

Received May 7, 2022, accepted May 29, 2022, date of publication June 8, 2022, date of current version June 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3181131

Field Evaluation of Path-Planning Algorithms for Autonomous Mobile Robot in Smart Farms

JEONGHYEON PAK^{1,2}, JEONGEUN KIM³, YONGHYUN PARK^{1,2},
AND HYOUNG IL SON^{1,2}, (Senior Member, IEEE)

¹Department of Convergence Biosystems Engineering, Chonnam National University, Gwangju 61186, South Korea

²Interdisciplinary Program in IT-Bio Convergence System, Chonnam National University, Gwangju 61186, South Korea

³Hyundai Robotics, Yongin-si 16891, South Korea

Corresponding author: Hyoung Il Son (hison@jnu.ac.kr)

This research was supported in part by the Korea Institute of Planning and Evaluation for Technology in Food, Agriculture, and Forestry (IPET) through the Smart Farm Innovation Technology Development Program, funded by the Ministry of Agriculture, Food and Rural Affairs (MAFRA), under Grant 421031-04.

ABSTRACT Path planning is crucial for several applications, including those in industrial facilities, network traffic, computer games, and agriculture. Enabling automated path-planning methods in smart farms is essential to the future development of agricultural technology. Path planning is divided into global and local planners. Global planners are divided into different types and use well-known grid-based and sampling-based algorithms. In this paper, we propose an algorithm suitable for smart farms in combination with simultaneous localization and mapping (SLAM) technology. The characteristics of the grid-based Dijkstra algorithm, the grid-based A* algorithm, the sampling-based rapidly exploring random tree (RRT) algorithm, and the sampling-based RRT* algorithm are discussed, and an algorithm suitable for smart farms is investigated through field tests. We hypothesized path planning for an agricultural harvesting robot, a spraying robot, and an agricultural transport robot, and conducted experiments in environments with static and dynamic obstacles. In addition, the set parameters are validated experimentally. The Shapiro–Wilk test is used to confirm the shape of the normal distribution, and the analysis of variance (ANOVA) and Kruskal–Wallis test are performed to confirm the significance of the experimental results. Smart farms aim to minimize crop damage; thus, it is vital to reach the goal point accurately rather than quickly. Based on the results, we determined that the A* algorithm is suitable for smart farms. The results also open the possibility of reaching the correct destination in the shortest time when working in smart farms.

INDEX TERMS Path planning, navigation, smart farm, autonomous mobile robot, simultaneous localization and mapping.

I. INTRODUCTION

Today, agriculture is industrialized, and economies of scale in agricultural production have been achieved. In addition, the integration of digital technology has contributed to the development of innovative farming techniques. In developed countries, agriculture has gone beyond human labor and currently operates in the form of smart farms (Fig. 1). Given the global push for sustainable agriculture, smart farms offer a science-based agricultural alternative that utilizes automated facilities and information and communication technology (ICT) to observe and optimally manage the agricultural environment, to overcome the constraints of time and

space. As current smart farms are based on the greenhouse environment, it is possible to control the environment by controlling factors such as temperature and humidity in the greenhouse [1].

Autonomous driving technology improves farmers' quality of life through simple repetitive tasks and helps reduce labor. This technology could be applied to mobile robots, which could be used to seed [29], harvest [5], [6], [8], spray [4], [10], and transport [32] crops in smart farms based on greenhouse operations.

Several machine vision-based agricultural robots that have greater utility when applying autonomous driving technology have been developed [13]. Mobile robots, such as unmanned ground vehicles (UGVs) and unmanned aerial vehicles (UAVs), are increasingly being used in agriculture,

The associate editor coordinating the review of this manuscript and approving it for publication was Liandong Zhu.

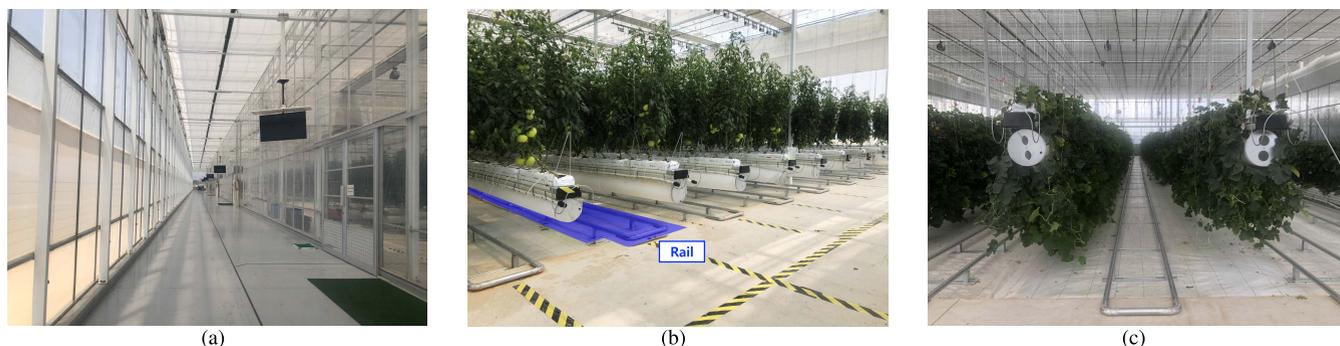


FIGURE 1. Smart farm at the national academy of agricultural sciences. (a) Smart farm passage. (b) Tomato greenhouse. (c) Korean melon greenhouse.

with a major focus on improving farm detection and autonomous navigation [11]. Using robots increase the efficiency and functionality of work in complex agricultural environments [12]. The basic motion of a mobile robot follows a row between crops, detects the end of a row, exits a row, and rotates and enters the next row. Repeating these steps allows the mobile robot to visit each row multiple times [14]. The range of applications of mobile robots in various fields has gradually been expanding. They can solve the aging problem in rural areas by minimizing the need for human labor on farms.

Mapping and localization are important aspects of advanced autonomous navigation systems for mobile robots [3]. The SLAM algorithm is being actively researched in the field of digital agriculture. Most SLAM developments have focused on outdoor mapping for unstructured environments. However, with the advent of smart farms, indoor SLAM has become increasingly important in the agricultural environment. This paper maps a smart farm using a UGV equipped with three-dimensional (3D) LiDAR and compares the performance of a path-planning algorithm.

Navigation problems are considered as path planning, localization, and motion control. Autonomous mobile robots can travel on an optimal path from their starting position to their goal without colliding with obstacles in their work environment through path planning [2]. Path planning can be divided into global and local path-planning methods. Global path planning considers the given map and searches for the entire travel path, essential for setting the optimal path rather than the time required for path calculation. On the other hand, a local path detects obstacles in real time with various sensors (e.g., LiDAR sensors, laser sensors, ultrasonic sensors, infrared sensors, and cameras) and sets the path.

The contribution of this paper is the selection of an algorithm suitable for smart farms by comparing various global planners: 1) We conducted field evaluations in the a special-purpose environment of a real smart farm, and 2) A more feasible study was conducted by adding a path-planning scenario in which static and dynamic obstacles were combined. The analyzed global planner can be applied for various purposes in various environments.

The remainder of this paper is structured as follows: Section 2 describes related work. Section 3 defines problems that require path planning in smart farms and explain the concepts of path-planning algorithms. Section 4 presents the system implemented in this experiment. Section 5 compares path-planning algorithms in field tests with combined static and dynamic obstacles. Section 6 discusses the limitations of the study, Section 7 concludes the paper, and Chapter 8 talks about future work.

II. RELATED WORK

As the interest in robotics increases, SLAM and path-planning algorithms are being actively studied for autonomous driving using mobile robots in atypical environments such as smart farms.

A. SIMULTANEOUS LOCALIZATION AND MAPPING

New technologies such as machine vision and SLAM have tremendous potential in advanced agricultural applications [13]. SLAM is divided into two main methods: visual simultaneous localization and mapping (V-SLAM), which depends on a camera, and light detection and ranging simultaneous localization and mapping (LiDAR-SLAM), which depends on a LiDAR sensor. V-SLAM is still being actively studied [64], [66], [67]. However, monocular cameras cannot provide 3D distance information, and the depth accuracy of stereo and RGB-D cameras is inversely proportional to their detection range, which is about 20 m. In addition, it is difficult to provide reliable localization due to poor performance in tracing feature points in an environment that lacks texture, such as a white wall without objects [65]. On the other hand, the LiDAR sensor not only provides centimeter-level distance information but also has a wide detection range and is unaffected by lighting conditions [33], [34]. LiDAR is more suitable because the outer wall of a smart farm is made of glass or vinyl and is easily affected by light during the day.

Simultaneous localization and mapping (SLAM) has been developed for use in various fields [15], including agriculture. Among LiDAR-based SLAM algorithms, Gmapping [16] and Google's Cartographer [17] are the most commonly used. The trajectory obtained from the Google's Cartographer system

TABLE 1. Summary of the characteristics of evaluated algorithms.

