Path Planning for Multiple Warehouse Autonomous Ground Vehicles

Abhiram Dapke
Maryland Robotics Center
University of Maryland
College Park, USA

Akshay Bapat
Maryland Robotics Center
University of Maryland
College Park, USA

Sanket Acharya
Maryland Robotics Center
University of Maryland
College Park, USA

***Abstract*— Since several years, the use of online shopping has grown exponentially, thanks to companies such as Amazon. In 2013, according to Amazon's statistics, customers ordered more than 36.8 million items globally on Cyber Monday alone, which equates to 426 items per second [11]. Managing such large numbers of deliveries requires a lot of planning and efficiency of work. This research paper draft aims at simulating a downsized version of such a warehouse, in which a fixed number of AGVs operate in a grid of fixed size to pick up items from shelves and deliver them to the employees for packing. In all, a total of 192 storage spaces are involved in this pick-up and delivery operation with pickup station on the left side and delivery station on the right. For the implementation, A* algorithm with priority planning is used. The results are simulated by using OpenCV for Python.**

I. INTRODUCTION

Warehouse layout directly affects the efficiency of any business operation, from manufacturing and assembly to order fulfilment [12]. Space utilization planning is one of the most crucial things considered in a warehouse. These can be handled by taking care of these three following tasks:

1. Equipment and surrounding workspace.
2. Production zones and workflow areas.
3. Storage areas.

In equipment and surrounding workspace, if the warehouse is primarily stock and ship, like in this case Amazon, then stock storage units are the primary equipment. These, usually in the form of shelving or bins, likely will take up the majority of the space. After that, allocating workspace for order packing and shipping, and stock receiving are the major concerns.

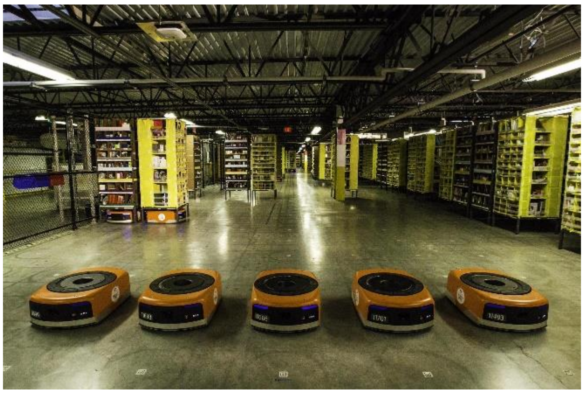

Figure 1: Amazon Robotics Warehouse Robots

The second major concern is the assembly stations. Assembly operations often combine the space needs of manufacturing and stock and ship. Assembly stations and related equipment make up the heart of your production zone. These can include workbenches or specialized stations, plus any needed bins for parts and finished goods.

This project aims at simulating a similar path planning model as Amazon warehouse for multiple automated guided vehicles in the amazon warehouse which amazon is currently using in most of their warehouses in the United States. The software used for simulating this project is python and A-star with prioritized planning is implemented. The animation of this whole scenario will be displayed by using OpenCV library in python.

