IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 22 February 2023, accepted 13 April 2023, date of publication 24 April 2023, date of current version 1 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3269981

RESEARCH ARTICLE

Multi-Level Indoor Path Planning and Clearance-Based Path Optimization for Search and Rescue Operations

JIHOON LEE[1], HEEIN PARK[2], YOUDAN KIM[1,3], (Senior Member, IEEE), CHAN GOOK PARK[1,3], (Member, IEEE), AND JAE HONG LEE[1], (Student Member, IEEE)

[1]Department of Aerospace Engineering, Seoul National University, Seoul 08826, South Korea
[2]Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea
[3]Institute of Advanced Aerospace Technology, Seoul National University, Seoul 08826, South Korea

Corresponding author: Youdan Kim (ydkim@snu.ac.kr)

This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant by the Korean Government through the National Fire Agency (NFA) (Development of wireless communication tracking-based location information system in disaster scene for firefighters and person who requested rescue) under Grant 2019-0-01325.

ABSTRACT In this study, a multi-level path planning system is proposed for indoor search and rescue operations. Requirements for the path planning system are derived based on the operational concept of the integrated indoor navigation system. Different aspects of various path planning algorithms are assessed, and their suitability to search and rescue operations in structured indoor environments is investigated. A five-step path planning system is proposed, which consists of map pre-processing, segment path planning, graph processing, route optimization, and path post-processing. The proposed method addresses a multi-goal path planning problem in a multi-story building in a computationally efficient way by adopting a graph-based approach while satisfying such requirements as clearance conditions in the pre- and post-processing steps. Furthermore, a multi-query approach is adopted to exploit the response time and earn flexibility with respect to environmental changes. The effectiveness of the proposed path planning system is demonstrated through numerical simulations. The proposed multi-level path planning system successfully adapts to complex indoor environments, enabling more effective navigation for search and rescue operations. Additionally, the system exhibits a high degree of flexibility in response to environmental changes, ensuring that the path planning remains robust and reliable even in dynamically changing situations.

INDEX TERMS Path planning, indoor navigation, path optimization, linear programming, dynamic environment.

I. INTRODUCTION

Searching the interior of a building on fire and rescuing victims is one of the most important and dangerous aspects of a firefighter's duty. Although there are many studies on autonomous or remotely controlled robots that help with mapping, situational assessment, monitoring, and searching for victims, deployment of firefighters is still inevitable in many cases. Completing the search and rescue (SAR) operation as quickly as possible is important to the safety of firefighters and victims. However, finding a way to reach the victims inside a building on fire, especially with a complicated structure, is extremely difficult due to blackout, flames, smoke, and noise. Furthermore, the fire situation may change rapidly, which makes it more difficult for firefighters to make the right decisions every time. Therefore, generating a safe and efficient path and providing it to firefighters can significantly contribute to SAR operations.

Recent studies have focused on high-dimensional path planning problems for robotic urban SAR (USAR) and wilderness SAR (WiSAR) operations [1], [2], [3], [4], [5]. In comparison, a systems perspective of manned indoor

The associate editor coordinating the review of this manuscript and approving it for publication was Ton Duc Do.

USAR operations has not been sufficiently studied. Human-like path planning in indoor environments [6], [7] and human-robot interaction [8], [9], [10] are studied. An indoor navigation and evacuation path planning framework based on internet of things (IoT) was proposed [11], and an individual-based framework for emergency evacuation guidance was proposed [12]. Note that an integrated indoor navigation system consists of several subsystems. Among them, path planning is a critical component of the integrated indoor navigation system for SAR operations. In addition, mapping, localization, and finding victims are all challenging tasks. In general, the floor plans of the building are available in advance so that they can be directly utilized to generate the map. Instead of global localization based on a global navigation satellite system (GNSS), relative localization with respect to the initial position is widely used in GNSS-denied environments. On the other hand, if no prior information is available about the structure, simultaneous localization and mapping (SLAM) approaches based on lidar or vision can be utilized [13].

Many algorithms have been developed that plan a path in an environment with obstacles. The most popular schemes for many applications are A* [14] and a rapidly exploring random tree (RRT) [15], [16], [17]. A* has useful characteristics, such as completeness, optimality, and an optimal efficiency, and RRTs are well known for their efficiency in a high-dimensional space. In the effort to improve RRTs, several variants have been proposed. RRT-connect improves the rate of convergence by building two RRTs rooted at the start and the goal [18], [19], and RRT* features asymptotic optimality [20]. A natural combination of the two variants is an RRT*-connect [21]. Other variants, such as T-RRT [22], T-RRT*, AT-RRT [23], and neural RRT* [24], have been proposed. Multi-query algorithms based on roadmaps such as the probabilistic roadmap (PRM) planner [25], [26] and its asymptotically optimal version PRM* [20] were also proposed. In recent years, learning-based algorithms were proposed [27], [28], [29], [30]. The path planning problem is also widely studied in relation to ground vehicles [31], [32], [33] and unmanned aerial vehicles [34], [35], [36], [37].

Reinforcement learning (RL) has become a prominent approach for addressing complex path planning problems, especially in dynamic and partially observable environments. Prominent RL-based path planning algorithms include Q-learning, deep Q-networks (DQNs) [38], proximal policy optimization (PPO) [39], and recent methods such as soft actor-critic (SAC) [40] and twin delayed deep deterministic policy gradient (TD3) [41]. Q-learning uses a Q-table to represent expected cumulative rewards for state-action pairs but suffers from scalability issues due to the curse of dimensionality. DQNs address this issue by integrating deep learning techniques to approximate Q-values. PPO, on the other hand, is a policy optimization method that balances exploration and exploitation. SAC and TD3 are advanced off-policy algorithms that improve sample efficiency and stability in learning. These RL-based algorithms have been applied to indoor path planning with varying degrees of success, offering different strengths and limitations compared to the proposed method in this paper.

Most methods are aimed at finding a feasible solution. However, it is also important to remove redundant motion while keeping the path length as short as possible. Path pruning is a simple technique used to decrease the path length, which considers all nodes of the path. On the other hand, the shortcut technique considers all configurations on the path, which results in shorter paths in general. The random shortcut technique is one of the most widely adopted methods, and several variants exist [42].

Maintaining a proper clearance throughout the path is also important in various aspects. Geraerts and Overmars proposed an algorithm that maximizes the path clearance for safety margin [43]. The generalized Voronoi diagram (GVD) was utilized to create a high-clearance roadmap [44], and the shortest path was extracted from the roadmap using Dijkstra's algorithm. However, creating the GVD may be impractical in high-degree-of-freedom (DOF) problems. Therefore, approximate algorithms have been proposed to reduce the computational burden of the GVD method. Amersdorfer and Meurer proposed an equidistant tool path planning strategy [45]. Another approach to maximize path clearance is called the retraction technique. The PRM nodes are randomly sampled on the medial axis (MAPRM) [46], [47], [48], [49], or the initial path produced by the PRM is modified [50], [51], [52]. The retraction technique can also be combined with RRTs, such as in retraction-based RRT [53], [54] and selective retraction-based RRT (SR-RRT) [55]. However, a large clearance can make it difficult to keep track of the wall, which is dangerous in buildings on fire, especially under low visibility conditions. Therefore, most of the existing clearance-based path planning and optimization methods cannot be directly applied to the problem of indoor SAR operations.

In this study, the operational concept of the integrated indoor navigation system is reviewed. Then the requirements for the path planning system are derived. Existing path planning algorithms are assessed in various aspects, and their suitability to indoor USAR operations is evaluated. The proposed path planning system consists of five steps: map preprocessing, segment path planning, graph processing, route optimization, and path post-processing. The effectiveness of the proposed path planning system is demonstrated through numerical simulations.

The main contributions of this study are as follows. First, detailed requirements for a path planning system to perform SAR operations are derived. The derived requirements can provide the specific direction of further research for an indoor disaster response system. Second, a bottom-up approach with segment path planning is proposed to overcome the limitations of the Euclidean distance metric. The proposed graph-building approach significantly simplifies a 3D multi-goal path planning problem into a graph problem.

Third, a graph reduction method is adopted to minimize the scale of the route optimization problem in a multi-story building. The size of an NP-hard route optimization problem can be substantially reduced by applying the proposed graph reduction method so that exact algorithms can be used instead of suboptimal or heuristic algorithms. Finally, a novel path post-processing algorithm is proposed to meet the SAR operation requirements. The proposed path post-processing algorithm can be used for any low-quality paths to ensure safety, optimality, and smoothness.

This paper is organized as follows. In Sec. II, the basic requirements and assumptions for the path planning system are described. In Sec. III, a graph-based path planning system for indoor search and rescue operations is designed. In Sec. IV, the performance of the path planning system is demonstrated using numerical simulations. Finally, Sec. V concludes this paper.