Algorithm	Advantage	Disadvantage
Dijkstra [44], [46], [47]	<ul style="list-style-type: none"> • Visits all nodes • Shortest distance 	<ul style="list-style-type: none"> • No negative weights • Take a long time
A* [41], [48], [49]	<ul style="list-style-type: none"> • Visits some nodes • Fast navigation speed • Infinite space exploration 	<ul style="list-style-type: none"> • No negative weights • Possibility of path navigation failure • Dependency of heuristic function
RRT [37], [39], [52]	<ul style="list-style-type: none"> • Small amount of calculation • Navigates complex environments 	<ul style="list-style-type: none"> • Sample dependencies • Unpredictable
RRT* [37], [38], [40]	<ul style="list-style-type: none"> • Optimality • Uniformity • Navigates complex environments 	<ul style="list-style-type: none"> • Sample Dependencies • Slow convergence speed • Unpredictable

was compared with the ground truth, and favorable results were obtained, relevant for indoor robotics applications [18].

B. PATH-PLANNING ALGORITHMS IN SMART FARMS

The basic algorithm is the basis for the application. The grid-based Dijkstra [44], [58] and A* [53] – [59] algorithms and the sampling-based RRT [62], [63] and RRT* [43], [60], [61] algorithms have been used in many recent studies. Path-planning algorithms are being studied for various applications, such as in surgical robots [50] and autonomous underwater vehicles [51]; a hybrid path-planning algorithm based on membranes [35], [36] is also being studied. In agricultural environments, research in navigation with mobile robots continues as well [7], [9]. To the best of our knowledge, there are not many navigation studies carried out in real, indoor smart farms. Research is mostly carried out in outdoor agricultural environments such as orchards and paddy fields [30], [31] and cannot be applied to indoor agricultural environments. We need to know the basic path-planning algorithm and scale it up for agriculture.

III. PATH-PLANNING ALGORITHMS

We need to select an appropriate path-planning algorithm considering the special environment of a smart farm. Path-planning algorithms have their individual characteristics. In this section, we identify the reasons for the need for path planning in a smart farm and analyze the characteristics of various path-planning algorithms.

A. PROBLEM DESCRIPTION

In smart farms, various robots, such as harvesting robots, spraying robots, and transportation robots, are used depending on the purpose. There may be many dynamic or static obstacles in the robot driving path. As only one robot can pass through the internal corridor and gates, for various robots to work, it is necessary to set the path efficiently. The path-planning algorithm searches for the number of paths the robot travels in various cases. The optimum path is then

searched according to the purpose of the algorithm. This study evaluates algorithms for setting the optimal path for almost every path between the start and goal points.

B. PATH-PLANNING ALGORITHMS

Grid-based and sampling-based algorithms are the most widely used global path planners; for example, Dijkstra and A* and the rapidly exploring random tree (RRT) and RRT*, respectively. A grid-based search efficiently expresses obstacles and nonobstacles by dividing the space into grids. Sampling-based path planning is used for finding a path by creating multiple sample points at random and searching point-wise space, without dividing the space into grids. The selected algorithms have many advantages and disadvantages; the results are summarized in Table 1. Obstacles may appear during navigation, and the robot attempts to avoid them [20]; obstacle avoidance is inevitable during navigation, and navigation is the basic operation task for a mobile robot. The path-planning algorithm must determine where the robot is, where it should be going, and how it will interact with its environment (such as static obstacles and dynamic obstacles). Therefore, to proceed with path planning, global and local indices must be considered simultaneously. The local planner mainly responds to unpredictable situations by reflecting the conditions around the robot in real time. The global planner, on the other hand, plans the entire path and takes precedence over the local planner in path planning. If the global planner is set up properly and no visible external action takes place, a small number of calculations from the local planner can allow it to reach the goal point. Both global and local planners are important, but this study focused on the global planner. In this study, the local planner was fixed to the dynamic window approach (DWA) algorithm; thus, only global path planners were compared.

1) DIJKSTRA ALGORITHM

The Dijkstra algorithm [20] is a typical method for determining the shortest path and solves the optimization problem.

Algorithm 1 Dijkstra Algorithm

```

Set OpenList as a priority queue
OpenList.push(START)
Set CloseList as an empty set
while OpenList.top() is not the GOAL do
  Current = OpenList.top()
  OpenList.pop(Current)
  CloseList.push(Current)
  for Neighbors of Current do
     $f(\text{Neighbor}) = g(\text{Neighbor})$ 
    Neighbor in OpenList and  $\text{cost} < f(\text{Neighbor})$ 
    select a Neighbor with a lower cost in all Neighbors
    if  $\text{Neighbor}_\alpha < \text{Neighbor}_\beta$  then
      Remove Neighborβ
    end if
  end for
end while
Get the solution by reversing path from GOAL to START

```

This is a greedy algorithm, which means that when a decision must be made, it selects the best choice available at the moment. Based on a breadth-first search (BSF), this algorithm searches the path from the starting point to all points; it is a method for exploring widely before exploring deeply. The Dijkstra algorithm was used because it is difficult to express the shortest distance when considering weights.

The Dijkstra algorithm is described in Algorithm 1. The basic logic of the Dijkstra algorithm is to add the shortest point based on the start point and update the shortest distance (Table 1). The Dijkstra algorithm is suitable for use when a single source and all nodes are given. The weights of points not yet connected are infinite. The Dijkstra algorithm is defined as

$$f(n) = g(n) \quad (1)$$

- n is the current node on the path.
- $f(n)$ is the cost of the shortest path of all possible paths that can go from the start node n_0 to the goal node G via the current node n .
- $g(n)$ is the cost of the path from the start node to current node n .

2) A* ALGORITHM

The A* algorithm, a grid-based path-planning algorithm, differs from other graph-based search algorithms in that it uses heuristic functions to evaluate the distance to the goal. The A* algorithm is useful when given one source and one object. An efficient algorithm needs a method for determining the fewest possible nodes when searching for the optimal path; choosing a node not on the optimal path is a waste of effort. The A* algorithm assumes that $f(n)$ can be computed for any node n [21]; it is described in Algorithm 2.

In this study, we used the Euclidean distance heuristic [22]. The shortest path is established by comparing the sum of the weights and the heuristic function. The A* algorithm aims to

Algorithm 2 A* Algorithm

```

Set OpenList as a priority queue
OpenList.push(START)
Set CloseList as an empty set
while OpenList.top() is not the GOAL do
  Current = OpenList.top()
  OpenList.pop(Current)
  CloseList.push(Current)
  for Neighbors of Current do
     $f(\text{Neighbor}) = g(\text{Neighbor}) + h(\text{Neighbor})$ 
    Neighbor in OpenList and  $\text{cost} < f(\text{Neighbor})$ 
    select a Neighbor with a lower cost in all Neighbors
    if  $\text{Neighbor}_\alpha < \text{Neighbor}_\beta$  then
      Remove Neighborβ
    end if
  end for
end while
Get the solution by reversing path from GOAL to START

```

find an approximation of the shortest path rather than finding the complete shortest path. In the A* algorithm, $h(n)$ has the greatest impact on performance (Table 1). Dijkstra is a special case of the A* algorithm with $h(n) = 0$. The A* algorithm is defined as

$$f(n) = g(n) + h(n) \quad (2)$$

- $h(n)$ is a heuristic function that estimates the cost of the lowest path from the current node n to the goal.

3) RRT ALGORITHM

RRT is a sampling-based path-planning method that randomly generates multiple sample points to find the path. It generates a sufficient number of sample points in space, checks whether they collide with obstacles, and detects the free space efficiently. The key result is that the distribution of the RRT vertices converges toward the sampling distribution, which is usually uniform [23]. However, it is unlikely that the sample will exactly match the goal point; thus, a threshold must be set. RRT allows one to move through a high-dimensional space, which is almost impossible to find with a grid-based search (Table 1).

The configuration consists of a tree made up of nodes and edges. An edge is a line-connecting node. RRT is often used for motion planning because it efficiently creates paths, even in complex environments with various obstacles. Recently, many improved algorithms have been developed to overcome the various shortcomings of RRT, RRT* is a typical example. RRT algorithm is described in Algorithm 3.

