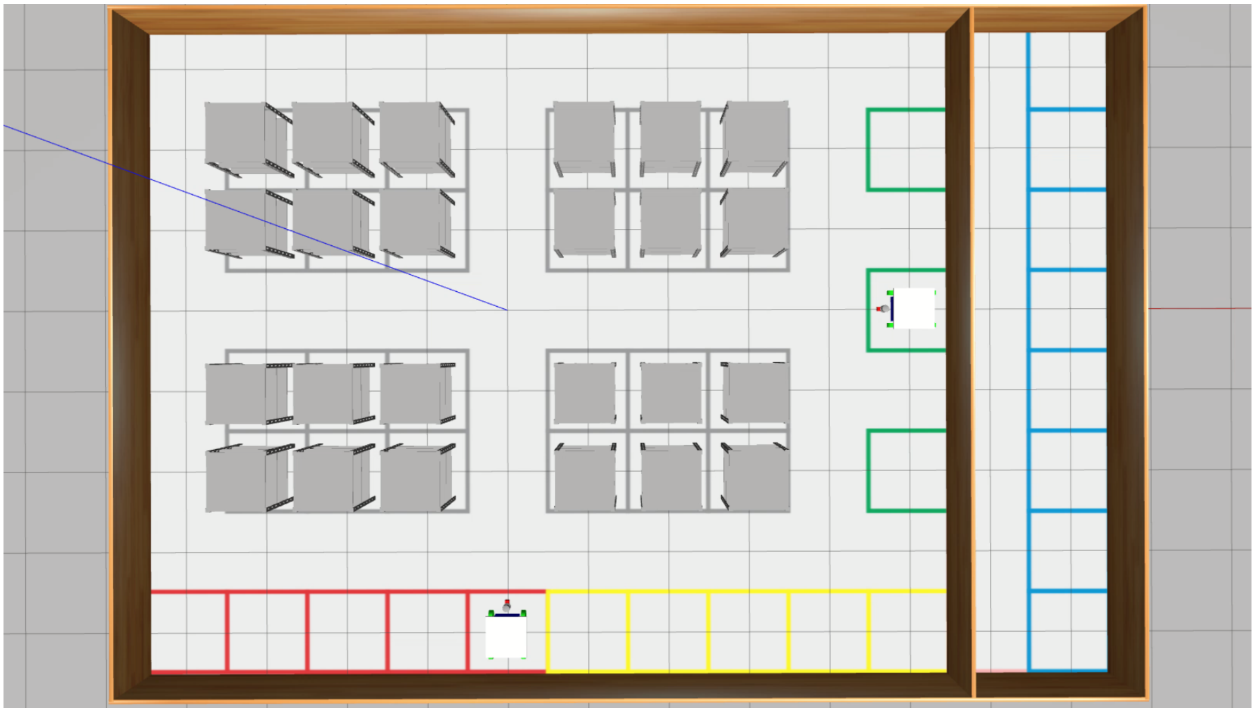

Figure 5.6: Two robots at their start positions

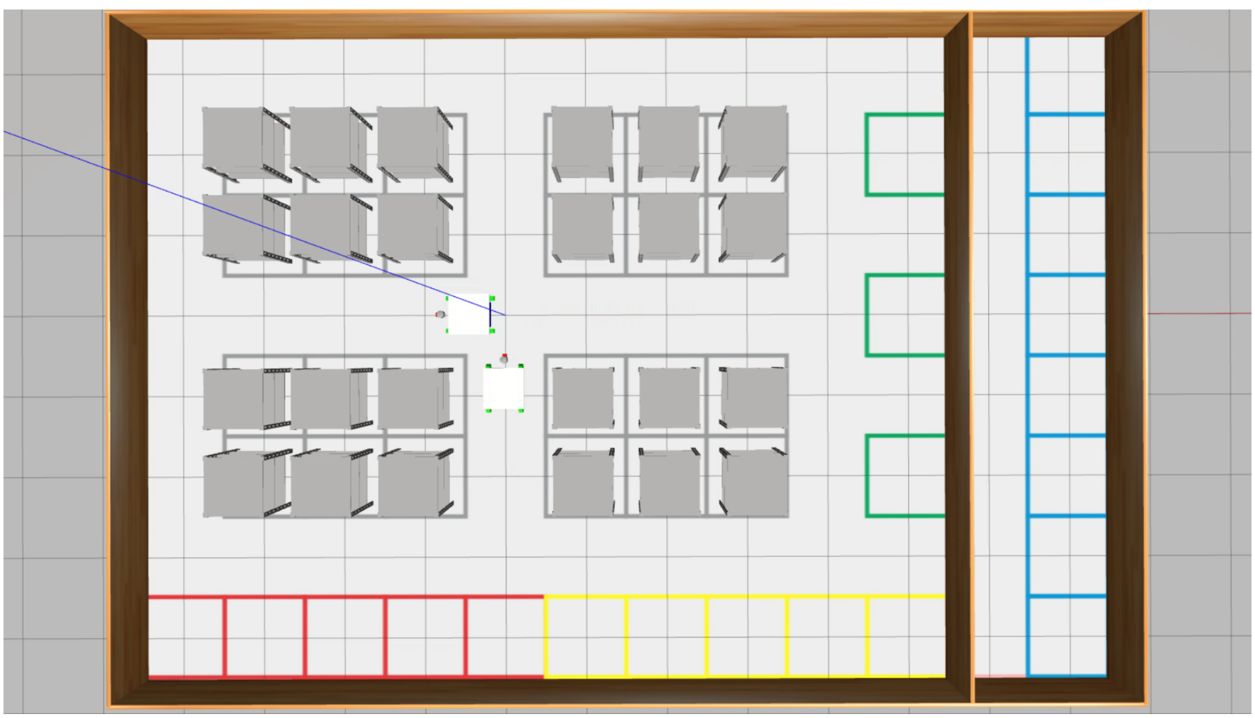

Figure 5.7: Collision Avoidance: One robot waits for the other robot to pass

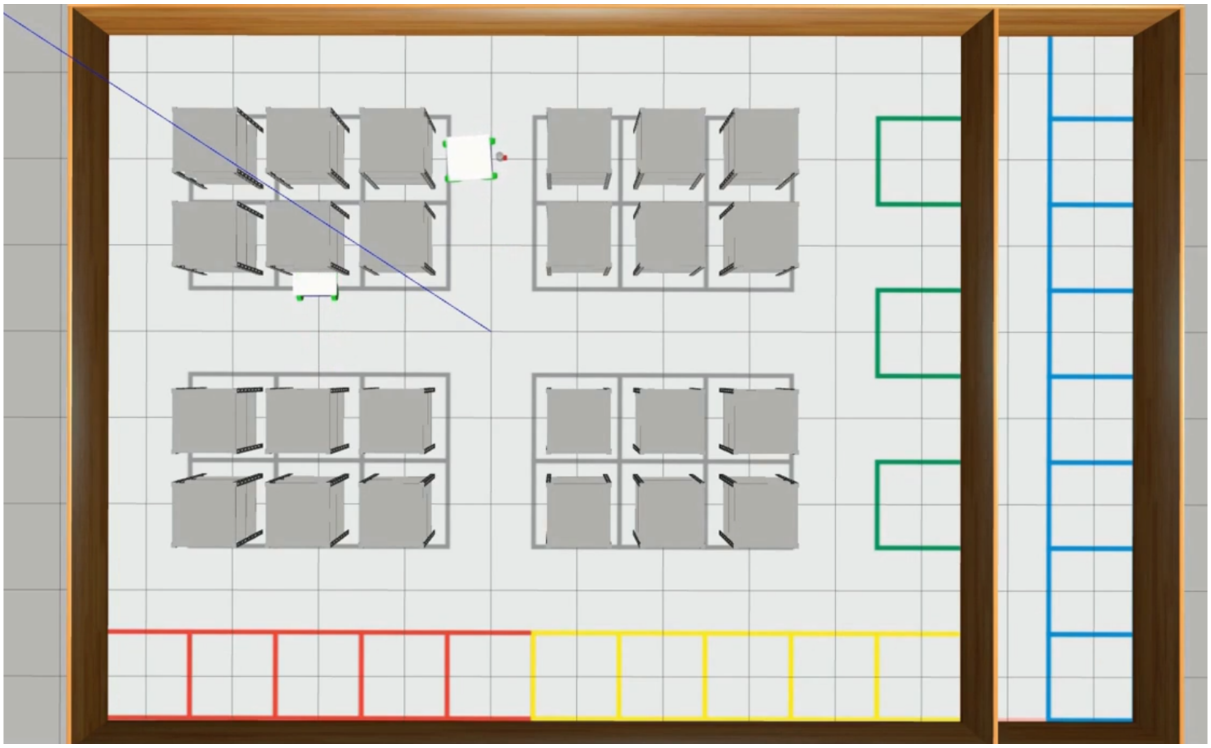

Figure 5.8: One robot reaches its designated shelf

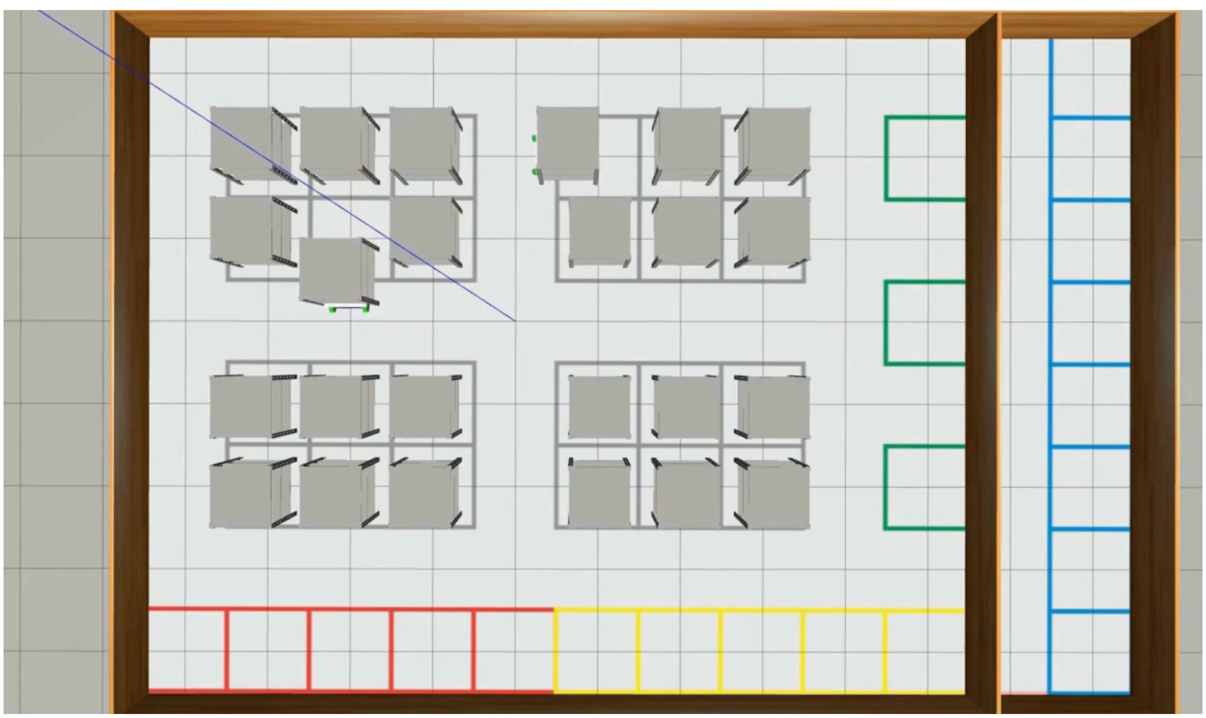

Figure 5.9: Both robots pick up their designated shelfs

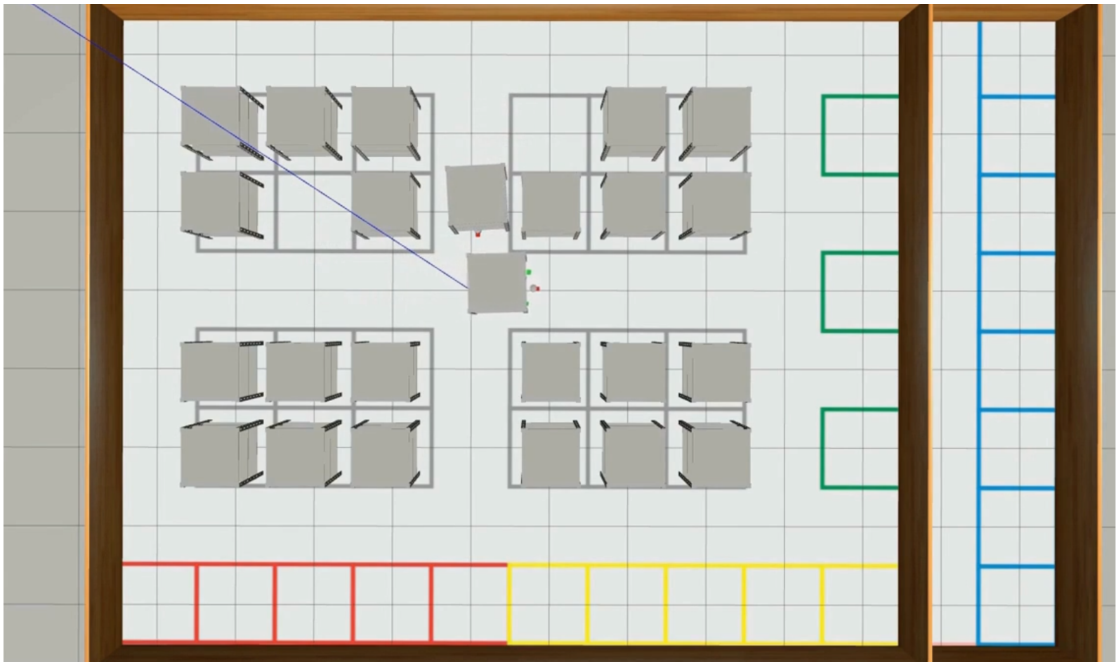

Figure 5.10: Collision Avoidance with Shelves

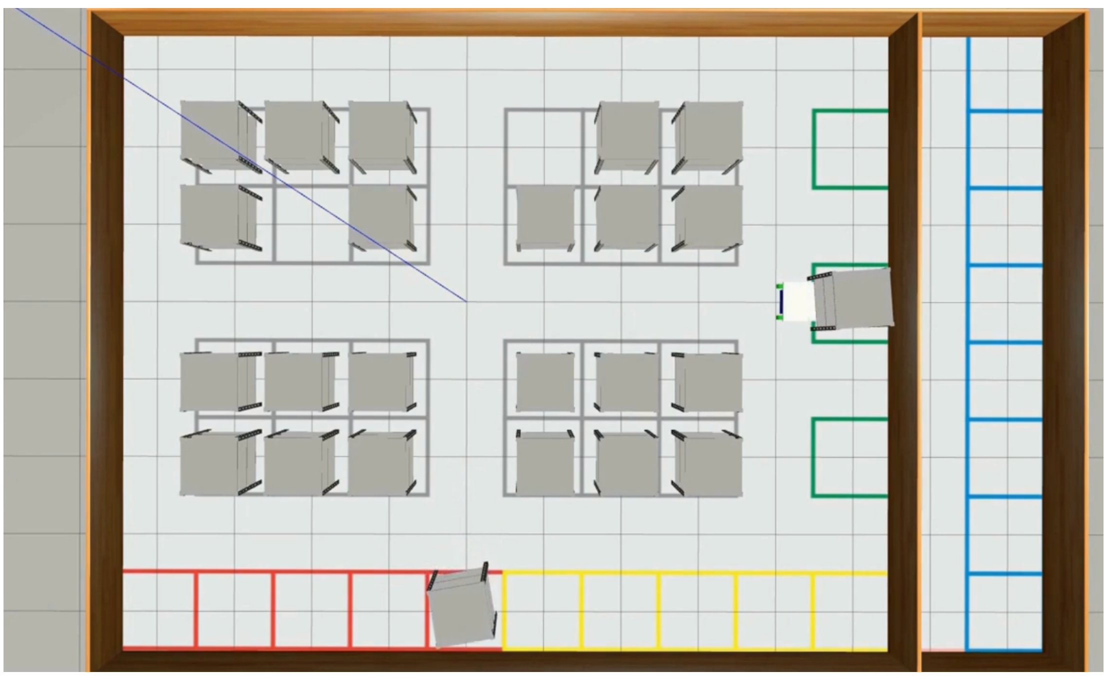

Figure 5.11: Both robots reach their respective goal positions

The video result can be viewed at:

https://www.youtube.com/playlist?list=PLtFsVQQo-YsP-UkStFe87VfC9i6lNJpq0

5.4 Future Work