II. PROBLEM STATEMENT

In this section, the basic requirements and assumptions for the path planning system are described. The path planning system operates in an integrated indoor navigation system, whose operational environment imposes various restrictions on the design of the path planning system. First, the distinct characteristics of the search and rescue operations are analyzed and translated into the formal requirements of the path planning system. Proper assumptions are made so that the problem can be formally formulated; therefore, suitable existing methods can be applied to solve the problem.

A. OPERATIONAL CONCEPT

Let us briefly introduce the operational concept of the integrated indoor navigation system, within which the path planning system works. Once a fire report is received, the search and rescue operation is initiated, and the rescue team is dispatched to the fire scene. In general, a few minutes are given for preparation before the team arrives (response time). In the meantime, the given map can be pre-processed and the roadmap can be generated for fast onsite path planning. Furthermore, some of the segments can be planned in advance at this stage. The rescue team enters the building with a portable device that can detect the signal from the mobile device of the targets, i.e., victims. The position of the team inside the building is continuously tracked by using the pedestrian dead reckoning (PDR) technique. These two pieces of information are sent to the command and control center and combined to compute the location of the targets. Once the targets are found, the path from the current location to the prescribed exit via all the targets can be generated. Then, the team follows the path displayed on the screen of the portable device. The path can be updated during the operation in case of a change in the situation.

B. SYSTEM REQUIREMENTS

The requirements for the path planning system are derived based on the operational concept. The requirements are related to i) limited visibility, ii) the variability and unpredictability of the indoor disaster scene, iii) the uncertainty of the sensor measurements, iv) the onsite communication capability, and v) the computing power of the portable device.

First, the clearance of the path against the wall should not exceed a prescribed limit. In general, a fire in the building is accompanied by thick smoke and an electrical shutdown. As a result, the visibility can be significantly reduced inside a building that is on fire. In such situations, the walls can serve as important landmarks helping the team locate themselves. Consequently, it is strongly preferred that the path maintains a small clearance against the wall while avoiding penetrating large empty spaces.

Second, it is desired that the path planning system exploit the response time. In general, it takes approximately 5-10 minutes for a team to arrive at the fire scene. Preliminary work performed during this period can contribute to reducing the onsite computational load. Hence, the path planning system should be able to utilize the available information.

Third, path planning should be completed promptly. Despite the pre-processing procedure, it may take a considerable amount of time to plan the entire path. The computational load tends to increase sharply as the number of targets, floors, and stairs increases, even though the path planning problem is 2-dimensional without any path constraints. Moreover, the internal structure of some buildings may be highly cluttered with many narrow passages, which will slow down the convergence rate of most path planning algorithms. The prolonged path calculation can lead to a dangerous situation in case of an emergency. Minimization and upper bounding of the computation time become even more important in case i), where the communication between the rescue team and the command and control center is lost, and in case ii), where the computation has to be performed solely on a portable device whose computing power is highly limited compared to the main computer. Therefore, it is important to keep track of the computational complexities of the individual algorithms used in the path planning system.

Fourth, the path planning algorithm should have flexibility and adaptability to a certain extent. Sometimes, the planned path should be modified or recreated during the operation under certain conditions. Note that the given floor plan may not be up-to-date, and the current building may significantly differ from the map due to maintenance or renovation. Moreover, the interior structure of the building can be damaged and deformed due to fire, so that the path created based on the original map is no longer valid. Therefore, the target information can be updated as the operation progresses. It is important to cope with such unpredictable incidents with minimal additional computation, which can be done by adjusting just a part of the entire path or utilizing the information obtained during previous path planning.

Finally, it is favorable that the path is consistent with human intuition and easy to recognize, even on a small screen. Some path planning algorithms result in a highly jagged path. Such unnecessary complexity of the path can be too difficult

TABLE 1. Summary of the system requirements.

Requirement	Description
Path clearance	A planned path should obey the prescribed clearance limit.
Response time	A path planning system should utilize the response time.
Planning time	Path planning should be completed promptly.
Flexibility	A path planning system should allow partial modification.
Intuitiveness	A planned path should be easy to recognize.

to understand instantly, especially in an urgent situation. Therefore, the resulting path should be smooth enough without any unnecessary local detours. The system requirements are summarized in Table 1.

C. PATH PLANNING ALGORITHMS

Let us briefly explain the path planning algorithms considered in this study. Various existing methods are investigated and evaluated for the specific problem considered in this study. The basic indoor path planning problem is to find a continuous path that connects the start position and the goal position while avoiding collision with known obstacles. The performance of the algorithms can be measured in terms of completeness, optimality, and computational efficiency.

First, grid-based algorithms can be very effective for low-dimensional problems due to their completeness and optimality. Furthermore, narrow passages, which are common in indoor path planning problems, can be easily handled. However, grid-based algorithms become computationally expensive as the grids become finer and the sheer size of the map increases. Second, geometric algorithms work intuitively, but they can no longer be useful when the obstacles are not polygonal or convex. Third, computationally efficient algorithms based on potential fields, such as [56], have been proposed, which are usually not suitable for indoor path planning problems because they can be trapped in local minima. Fourth, sampling-based algorithms are well known for their performance in high-dimensional problems, but they can also be useful in low dimensions. These are typically computationally more efficient than grid-based algorithms at the partial sacrifice of completeness and optimality.

In this study, a few of the most popular algorithms are considered for the indoor path planning problem: A*, RRT, RRT-connect, RRT*, and PRM. Example paths generated on a simple 52 × 41 grid map are shown in Fig. 1, in which the gray zone is the area where the clearance against the wall exceeds the limit.

1) A*

The A* algorithm was developed in 1968 and has become one of the most popular methods for path planning problems due to its completeness, optimality, and optimal efficiency, which can be applied to either a graph or a grid map. It can also be viewed as a combination of the greedy best-first search algorithm and Dijkstra's algorithm. At each iteration, the A* algorithm calculates the cost of the path $g(n)$ and an estimate of the cost-to-go based on the heuristic function $h(n)$, and selects the next node n that minimizes

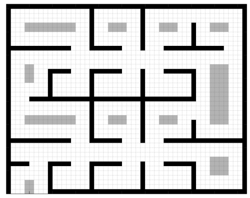


FIGURE 1. A simple map.

$$f(n) = g(n) + h(n) \tag{1}$$

However, despite its optimal efficiency, direct application of the A* algorithm to the grid map of a large multistory building can be computationally intensive. The example path is shown in Fig. 2(a).

2) RRT

The RRT algorithm is well known for its efficiency in nonconvex high-dimensional spaces. The tree grows incrementally by extending to a randomly sampled point, which tends to be biased towards the unexplored area. However, the fact that optimality is not considered at all in the RRT algorithm is a major disadvantage, especially when the path length is important. Jagged path lengths can be significantly different from the actual path, which results in a suboptimal choice in route optimization. Moreover, the path may not be found rapidly if there are narrow passages between the start and the goal points. The resulting path is often so jagged that it has to be post-processed for improved optimality and recognizability. The example path is shown in Fig. 2(b).

3) RRT-CONNECT

The RRT-connect algorithm can be a good alternative, especially for buildings where rooms are accessed through the corridor. The original RRT may have difficulty extending towards the rooms through narrow passages. To overcome this difficulty, the RRT-connect algorithm was developed, where two trees grow from both ends and encounter on the open corridor.

However, faster convergence can yield another problem of exploration and exploitation, which means that the resulting path may be a long detour. The example path is shown in Fig. 2(c).

4) RRT*

The RRT* algorithm is the most popular variant of the RRT algorithm, which generates a remarkably straight path. It achieves an improved optimality by adopting a process called parent selection and rewiring. The improvement can be continued even after the feasible path is found, which allows for asymptotic optimality at the expense of extra computational load. Improved optimality can increase the probability

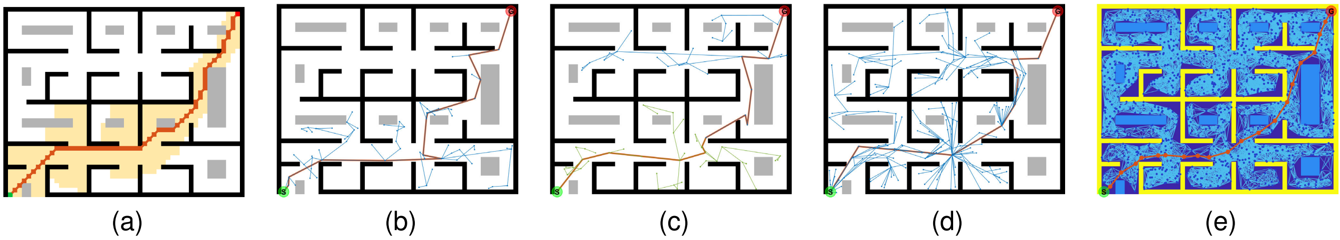


FIGURE 2. Example paths using various path planning algorithms. (a) A*. (b) RRT. (c) RRT-connect. (d) RRT*. (e) PRM.