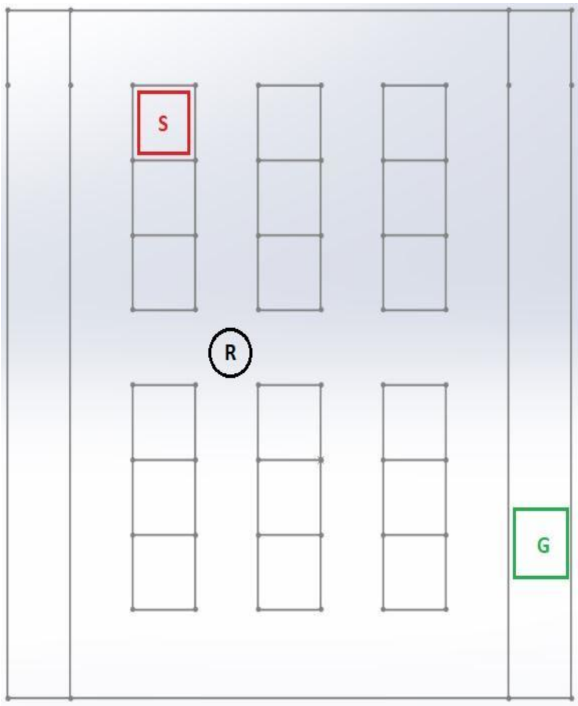

Figure 2: Workspace to be used for this project

The workspace is inspired by the one used by Kiva Systems (now a part of Amazon Robotics) and the layout comprises of a workspace having dimensions 2820 cm x 1700 cm including pickup and delivery stations. The pickup stations are on the left-hand side of the workspace and the delivery stations are on the right. There is a total of 35 pick-up and drop points and 192 storage units which are arranged across the whole workspace with three units together. These combination of three storage units are varied across 8 horizontal spaces and 8 vertical spaces in the whole workspace which are called as shelves (3 boxes in one shelf). These boxes have dimensions of 80 cm x 80 cm each. This can be clearly understood from the figure below which depicts a sample of the actual workspace used in this project. There would be 8 robots working at a time in the workspace. The robot or AGV would have a diameter of 80cm each..

II. Previous Studies

In a study from the research paper by van den Berg et al. [1], the problem of path planning for multiple robots in a common workspace is discussed. This paper assumes that the exact representation of the geometry and the workspace is already given. The results are reported as a result of three scenarios implemented in the paper. The first scenario has relatively few robots but a potentially high degree of coupling among them. Some of the challenges of this scenario are the domain of the coupled planners as the decoupled planners are most likely to fail in these cases. In the next scenario, the number of robots as well as the degree of coupling is varied. Some of the challenges faced by this type of scenario are the domain of decoupled planners as for a coupled planner, there would be too many robots to be able to compute a solution. The third scenario involved a randomly generated environment containing as many as 65 robots, giving a full composite configuration space of 130 dimensions. Each robot was assigned a random start and goal configuration.

This means that this algorithm found a solution by only planning in the 65 two-dimensional configuration spaces of each of the robots. Even though the number of robots is high, the degree of coupling in this example is very low. Their algorithm exploits this; it solved this example in only 73 seconds. After the last iteration of the algorithm, the constraint expression contained 4104 conjunctions, and the conjunction that provided the solution sequence contained 27 atomic constraints. Finally, these results show that their approach is able to solve problems which could not be solved by either fully coupled planners or decoupled planners.

van den Berg et al. [1] address the problem of trajectory planning in multi-robot systems where not only the robot-obstacle collisions are to be avoided, but also the robot-robot collisions. It discusses the prioritized multi-robot path planning algorithm, methods of implementing the same and the instances when the problem is provably solvable but the algorithm fails to provide a valid solution to the path-planning problem. Further, it proposes an asynchronous decentralized variant of the prioritized path-planning algorithm. The paper also discusses the biggest application of this algorithm: warehouse robotics, which is directly related to this project. In all, the paper discusses two algorithms existing in literature and proposes four new variants of the prioritized planning algorithm and compares the scenarios in which each algorithm may perform better than the others.

Svetska et al. [2] discuss a new approach to multi-robot path planning in which a coordinated approach is used instead of the decoupled approach that is generally used in multi-robot path planning algorithms. The paper discusses this approach for both holonomic and non-holonomic robots. The method described in this paper is called a “super-graph” method in which all the robots are collectively treated as a composite entity. Graphs are generated individually for each robot and then combined into a single roadmap for the composite robot. The implementation recorded a computation time of the order of seconds for a case with five non-holonomic robots.

According to Cap et al. [8], one possible approach to solve the path planning problem for warehouse robots is to plan the routes for each robot individually and independently and handle the robot-robot conflicts only when they occur. Although this is a valid approach, it is prone to deadlocks, that is, cases when two robots come in front of each other and try to move in opposite directions, thus blocking each other. Solving such cases is possible using the above approach but is computationally very expensive.

III. Method

According to the literature survey discussed in the previous section, planning for robot-robot conflicts on-the-fly is computationally expensive [8]. Hence, prioritized path planning is a better approach because it is a straightforward method and avoids such deadlocks by planning the paths for robots beforehand.

In prioritized planning, the robots are assigned priorities beforehand. Then, trajectories for robots are planned for higher priority robots first, followed by the lower priority robots. The pseudo-code for the prioritized path-planning algorithm is shown below:

$\Delta \leftarrow \phi$

for $i \leftarrow 1, 2, \cdots, n$ *do:*

  $\pi i \leftarrow$ *shortest_path* (Si, Gi, Δ)

  if $\pi i = \phi$ *then:*

    report failure and terminate

  $\Delta \leftarrow \Delta \cup Ri\Delta(\pi i)$

Function shortest_path (Si, Gi, Δ):

return shortest path for robot i from start node Si to goal node Gi while avoiding the obstacle space Δ

In the pseudo-code provided above, the notation used is:

Δ - Obstacle space

π i - Path planned for robot i

RiΔ(πi) - A function that maps the trajectories of robot i to regions of space-time (that is, coordinates of the nature (x,y,t)) when its center point follows a trajectory πi.