There are some drawbacks or deadends in our project. A mobile robot in a warehouse can encounter many situations where it can stall. These situations include when a robot is encapsulated by other robots on the front and back. Another situation might be where there are two robots obstructing the passage for another robot. All these test cases are not handled at the moment by our system. This leaves the scope for future improvements where all these edge cases are tested and solved.

Another valuable addition to this project will be the generation of complex paths using splines in a multi-robot environment. This allows the robots to perform complex maneuvers during navigation which can help in obstacle avoidance and release from stall conditions.

But planning complex paths require heavy computation which cannot be done on-board. This demands for the use of cloud environments for the computation and processing of data. On average a multi-robot path planning with on-board computation takes about nearly 30 seconds, which is dead slow. On the contrary, given that network connectivity is good, cloud platforms can do this computation in a matter of a few seconds.

With that being said, warehouses pose complex challenges and the efficiencies of which can be altered by proposing small changes. There are many opportunities where robots can be incorporated into a warehouse for helping with various tasks.

Chapter 6

Conclusion

We are able to successfully navigate the mobile robot with a more efficient approach than the robots using a fusion of IMU and Wheel odometer that is traditionally used in warehouse environment, with a single 3D Lidar sensor which is redundant to wheel slippages and electromagnetic interfaces that can exist in warehouses. The robot we proposed can be enhanced with more advanced algorithms for decentralized robots traversing in warehouses for inventory management. In low-light environments, a lidar sensor can provide the robot with the necessary information to navigate even if it is unable to see its surroundings using other sensors such as cameras. There are a few challenges to using lidar for mobile robot navigation in dark environments. One challenge is that the laser pulses may not be able to reach objects that are too far away or that are partially occluded by other objects, which may not be much required for a warehouse situation. Thus the proposed solution allows autonomous operation in indoor environments with insufficient illumination for camera-based solutions, where our mobile robot platforms were deployed in different kinds of environments. This allows our solution to be deployed in a large number of applications.

REFERENCES

[1] E. Royal, K. H. Lee and P. K. Sen, "Electrical Energy Consumption, Energy Efficiency and Energy Conservation in Commercial Sector and “LEED” Certification", 2019 North American Power Symposium (NAPS), 2019, pp. 1-6,

[2] H. Zhang, C. Zhang, W. Yang, and C.-Y. Chen, “Localization and navigation using qr code for mobile robot in indoor environment”, in 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, 2015, pp. 2501–2506.

[3] E. H. C. Harik, F. Guerin, F. Guinand, J. Breth ´ e, and H. Pelvillain, ´ “Towards an autonomous warehouse inventory scheme,” in SSCI, 2016, pp. 1–8.