TABLE 2. Summary of the path planning algorithms.

Algorithm	Strength	Weakness
A*	completeness, optimality, optimal efficiency	space complexity, memory requirement
RRT	fast convergence	incompleteness, non-optimality
RRT-connect	faster convergence	incompleteness, non-optimality
RRT*	asymptotic optimality	incompleteness, slower convergence
PRM	pre-processing, fast convergence, probabilistical completeness	incompleteness

that the found route is optimal. However, there still remains a problem of exploration and exploitation. Furthermore, path optimality can also be accomplished in a more sophisticated way by using various path optimization algorithms. The example path is shown in Fig. 2(d).

5) PRM

Roadmap-based multi-query path planning can be a solution to the exploration and exploitation problem of single-query path planning [57]. The PRM algorithm is one of the most popular multi-query path planning algorithms. It consists of two phases: the construction phase and the query phase. In the construction phase, random samples are taken from the free space and connected to the graph. In the query phase, the shortest path is obtained using Dijkstra's algorithm. It is known that the PRM algorithm is provably probabilistically complete, depending on the number of sampled points. Even though the construction can take a long time as the number of nodes and the maximum connection distance increase, the fact that the construction phase can be done within the response time of the rescue team is one of the most substantial advantages of this approach. In fact, the computational load of the query phase is significantly low compared to the other algorithms. Flexible algorithms based on PRM for dynamic environments can be found in [58] and [59]. There are also many variants adopting more sophisticated sampling and connection strategies. The example path is shown in Fig. 2(e).

The strengths and weaknesses of the algorithms are summarized in Table 2. In summary, the PRM is regarded as the most suitable algorithm for the considered problem.

Algorithm 1 Prune

$i \leftarrow 2$
while $i < n$ **do**
  if IsMotionValid(q_{i-1}, q_{i+1}) **then**
    $\Pi \leftarrow \Pi \backslash q_i$
    $n \leftarrow n - 1$
  else
    $i \leftarrow i + 1$
return Π

D. PATH OPTIMIZATION ALGORITHMS

The planned path may not meet all the requirements. Therefore, a path optimization technique can be applied to improve the quality of the path. Three path optimization techniques are briefly reviewed. One of the simplest techniques is path pruning. The path pruning algorithm removes all redundant nodes, which results in a considerable decrease in the path length. This technique is easy to implement, fast, and deterministic. However, improving the path quality is not dramatic, in that only the nodes of the path are considered. See Algorithm 1 for more details, where $\Pi = \{q_1, \ldots, q_n\}$ is a discrete path and IsMotionValid(a, b) is a function that determines if motion from a to b is collision-free.

The random shortcut algorithm is a highly effective technique with a relatively simple structure. Before the shortcut algorithm is applied, the path is filled with additional nodes so that the distance between the neighboring nodes is at most one predefined step. See Algorithm 2 for more details, where Rem(a, b) returns the remainder after the division of a by b, and Floor(x) rounds x to the nearest integer less than or equal to x. This method randomly selects two configurations on the path and checks if there is a shorter path between them. If the shortcut is found then that part of the path is replaced with the shortcut. This process is repeated until the number of iterations reaches the predefined maximum. The resulting path is asymptotically shortest as the iteration continues. See Algorithm 3 for more details, where MaxNumIter is the maximum number of iterations and RandIt$([a, b])$ returns a random integer between a and b.

E. PATH CLEARANCE

There are many techniques to increase path clearance. Here the retraction algorithm is explained. The method moves

Algorithm 2 Impose Adjacency

$i \leftarrow 1$
while $i < n$ **do**
  $\Delta = q_{i+1} - q_i$
  if $||\Delta|| > k$ **then**
    $\Pi_1 \leftarrow \{q_1, \ldots, q_i\}$
    $\Pi_2 \leftarrow \{q_{i+1}, \ldots, q_n\}$
    if $\mathrm{Rem}(||\Delta||, k) = 0$ **then**
      $m \leftarrow \mathrm{Floor}(||\Delta||/k) - 1$
    else
      $m \leftarrow \mathrm{Floor}(||\Delta||/k)$
    $\Pi' \leftarrow \{q_i + \frac{1}{m+1}\Delta, \ldots, q_i + \frac{m}{m+1}\Delta\}$
    $\Pi \leftarrow \Pi_1 \cup \Pi' \cup \Pi_2$
    $i \leftarrow i + m + 1$
  else
    $i \leftarrow i + 1$
return Π

Algorithm 3 RandomShortcut

$i \leftarrow 0$
while $i <$ MaxNumIter **do**
  $a, b \leftarrow \mathrm{RandInt}([1, n])$ s.t. $a + 1 < b$
  $\Pi_1 \leftarrow \{q_1, \ldots, q_{a-1}\}$ $\triangleright\ \Pi_1 \leftarrow \emptyset$ if $a = 1$
  $\Pi' \leftarrow \mathrm{ImposeAdjacency}(\{q_a, q_b\})$
  $\Pi_2 \leftarrow \{q_{b+1}, \ldots, q_n\}$ $\triangleright\ \Pi_2 \leftarrow \emptyset$ if $b = n$
  $\Pi \leftarrow \Pi_1 \cup \Pi' \cup \Pi_2$
  $i \leftarrow i + 1$
return Π

the selected configuration to the medial axis. The medial axis for an n-DOF object is defined as the set of all $(3 - n - k)$-equidistant locations, where $0 \leq k < n$. For a 2D problem where the configuration space and the workspace coincide, a location on the medial axis has at least 2 equidistant obstacles. As a result, each configuration has a maximum clearance, given the obstacles. See [42] for more details. Applying retraction along the path results in a Voronoi-like path with redundant branches. These branches can be removed by using a relatively simple algorithm.

III. MULTI-LEVEL PATH PLANNING SYSTEM

In this section, a multi-level path planning methodology with clearance-based path optimization for indoor search and rescue operations is proposed. The proposed method maintains the size of the multi-goal path planning problem, even in a multi-story building, by adopting a graph-based approach. Let us explain the overall structure of the multi-level path planning system. Formulating and solving the problem considering requirements can be extremely expensive. Therefore, it is more advantageous to consider several steps in order to meet the various requirements separately. Each subsystem receives the processed information from the preceding subsystem, performs a specific task, and then delivers the result to the following subsystem. Figure 3 shows the structure of the proposed multi-level path planning system, which consists of five steps: map pre-processing, segment path planning, graph processing, route optimization, and path post-processing.

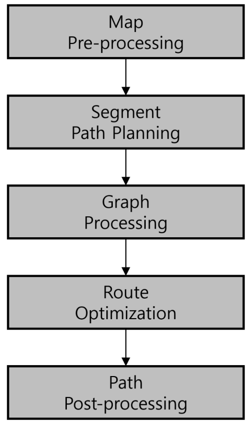


FIGURE 3. Structure of the path planning system.

The ‘Map Pre-processing’ step ensures that no path will be generated over low visibility regions for safer operations. In the ‘Segment Path Planning’ step, the detailed paths corresponding to each segment are generated for precise route determination in the later step. The ‘Graph Processing’ step reduces the size of the graph for a multi-story building. In the ‘Route Optimization’ step, the shortest route that visits all the important nodes is found. The ‘Path Post-processing’ step ensures that the path meets the requirements. Detailed descriptions of each step are as follows.

A. MAP PRE-PROCESSING

In the ‘Map Pre-processing’ step, low visibility regions on the map are marked as occupied so that no other path will be generated over them. In general, the visibility range is very short inside a building on fire, and the walls are the most important and usually the only landmarks. Therefore, it is important to prevent the generated path from passing too far away from the walls. The low visibility regions can be extracted by converting the regions into obstacles, i.e., inflating the obstacle regions. The inflation factor is determined based on the clearance limit, which depends on the visibility conditions. The extracted regions are then added to the original map so that no path can be generated over those regions. This process can be performed during the response time, which contributes to reducing the time for onsite path planning.