4) RRT* ALGORITHM

RRT* seeks to solve the optimality problem based on RRT. It has been proven that the costs obtained using RRT* converge to the optimum solution when the number of samples is increased infinitely (Table 1). The optimal motion planning

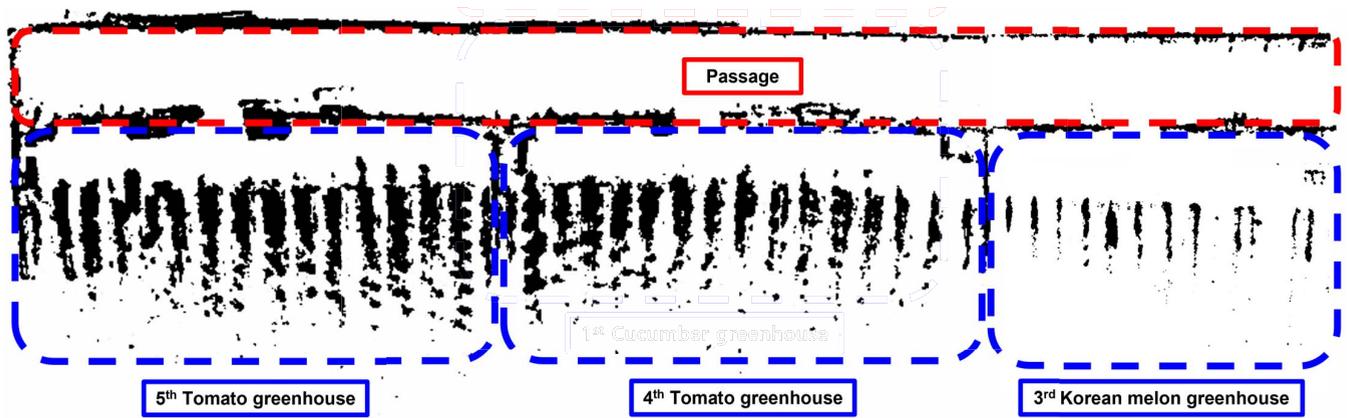


FIGURE 4. Smart farm map of path planning with static obstacles.

TABLE 2. Performance metrics.

Task	Item	
Navigation	• Time to reach goal point (s)	• $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$
	• Robot position (m) and orientation (rad)	• $S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$
Path planning	• Success rate (%)	• $\sqrt{S^2}$
		• $\frac{(\text{Number of successful autonomous driving})}{(\text{Total number of experiments})} * 100$

• The localization error is calculated as (Coordinates of the goal point) – (Coordinates after autonomous driving end).

1) SIMULTANEOUS LOCALIZATION AND MAPPING

SLAM was required before proceeding with the path-planning algorithm comparison test. SLAM technology uses a 3D cartographer algorithm combined with the A* algorithm. Cartographer is a LiDAR SLAM method that uses inertial measurement unit (IMU) and LiDAR data. IMU data are used to generate an initial estimate of robot motion, while LiDAR data updates the IMU-based estimate through scan matching against the constructed map. Among SLAM algorithms, Cartographer accumulates drift but corrects for drift by closing the loop [25].

The smart farm used 3D SLAM because the outer wall is made of transparent material and allows the passage of light. The map generated using Cartographer before the experiment is shown in Fig. 4 and Fig. 5.

2) SELECTION OF PARAMETERS

For a fully fair comparison of the performances of the algorithms being compared, the control parameters were tuned to produce the same magnitude of force for each algorithm. Algorithms are partially random, so the result is not always the same. We fixed the velocity parameters of the algorithms to either 0.5 m/s or 1.0 m/s. As a result, in Fig. 6, the accelerations of the algorithms are different, but they converge to the similar velocity. We finally set the velocity to 0.5 m/s to

minimize localization errors such as robot slippage that may occur in the experimental environment.

We set the conversion tolerance to 0.5 as a common condition. Additionally, in the case of the sampling algorithm, the number of samples was set equally to 200,000 or more because many samples are needed to search a wide environment.

3) PERFORMANCE METRIC

Algorithm performances were compared based on the following properties: time to reach the goal point [69], localization accuracy (robot position and orientation), and success rate (Table 2). For the experimental results of all algorithms, we computed the mean and standard deviation of the time to path the goal point and the position and orientation of the robot. The time to reach the goal point includes the time taken to calculate the path and drive to the destination. The localization accuracy was calculated as the relative distance between the endpoint and the goal point. We used position and orientation coordinates as performance metrics more precisely than in previous studies because location accuracy is important [39], [68]. To the best of our knowledge, few studies have used position and orientation coordinates as performance indicators. The success rate was defined as the number of successful autonomous driving cycles divided by the total number of experiments [70], [71]. In this study,

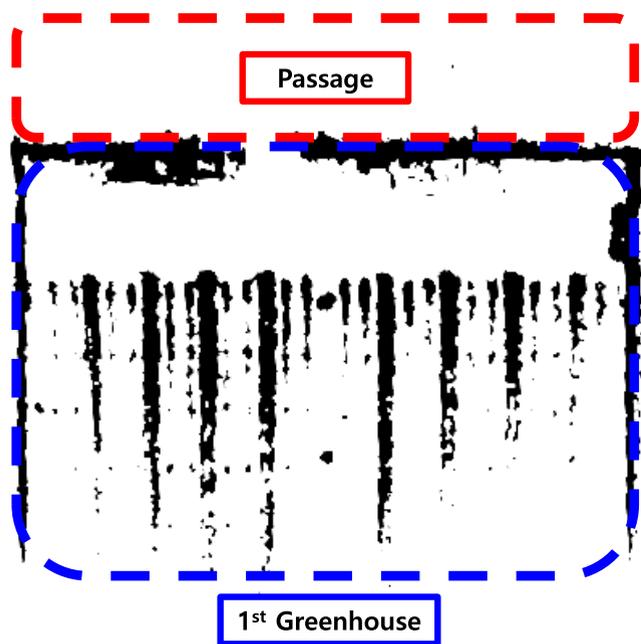


FIGURE 5. Smart farm map of path planning with dynamic obstacles.

path failure means that the autonomous mobile robot cannot reach the goal point due to internal or external problems. The internal problem refers to the robot falling into an infinite loop if it cannot create a path to the goal point or update the path because it does not know its current location while driving. The external problem refers to when the robot collides with an obstacle. When this occurs, in most cases, the robot cannot reach the goal point.

B. EXPERIMENT 1: PATH PLANNING WITH STATIC OBSTACLES

A static obstacle is a fixed obstacle reflected on the map. The robot creates a map with static obstacles through SLAM and plans a global path.

1) DESCRIPTION OF THE EXPERIMENTAL ENVIRONMENT

The algorithm performance was tested at the smart farm (Fig. 1) of the National Academy of Agricultural Sciences in Wanju-gun, Jeollabuk-do, Republic of Korea. This smart farm comprises a control room, loading docks for storing harvested crops, warehouses, and five greenhouses. It is an indoor smart farm, so there is no great challenge in moving the mobile robot, but robot slippage might occur.

2) SCENARIO

Short-distance path planning with static obstacles (Fig. 7(a)) assumed a harvesting or spraying robot. The experiment was conducted by designating the 10th rail (Table 3) of the fifth tomato greenhouse as the goal point. The robot was to move to the 6th and 10th rails of the fifth tomato greenhouse designated by the operator and harvest crops. Long-distance path planning with static obstacles (Fig. 7(b)) assumed a

TABLE 3. Goal point of field test.

Goal point	Case 1	Case 2	Case 3
Position x (m)	7.48	28.28	9.88
Position y (m)	4.97	7.84	-7.44
Orientation z (rad)	0.73	0.78	-0.73
Orientation w (rad)	0.69	0.63	0.68

transportation robot. The experiment was performed by designating the 12th rail (Table 3) of the fourth Korean melon greenhouse as the goal point. According to the path specified by the operator, the crops were to be transported past the 5th rail of the fifth tomato greenhouse and moved to the 3rd and 12th rails of the fourth Korean melon greenhouse.

C. EXPERIMENT 2: PATH PLANNING WITH DYNAMIC OBSTACLES

Dynamic obstacles are unexpected objects that are not reflected on the map. The robot avoids unpredictable obstacles that appear in real time through local path planning.

1) DESCRIPTION OF THE EXPERIMENTAL ENVIRONMENT

The algorithm performance was tested at the smart farm (Fig. 8) of the Korea Agricultural Technology Practicalization Agency in Iksan, Jeollabuk-do, Republic of Korea. This is a semi-indoor smart farm with sand and gravel on the floor, which is simply covered with sackcloth. In such an environment, it is difficult for a mobile robot to drive and an odometry error is easy to occur.

2) SCENARIO

The field test with dynamic obstacles used heterogeneous robots that moved in and out of the greenhouse for work (Fig. 9). The experiment was conducted by designating the 6th rail (Table 3) of the greenhouse as the goal point. According to the path specified by the operator, the robot was to move past the 4th rail of the greenhouse to the 6th rail. Here, when the robot entered the greenhouse to move to rail 4, it would encounter another robot that had finished its work. Dynamic obstacle avoidance would occur at the point where the paths of the robots intersect.