The algorithm starts with an empty obstacle space. This is under the assumption that there are no static obstacles. If there are static obstacles, then that space would be the initial obstacle space. Then, for every robot i (in the decreasing order of priority), the shortest path πi is computed from start to goal nodes while avoiding the obstacle space Δ. Here, if no path is feasible, the program returns an error and terminates. If the path is feasible, the map from the space-time coordinates (x, y, t) to the trajectory of the robot is added to the obstacle space as a dynamic obstacle. Note that, for i = 1, the robot will only need to take care of static obstacles. For i = 2, the robot will need to take care of static obstacles as well as the dynamic obstacle which is robot 1, and so on. Thus, as the algorithm iterates over i, the robot needs to take care of dynamic obstacles which are the robots numbered 1, 2, … i-1.

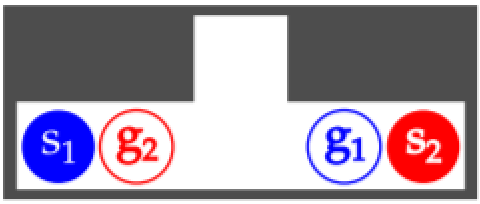

Figure 3

Prioritized planning is in general incomplete. It does not guarantee a solution to every solvable case. In figure 3, robot 1 travels from s_1 to g_1 and robot 2 travels from s_2 to g_2 in a corridor that is only slightly wider than a body of a single robot. The scenario assumes that both robots have identical speeds. In this case, trajectory of the robot that plans first will be in conflict with all satisfying trajectories of the robot that plans second, irrespective of the sequence. [8] However, in our workspace this scenario will never be observed.

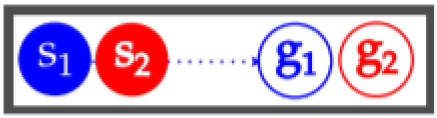

Figure 4

Figure 4 shows two robots moving from s_1 to g_1 and s_2 to g_2 respectively. The scenario assumes that robot 1 can travel faster than robot 2. Robot 1 will plan first and adopts a straight-line trajectory from s_1 to g_1 at the maximum speed, because it ignores the task of robot 2. Robot 2 plans second, but all satisfying trajectories for robot 2 are in conflict with robot 1. [8] In our case, as speeds of all robots are same, this situation will never arise.

References

[1] J van den Berg, J Snoeyink, M Lin, D Manocha, “Centralized Path Planning for Multiple Robots: Optimal Decoupling into Sequential Plans”, Robotics: Science and Systems, 2009

[2] P Svetska, MH Overmars, “Coordinated path planning for multiple robots”, Robotics and Autonomous Systems, Vol. 23, 125-152, 1998

[3] M Saha, P Isto, “Multi-Robot Motion Planning by Incremental Coordination”

[4] R D’Andrea, “A Revolution in the Warehouse: A Retrospective on Kiva Systems and the Grand Challenges Ahead”, IEEE Transactions on

Automation Science and Engineering, Vol. 9, No. 4, October 2012

[5] W Honig, S Kiesel, A Tinka, JW Durham, N Ayanian, “Persistent and Robust Execution of MAPF Schedules in Warehouses”, IEEE Robotics and Automation Letters, Vol. 4, No. 2, 2019

[6] RKA Sakir, A Rusdinar, S Yuwono, AS Wibowo, Silvirianti, NT Jayanti, “Movement Control Algorithm of Weighted Automated Guided Vehicle Using Fuzzy Inference System”, International Conference on Control and Robotics Engineering, 2017

[7] D Milner, “Swarm Robotics Algorithms: A Survey”, May 2007

[8] Michal Cáp, Peter Novák, Alexander Kleiner, Martin Selecký, “Prioritized Planning Algorithms for Trajectory Coordination of Multiple Mobile Robots”

[9] Jun-Han-Oh, Joo-Ho Park and Jong Tae-Lim, “Centralized Decoupled Path Planning Algorithm for Multiple Robots Using the Temporary Goal Configurations

[10] Fatih Okumus, Adnan Fatih Kocamaz, “Comparing Path Planning Algorithms for Multiple Mobile Robots

[11] https://www.techspot.com/news/55134-amazon-sold-368-million-items-on-cyber-monday-en-route-to-best-holiday-season-ever.html

[12] https://fitsmallbusiness.com/warehouse-layout/