B. SEGMENT PATH PLANNING (SINGLE-STORY)

In the ‘Segment Path Planning’ step, detailed segment paths between two positions on the same floor are created. Note that the segment paths correspond to the innermost layer, in that it contains the detailed paths, and the outermost layer, in that the abstract graph is constructed based on the segment path lengths. This step ensures that route finding on a graph to be constructed is performed based on precise edge weights, instead of a naive Euclidean distance. First, the paths connecting the stairs and the entry and exit points can be created and post-processed during the response time. Five path planning algorithms introduced in Table 1 were implemented. If a roadmap-based multi-query path planning algorithm such as PRM is used, then the construction phase, which is computationally heavy in general, can be completed during the response time. Once the target locations are determined, the segments connecting them and their existing nodes are planned. As the operation progresses, the locations of the victims can be added, excluded, or corrected. When the target information changes during the operation, only the segments directly associated with the changed node are affected. Note that the actual structure inside the building can be different from the map due to maintenance or damage and is therefore incompatible with the planned segments. In such a case, only the conflicting segments need to be re-planned. This partial modification adds substantial flexibility and efficiency to the entire system. Once all the segments are planned, the 3-dimensional relation between the nodes can be represented as a weighted graph whose edge weights are segment lengths.

C. GRAPH PROCESSING (MULTI-STORY)

In the 'Graph Processing' step, the information on a multi-story building is represented as an abstract graph. Note that graph representation allows for handling the 2.5D large path planning problem with 2D path planning methods. The graph is further condensed so that the route optimization problem is tractable. Some of the graph nodes are classified as essential nodes, which are entry and exit points and victim locations. The essential nodes must be visited once. The set of nodes consists of the essential nodes and the stairs. Note that each set of stairs can be visited as many times as required or not be visited at all. The aim of the graph processing step is to remove the nonessential nodes from the graph. First, the distance between the essential nodes is obtained by solving the shortest path problem using Dijkstra's algorithm. The reduced graph only has the essential nodes as its nodes, and the nodes are fully connected. The new edges contain information about the intermediate nodes and the detailed segment paths.

D. ROUTE OPTIMIZATION

In the 'Route Optimization' step, the shortest route that i) departs from the entry point, ii) visits each node exactly once, and iii) arrives at the exit point is found. This route-finding problem can be solved accurately because the edge weights are obtained from the actual segment path lengths. Furthermore, the sheer size of the problem is significantly reduced by applying the graph processing step. The simplest way to find the solution is to consider all possible permutations and select the shortest one. If there are $(n+2)$ nodes, then $n!$ cases should be compared, which becomes impractical as the number of nodes becomes large. As an alternative, the problem can be solved based on the integer linear programming (ILP) formulation. In fact, all the variables are binary in this problem. The constraints are very similar to those of the famous traveling salesman problem (TSP). In this study, the Dantzig–Fulkerson–Johnson (DFJ) formulation is adopted [60]. The nodes are labelled with the numbers $1, \ldots, n$, where node 1 is the entry point and node n is the exit point. Let us define x_{ij} as

$$x_{ij} = \begin{cases} 1, & \text{if the path goes from node i to node j} \\ 0, & \text{if } j = 1 \\ 0, & \text{if } i = n \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Take $c_{ij} > 0$ to be the distance from node i to node j. Then, the shortest route problem can be written as a following integer linear programming (ILP) problem.

$$\underset{x_{ij}}{\text{minimize}} \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} c_{ij} x_{ij} \tag{3a}$$

$$\text{subject to} \sum_{i=1, i\neq j}^{n-1} x_{ij} = 1, \quad j = 1, \ldots, n \tag{3b}$$

$$\sum_{j=2, j\neq i}^{n} x_{ij} = 1, \quad i = 1, \ldots, n \tag{3c}$$

$$\sum_{i \in Q} \sum_{j \neq i, j \in Q} x_{ij} \leq |Q| - 1, \quad \forall Q \subseteq 1, \ldots, n, |Q| \leq 2 \tag{3d}$$

The first constraint ensures that the path enters each node once. The second constraint ensures that the path leaves each node once. The last constraint ensures that no sub-tour forms. Note that the constraints become much simpler by applying the graph processing step. Furthermore, the sheer size of the ILP is significantly reduced. Without the graph processing step, the problem can be extremely large when the building has many stories and sets of stairs. The problem can become much larger if multiple candidate positions are considered for each victim, which may be necessary for robust path planning under low-accuracy target localization.

In the route optimization step, the proposed method utilizes an integer linear programming formulation to find the optimal order of visitation for the given targets. This approach is capable of handling multiple constraints while optimizing the objective function, which in this case, is the minimization of the total path length. The formulation considers both the horizontal and vertical connections between nodes and takes into account the edge weights representing the path lengths between them. By solving the ILP problem, we obtain the optimal visitation sequence for the targets, ensuring an efficient path for the search and rescue operation.

E. PATH POST-PROCESSING

In the 'Path Post-processing' step, the jagged paths are optimized in terms of length and clearance. This step ensures that all the path requirements are met, even when the quality of

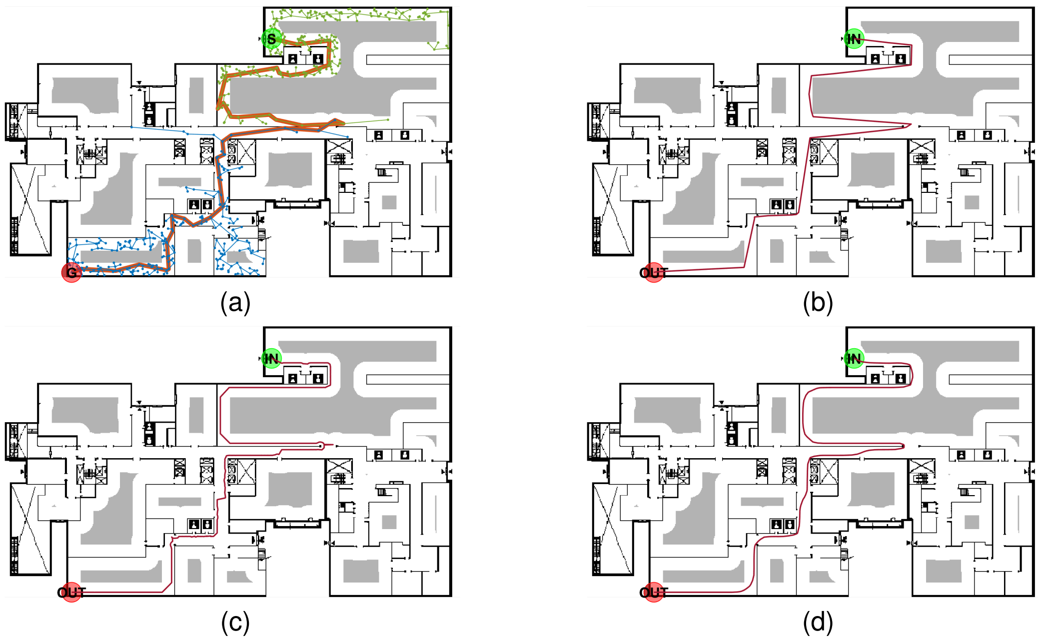


FIGURE 4. Multi-step path post-processing procedure. (a) Raw path based on RRT-connect. (b) A path after a random shortcut is applied. (c) A path after a conditional retraction is applied. (d) A path after a bounded random shortcut is applied.

the raw paths is relatively low. At this step, only the active segments are processed in consecutive order. Early parts of the segments can be processed and provided to the team first, and the latter parts can be processed during the operation in the background. Note that some portions of the segments can already be processed during the response time. Once the post-processing is completed with respect to the active segments, the inactive segments can also be processed in the background, which allows for swift replanning when the situation changes. Moreover, route optimization can be resolved during spare time with the post-processed set of unvisited segments. Because the original route is determined based on the length of the raw paths, the new route can be shorter and more accurate.