V. EXPERIMENTAL RESULTS

The results of the field test path planning for the smart farm are presented in Table 5 and Figure 10. In the resulting Table 5, the best values are presented in bold. The experiment investigated which of the grid-based algorithms, Dijkstra, A*, and the sampling-based algorithms RRT and RRT* are suitable for smart farm environments. Grid-based algorithms frequently generate paths closer to obstacles compared to sampling-based algorithms (Fig. 11). They can potentially generate the shortest path, and the search speed is relatively high. Sampling-based algorithms can generate a variety of unpredictable paths and can navigate complex environments. Path planning with dynamic obstacles generates new paths

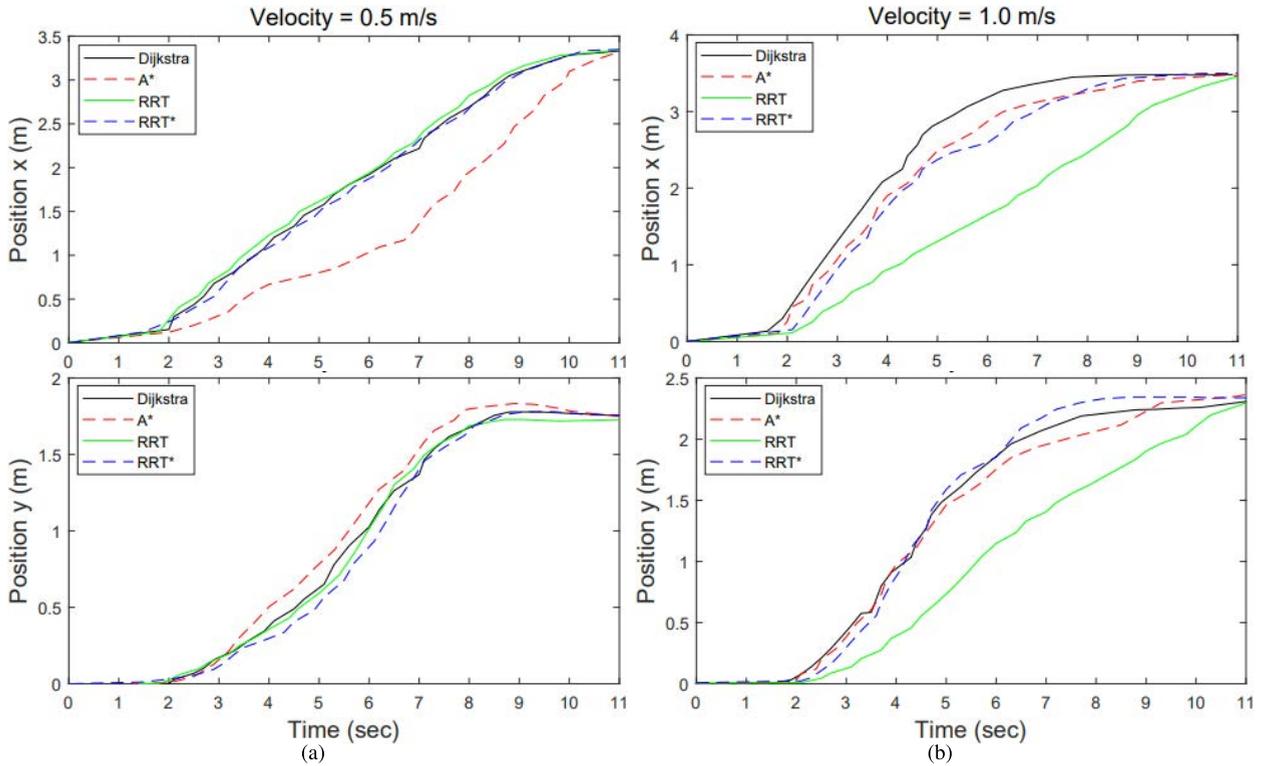


FIGURE 6. Experimental results of control parameter tuning. (a) Velocity is 0.5 m/s. (b) Velocity is 1.0 m/s.

when unpredictable objects are captured by LiDAR based on maps generated using SLAM (Fig. 12).

Based on Table 2, performance metrics are set up for evaluation. We are interested in the time and localization accuracy at which the autonomous mobile robot reaches its goal point when using each algorithm. Through the experimental results, it is possible to understand the characteristics of the algorithm and to check the differences according to the experimental environment.

To evaluate the algorithm proposed in this study, the benchmark function was used, as listed in Table 4. Data were analyzed using the Shapiro–Wilk test [26] to evaluate the normality of the sample distribution. The Shapiro–Wilk test, a test of normality, rejects the null hypothesis of the test of normality at the smallest sample size compared to other tests at all levels of skewness and kurtosis. If the experimental data were in the form of a normal distribution, ANOVA analysis was performed, and the other Kruskal–Wallis test was performed. If the p -value > 0.05 , marked with *, the mean of all groups was considered equal, i.e., the difference between the averages of all groups was not large enough to be statistically significant. However, if p -value < 0.05 , the difference between the average rankings of some groups is sufficiently large to be statistically significant.

A. EXPERIMENT 1: PATH PLANNING WITH STATIC OBSTACLES

In short-distance path planning with static obstacles, the RRT* algorithm reached the goal point the fastest, followed

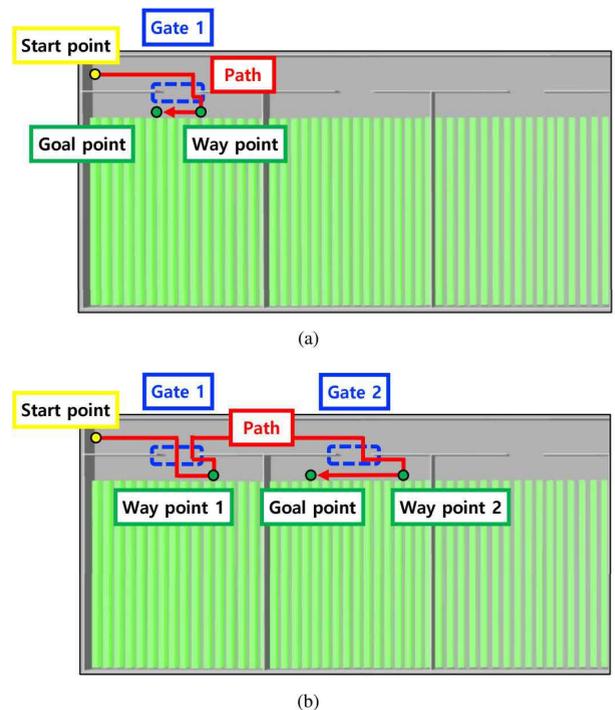


FIGURE 7. Two cases for path planning with static obstacles. (a) Short-distance path planning with static obstacles, where tomatoes are harvested or sprayed while moving rails in the first greenhouse. (b) Long-distance path planning with static obstacles, where Korean melons are transported in second greenhouse after transporting tomatoes in first greenhouse.

by the A* algorithm, and the Dijkstra algorithm took the longest time to reach the goal. These results indicate that the



(a)



(b)

FIGURE 8. Semi-indoor smart farm at the Korea agricultural technology practicalization agency. (a) Outside the greenhouse. (b) Inside the greenhouse.

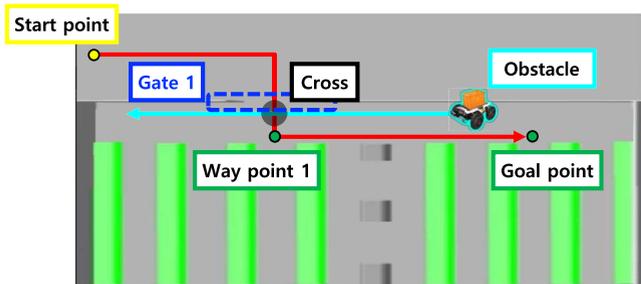


FIGURE 9. Path planning with dynamic obstacle intersecting with other working robots when entering the greenhouse.

Dijkstra algorithm’s search speed is slow because it searches all nodes. The localization accuracy was relatively high in the A* and RRT algorithms. For short-distance path planning with static obstacles, position x was sufficiently valuable to be statistically significant.

For long-distance path planning with static obstacles, the Dijkstra algorithm reached the goal point the fastest. The A* algorithm took a long time to find the long path, indicating that the A* algorithm relies on heuristic functions and is not always faster than the Dijkstra algorithm [42], [43]. In this experiment, the RRT algorithm took the longest time to reach its goal point. The localization accuracy was relatively high in the A* and RRT algorithms. For long-distance path planning with static obstacles, the time to reach

TABLE 4. Statistical analysis of results.