[4] M. Saska, V. Kratk ´ y, V. Spurn ´ y, and T. B ´ a´ca, “Documentation of dark ˇ areas of large historical buildings by a formation of unmanned aerial vehicles using model predictive control,” in ETFA, 2017, pp. 1–8.

[5] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” IEEE Transactions on Robotics, vol. 32, no. 6, pp. 1309–1332, 2016.

[6] J. Delmerico and D. Scaramuzza, “A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots,” in ICRA, 2018, pp. 2502–2509.

[7] H. Alismail, M. Kaess, B. Browning, and S. Lucey, “Direct visual odometry in low light using binary descriptors,” RAL, vol. 2, no. 2, pp. 444–451, 2017.

[8] R. Horaud, M. Hansard, G. Evangelidis, and C. Menier, “An overview ´ of depth cameras and range scanners based on time-of-flight technologies,” Machine Vision and Applications, vol. 27, no. 7, pp. 1005–1020, 2016.

[9] R. Opromolla, G. Fasano, G. Rufino, M. Grassi, and A. Savvaris, “Lidar-inertial integration for uav localization and mapping in complex environments,” in ICUAS, 2016, pp. 649–656.

[10] F. WANG, J.-Q. CUI, B.-M. CHEN, and T. H. LEE, “A comprehensive uav indoor navigation system based on vision optical flow and laser fastslam,” Acta Automatica Sinica, vol. 39, no. 11, pp. 1889 – 1899, 2013.

[11] M. Montemerlo and S. Thrun, FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics. Springer-Verlag, 2007.

[12] J. Q. Cui, S. Lai, X. Dong, P. Liu, B. M. Chen, and T. H. Lee, “Autonomous navigation of uav in forest,” in ICUAS, 2014, pp. 726– 733.

[13] S. Thrun and M. Montemerlo, “The graph slam algorithm with applications to large-scale mapping of urban structures,” The International Journal of Robotics Research, vol. 25, no. 5-6, pp. 403–429, 2006.

[14] F. Lu and E. Milios, “Robot pose estimation in unknown environments by matching 2d range scans,” Journal of Intelligent and Robotic Systems, vol. 18, no. 3, pp. 249–275, 1997.

[15] M. Bosse, R. Zlot, and P. Flick, “Zebedee: Design of a springmounted 3-d range sensor with application to mobile mapping,” IEEE Transactions on Robotics, vol. 28, no. 5, pp. 1104–1119, 2012.

[16] J. Razlaw, D. Droeschel, D. Holz, and S. Behnke, “Evaluation of registration methods for sparse 3d laser scans,” in ECMR, 2015, pp. 1–7.

[17] P. J. Besl and N. D. McKay, “A method for registration of 3-d shapes,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239–256, 1992.

[18] A. Segal, D. Hahnel, and S. Thrun, “Generalized-icp,” 2009.

[19] M. Magnusson, A. Lilienthal, and T. Duckett, “Scan registration for autonomous mining vehicles using 3d-ndt,” Journal of Field Robotics, vol. 24, no. 10, pp. 803–827, 2007.

[20] T. Stoyanov, M. Magnusson, H. Andreasson, and A. J. Lilienthal, “Fast and accurate scan registration through minimization of the distance between compact 3d ndt representations,” The International Journal of Robotics Research, vol. 31, no. 12, pp. 1377–1393, 2012.

[21] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgb-d slam systems,” in IROS, 2012.

[22] D. Droeschel, J. Stuckler, and S. Behnke, “Local multi-resolution rep- ¨ resentation for 6d motion estimation and mapping with a continuously rotating 3d laser scanner,” in ICRA, 2014, pp. 5221–5226.

[23] Lewczuk K, Kłodawski M, Gepner P. Energy Consumption in a Distributional Warehouse: A Practical Case Study for Different Warehouse Technologies. Energies. 2021; 14(9):2709. https://doi.org/10.3390/en14092709

[24] H. Wang, C. Wang, C. -L. Chen and L. Xie, "F-LOAM : Fast LiDAR Odometry and Mapping," 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021