In this study, a path optimization technique called 'Conditional Retraction' is proposed. Even though the proposed path post-processing method can be applied to any path, RRT-connected paths with a clearance limit of 4 m are given as an example. It is shown in Fig. 4(a) that the RRT-connect path has many jagged motions. The paths created using A* and PRM, as shown in Figs. 5(a) and 5(b), can also be post-processed in the same way. The path has to be modified to follow the walls, but the direct application of the retraction algorithm makes it impossible to adjust the clearance independently. The clearance is then automatically set to 2 m in the open area when the clearance limit is 4 m. However, it is more desirable to be able to adjust the clearance and its limit independently, depending on the visibility condition. As an alternative, the node can be moved toward or away from the nearest wall so that the clearance becomes a predefined value,

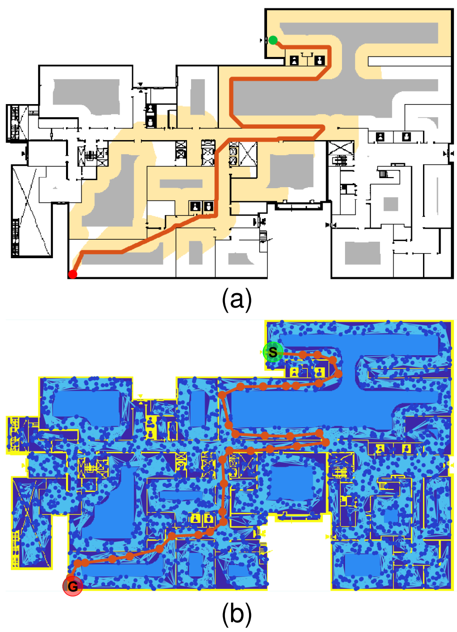


FIGURE 5. A* and PRM path examples. (a) A* path. (b) PRM path.

for example, 1 m in this case. This shift can be performed without any iteration, unlike the retraction, as follows.

$$q' = q + \frac{(q - q_o)}{c}(c_{des} - c) \tag{4}$$

or, equivalently,

$$q' = q_o + \frac{(q - q_o)}{c}c_{des} \tag{5}$$

Algorithm 4 Conditional Retraction

$i \leftarrow 0$
while $i < n$ **do**
  $q \leftarrow q_i$
  $q_o \leftarrow \text{ClosestPair}(q)$
  $c \leftarrow ||q - q_o||$
  $q' \leftarrow q + \frac{(q-q_o)}{c}(c_{des} - c)$
  $q'_o \leftarrow \text{ClosestPair}(q')$
  if $\text{IsMotionValid}(q, q')$ **then**
    if $q_o = q'_o$ **then**
      $q_i \leftarrow q'$
    else
      $q_i \leftarrow q$
  else
    $q_i \leftarrow \text{Retract}(q)$
  $i \leftarrow i + 1$
return Π

Algorithm 5 Bounded Random Shortcut

$i \leftarrow 0$
while $i <$ MaxNumIter **do**
  $a, b \leftarrow \text{RandInt}([1, n])$ s.t. $a + 1 < b$, $||q_a - q_b|| \leq r$
  $\Pi_1 \leftarrow \{q_1, \ldots, q_{a-1}\}$ $\triangleright$ $\Pi_1 \leftarrow \emptyset$ if $a = 1$
  $\Pi' \leftarrow \text{ImposeAdjacency}(\{q_a, q_b\})$
  $\Pi_2 \leftarrow \{q_{b+1}, \ldots, q_n\}$ $\triangleright$ $\Pi_2 \leftarrow \emptyset$ if $b = n$
  $\Pi \leftarrow \Pi_1 \cup \Pi' \cup \Pi_2$
  $i \leftarrow i + 1$
return Π

where q is the subject node, q' is the shifted node, q_o is the nearest obstacle, $c = ||q - q_o||$ is the minimum clearance, and c_{des} is the desired clearance, which should satisfy $c_{des} \leq c_{lim}/2$. The shift result falls into one of the following three cases depending on the local width of the passage w.

1) If and only if $w \leq c_{des}$, $\text{IsMotionValid}(q, q') = 1$. That is, the shift results in wall penetration. Therefore, a retraction is applied instead of a shift.
2) If and only if $c_{des} < w < 2c_{des}$, $\text{IsMotionValid}(q, q') = 1$ and $q_o \neq q'_o$. That is, both sides of the wall are within c_{des}. Therefore, there is no need to shift. Or, a retraction can be applied if required.
3) If and only if $w \geq 2c_{des}$, $\text{IsMotionValid}(q, q') = 1$ and $q_o = q'_o$. In this case, no more action is required.

Therefore, retraction is performed only under the first condition, which involves iteration; otherwise, shifting is performed. See Algorithm 4 for more details.

However, applying conditional retraction to jagged paths can result in a path that unnecessarily alternates between both sides of the wall. One remedy is to apply the random shortcut algorithm before the conditional retraction algorithm is applied. The shortcutting algorithm is then applied to make the path as straight as possible, as shown in Fig. 4(b). Fig. 4(b) shows that the path intersects the medial axis exactly once for each long passage. The conditional retraction algorithm is now applied, as shown in Fig. 4(c). It is shown in Fig. 4(c) that a 1 m clearance is maintained along the path, except for the medial axis crossing and the corners. Note that the temporarily increased clearance at the corners leads to a smoother turn (e.g., 45 deg instead of 90 deg at the right-angled passage), which is preferable for the field agent to follow. However, the path has some redundant branches and two successive sharp turns around the medial axis crossing. Additionally, minor zigzags are caused by the irregularity of the walls. In this study, a 'Bounded Random Shortcut' technique is proposed as a solution. The method is slightly modified from the random shortcut technique in that it only takes a shortcut when the two randomly selected nodes are within some radius. As a result, a much smoother and shorter path without a local detour is obtained, as shown in Fig. 4(d). See Algorithm 5 for more details.

The clearance along the path is shown in Fig. 6. In Fig. 6, it is shown that the raw path has the longest length and random clearance. The path length is significantly decreased after applying random shortcuts. The clearance then becomes highly concentrated to 1 m after applying conditional retraction at the expense of a slight increase in length. Finally, a bounded random shortcut results in a decreased path length, which is almost comparable with that of the shortest path.

The functions and the methods of each step are summarized in Table 3.

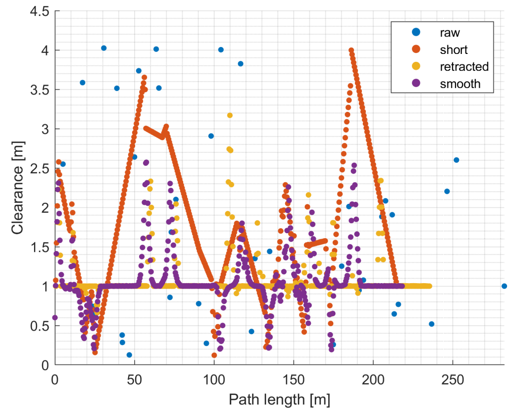


FIGURE 6. Path clearance.

TABLE 3. Summary of the path planning steps.

Step	Function	Method
A. Map pre-processing	Prevent path over a hazardous area	Exclusion of empty space based on the distance to the wall
B. Segment path planning	Connect nodes	PRM-based path planning
C. Graph processing	Reduce the size of a graph	Shortest route finding based on Dijkstra's algorithm
D. Route optimization	Find a shortest route	ILP based on the DFJ formulation
E. Path post-processing	Optimize paths	Random shortcut, conditional retraction, and bounded random shortcut

FIGURE 7. Gunsan medical center.

TABLE 4. Edge weights of the full graph.

Edge	Weight	Edge	Weight	Edge	Weight
IN-T1	125.8	T5-T6	116.6	OUT-S1	113.1
IN-OUT	100.8	T5-S7	140.6	OUT-S2	28.7
IN-S1	73.4	T5-S8	56.9	OUT-S3	25.7
IN-S2	88.2	T5-S9	61.8	S1-S2	100.5
IN-S3	87.0	T6-S7	88.7	S1-S3	99.3
T1-OUT	39.9	T6-S8	60.7	S1-S4	3.0
T1-S1	137.7	T6-S9	63.9	S2-S3	8.6
T1-S2	51.0	T7-T8	59.9	S2-S5	3.0
T1-S3	47.9	T7-T9	52.2	S3-S6	3.0
T2-T3	86.3	T7-T10	57.0	S4-S5	91.8
T2-S4	53.8	T7-S10	32.1	S4-S6	100.5
T2-S5	105.5	T7-S11	65.7	S4-S7	3.0
T2-S6	114.2	T8-T9	97.4	S5-S6	18.1
T3-S4	72.5	T8-T10	102.1	S5-S8	3.0
T3-S5	42.9	T8-S10	30.4	S6-S9	3.0
T3-S6	51.6	T8-S11	110.8	S7-S8	93.3
T4-T5	172.1	T9-T10	24.9	S7-S9	97.1
T4-T6	103.2	T9-S10	67.6	S7-S10	3.0
T4-S7	53.1	T9-S11	33.6	S8-S9	7.8
T4-S8	116.7	T10-S10	74.5	S8-S11	3.0
T4-S9	119.9	T10-S11	10.7	S10-S11	83.2

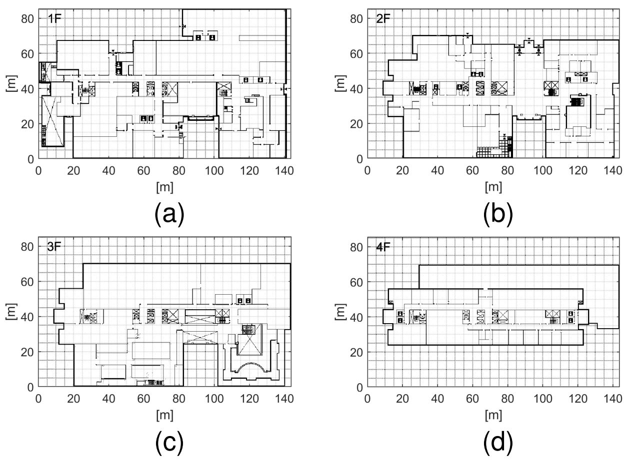


FIGURE 8. Building floor plans. (a) 1F. (b) 2F. (c) 3F. (d) 4F.