	ANOVA			Kruskal-Wallis test			
	F	d.f.	P-value	H	d.f.	P-value	
Case 1	Time to reach goal point (s)	2.5269	3	0.0942	-	-	-
	Position x (m)	3.2552	3	0.0493*	-	-	-
	Position y (m)	1.5652	3	0.2367	-	-	-
	Orientation z (rad)	0.4880	3	0.6955	-	-	-
	Orientation w (rad)	0.4464	3	0.7232	-	-	-
Case 2	Time to reach goal point (s)	5.3535	3	0.0096*	-	-	-
	Position x (m)	3.1779	3	0.0527	-	-	-
	Position y (m)	0.7238	3	0.5524	-	-	-
	Orientation z (rad)	1.7179	3	0.2035	-	-	-
	Orientation w (rad)	1.8976	3	0.1707	-	-	-
Case 3	Time to reach goal point (s)	2.2426	3	0.1227	-	-	-
	Position x (m)	-	-	-	6.4057	3	0.0935
	Position y (m)	1.6881	3	0.2096	-	-	-
	Orientation z (rad)	1.2852	3	0.3134	-	-	-
	Orientation w (rad)	1.3217	3	0.3020	-	-	-

the goal point was sufficiently valuable to be statistically significant.

B. EXPERIMENT 2: PATH PLANNING WITH DYNAMIC OBSTACLE

In path planning with dynamic obstacles, the A* algorithm reached the goal point the fastest. The localization accuracy was relatively high in the A* and RRT algorithms. If the A* algorithm creates the shortest path according to the heuristic function, it can significantly reduce the drive time compared to other algorithms. Further, the shorter the distance is, the better the search. The A* algorithm has been studied and applied according to its purpose until recently [53], [72]. We discovered experimentally the algorithm suitable for smart farms.

VI. DISCUSSION

We considered the autonomous driving problem of smart farms from various perspectives.

A. PRACTICAL AUTONOMOUS DRIVING FOR SMART FARM

As there are various movements inside a smart farm, we considered application scenarios of practical autonomous driving technology in a smart farm environment. For the case of repetitive movements (e.g., harvesting operation) in a stationary environment, we propose a method for combining a technology that enables followings a magnetic guide-line and

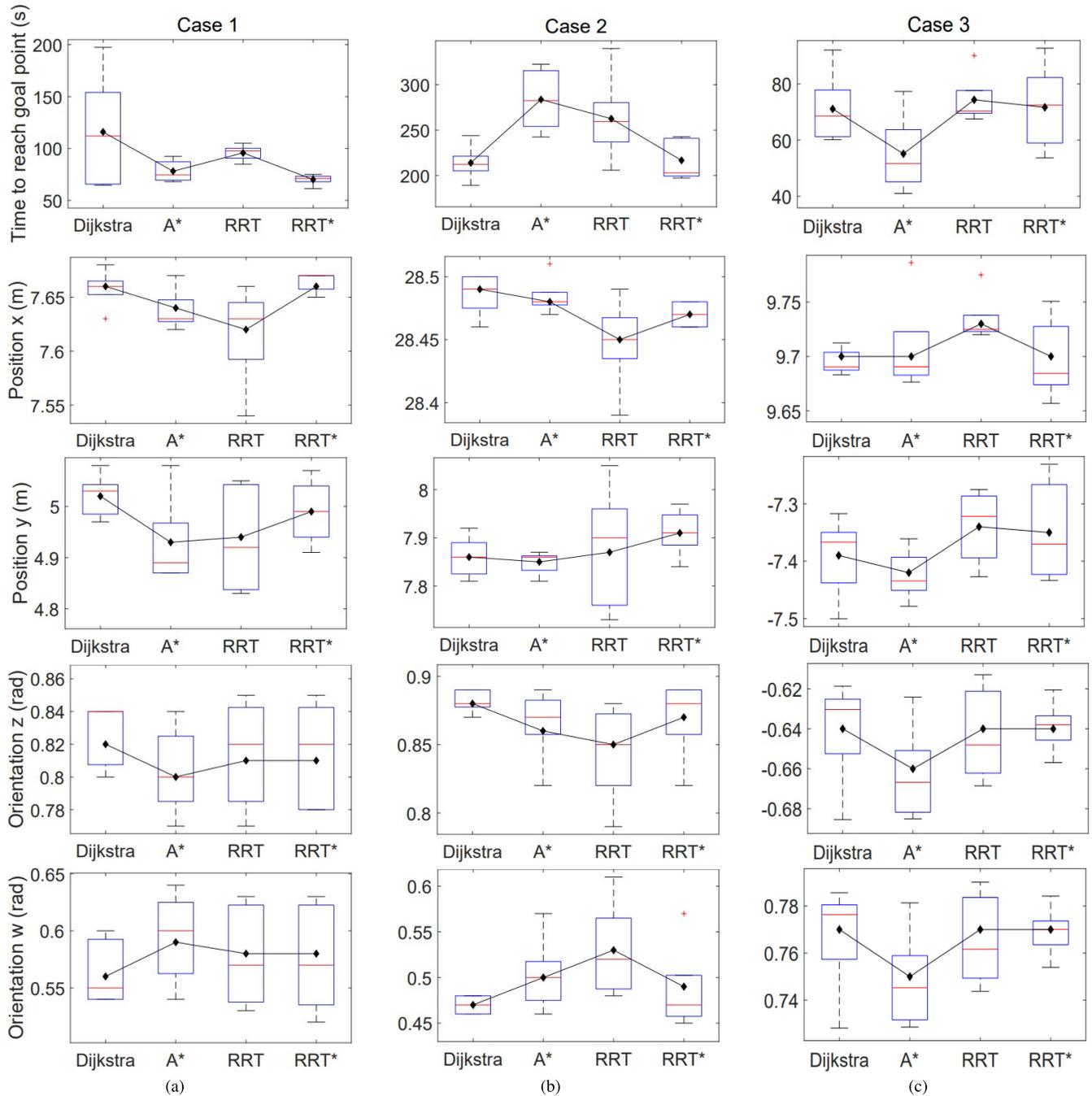


FIGURE 10. Results of field test. From the left: Time to reach goal point (s), position x (m), position y (m), orientation z (rad), orientation w (rad). (a) Short-distance path planning with static obstacles. (b) Long-distance path planning with static obstacles. (c) Path planning with dynamic obstacle.

autonomous driving technology that enables movement to the desired location in real time.

B. COMPLEMENTED A* ALGORITHM

We devised a hypothesis for addressing the limitations of the A* algorithm. The Bellman–Ford algorithm is negatively weighted. The Dijkstra and A* algorithms are not capable of handling the negative weights that the Bellman–Ford algorithm can handle. The Bellman–Ford algorithm was not considered because this study has no negative weights [27].

Although the fast search speed is an advantage of the A* algorithm, the actual shortest path search has a high probability of failure in a complex and irregular environment [28]. Path planning failures and heuristic-dependent problems were mitigated by applying an inverse algorithm.

C. IMPROVED LOCALIZATION ACCURACY

The localization accuracy can be increased by using landmarks that use Quick Response (QR) codes or AprilTags; this was confirmed in a previous study. Using QR codes as

TABLE 5. Results of field test.

Case	Algorithm	Time (s)	Localization accuracy				Success rate (%)
			Position x (m)	Position y (m)	Orientation z (rad)	Orientation w (rad)	
1	Dijkstra	115.89(±55.59)	7.66(±0.02)	5.02(±0.04)	0.82(±0.02)	0.56(±0.03)	100
	A*	78.04(±10.46)	7.64(±0.02)	4.93(±0.09)	0.80(±0.03)	0.59(±0.04)	100
	RRT	95.73(±7.53)	7.62(±0.04)	4.94(±0.10)	0.81(±0.03)	0.58(±0.05)	100
	RRT*	70.04(±5.23)	7.66(±0.01)	4.99(±0.06)	0.81(±0.03)	0.58(±0.05)	100
2	Dijkstra	213.89(±35.39)	28.49(±0.02)	7.86(±0.04)	0.88(±0.00)	0.47(±0.01)	100
	A*	283.64(±56.41)	28.48(±0.01)	7.85(±0.03)	0.86(±0.03)	0.50(±0.04)	100
	RRT	262.55(±48.59)	28.45(±0.04)	7.87(±0.13)	0.85(±0.04)	0.53(±0.05)	100
	RRT*	216.65(±22.80)	28.47(±0.01)	7.91(±0.05)	0.87(±0.03)	0.49(±0.05)	100
3	Dijkstra	71.11(±12.87)	9.70(±0.01)	-7.39(±0.07)	-0.64(±0.03)	0.77(±0.02)	100
	A*	55.13(±14.12)	9.70(±0.04)	-7.42(±0.04)	-0.66(±0.02)	0.75(±0.02)	100
	RRT	74.37(±9.06)	9.73(±0.02)	-7.34(±0.06)	-0.64(±0.02)	0.77(±0.02)	100
	RRT*	71.69(±15.33)	9.70(±0.04)	-7.35(±0.09)	-0.64(±0.01)	0.77(±0.01)	100

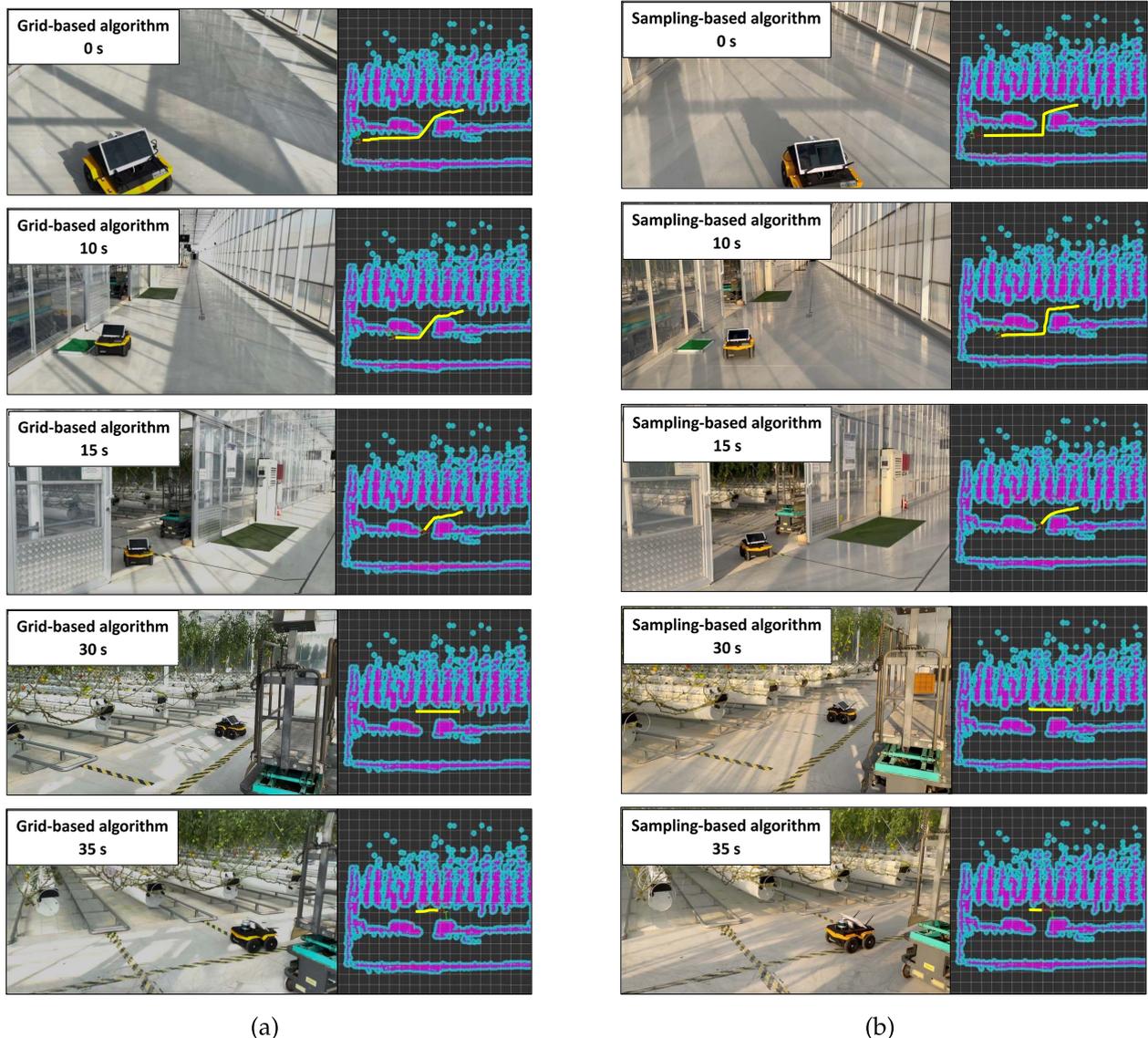


FIGURE 11. Grid-based algorithms generate a path close to the obstacle and arrive at the goal point in the relatively shortest path. A sampling-based algorithm that randomly generates nodes creates different paths. (a) Path of A* algorithm. (b) Path of RRT* algorithm.

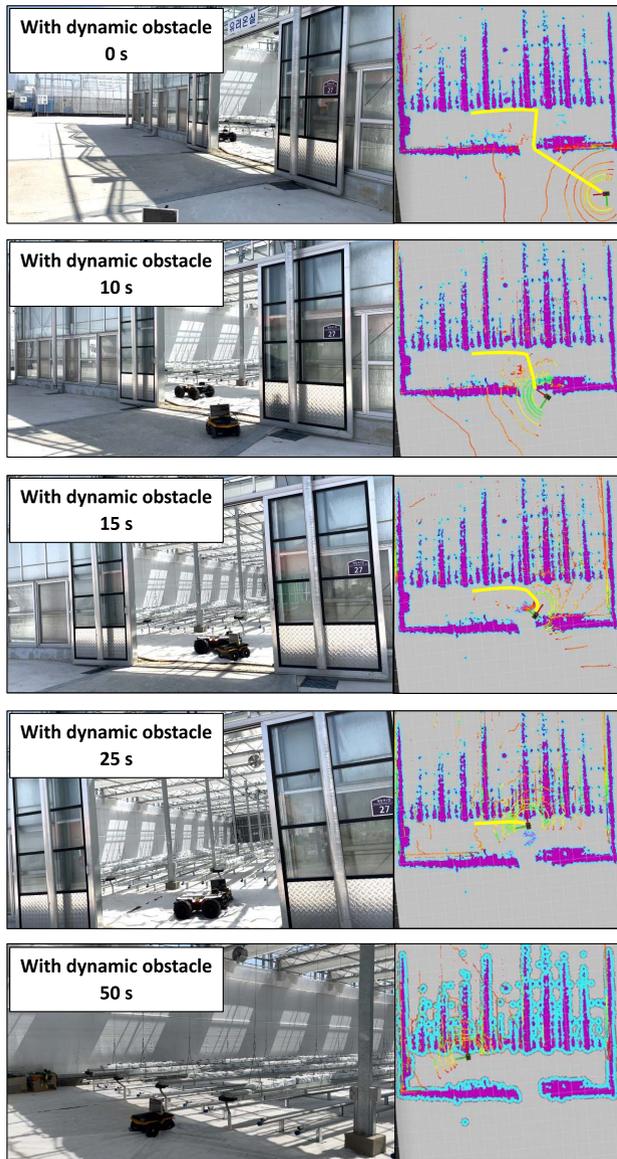


FIGURE 12. When an unpredictable object not on the map is detected on the LiDAR, the robot recognizes the obstacle and creates a new path.

landmarks in the smart farm environment, two landmarks were used over a short distance and three landmarks were used over a long distance. The goal point could not be reached without using landmarks, and using them allowed the goal point to be reached with a small margin of error.

D. EXPERIMENTAL ENVIRONMENT CHARACTERISTICS

Smart farms exist in various forms. We visited a well-maintained indoor smart farm and a semi-indoor smart farm that somewhat resembled the outdoors. A well-maintained indoor smart farm should consider slippage and narrow entrances to and from one vehicle. Additionally, in a semi-indoor smart farm, if the floor is similar to a field and the sackcloth on the floor is not fixed well, the localization error becomes more severe. In a variety of environments, both flat

and bumpy road surfaces can hit the wheels, accumulating errors and resulting in inaccurate navigation [45].

VII. CONCLUSION

In this study, we evaluated path planning algorithms based on their suitability for smart farms. We compared and analyzed the well-known grid-based Dijkstra and A* algorithms and sampling-based RRT and RRT* algorithms. The three analyses were based on the following: 1) Short-distance path planning with static obstacles 2) Long-distance path planning with static obstacles 3) Path planning with dynamic obstacles. In addition, the reliability was increased by conducting the field experiment under the same conditions through the separately conducted parameter setting experiment.

In special environments such as smart farms, it is essential to minimize damage to crops, so it is more important to accurately reach the target rather than quickly. The A* algorithm showed high localization accuracy in all cases and is more predictable than the sampling-based algorithm. Considering the multi-robot, it showed the best performance in terms of time and location even in an environment with dynamic obstacles. The results of the experiment confirmed that the A* algorithm is suitable for smart farms. Furthermore, the A* algorithm is still frequently studied, and we can apply it in many different ways.

VIII. FUTURE WORK

Path planning is important for scaling to multi-robots in a smart farm. Smart farms of the future will require multiple robots to perform different roles and collaborate for various functions. In a multi-robot environment, different robots become dynamic obstacles, so we must devise better applications for controlling multiple single robots based on the current evaluation. The problem of path planning extends to the multi-UGV path planning and coordination problem. Each robot needs to implement SLAM and navigation, further considering planning between multiple robots. We plan to investigate a robot that harvests by placing a robot arm on a platform [6], a robot that controls spraying by attaching a sprayer, and a robot that transports harvested crops to a loading box.

The main contribution of this study is that it identified a basic algorithm suitable for autonomous driving in a smart farm, which has infinite potential for development in a specialized field. This will motivate subsequent studies to improve the overall performance of control systems from a human perspective of the design and application of system parameters.