IV. PERFORMANCE DEMONSTRATION

In this section, the performance of the proposed path planning system is demonstrated. The lower part (from 1F to 4F) of the main building of the Gunsan Medical Center (Fig. 7) located in Gunsan, Korea, is considered. The original map is moderately refined to suit the path planning problem.

First, the map is pre-processed with a clearance limit of 4 m. The floor plan of the building is shown in Fig. 8, and the scenario for path planning is shown in Fig. 9. There

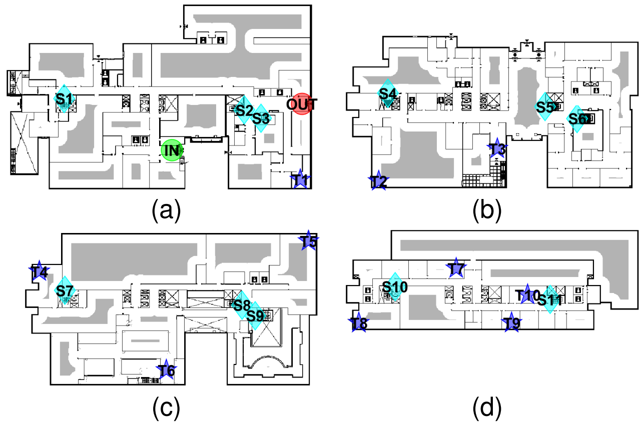


FIGURE 9. Scenario: Visit from T1 (1F) to T10 (4F). (a) 1F. (b) 2F. (c) 3F. (d) 4F.

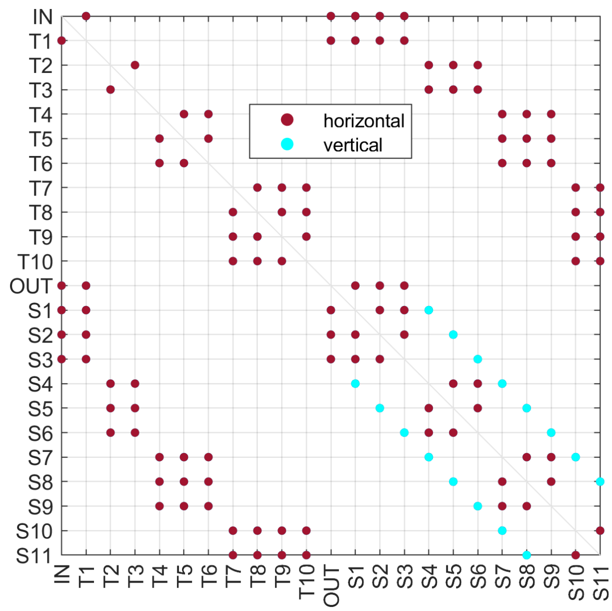


FIGURE 10. Adjacency of the nodes.

are 11 access points to the staircases (S1-S11), which are vertically connected, as shown in Fig. 10. It is assumed that there are 10 targets: 1 target on 1F (T1), 2 targets on 2F (T2, T3), 3 targets on 3F (T4-T6), and 4 targets on 4F (T7-T10). In summary, the total number of nodes is 23, and 12 nodes must be visited.

Then, 55 segment paths corresponding to the horizontal connections in Fig. 10 are planned. Note that the adjacency matrix is symmetric, and the opposite triangle can be filled out by simply reversing the corresponding path. All segments generated using the PRM are shown in Fig. 11. The full graph can be built based on the connectivity and path lengths, as shown in Fig. 12. The edge weights of the full graph are summarized in Table 4. The edge weights for the vertical paths are set to 3 m, which can affect the route optimization results, to some extent.

In the graph processing step, the nonessential nodes are eliminated by directly connecting all essential nodes to one

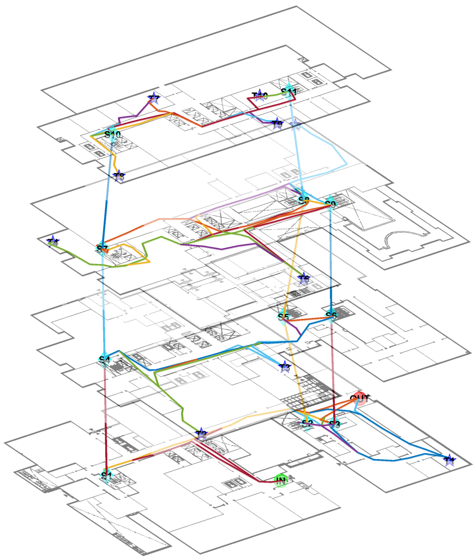


FIGURE 11. All generated segments.

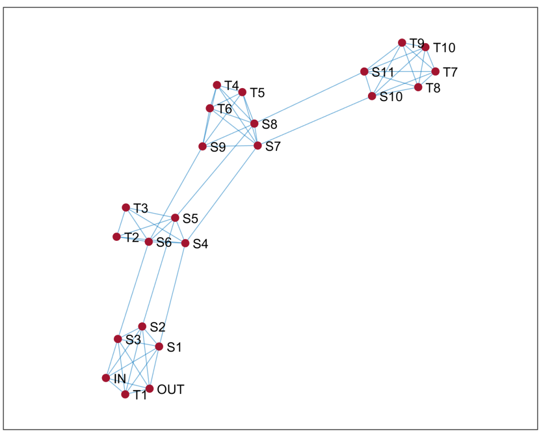


FIGURE 12. Full graph.

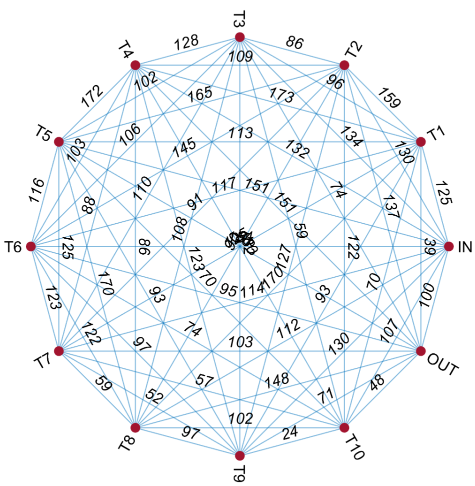


FIGURE 13. Reduced graph.

another. The reduced graph is shown in Fig. 13. The optimal route on the reduced graph can be found based on ILP. The

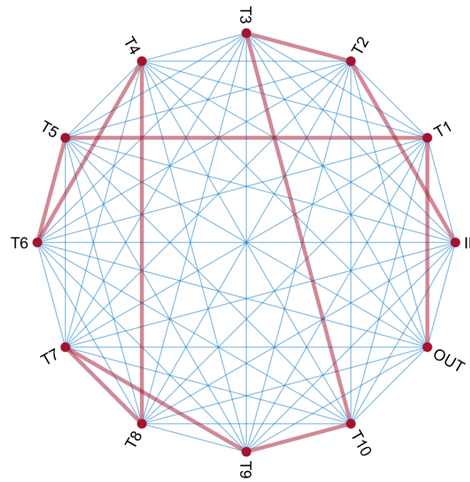


FIGURE 14. The solution on the reduced graph.

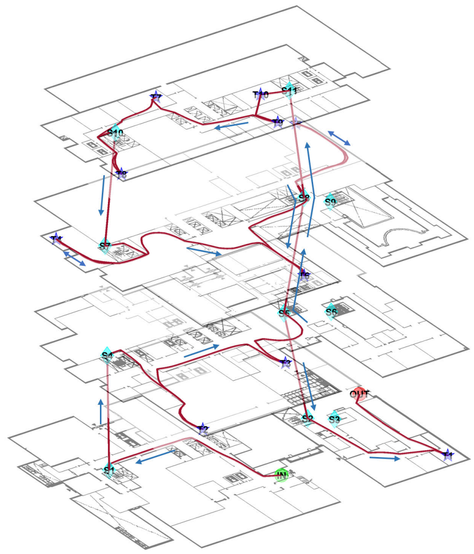


FIGURE 15. The entire path represented on the 3D map.

TABLE 5. Computation time for each step.

Task	Computation time [s]
Map pre-processing (4 floors)	7.95, 7.95, 7.84, 7.70
PRM generation (4 floors)	1.88, 1.99, 1.92, 2.48
Segment path planning (55 segments)	4.58
Graph processing	0.05
Route optimization	1.64
Path post-processing	22.04

optimal order of visitation is obtained as {IN, T2, T3, T10, T9, T7, T8, T4, T6, T5, T1, OUT}, which is shown on the reduced graph in Fig. 14. The full route can be obtained by expanding the order as {IN, S1, S4, T2, T3, S5, S8, S11, T10, T9, T7, T8, S10, S7, T4, T6, T5, S8, S5, S2, T1, OUT}. The post-processed path represented in the 3D map is shown in Fig. 15, whose total length is 865 m.

The algorithms are executed based on MathWorks MATLAB R2021b on an *Intel Core i7-4790* (3.60 GHz) processor. The computation time required to plan 55 segments is 4.58 s, and the average time of segment path planning is approximately 0.1 s. The computation times at each step are summarized in Table 5.

In summary, the proposed path planning system successfully generated a path with a total length of 865 meters, visiting all ten target locations within the multi-story building. The average computation time for segment path planning was approximately 0.1 seconds, with the overall computation time for each step detailed in Table 5. The case study demonstrates the effectiveness of the proposed multi-level path planning system in a real-world indoor environment. The system is able to efficiently navigate through the complex structure of the Gunsan Medical Center's main building, visiting all the required target locations while considering clearance limits and optimizing the route. This showcases the system's ability to adapt to and effectively address the challenges of indoor search and rescue operations.

V. CONCLUSION

In this study, a multi-level path planning system was proposed for indoor search and rescue operations. Requirements for the path planning system were derived, and a five-stage path planning system was proposed, which consisted of map pre-processing, segment path planning, graph processing, route optimization, and path post-processing. The proposed method provides systematic and efficient multi-goal shortest path planning for search and rescue operations inside buildings on fire. The multi-query approach exploits the response time and has flexibility with respect to dynamic environment changes. For future work, the effectiveness of the proposed scheme will be verified through a field test with a real-time integrated indoor navigation system.

In anticipation of the field experiments, we plan to carefully design and implement a rigorous testing protocol to ensure that the effectiveness of the proposed multi-level path planning system is thoroughly assessed under realistic conditions. These field experiments will simulate indoor search and rescue operations in various building layouts and environments, incorporating dynamic obstacles and hazards, such as fire, smoke, and structural damage. Key performance metrics, such as success rate, response time, and path optimality, will be measured and compared to existing path planning solutions. Furthermore, we will collaborate with professional search and rescue teams to gain valuable insights into the practical challenges faced during real operations. Their feedback will enable us to refine and improve our system, ensuring that it is both effective and usable in real-world scenarios. Ultimately, these field experiments will serve as a crucial step toward the broader adoption of advanced path planning systems for indoor search and rescue operations, with the potential to save lives and improve overall emergency response efficiency.

REFERENCES

[1] M. Ashish, J. Vilela, G. Nejat, and B. Benhabib, "A multirobot path-planning strategy for autonomous wilderness search and rescue," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1784–1797, Sep. 2014.

[2] R. Cui, Y. Li, and W. Yan, "Mutual information-based multi-AUV path planning for scalar field sampling using multidimensional RRT," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 7, pp. 993–1004, Jul. 2016.

[3] Y. Liu, Q. Wang, H. Hu, and Y. He, "A novel real-time moving target tracking and path planning system for a quadrotor UAV in unknown unstructured outdoor scenes," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 11, pp. 2362–2372, Nov. 2019.

[4] J. Wang and M. Q.-H. Meng, "Real-time decision making and path planning for robotic autonomous luggage trolley collection at airports," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 4, pp. 2174–2183, Apr. 2022.

[5] R. G. Ribeiro, L. P. Cota, T. A. M. Euzebio, J. A. Ramirez, and F. G. Guimaraes, "Unmanned-aerial-vehicle routing problem with mobile charging stations for assisting search and rescue missions in postdisaster scenarios," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 11, pp. 6682–6696, Nov. 2022.

[6] C. Wang, J. Cheng, W. Chi, T. Yan, and M. Q.-H. Meng, "Semantic-aware informative path planning for efficient object search using mobile robot," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 8, pp. 5230–5243, Aug. 2021.

[7] X. Zhang, J. Wang, Y. Fang, and J. Yuan, "Multilevel humanlike motion planning for mobile robots in complex indoor environments," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1244–1258, Jul. 2019.

[8] M. P. Linegang, H. A. Stoner, M. J. Patterson, B. D. Seppelt, J. D. Hoffman, Z. B. Crittendon, and J. D. Lee, "Human-automation collaboration in dynamic mission planning: A challenge requiring an ecological approach," in *Proc. Hum. Factors Ergon. Soc. Annu. Meeting*, 2006, pp. 2482–2486.

[9] S.-Y. Chien, H. Wang, and M. Lewis, "Human vs. algorithmic path planning for search and rescue by robot teams," *Proc. Hum. Factors Ergonom. Soc. Annu. Meeting*, vol. 54, no. 4, pp. 379–383, Sep. 2010.

[10] J. Yuan, S. Zhang, Q. Sun, G. Liu, and J. Cai, "Laser-based intersection-aware human following with a mobile robot in indoor environments," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 1, pp. 354–369, Jan. 2021.

[11] L.-W. Chen and J.-X. Liu, "Time-efficient indoor navigation and evacuation with fastest path planning based on Internet of Things technologies," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 5, pp. 3125–3135, May 2021.

[12] L.-W. Chen, J.-H. Cheng, and Y.-C. Tseng, "Optimal path planning with spatial–temporal mobility modeling for individual-based emergency guiding," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 12, pp. 1491–1501, Dec. 2015.

[13] I. Maurović, M. Seder, K. Lenac, and I. Petrović, "Path planning for active SLAM based on the D* algorithm with negative edge weights," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 8, pp. 1321–1331, Aug. 2018.

[14] 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.

[15] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Iowa State Univ., Ames, IA, USA, Tech. Rep. 98-11, Oct. 1998.

[16] S. M. LaValle and J. J. Kuffner, Jr., "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.

[17] G. Noh, J. Park, D. Han, and D. Lee, "Selective goal aiming rapidly exploring random tree path planning for UAVs," *Int. J. Aeronaut. Space Sci.*, vol. 22, no. 6, pp. 1397–1412, Dec. 2021.

[18] S. M. LaValle and J. J. Kuffner, Jr., "Randomized kinodynamic planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, Detroit, MI, USA, vol. 1, Jul. 1999, pp. 473–479.

[19] 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)*, San Francisco, CA, USA, vol. 2, Apr. 2000, pp. 995–1001.

[20] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.

[21] S. Klemm, J. Oberländer, A. Hermann, A. Roennau, T. Schamm, J. M. Zollner, and R. Dillmann, "RRT*-connect: Faster, asymptotically optimal motion planning," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Zhuhai, China, Dec. 2015, pp. 1670–1677.

[22] L. Jaillet, J. Cortés, and T. Siméon, "Sampling-based path planning on configuration-space costmaps," *IEEE Trans. Robot.*, vol. 26, no. 4, pp. 635–646, Aug. 2010.

[23] 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.

[24] 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.

[25] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[26] L. E. Kavraki, M. N. Kolountzakis, and J.-C. Latombe, "Analysis of probabilistic roadmaps for path planning," *IEEE Trans. Robot. Autom.*, vol. 14, no. 1, pp. 166–171, Feb. 1998.

[27] A. Konar, I. G. Chakraborty, S. J. Singh, L. C. Jain, and A. K. Nagar, "A deterministic improved Q-learning for path planning of a mobile robot," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 5, pp. 1141–1153, Sep. 2013.

[28] Y. Wang, X. Li, and R. Ruiz, "An exact algorithm for the shortest path problem with position-based learning effects," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 11, pp. 3037–3049, Nov. 2017.

[29] Y. Liu, Z. Chen, Y. Li, M. Lu, C. Chen, and X. Zhang, "Robot search path planning method based on prioritized deep reinforcement learning," *Int. J. Control, Autom. Syst.*, vol. 20, no. 8, pp. 2669–2680, Aug. 2022.

[30] S. A. Mokhtari, "FOPID control of quadrotor based on neural networks optimization and path planning through machine learning and PSO algorithm," *Int. J. Aeronaut. Space Sci.*, vol. 23, no. 3, pp. 567–582, Jul. 2022.