REFERENCES

- [1] C. Yoon, M. Huh, S.-G. Kang, J. Park, and C. Lee, "Implement smart farm with IoT technology," in *Proc. 20th Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2018, pp. 749–752.
- [2] N. Sariff and N. Bunyamin, "An overview of autonomous mobile robot path planning algorithms," in *Proc. 4th Student Conf. Res. Develop.*, Jun. 2006, pp. 183–188.

- [3] M. Pierzchała, P. Giguère, and R. Astrup, "Mapping forests using an unmanned ground vehicle with 3D LiDAR and graph-SLAM," *Comput. Electron. Agricult.*, vol. 145, pp. 217–225, Feb. 2018.
- [4] J. Kim, J. Seol, S. Lee, S.-W. Hong, and H. I. Son, "An intelligent spraying system with deep learning-based semantic segmentation of fruit trees in orchards," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 3923–3929.
- [5] J. Zhuang, C. Hou, Y. Tang, Y. He, Q. Guo, Z. Zhong, and S. Luo, "Computer vision-based localisation of picking points for automatic litchi harvesting applications towards natural scenarios," *Biosyst. Eng.*, vol. 187, pp. 1–20, Nov. 2019.
- [6] J. Jun, J. Kim, J. Seol, J. Kim, and H. I. Son, "Towards an efficient tomato harvesting robot: 3D perception, manipulation, and end-effector," *IEEE Access*, vol. 9, pp. 17631–17640, 2021.
- [7] H. Masuzawa, J. Miura, and S. Oishi, "Development of a mobile robot for harvest support in greenhouse horticulture—Person following and mapping," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Dec. 2017, pp. 541–546.
- [8] S. Kitamura and K. Oka, "Recognition and cutting system of sweet pepper for picking robot in greenhouse horticulture," in *Proc. IEEE Int. Conf. Mechatronics Automat.*, vol. 4, Jul. 2005, pp. 1807–1812.
- [9] G. Belforte, R. Deboli, P. Gay, P. Piccarolo, and D. Ricauda Aimonino, "Robot design and testing for greenhouse applications," *Biosyst. Eng.*, vol. 95, no. 3, pp. 309–321, Nov. 2006.
- [10] P. J. Sammons, T. Furukawa, and A. Bulgin, "Autonomous pesticide spraying robot for use in a greenhouse," in *Proc. Austral. Conf. Robot. Automat.*, Dec. 2005, vol. 1, no. 9, pp. 1–9.
- [11] P. Tokekar, J. V. Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic UAV and UGV system for precision agriculture," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1498–1511, Dec. 2016.
- [12] J. Kim and H. I. Son, "A Voronoi diagram-based workspace partition for weak cooperation of multi-robot system in orchard," *IEEE Access*, vol. 8, pp. 20676–20686, 2020.
- [13] Y. Tang, M. Chen, C. Wang, L. Luo, J. Li, G. Lian, and X. Zou, "Recognition and localization methods for vision-based fruit picking robots: A review," *Frontiers Plant Sci.*, vol. 11, p. 510, May 2020.
- [14] M. Bergerman, S. M. Maeta, J. Zhang, G. M. Freitas, B. Hammer, S. Singh, and G. Kantor, "Robot farmers: Autonomous orchard vehicles help tree fruit production," *IEEE Robot. Autom. Mag.*, vol. 22, no. 1, pp. 54–63, Mar. 2015.
- [15] 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 Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.
- [16] J. M. Santos, D. Portugal, and R. P. Rocha, "An evaluation of 2D SLAM techniques available in robot operating system," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Oct. 2013, pp. 1–6.
- [17] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LiDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 1271–1278.
- [18] M. Filipenko and I. Afanasyev, "Comparison of various SLAM systems for mobile robot in an indoor environment," in *Proc. Int. Conf. Intell. Syst. (IS)*, Sep. 2018, pp. 400–407.
- [19] L. Grimstad, R. Zakaria, T. Dung Le, and P. J. From, "A novel autonomous robot for greenhouse applications," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1–9.
- [20] H. Wang, Y. Yu, and Q. Yuan, "Application of Dijkstra algorithm in robot path-planning," in *Proc. 2nd Int. Conf. Mechanic Automat. Control Eng.*, Jul. 2011, pp. 1067–1069.
- [21] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.
- [22] P.-E. Danielsson, "Euclidean distance mapping," *Comput. Graph. Image Process.*, vol. 14, no. 3, pp. 227–248, 1980.
- [23] J. J. Kuffner, Jr., and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, vol. 2, Apr. 2000, pp. 995–1001.
- [24] S. Karaman, M. R. Walter, A. Perez, E. Frizzoli, and S. Teller, "Anytime motion planning using the RRT*," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2011, pp. 1478–1483.
- [25] R. Milić, L. Marković, A. Ivanović, F. Petric, and S. Bogdan, "A comparison of LiDAR-based SLAM systems for control of unmanned aerial vehicles," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2021, pp. 1148–1154.
- [26] N. A. Ahad, T. S. Yin, A. R. Othman, and C. R. Yaacob, "Sensitivity of normality tests to non-normal data," *Sains Malaysiana*, vol. 40, no. 6, pp. 637–641, 2011.
- [27] P.-Y. Kong, C.-W. Liu, and J.-A. Jiang, "Cost-efficient placement of communication connections for transmission line monitoring," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4058–4067, May 2017.
- [28] Y. G. Ryu and Y. Park, "A study on A* algorithm applying reversed direction method for high accuracy of the shortest path searching," *J. Korea Inst. Intell. Transp. Syst.*, vol. 12, no. 6, pp. 1–9, 2013.
- [29] T. Blender, T. Buchner, B. Fernandez, B. Pichlmaier, and C. Schlegel, "Managing a mobile agricultural robot swarm for a seeding task," in *Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2016, pp. 6879–6886.
- [30] H. Gan and W. Lee, "Development of a navigation system for a smart farm," *IFAC-PapersOnLine*, vol. 51, no. 17, pp. 1–4, 2018.
- [31] S. P. Adhikari, G. Kim, and H. Kim, "Deep neural network-based system for autonomous navigation in paddy field," *IEEE Access*, vol. 8, pp. 71272–71278, 2020.
- [32] L. Shangguan, J. A. Thomasson, and S. Gopalswamy, "Motion planning for autonomous grain carts," *IEEE Trans. Veh. Technol.*, vol. 70, no. 3, pp. 2112–2123, Mar. 2021.
- [33] Y. Li, L. Ma, Z. Zhong, F. Liu, M. A. Chapman, D. Cao, and J. Li, "Deep learning for LiDAR point clouds in autonomous driving: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3412–3432, Aug. 2021.
- [34] Z. Zhao, W. Zhang, J. Gu, J. Yang, and K. Huang, "LiDAR mapping optimization based on lightweight semantic segmentation," *IEEE Trans. Intell. Vehicles*, vol. 4, no. 3, pp. 353–362, Sep. 2019.
- [35] U. Orozco-Rosas, K. Picos, and O. Montiel, "Hybrid path planning algorithm based on membrane pseudo-bacterial potential field for autonomous mobile robots," *IEEE Access*, vol. 7, pp. 156787–156803, 2019.
- [36] U. Orozco-Rosas, O. Montiel, and R. Sepúlveda, "Mobile robot path planning using membrane evolutionary artificial potential field," *Appl. Soft Comput. J.*, vol. 77, pp. 236–251, Apr. 2019.
- [37] D. Devaurs, T. Siméon, and J. Cortés, "Optimal path planning in complex cost spaces with sampling-based algorithms," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 415–424, Apr. 2016.
- [38] J. Jiang and K. Wu, "Cooperative pathfinding based on memory-efficient multi-agent RRT*," *IEEE Access*, vol. 8, pp. 168743–168750, 2020.
- [39] W. Chi, Z. Ding, J. Wang, G. Chen, and L. Sun, "A generalized Voronoi diagram-based efficient heuristic path planning method for RRTs in mobile robots," *IEEE Trans. Ind. Electron.*, vol. 69, no. 5, pp. 4926–4937, May 2022.
- [40] R. Mashayekhi, M. Y. I. Idris, M. H. Anisi, I. Ahmedy, and I. Ali, "Informed RRT*-connect: An asymptotically optimal single-query path planning method," *IEEE Access*, vol. 8, pp. 19842–19852, 2020.
- [41] D. Drake, S. Koziol, and E. Chabot, "Mobile robot path planning with a moving goal," *IEEE Access*, vol. 6, pp. 12800–12814, 2018.
- [42] N. Wang and H. Xu, "Dynamics-constrained global-local hybrid path planning of an autonomous surface vehicle," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 6928–6942, Jul. 2020.
- [43] J. Wang, W. Chi, C. Li, C. Wang, and M. Q.-H. Meng, "Neural RRT*: Learning-based optimal path planning," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 4, pp. 1748–1758, Oct. 2020.
- [44] M. Luo, X. Hou, and J. Yang, "Surface optimal path planning using an extended Dijkstra algorithm," *IEEE Access*, vol. 8, pp. 147827–147838, 2020.
- [45] X. Gao, J. Li, L. Fan, Q. Zhou, K. Yin, J. Wang, C. Song, L. Huang, and Z. Wang, "Review of wheeled mobile robots' navigation problems and application prospects in agriculture," *IEEE Access*, vol. 6, pp. 49248–49268, 2018.
- [46] M. Noto and H. Sato, "A method for the shortest path search by extended Dijkstra algorithm," in *Proc. SMC Conf., IEEE Int. Conf. Syst., Man Cybern. Evolving Syst., Hum., Org., Their Complex Interact.*, vol. 3, Oct. 2000, pp. 2316–2320.
- [47] G. Qing, Z. Zheng, and X. Yue, "Path-planning of automated guided vehicle based on improved Dijkstra algorithm," in *Proc. 29th Chin. Control Decis. Conf. (CCDC)*, May 2017, pp. 7138–7143.
- [48] G. Tang, C. Tang, C. Claramunt, X. Hu, and P. Zhou, "Geometric A-star algorithm: An improved A-star algorithm for AGV path planning in a port environment," *IEEE Access*, vol. 9, pp. 59196–59210, 2021.
- [49] Z. Liu, H. Liu, Z. Lu, and Q. Zeng, "A dynamic fusion pathfinding algorithm using Delaunay triangulation and improved A-star for mobile robots," *IEEE Access*, vol. 9, pp. 20602–20621, 2021.