[31] H. Ma, W. Zou, S. Sun, Z. Zhu, and Z. Kang, "FOV constraint region analysis and path planning for mobile robot with observability to multiple feature points," *Int. J. Control, Autom. Syst.*, vol. 19, no. 11, pp. 3785–3800, Nov. 2021.

[32] J. Shin, D. Kwak, and K. Kwak, "Model predictive path planning for an autonomous ground vehicle in rough terrain," *Int. J. Control, Autom. Syst.*, vol. 19, no. 6, pp. 2224–2237, Jun. 2021.

[33] X. Lv, W. Li, and J. Wang, "Safety-field-based path planning algorithm of lane changing for autonomous vehicles," *Int. J. Control, Autom. Syst.*, vol. 20, no. 2, pp. 564–576, Feb. 2022.

[34] C. Hu, Z. Meng, G. Qu, H. S. Shin, and A. Tsourdos, "Distributed cooperative path planning for tracking ground moving target by multiple fixed-wing UAVs via DMPC-GVD in urban environment," *Int. J. Control, Autom. Syst.*, vol. 19, no. 2, pp. 823–836, 2021.

[35] D. Lim, J. Park, D. Han, H. Jang, W. Park, and D. Lee, "UAV path planning with derivative of the heuristic angle," *Int. J. Aeronaut. Space Sci.*, vol. 22, no. 1, pp. 140–150, Feb. 2021.

[36] D. Seo and J. Kang, "Collision-avoided tracking control of UAV using velocity-adaptive 3D local path planning," *Int. J. Control, Autom. Syst.*, vol. 21, no. 1, pp. 231–243, Jan. 2023.

[37] X. Liu, Z.-R. Peng, L.-Y. Zhang, and Q. Chen, "Real-time and coordinated UAV path planning for road traffic surveillance: A penalty-based boundary intersection approach," *Int. J. Control, Autom. Syst.*, vol. 20, no. 8, pp. 2655–2668, Aug. 2022.

[38] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[39] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[40] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," 2018, *arXiv:1812.05905*.

[41] S. Fujimoto, H. Van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. 35th Int. Conf. Mach. Learn.*, Stockholm, Sweden, Jul. 2018, pp. 1587–1596.

[42] R. Geraerts and M. H. Overmars, "Creating high-quality paths for motion planning," *Int. J. Robot. Res.*, vol. 26, no. 8, pp. 845–863, Aug. 2007.

[43] R. Geraert and M. H. Overmars, "Clearance based path optimization for motion planning," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, New Orleans, LA, USA, vol. 3, Apr./May 2004, pp. 2386–2392.

[44] H. Choset and J. Burdick, "Sensor-based exploration: The hierarchical generalized Voronoi graph," *Int. J. Robot. Res.*, vol. 19, no. 2, pp. 96–125, Feb. 2000.

[45] M. Amersdorfer and T. Meurer, "Equidistant tool path and Cartesian trajectory planning for robotic machining of curved freeform surfaces," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 4, pp. 3311–3323, Oct. 2022.

[46] S. A. Wilmarth, N. M. Amato, and P. F. Stiller, "MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, vol. 2, May 1999, pp. 1024–1031.

[47] C. Holleman and L. E. Kavraki, "A framework for using the workspace medial axis in PRM planners," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, San Francisco, CA, USA, vol. 2, Apr. 2000, pp. 1408–1413.

[48] J.-M. Lien, S. L. Thomas, and N. M. Amato, "A general framework for sampling on the medial axis of the free space," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Taipei, Taiwan, vol. 3, Sep. 2003, pp. 4439–4444.

[49] J. Denny, E. Greco, S. Thomas, and N. M. Amato, "MARRT: Medial axis biased rapidly-exploring random trees," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Hong Kong, May 2014, pp. 90–97.

[50] P. Isto, "Constructing probabilistic roadmaps with powerful local planning and path optimization," in *Proc. IEEE Int. Conf. Intell. Robots Syst. (IROS)*, Lausanne, Switzerland, vol. 3, Sep./Oct. 2002, pp. 2323–2328.

[51] M. Hasan, M. L. Gavrilova, and J. G. Rokne, "A geometric approach to clearance based path optimization," in *Proc. Int. Conf. Comput. Sci.*, Kuala Lumpur, Malaysia, Aug. 2007, pp. 136–150.

[52] B. Park, J. Choi, and W. K. Chung, "Sampling-based retraction method for improving the quality of mobile robot path planning," *Int. J. Control, Autom. Syst.*, vol. 10, no. 5, pp. 982–991, Oct. 2012.

[53] L. Zhang and D. Manocha, "An efficient retraction-based RRT planner," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Pasadena, CA, USA, Sep. 2008, pp. 3743–3750.

[54] J. Pan, L. Zhang, and D. Manocha, "Retraction-based RRT planner for articulated models," in *Proc. IEEE ICRA*, Anchorage, AK, USA, May 2010, pp. 2529–2536.

[55] J. Lee, O. Kwon, L. Zhang, and S.-E. Yoon, "SR-RRT: Selective retraction-based RRT planner," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Saint Paul, MN, USA, May 2012, pp. 2543–2550.

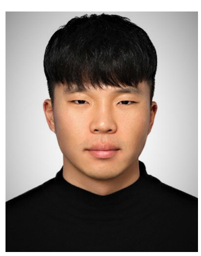
[56] A. K. Pamosoaji and K.-S. Hong, "A path-planning algorithm using vector potential functions in triangular regions," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 43, no. 4, pp. 832–842, Jul. 2013.

[57] P. Bhattacharya and M. L. Gavrilova, "Roadmap-based path planning-using the Voronoi diagram for a clearance-based shortest path," *IEEE Robot. Automat. Mag.*, vol. 15, no. 2, pp. 58–66, Jun. 2008.

[58] L. Jaillet and T. Siméon, "A PRM-based motion planner for dynamically changing environments," in *Proc. IEEE Int. Conf. Intell. Robots Syst. (IROS)*, Sendai, Japan, vol. 2, Sep./Oct. 2004, pp. 1606–1611.

[59] K. Belghith, F. Kabanza, L. Hartman, and R. Nkambou, "Anytime dynamic path-planning with flexible probabilistic roadmaps," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Orlando, FL, USA, May 2006, pp. 2372–2377.

[60] G. B. Dantzig, "Linear programming and extensions," RAND Corp., Santa Monica, CA, USA, Tech. Rep. R-366-PR, Aug. 1963.



JIHOON LEE received the B.S. degree in mechanical engineering and astronomy from Yonsei University, Seoul, Republic of Korea, in 2014. He is currently pursuing the Ph.D. degree in aerospace engineering with Seoul National University, Seoul. His current research interests include motion planning, target tracking systems, robust and gain-scheduled control of linear parameter-varying systems, and intelligent flight control systems for morphing aircraft.

HEEIN PARK received the B.S. and M.S. degrees in mechanical and aerospace engineering from Seoul National University, Republic of Korea, in 2020, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. His current research interests include optimal control theory and path planning.

YOUDAN KIM (Senior Member, IEEE) received the B.S. and M.S. degrees in aeronautical engineering from Seoul National University, Republic of Korea, in 1983 and 1985, respectively, and the Ph.D. degree in aerospace engineering from Texas A&M University, in 1990. He joined Seoul National University, in 1992, where he is currently a Professor with the Department of Aerospace Engineering. His current research interests include aircraft control system design, reconfigurable control system design, path planning, and guidance techniques for aerospace systems.

CHAN GOOK PARK (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in control and instrumentation engineering from Seoul National University, Seoul, Republic of Korea, in 1985, 1987, and 1993, respectively. From 1994 to 2003, he was an Associate Professor with Kwangwoon University, Seoul. He was a Postdoctoral Fellow with Prof. J. L. Speyer for peak seeking control for a formation flight with the University of California at Los Angeles, Los Angeles, CA, USA, in 1998. In 2003, he joined the School of Mechanical and Aerospace Engineering, Seoul National University, where he is currently a Professor. In 2009, he was a Visiting Scholar with the Department of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA. His current research interests include advanced filtering techniques, high-precision inertial navigation systems (INSs), visual-inertial odometry (VIO), INS/GNSS/IBN integration, and smartphone-based/foot-mounted pedestrian dead reckoning (PDR) systems. He served as the Chair for the IEEE AES Korea Chapter, until 2009.

JAE HONG LEE (Student Member, IEEE) received the B.S. degree from the School of Mechanical and Electrical Control Engineering, Handong Global University, in 2017, and the M.S. degree from the Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, Republic of Korea, in 2019, where he is currently pursuing the Ph.D. degree. His research interests include pedestrian dead reckoning and inertial navigation systems.