- [50] D. Li, P. Wang, and L. Du, "Path planning technologies for autonomous underwater vehicles-A review," *IEEE Access*, vol. 7, pp. 9745–9768, 2019.
- [51] K. Leibrandt, C. Bergeles, and G.-Z. Yang, "Concentric tube robots: Rapid, stable path-planning and guidance for surgical use," *IEEE Robot. Autom. Mag.*, vol. 24, no. 2, pp. 42–53, Jun. 2017.
- [52] H. Zhang, Y. Wang, J. Zheng, and J. Yu, "Path planning of industrial robot based on improved RRT algorithm in complex environments," *IEEE Access*, vol. 6, pp. 53296–53306, 2018.
- [53] H. Kuang, Y. Li, Y. Zhang, and Y. Feng, "Improved A-star algorithm based on topological maps for indoor mobile robot path planning," in *Proc. IEEE 6th Inf. Technol. Mechatronics Eng. Conf. (ITOEC)*, Mar. 2022, pp. 1236–1240.
- [54] M. Alireza, D. Vincent, and W. Tony, "Experimental study of path planning problem using EMCOA for a holonomic mobile robot," *J. Syst. Eng. Electron.*, vol. 32, no. 6, pp. 1450–1462, Dec. 2021.
- [55] D. Li, W. Yin, W. E. Wong, M. Jian, and M. Chau, "Quality-oriented hybrid path planning based on A* and Q-learning for unmanned aerial vehicle," *IEEE Access*, vol. 10, pp. 7664–7674, 2022.
- [56] O. A. A. Salama, M. E. H. Eltaib, H. A. Mohamed, and O. Salah, "RCD: Radial cell decomposition algorithm for mobile robot path planning," *IEEE Access*, vol. 9, pp. 149982–149992, 2021.
- [57] Z. Chen, Y. Zhang, Y. Zhang, Y. Nie, J. Tang, and S. Zhu, "A hybrid path planning algorithm for unmanned surface vehicles in complex environment with dynamic obstacles," *IEEE Access*, vol. 7, pp. 126439–126449, 2019.
- [58] J. Votion and Y. Cao, "Diversity-based cooperative multivehicle path planning for risk management in costmap environments," *IEEE Trans. Ind. Electron.*, vol. 66, no. 8, pp. 6117–6127, Aug. 2019.
- [59] M. G. B. Atia, H. El-Hussieny, and O. Salah, "A supervisory-based collaborative obstacle-guided path refinement algorithm for path planning in wide terrains," *IEEE Access*, vol. 8, pp. 214672–214684, 2020.
- [60] H. Nichols, M. Jimenez, Z. Goddard, M. Sparapany, B. Boots, and A. Mazumdar, "Adversarial sampling-based motion planning," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4267–4274, Apr. 2022.
- [61] T. Zhang, J. Wang, and M. Q.-H. Meng, "Generative adversarial network based heuristics for sampling-based path planning," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 1, pp. 64–74, Jan. 2022.
- [62] Y. A. Prabowo, B. R. Trilaksono, E. M. I. Hidayat, and B. Yulianto, "Utilizing a rapidly exploring random tree for hazardous gas exploration in a large unknown area," *IEEE Access*, vol. 10, pp. 15336–15347, 2022.
- [63] R. Mashayekhi, M. Y. I. Idris, M. H. Anisi, and I. Ahmedy, "Hybrid RRT: A semi-dual-tree RRT-based motion planner," *IEEE Access*, vol. 8, pp. 18658–18668, 2020.
- [64] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.
- [65] Q. Zou, Q. Sun, L. Chen, B. Nie, and Q. Li, "A comparative analysis of LiDAR SLAM-based indoor navigation for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, early access, Mar. 18, 2021, doi: 10.1109/TITS.2021.3063477.
- [66] Y. Jang, C. Oh, Y. Lee, and H. J. Kim, "Multirobot collaborative monocular SLAM utilizing rendezvous," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1469–1486, Oct. 2021.
- [67] B. Bescos, C. Campos, J. D. Tardós, and J. Neira, "DynaSLAM II: Tightly-coupled multi-object tracking and SLAM," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5191–5198, Jul. 2021.
- [68] H. Xie, K. Jiang, and K. Song, "An optimal path-planning algorithm for unmanned rollers with constraints on roller attitude," in *Proc. IEEE 10th Data Driven Control Learn. Syst. Conf. (DDCLS)*, May 2021, pp. 964–971.
- [69] Y. Quan, H. Ouyang, C. Zhang, S. Li, and L.-Q. Gao, "Mobile robot dynamic path planning based on self-adaptive harmony search algorithm and morphin algorithm," *IEEE Access*, vol. 9, pp. 102758–102769, 2021.
- [70] Q. Li, W. Lin, Z. Liu, and A. Prorok, "Message-aware graph attention networks for large-scale multi-robot path planning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5533–5540, Jul. 2021.
- [71] Q. Li, F. Gama, A. Ribeiro, and A. Prorok, "Graph neural networks for decentralized multi-robot path planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 11785–11792.
- [72] A. Candra, M. A. Budiman, and K. Hartanto, "Dijkstra's and A-star in finding the shortest path: A tutorial," in *Proc. Int. Conf. Data Sci., Artif. Intell., Bus. Analytics (DATABIA)*, Jul. 2020, pp. 28–32.



JEONGHYEON PAK received the B.S. degree from the Department of Rural and Biosystems Engineering, Chonnam National University, South Korea, in 2022, where she is currently pursuing the M.S. degree in biosystems engineering. Her research interests include field robotics, SLAM, and autonomous control of UGV and UAV.



JEONGEUN KIM received the B.S. and M.S. degrees from the Department of Rural and Biosystems Engineering, Chonnam National University, South Korea, in 2019 and 2021, respectively. She is currently an Associate Researcher with Hyundai Robotics. Her research interests include mobile robots, SLAM, and multi-robot operation systems.



YONGHYUN PARK received the B.S. and M.S. degrees from the Department of Biosystems Engineering, Gyeongsang National University, South Korea, in 2019 and 2021, respectively. He is currently pursuing the Ph.D. degree in biosystems engineering with Chonnam National University. His research interests include field robotics and agricultural robotics.



HYOUNG IL SON (Senior Member, IEEE) received the B.S. and M.S. degrees from the Department of Mechanical Engineering, Pusan National University, South Korea, in 1998 and 2000, respectively, and the Ph.D. degree from the Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2010. He had several appointments both academia and industry as a Senior Researcher with LG Electronics, Pyeongtaek-si, South Korea, from 2003 to 2005, and Samsung Electronics, Cheonan-si, South Korea, from 2005 to 2009; a Research Associate with the Institute of Industrial Science, The University of Tokyo, Tokyo, Japan, in 2010; and a Research Scientist with the Max Planck Institute for Biological Cybernetics, Tübingen, Germany, from 2010 to 2012. From 2012 to 2015, he was the Principal Researcher with the Telerobotics Group, Central Research Institute, Samsung Heavy Industries, Daejeon, South Korea. In 2015, he joined as the Faculty Member of the Department of Convergence Biosystems Engineering, Chonnam National University, Gwangju, South Korea, where he is currently an Associate Professor. He is also an Adjunct Professor with the Department of Robotics Engineering, Chonnam National University. His research interests include field robotics, hybrid systems, haptics, teleoperation, and agricultural robotics.

...